

# Simulation of Rolling Element Bearings

Lars-Erik Stacke

Dag Fritzon

SKF Nova AB

Chalmers Teknikpark, S-412 88 Göteborg, Sweden

October 1, 1999

## Abstract

Rolling bearings are high precision, low cost machine elements, used in all kinds of rotating machinery. Simulations of rolling bearings bring increased understanding of their dynamic behaviour, and shortens product development time. A rolling bearing simulation model called BEAST (BEARING Simulation Tool), has been developed by SKF. Due to the high demands on contact geometry description and contact force calculations (including traction), simulations are computationally intensive and are normally run on parallel computers. The simulations have been verified against several experiments, e.g., cage forces in a deep groove ball bearing.

## Notation

$\mathbf{F}_i^b$	vector of generalised boundary reaction forces acting on body $i$ , [N, Nm]
$i, j$	body numbers, [-]
$\mathbf{M}_i$	matrix of inertia for body $i$ , [kg, kg m <sup>2</sup> ]
$N$	number proportional to the number of rolling elements, [-]
$n_B$	number of bodies, [-]
$n_W$	number of rolling elements, [-]
$n_S$	number of contact segments, [-]
$\mathbf{q}_i$	vector of generalised positions for body $i$ , [m, rad]
$t$	time, [s]
$t^*$	arbitrary time instance, [s]
$t_{\text{start}}$	simulation start time, [s]
$\mathbf{v}_i$	vector of time derivatives of $\mathbf{q}_i$ , [m/s, rad/s]
$\mathbf{F}_i$	vector of generalised forces acting on body $i$
$\mathbf{F}_{i,j}^c$	vector of generalised contact forces acting on body $i$ due to contact with body $j$ , [N, Nm]
$\mathbf{F}_i^e$	vector of generalised external forces acting on body $i$ , [N, Nm]
$\mathbf{z}_i$	vector of generalised state variables for body $i$ , [m, rad]
$\mathbf{z}_0$	vector of generalised initial state variables, [m, rad]

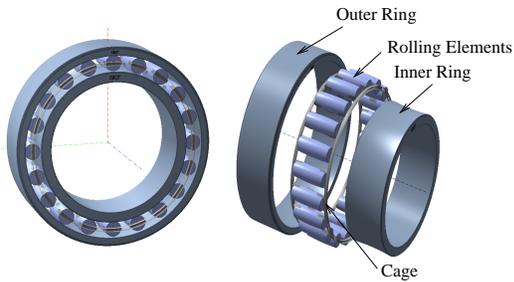


Figure 1: A rolling bearing consist of an inner ring, an outer ring, rolling elements, and a cage. Here exemplified by a CARB™ toroidal roller bearing.

## 1 Introduction

Rolling bearings are machine elements with demands for high precision and load carrying capacity, together with low friction torque, vibration and noise emissions. At the same time they are mass produced. The role of a rolling bearing is to act, without being seen or heard. A normal household contains about 150 rolling bearings in different kinds of machinery.

Rolling bearings generally consist of four types of parts: an inner ring, an outer ring, a set of rolling elements (balls or rollers), and a cage (see Figure 1). These components interact with each other in a dynamic way. Even though rolling bearings have been around for more than a hundred years, there are still aspects of them that are not sufficiently understood. This is especially true for their dynamic behaviour, most certainly when it comes to the cage.

Rolling bearings are thus a perfect candidate to be studied with multi-body dynamics techniques. However, the traditional tools for multi-body dynamics cannot readily be used, since the contacting surfaces cannot be described with sufficient detail, and the routines for calculating the contacting forces are too rudimentary and slow.

