Efficiency Evaluation of the Real-time Multibody Analysis with Matrix Libraries

Taichi Shiiba
Department of Mechanical Engineering, Meiji University
1-1-1, Higashi-mita, Tama-ku, Kawasaki, Kanagawa, 214-8571 JAPAN
Email: shiiba@isc.meiji.ac.jp

Naoya Machida
Meiji University Graduate School
1-1-1, Higashi-mita, Tama-ku, Kawasaki, Kanagawa, 214-8571 JAPAN
Email: ce92063@isc.meiji.ac.jp

Multibody dynamics is an effective method to reduce the cost and the period for the development of mechanical products. With the progress of CPU performance, the computational speed of multibody dynamic analysis has been improved, and several applications of real-time multibody analysis such as a driving simulator and a HILS application were proposed. The computational speed of multibody dynamic analysis depends on the number of bodies and constraints and the eigenfrequency of the system to be analyzed, and it is necessary to enhance the computational efficiency to realize real-time analysis of a large size system or of high frequency phenomena. The main part of multibody dynamic analysis is solving the differential algebraic equation (DAE), and it is known that the performance of solving matrix equation can be improved by using matrix libraries.

It is also important to consider the environment of real-time calculation. Even though Windows and Linux are commonly and widely used, the periodicity of the calculation cannot be assured on these operational systems, for they are not real-time operational system (RTOS). In addition, real-time calculation is generally used with some input/output devices, and these devices are used for the control of some hardwares. Hence the use of RTOS is necessary for the periodic and stable real-time calculation and hardware control. Several kinds of RTOSs such as RTLinux and VxWorks have been developed. However, the use of matrix libraries is not taken into account when those RTOSs were developed. Therefore the compatibility and the performance of matrix libraries on RTOSs should be examined carefully for real-time calculation.

In this study, we compared several kinds of matrix libraries on our real-time analysis environment. Our real-time simulation environment of this study was with xPC Target, which is one of the MATLAB products. First, we evaluated the performance of matrix libraries by solving large-size linear equation consisting of random numbers. Next, the matrix libraries were embedded to real-time multibody dynamics code developed by authors and the computational time was evaluated. The simulation model was multiple pendulums model and we investigated relationship between calculation performance and the number of bodies. Finally we analyzed vehicle dynamics as an example of actual multibody system, and the performance of using matrix libraries is discussed.
Efficiency Evaluation of the Real-time Multibody Analysis with Matrix Libraries

Taichi Shiiba
Department of Mechanical Engineering, Meiji University
1-1-1, Higashi-mita, Tama-ku, Kawasaki, Kanagawa, 214-8571 JAPAN
Email: shiiba@isc.meiji.ac.jp

Naoya Machida
Meiji University Graduate School
1-1-1, Higashi-mita, Tama-ku, Kawasaki, Kanagawa, 214-8571 JAPAN
Email: ce92063@isc.meiji.ac.jp

ABSTRACT
In this paper, we compared several kinds of matrix libraries on a real-time analysis environment. The computational speed of multibody dynamic analysis depends on the number of bodies and constraints and the eigenfrequency of the system to be analyzed, and it is necessary to enhance the computational efficiency to realize real-time analysis of a large size system or of high frequency phenomena. Our real-time simulation environment of this study was with xPC Target, which is one of the MATLAB products. First, we evaluated the performance of matrix libraries by solving large-size linear equation consisting of random numbers. Next, the matrix libraries were embedded to real-time multibody dynamics code developed by authors and the computational time was evaluated. The simulation model was multiple pendulums model and we investigated relationship between calculation performance and the number of bodies. Finally we analyzed vehicle dynamics as an example of actual multibody system, and the performance of using matrix libraries is discussed.