SKF, a leading manufacturer of rolling bearings, has therefore developed a bearing simulation tool called BEAST (BEARING Simulation Tool). Thanks to efficient contact algorithms, and utilization of parallel computing, rolling bearing simulation is now possible. BEAST is specifically designed to simulate the behaviour of a complete bearing, including the cage, and handles

all kinds of rolling bearings. Other rolling bearing simulation codes usually either leave out the cage, or describe it in a very basic way, e.g., two-dimensional models. An exception is ADORE, a three-dimensional bearing simulation model developed by Gupta [8]. However, ADORE was developed very much with calculation speed in mind, leading to simplified contact force calculations, especially for the cage and flange contacts. This limits the accuracy of the simulations of, e.g., roller skew behaviour. The bearing geometries are restricted to a number of standard types, with limited possibilities for variations.

BEAST is a fully three-dimensional model, and no assumptions are made as to the static or dynamic behaviour of the bearing components. The geometric description is also very general in BEAST, which is important, especially for the cage.

The simulation code has been ported to various computer systems, ranging from PCs with Windows NT to Cray Supercomputers. Even if most production simulations are run on parallel computers, they are easily accessible to SKF's engineers via computer networks (intranet). A graphical front end makes it easy to submit and monitor simulations, and to retrieve output data from any SKF site.

## 2 Simulation model

A bearing is modeled as a multi-body system. Bearing simulation involves the simultaneous solution of Newton's equations of motion for each body in the bearing. Newton's second and third laws are written as second order ordinary differential equations (ODEs). Typical characteristics of ODEs for rolling bearings are: stiffness, very high demands on numerical precision, and computationally expensive evaluation of the derivatives.

Efficient and accurate contact calculations are important in bearing simulation. The contact force and moment calculations are mainly based on elastohydrodynamic lubrication theory (EHL). Detailed geometric description is also very important, since features of the order of  $0.1 \mu\text{m}$  may have a significant effect on bearing performance.

## 2.1 Equation system for multi-body dynamics

This article considers one-row rolling bearings with one cage. It is modelled with  $n_B$  number of structurally rigid bodies, where  $n_B = n_W + 3$ ,  $n_W$  is the number of rolling elements, and the number 3 comes from the inner ring, the outer ring, and the cage.

Let  $\mathbf{q}_i$  be a vector of position and angle variables for body  $i$ ,  $\mathbf{M}_i$  a matrix of inertia, and  $\mathbf{F}_i$  a vector of generalised forces acting on body  $i$ . The elements in  $\mathbf{F}_i$  consist of the sum of:

$\mathbf{F}_{i,j}^c$ , the forces acting on body  $i$  due to *contact* with body  $j$ , where  $j \in n_B$  and  $j \neq i$ ,

$\mathbf{F}_i^e$ , *external* forces acting on body  $i$ ,

$\mathbf{F}_i^b$ , reaction forces from the *boundary* conditions acting on body  $i$ .

The laws of Newton give for body  $i$

$$\mathbf{M}_i \ddot{\mathbf{q}}_i = \mathbf{F}_i(\mathbf{q}, \dot{\mathbf{q}}, t), \quad (1)$$

With  $\dot{\mathbf{q}}_i = \mathbf{v}_i$  and  $\dot{\mathbf{v}}_i = \ddot{\mathbf{q}}_i$ , and by setting  $\mathbf{z} = (\mathbf{q}, \mathbf{v})$  the ODEs can be written in canonical form:

$$\dot{\mathbf{z}} = \text{RHS}(\mathbf{z}, t), \quad \mathbf{z}(t_{\text{start}}) = \mathbf{z}_0, \quad (2)$$

where RHS denotes the right hand side. This is a standard form for an *initial value problem* (IVP) with initial value  $\mathbf{z}_0$ , the state variable  $\mathbf{z}$ , and the independent variable  $t$ .

## 2.2 Contact force calculations, RHS

The contact force model is the most important part of a rolling bearing simulation model. High demands are put on the contact model when it comes to:

**Speed:** A typical simulation of a rolling bearing involves millions of contact force calculations. Even if they are very fast they will still account for the majority of the computational work.

**Generality:** The contact model has to handle all kinds of contact situations in a rolling bearing. Even if many surfaces in rolling bearings can be described with simple curva-

tures, deviations from these simple surfaces are very important and must be handled in a good way. Moreover, the geometry of cages varies widely. The contact model must be able to handle general contacts, not only Hertzian contacts.