1. INTRODUCTION
Multibody dynamics is an effective method to reduce the cost and the period for the development of mechanical products. With the progress of CPU performance, the computational speed of multibody dynamic analysis has been improved, and several applications of real-time multibody analysis such as a driving simulator and a HILS application were proposed, for example, by Kim (2009) and Shiiba (2007 and 2009). The computational speed of multibody dynamic analysis depends on the number of bodies and constraints and the eigenfrequency of the system to be analyzed, and it is necessary to enhance the computational efficiency to realize real-time analysis of a large size system or of high frequency phenomena. The main part of multibody dynamic analysis is solving the differential algebraic equation (DAE), and it is known that the performance of solving matrix equation can be improved by using matrix libraries.

It is also important to consider the environment of real-time calculation. Even though Windows and Linux are commonly and widely used, the periodicity of the calculation cannot be assured on these operational systems, for they are not real-time operational system (RTOS). In addition, real-time calculation is generally used with some input/output devices, and these devices are used for the control of some hardwares. Hence the use of RTOS is necessary for the periodic and stable real-time calculation and hardware control. Several kinds of RTOSs such as RTLinux and VxWorks have been developed. However, the use of matrix libraries is not taken into account when those RTOSs were developed. Therefore the compatibility and the performance of matrix libraries on RTOSs should be examined carefully for real-time calculation.

In this study, we compared several kinds of matrix libraries on our real-time analysis environment. Our real-time simulation environment of this study was with xPC Target, which is one of the MATLAB products. First, we evaluated the performance of matrix libraries by solving large-size linear equation consisting of random numbers. Next, the matrix libraries were embedded to real-time multibody dynamics code developed by authors and the computational time was evaluated. The simulation model was multiple pendulums model and we investigated relationship between calculation performance and the number of bodies. Finally we analyzed vehicle dynamics as an example of actual multibody system, and the performance of using matrix libraries is discussed.

2. REAL-TIME ANALYSIS ENVIRONMENT
The assurance of periodic calculation is the most important factor in real-time analysis. Therefore, some specialized operating system which is originally designed for real-time calculation is required. In our research, xPC target was used as a real-time analysis environment. xPC Target is
one of the MATLAB products that is used together with Real-time Workshop, and several input/output devices can be installed in the target PC in which real-time calculation is running. The configuration of xPC target system is shown in Figure 1, and Table 1 shows the performance of the target PC. By using Real-time Workshop, real-time executable code is generated from Simulink block diagram program. The purpose of this paper is to evaluate the performance of matrix libraries in this environment. Hence the matrix calculation and multibody dynamic analysis for this research is described as S-functions, which is a kind of an user subroutine for Simulink.

![Configuration of xPC Target System](image)

Table 1 SPECIFICATIONS OF THE PC FOR REAL-TIME ANALYSIS

<table>
<thead>
<tr>
<th>Target PC</th>
<th>DELL Dimension 8300</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel Pentium IV 2.80 GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>1024 MB</td>
</tr>
</tbody>
</table>

3. CALCULATION WITH MATRIX LIBRARIES

3.1 Matrix Libraries

Matrix calculations of solving a linear equations and the eigenvalue analysis are the fundamentals in numerical simulations. Various matrix libraries have been developed to enhance the performance of simulations (see references by Anderson (1995), Dongarra (1998), and Gonzalez (2008 and 2009)). There are two main types of matrix libraries. The first type is LAPACK families, and the other is sparse matrix libraries. LAPACK is originally written in Fortran as a replacement of LINPACK and EISPACK, and used together with BLAS (Basic Linear Algebra Subprograms) so that as much as possible of the computation is performed. Several kinds of matrix libraries are derived from LAPACK. In this paper, CLAPACK (http://www.netlib.org/clapack/), CPPLAPACK (http://sourceforge.net/projects/cpplapack/), Boost (http://www.boost.org/) were examined as LAPACK family libraries. Sparse matrix libraries compress the data of sparse matrices by using the indices of non-zero entries. A lot of algorithms for "fill reducing ordering" have been proposed. As sparse matrix libraries, KLU (http://www.cise.ufl.edu/research/sparse/) and SuperLU (http://crd.lbl.gov/~xiaoye/SuperLU/) were evaluated in our research.