The geometric description must allow the user to specify the geometry he wants, since very small geometrical deviations may affect the performance of a bearing.

**Stability/Continuity:** The contact forces have to have a high order of continuity (i.e., several times differentiable) with respect to the *contact variables*, e.g., the relative motion of the contacting surfaces, and other state variables. Discontinuous forces are handled by the solver with shorter time steps, giving longer simulation times.

**Accuracy:** The traction forces are just as important as the normal forces in bearing simulation. Bearing behaviour is often governed by the traction forces, e.g., roller guidance. Various types of damping may have a profound effect on the dynamic behaviour of a bearing.

The contact model in BEAST includes three-dimensional elastic effects of arbitrary geometry, truncation, oil film influence, traction, and squeeze. The oil film thickness and the additional pressure distribution from the oil are calculated with a simplified algorithm, based on the EHL model by Venner [10], since run-time EHL calculations are far too slow for dynamic simulations. For each point in the contact, the oil flow is assumed to be in the principal rolling/sliding direction, which is well defined for most contacts in a rolling bearing.

Semi-empirical models for material damping and squeeze are used for all contacts. The models are based on experiments with balls bouncing on plates made of various materials and with different oil layer thicknesses.

The traction forces are calculated with a non-Newtonian rheological model, using the Newtonian film thickness, slip speeds, and pressure (with influence from elastic deformation, oil film, and squeeze effects).

## 2.3 ODE solver

For a system of ordinary differential equations where the right hand side is expensive to evaluate, the use of linear multi-step methods is preferred over one-step methods, i.e., Runge-Kutta methods. Furthermore, the system of ODEs is “stiff”, which makes implicit Backward Differentiation Formula (BDF) methods suitable. They are also good for problems where the RHS is expensive to evaluate since not more than one or two evaluations of the RHS are normally needed for each step. However, this method requires a Jacobian to be calculated, though not necessarily for every step.

The use of rather small tolerances (needed for bearing simulation), makes it beneficial to use methods of high order (four-five). The BDF methods of such orders could, however, have problems with eigenvalues lying on the imaginary axis. The choice of integration method is at the moment CVODE, by Cohen and Hindmarsh [4], which uses BDF methods. The CVODE solver has been extended to handle the new type of Jacobian (see below). The BDF methods are well proven methods for this application, and give results in which the engineers at SKF have confidence.

## 2.4 Calculation scheme

The code in the bearing simulation tool consists of two major blocks [5], the code implementing rolling bearing model, and the numerical integration routines. The model defines the system of ODEs and the Jacobian (JAC) to the system. The integration routine uses the bearing model and some linear algebra to do the numerical integration over time.

The crucial parts in the rolling bearing code are the evaluation of the right hand side (RHS) of the ODEs, and the calculation of the Jacobian. Both involve computationally intensive contact force calculations (each rolling element can be in contact with the inner ring, the outer ring, and the cage).

The computationally intensive parts of the numerical integration routine, CVODE, are the LU factorization of the Newton matrix (a matrix of the

same structure as the Jacobian, see below), and the forward/backward elimination that is needed to obtain a solution to the linear system within the CVODE solver.

The numerical integration in the CVODE solver is roughly structured as follows:

- The solver has a number of state vectors, which are the positions and velocities of the bodies, i.e.,  $z(t)$  in Equation 2.
- A new value for the state vector is calculated, at a new point in time  $t^*$ , depending on the previous history.
- The RHS is called in order to obtain the derivatives at  $t^*$ . The call to the RHS involves determining the contact variables (i.e., the relative positions, rotations, and velocities of the two bodies, and other state variables) and integration of the contact forces. The solver might also call the JAC function in order to update the Jacobian. For typical bearing simulations the ratio RHS/JAC is of the order 50/1.
- If the time integration was successful, a new state vector is returned. This state vector is normally not the state vector where the contact forces were last calculated. Therefore, the current state vector is used to evaluate the RHS at regular intervals, in order to obtain the contact forces and other output data of this state. The state vector, contact forces, and other interesting variables are then written to a file. Typically, this is done every 10th step.

## 2.5 Jacobian

The Jacobian is involved in several very time consuming tasks, the assembly (calculation of the partial derivatives), the LU factorization of the Newton matrix, and forward/backward elimination in the solving of the linear system of equations. By using the knowledge of the rolling element system, this computational effort could be drastically reduced, especially for bearings with many rolling elements [5].

Since there is no analytical Jacobian available, the traditional way to calculate the partial derivatives is the finite difference method. The compu-

tational work for this method grows quadratically with the number of rolling elements. By using a novel strategy, the computational work could be reduced to order  $N$  (where  $N$  is proportional to the number of rolling elements).

The topological structure of the Jacobian for the rolling bearing system with a cage can be classified as a block diagonal with borders. This is because the rolling elements cannot be in contact with each other, only with the rings and the cage (see Figure 2). By utilizing this structure of the Jacobian, the number of operations required for the LU factorization, and the forward/backward elimination could be reduced with respect to the number of rolling elements. Compared to the built-in algorithm the reduction was:

- LU factorization:  $N^3 \rightarrow N$ ,
- forward/backward elimination:  $N^2 \rightarrow N$ .

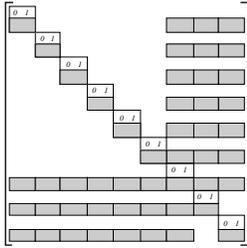


Figure 2: The structure of the Jacobian for single row rolling bearing with a cage. The bearing has six rolling elements.

## 2.6 Parallel solution of the ODEs

As mentioned earlier the computation of the right hand side and the Jacobian are quite time consuming. A good way to shorten the wall clock time for the simulations is to split the computational work on several parallel processors.

It is common to categorize parallel methods of solving ODEs into three classes according to what kind of parallelism they exploit; parallelism across the system, parallelism across the method, and parallelism across time (for more information see [1–3, 7, 9]). The first method, parallelism across the system, i.e., the bearing, is used in BEAST. Since it is the contact force calculations that are the most time consuming, it is natural to parallelize over them.

One processor is assigned to steer the computations, the so called *master*. The master distributes the work on the other processors, the *slaves*.

The following levels of granularity in the parallelization have been identified, (where  $n_W$  is the number of rolling elements):

1. Parallelization over the rolling elements. The tasks for the slaves will be to evaluate the RHS for one specific rolling element, or to calculate the sub-Jacobian belonging to a rolling element. The slave task will include all the contact force calculations between the rolling elements and the other bodies. The maximum number of processors that can be utilized is limited by the number of rolling elements and is  $n_W + 1$ , where the extra processor is used for the master.
2. Parallelization over the contact force calculations at the *body level*. Every rolling element can be in contact with several bodies and the contact calculations can be done for bodies in parallel for every rolling element. If  $n_B$  is the number of bodies that the rolling elements can interact with, then the maximum number of processors that can be used is  $n_B \cdot n_W + 1$ .
3. Parallelization over the contact force calculations at the *segment level*. A segment is defined in the model as an area where physical contact is possible. Every possible contact that can occur in the bearing is considered for the parallelization. If  $n_S$  is the number of segments each roller is in contact with then the maximum number of processors that can be used is  $n_S \cdot n_W + 1$ .

In order to get good utilization of the parallel computation, a number of factors have to be considered; the amount of work done in serial mode has to be minimized, the communication time between the processors should be short, and the processors should be equally loaded.

**Scheduling**, is the task of distributing the computational load, as evenly as possible, on the different processors of the parallel machine.

This has turned out to be a difficult task [6]. The computational conditions may change during the simulation. It is thus a dynamic optimization problem, where the main influencing factors are:

**The platform** will influence the choice of scheduling strategy. Different computers have different characteristics, e.g., communication bandwidth, latency, and nodes with different speeds.