3.2 Evaluation of the Calculation Performance with Matrix Libraries in Real-Time Environment

In multibody dynamics simulations, the biggest time consumption is in the process of solving the differential algebraic equations, which contains the calculation of solving a linear equation. Hence we evaluated the performance of solving simple linear equations (1) with several matrix libraries on xPC target environment as a first step of the evaluation.

\[ Ax = b \]

The elapsed time on xPC target environment to solve the linear equation with the matrix size \( N = 50, 100, 250 \) was examined for several sparsities of the matrix \( A \). In these calculations, non-zero entries were filled with random numbers. The calculation of solving the linear equation was repeated for 1000 times, and the average calculation time was evaluated. The results of the average time are shown in Figure 1. The sparsity of the matrix is indicated in these figures, which means the ratio of non-zero entries to the total number of matrix entries. The calculation results without matrix libraries are also shown in these figures as "w/o library", which was obtained with the widely-used LU factorization method. From these results, the dependency of the elapsed time to the sparsity of matrix \( A \) can be observed in the results with every matrix library, especially in the results with KLU that is classified to a sparse matrix library. Meanwhile the elapsed time is almost same in the results without matrix library.

In addition, the elapsed time with the matrix size \( N = 100 \) are compared in Table 2 and Figure 3, for the distribution of the calculation time has a meaning in real-time calculation. The real-time calculation is failed if the calculation time exceeds the step size for real-time calculation, even if the average elapsed time is low. In Figure 3, the distributions of the difference from the average elapsed time were plotted. It is shown that the average elapsed time with CLAPACK was quite low in Table 2. However, the elapsed time with CLAPACK was widely distributed as shown in Figure 3 (i), which is undesirable for real-time calculation. Thus the performance of the matrix libraries should be discussed with not only the average value of the elapsed time, but also its distribution.
Figure 2 ELAPSED TIME

Table 2 ELAPSED TIME TO SOLVE LINEAR EQUATION (N=100)

(i) SPARSITY: 10 %

<table>
<thead>
<tr>
<th></th>
<th>Ave.</th>
<th>Min.</th>
<th>Max.</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLAPACK</td>
<td>0.8142</td>
<td>0.6727</td>
<td>1.9045</td>
<td>0.1598</td>
</tr>
<tr>
<td>CPPLAPACK</td>
<td>0.7586</td>
<td>0.6932</td>
<td>1.0788</td>
<td>0.1198</td>
</tr>
<tr>
<td>KLU</td>
<td>1.2797</td>
<td>1.2612</td>
<td>1.3262</td>
<td>0.0136</td>
</tr>
<tr>
<td>SuperLU</td>
<td>1.3156</td>
<td>1.3015</td>
<td>1.3535</td>
<td>0.0113</td>
</tr>
<tr>
<td>Boost</td>
<td>2.6544</td>
<td>2.6228</td>
<td>2.6912</td>
<td>0.0135</td>
</tr>
<tr>
<td>w/o library</td>
<td>1.8678</td>
<td>1.7671</td>
<td>2.0305</td>
<td>0.0320</td>
</tr>
</tbody>
</table>

Unit: ms

(ii) SPARSITY: 100 %

<table>
<thead>
<tr>
<th></th>
<th>Ave.</th>
<th>Min.</th>
<th>Max.</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLAPACK</td>
<td>0.9318</td>
<td>0.9017</td>
<td>2.5114</td>
<td>0.1251</td>
</tr>
<tr>
<td>CPPLAPACK</td>
<td>0.9731</td>
<td>0.9241</td>
<td>1.6534</td>
<td>0.1138</td>
</tr>
<tr>
<td>KLU</td>
<td>3.1032</td>
<td>3.0732</td>
<td>3.2201</td>
<td>0.0218</td>
</tr>
<tr>
<td>SuperLU</td>
<td>1.5712</td>
<td>1.5410</td>
<td>1.6130</td>
<td>0.0180</td>
</tr>
<tr>
<td>Boost</td>
<td>2.9080</td>
<td>2.8686</td>
<td>2.9603</td>
<td>0.0166</td>
</tr>
<tr>
<td>w/o library</td>
<td>1.8688</td>
<td>1.7893</td>
<td>2.1025</td>
<td>0.0340</td>
</tr>
</tbody>
</table>