Additionally, if the machine is a multi-user system (such as a workstation cluster), the computational performance/availability of the processors might vary during the simulation.

**The bearing types** have different “computational” characteristics. This is mainly due to different contact situations. A bearing model having many “large” rolling element/cage contacts will benefit from a large parallel machine. The opposite is true for a bearing model with a few “small” contact segments.

**The application**, described by the bearing and its loading situation is very important. One of the most important parameters (from a scheduling point of view) is the number of rolling elements in the bearing.

Even if a very good constant scheduling strategy could be found with respect to the whole simulation, it may be possible to do better. The explanation for this is that there are phases when there are a lot of contacts, for which a certain scheduling strategy is the best, and there are other phases where there are a few contacts where another scheduling strategy is best. This feature is highly dependent on input data and bearing type.

BEAST uses a two-level scheduling algorithm, where the first level of granularity (see above) is used at the first scheduling level, and the third level of granularity is used at the second scheduling level. The scheduling scheme is continually evaluated and updated during the simulations, using a dynamic optimization method [6].

The speed-up, due to the parallel processors, may vary depending on morphology of the bearing (e.g., number of rolling elements, flange contacts, or type of cage pocket), load case, and computer configuration. A typical speed-up is 60–70% of linear speed-up, where linear speed-up is proportional to the number of processors.

## 3 Simulation example

### 3.1 Input data

BEAST is very versatile when it comes to specifying bearing geometry and loading conditions. Basically any bearing geometry can be specified, including tolerances. The bearing load can be applied either with prescribed displacement, prescribed force, or a combination of the two. All load input can be given as functions of time, making it possible to define any load cycle.

Depending on the application, the inertial properties of the shaft and the housing have to be included in the inner and outer ring, respectively.

The user can also define the lubrication conditions, e.g., the type of oil and temperature.

### 3.2 Output data

The main output from the BEAST simulations are time series of the positions, velocities, and accelerations of all bearing components, as well as the contact forces between the components. The data can be visualized both as 2D plots and as animations.

### 3.3 Simulation verification

BEAST has been validated in many different ways, from single contacts to the whole system. One such test series was carried out on a specially built test rig, named CATRIONA (see Figure 4), in order to determine the forces from the balls on the cage. The tests were conducted on a deep groove ball bearing, 6309 (see Figure 3 and Table 1).

The test set-up is a very specific one, with the inner ring mounted on a hydrostatic spindle, the loads applied on the outer ring via a yoke, and with an instrumented cage mounted on a separate aero-static spindle, colinear with the inner ring spindle (see Figure 5). The cage was mounted on a spindle for two main reasons: it gave the cage a well defined position, and it was easier to instrument the cage. Measurements were done for a number of load cases including pure radial

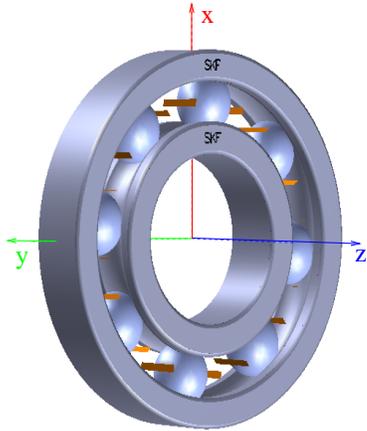


Figure 3: The test bearing, a Deep Groove Ball Bearing 6309 as seen in Beauty, the BEAST viewing tool, with the bearing coordinate system. Only the contacting surfaces of the cage are shown.

load, pure axial load, and combined radial and axial loads (no misalignment), and moment loads, resulting in a misaligned bearing.

### 3.4 Test set-up - Experimental

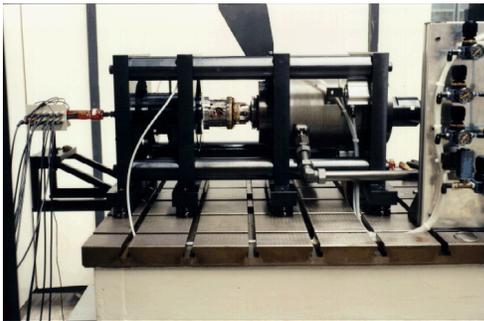


Figure 4: The test rig CATRIONA, with the hydrostatic spindle (right) and aerostatic spindle (left). The outer ring and its loading arrangement is not shown on the picture.

A radial load can be generated using a pneumatic cylinder (pulling upwards) acting on the outer ring of the bearing, whereas an axial load can be applied to the outer ring using three pneumatic cylinders attached at three equally spaced points.

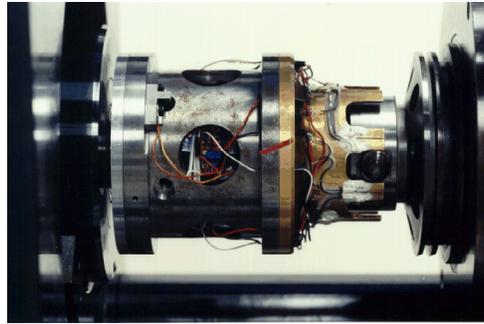


Figure 5: The artificial cage with the instrumented cage bars.

Table 1: Some geometrical parameters for the bearing used (a DGBB 6309).

Number of balls	8
Outer ring outer diameter	0.100 m
Inner ring inner diameter	0.045 m
Bearing width	0.025 m

The ball forces experienced by the cage web were recorded using a data logger operating with two channels, registering the forces on the “front” and “rear” sections of a cage bar, respectively (see Figure 5).

### 3.5 Test set-up - BEAST

BEAST computer simulations of selected test results were carried out. The geometry, materials used, and the loading conditions were all described to the BEAST program so that they would represent as closely as possible the real test conditions. For the individual loads, inner ring speeds, and misalignments applied, see Table 2.

To describe the actual arrangement of the test rig, the inner ring was modelled together with the shaft of the supporting spindle, using stiffness and damping values calculated for the spindle.

The outer ring, being in principle free to move in all directions together with the load yoke, was modelled with the combined mass and moments of inertia.

The cage, finally, was modelled with the inertia, stiffness and damping values of the aerostatic

spindle.

### 3.6 Evaluated tests

During the testing a big number of combinations of load type (radial/axial/combined), load magnitude, and inner ring speeds were tested. Of these tests the ones given in Table 2 are referred to here.

Table 2: Tests and test conditions used for the verification.

Case No.	$n_{IR}$ (r/min)	$F_r$ (N)	$F_a$ (N)	$\Phi_{mis}$ (mrad)
R3	3000	1000	0	0
R5	5800	1000	0	0
A12	3000	126	-903	0
M24	3000	126	-463	1.42

### 3.7 Working Methodology

The results from the simulations were compared to the test results with respect to the cage pocket contact forces. The following parameters were used in the comparison:

- the force level (magnitude),
- the force timing, and
- the nature of the forces.

The simulated cage bar forces are compared to the measured forces in the time domain. Since the time will have arbitrary values, the measured curves have been moved so that they cover the same time domain as the simulations. Both the simulations and the measurements are believed to have run long enough to reach steady-state conditions.

### 3.8 Results

#### 3.8.1 Pure Radial Loading, cases R3 (3000 r/min) and R5 (5400 r/min)

Two radial load conditions are presented here; cases R3 and R5. Both cases have a pure radial load of 1000 N, and zero misalignment. The only difference is the rotational speed of the inner ring.

With a pure radial load there will be one well defined loaded zone of the balls. Since the radial load is acting upwards on the outer ring, the loaded zone will be at the bottom of the bearing.

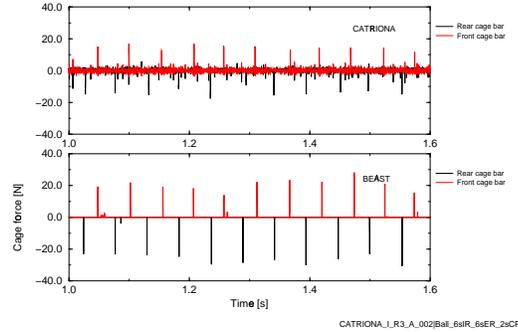


Figure 6: BEAST simulation (bottom) and test signals (top) of cage-ball impact for a bearing running at 3000 r/min under a pure radial load of 1 kN, case R3.

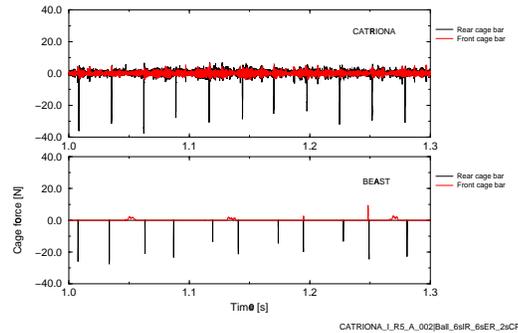


Figure 7: BEAST simulation (bottom) and test signals (top) of cage-ball impact for a bearing running at 6000 r/min under a pure radial load of 1 kN, case R5.

Figure 6 shows the typical force pattern for a bearing under radial load. Before entering the loaded zone, the balls “falls down” and hits the front cage bar. After the loaded zone, when the ball travels upwards, it will loose speed and the cage will hit the ball from behind. The lower the bearing speed, the clearer this mechanism is seen. At a higher speed, as in Figure 7, the cage only hits the ball on the rear cage bar, i.e., after the loaded zone. It is here the relative speed difference between the balls and the cage is largest.

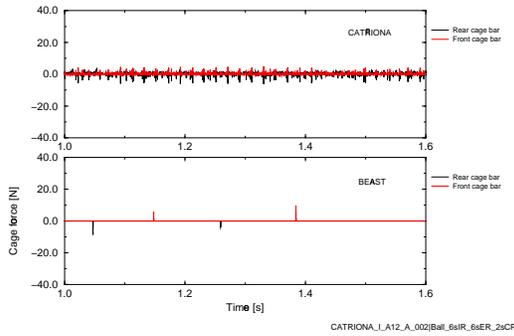


Figure 8: *BEAST* simulation (bottom) and test signals (top) of cage-ball impact for a bearing running at 3000 r/min under a pure axial load of 900 N, case A12.

### 3.8.2 Pure Axial Loading, case A12, 3000 r/min

Figure 8 shows the comparison between test and simulation for a bearing loaded with a pure axial load of 900 N, rotating at 3000 r/min.

In cases with pure axial load all balls will travel with the same speed, which will be the same as the cage speed. No, or very little, cage forces are to be expected. This is also the fact for both the simulations and the measurements.

### 3.8.3 Misaligned Loading, case M24 (3000 r/min)

The misaligned cases have quite a different force pattern. Here the forces are not just impacts, but of longer duration. This is due to the fact that the balls are differently loaded at different positions in the bearing. This will give them different tangential speed. If the difference in tangential position due to the difference in speed is larger than the cage pocket clearance, the balls will come in “conflict” with the cage, i.e., the loaded balls will be in contact with the cage for longer period of time.

In the measurements for these cases it can be seen that the cage bar “springs back” (negative “force” on the front cage bar, positive “force” on the rear cage bar) after a large force. This is an effect of the elasticity of the cage bar, and is not due to a real force.

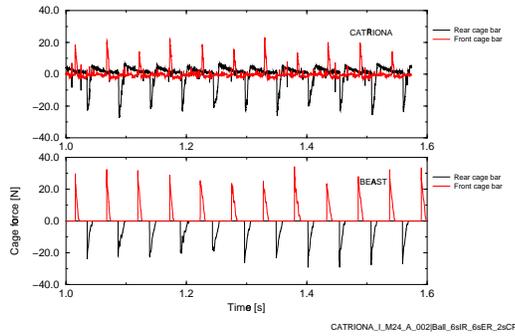


Figure 9: *BEAST* simulation (bottom) and test signals (top) of cage-ball impact for a bearing running at 3000 r/min under a misaligned axial load of 463 N, case M24.

At higher speed misaligned cases, like M24 (3000 r/min), Figure 9, the force peaks starts with an impact, but are to the largest extent friction driven. The impact peaks are not as visible in the measured cases, due to the elasticity of the real cage bars.

## 3.9 Example summary

On the whole there is a very good correlation between the CATRIONA measurements and the BEAST simulations. This is certainly true for the nature of the forces, but also for the force timing and force amplitudes. Impact force, however, tends to be overestimated in BEAST, due to the structurally rigid cage model.

## 4 Conclusions

The rolling bearing simulation model called BEAST is a “virtual test rig”, which makes it possible to study a wide range of performance parameters for rolling bearings. The example above, and the list below, give some insight in what BEAST can be used for:

In normal applications the cage is free to move, and **cage motion** may influence the noise and vibration characteristics of a bearing, as well as the forces acting on the cage. The cage motions calculated by BEAST has also been verified in the

CATRIONA test rig, in another test series.

**Cage pocket forces** are very difficult to measure in real applications, and simulation is therefore a good way to evaluate cage designs under various running conditions. By using BEAST, it may be possible to gain a profound understanding of the mechanisms behind cage behaviour, making it possible to design bearings for increased robustness and longer service life.

BEAST simulations **shortened the time to market** for the new toroidal roller bearing, CARB. The simulations gave the design engineers confidence in CARB performance and helped formulate design rules and application limits. The need for expensive laboratory tests was thereby reduced.

BEAST was used to solve an **application problem** for a large spherical roller thrust bearing. BEAST can be used to give application support, and help convince customers in the choice of bearings and running conditions.

The examples mentioned above show some specific aspects of what BEAST can be used for. More generally, BEAST is being used daily by SKF engineers for:

- increasing of fundamental knowledge of bearings,
- process and product development,
- solving application cases,
- marketing support.

The model utilizes modern multi-processor computers and parallel software, for faster turn-around times.

## Acknowledgments

The authors wish to thank Dr H. Wittmeyer, Director, SKF Group Technology Development, and Aurelio Nervo, Managing Director, SKF Engineering & Research Centre, B.V., for their permission to publish this article.

## References

- [1] A. Bellen and M. Zennaro. *Applied Numerical Mathematics*, volume 11 (1–3). North-Holland, January 1993.
- [2] Kevin Burrage. Parallel methods for initial value problems. *Applied Numerical Mathematics*, 11, 1993.
- [3] Kevin Burrage. *Parallel and Sequential methods for ODEs*. Oxford University Press, 1995.
- [4] Scott D. Cohen and Alan C. Hindmarsh. CVODE User Guide. [www.netlib.org/ode/cvode.tar.gz](http://www.netlib.org/ode/cvode.tar.gz), October 1994.
- [5] D. Fritzson, P. Fritzson, P. Nordling, and T. Persson. Rolling Bearing Simulation on MIMD Computers. *The International Journal of Supercomputer Application and High Performance Computing*, 11(4), 1997.
- [6] P. Fritzson and P. Nordling. Adaptive Scheduling Strategy Optimizer for Parallel Rolling Bearing Simulation. To be presented at HPCN Europe '99, Amsterdam, April 1999.
- [7] C.W. Gear and Xu Xuhai. Parallelism across time in ODEs. *Applied Numerical Mathematics*, 11, 1993.
- [8] P.K. Gupta. *Advanced Dynamics of Rolling Elements*. Springer-Verlag, New York, 1984.
- [9] P.J. van der Houwen and B.P. Sommeijer. Analysis of parallel diagonally implicit iteration of Runge-Kutta methods. *Applied Numerical Mathematics*, 11, 1993.
- [10] C.H. Venner. *Multilevel Solution of the EHL Line and Point Contact Problems*. PhD thesis, Universiteit Twente, The Netherlands, 1991.