Unit: ms

Figure 3 HISTOGRAM OF THE ELAPSED TIME (N=100)
4. CALCULATION PERFORMANCE EVALUATION IN MULTIBODY DYNAMICS PROBLEM

4.1 Multiple Pendulums

The calculation performance of matrix libraries in practical real-time multibody dynamics was evaluated in this chapter. Three dimensional multiple pendulums problem (Figure 4) was examined as a first example. Each body is connected by ball joint element. The number of pendulums can be modified in this problem, and the change of elapsed time to the number of the pendulums N was evaluated.

The differential algebraic equation of a three dimensional multibody system can be written with Euler parameters as follows (Haug (1989)):

\[
\begin{bmatrix}
M & 0 & \Phi^p_r & 0 & 0 \\
0 & 4G^TJ^*G & \Phi^p_r & 0 & 0 \\
\Phi^p_r & 0 & 0 & 0 & 0 \\
0 & \Phi^p_r & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
\dot{r} \\
\dot{p} \\
\lambda^e \\
\lambda^p \\
\end{bmatrix}
= \begin{bmatrix}
F^d \\
G^Tn^d + 8G^TJ^*Gp \\
\gamma \\
\gamma^p \\
\end{bmatrix}
\]  \tag{2}

where

- \(M\) : Mass matrix
- \(J^*\) : Inertia tensor
- \(F^d, n^d\) : External force and torque
- \(r\) : Position of centers of gravity
- \(p\) : Euler parameters
- \(\lambda^e, \lambda^p\) : Lagrange multipliers
- \(\Phi^p_r, \Phi^p_p\) : Jacobian of constraint equations
- \(\gamma, \gamma^p\) : Acceleration equations of constraint

The matrix size in this equation is shown in Table 3, and the sparsity of the coefficient matrix in Eq. (2) is plotted in Figure 5. The ratio of non-zero elements in the coefficient matrix decreases as the number of pendulums increases.

Numerical integration with this equation was executed on our xPC target system. The step size of real-time calculation was fixed to 5 ms, and Adams-Bashforth-Moulton method was used for the numerical integration. The elapsed time in each time step was recorded. The result is shown in Figure 6. In the result of CPPLAPACK, task overrun error was observed at N = 20, even though the average elapsed time did not have big difference to CLAPACK, KLU, and SuperLU at N = 16. It would appear that the wide distribution of the elapsed time of CPPLAPACK caused this result. In addition, the statistics of the results at N = 7 for 10 second simulation is compared in Table 4 and Figure 7.

Table 3  MATRIX SIZE IN DIFFERENTIAL ALGEBRAIC EQUATION

<table>
<thead>
<tr>
<th>Number of pendulums N</th>
<th>Number of bodies N+1</th>
<th>Number of generalized coordinates 7(N+1)</th>
<th>Number of constraint equation 4N+7</th>
<th>Number of rows of coefficient matrix 11N+14</th>
</tr>
</thead>
</table>

![Figure 5 THE SPARSITY OF COEFFICIENT MATRIX](image)

Table 4  ELAPSED TIME

<table>
<thead>
<tr>
<th></th>
<th>Ave.</th>
<th>Min.</th>
<th>Max.</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLAPACK</td>
<td>0.2271</td>
<td>0.2177</td>
<td>0.3314</td>
<td>0.0131</td>
</tr>
<tr>
<td>CPPLAPACK</td>
<td>0.3360</td>
<td>0.3124</td>
<td>0.3791</td>
<td>0.0073</td>
</tr>
<tr>
<td>KLU</td>
<td>0.4833</td>
<td>0.4378</td>
<td>0.4918</td>
<td>0.0025</td>
</tr>
<tr>
<td>SuperLU</td>
<td>0.4927</td>
<td>0.4793</td>
<td>0.5072</td>
<td>0.0039</td>
</tr>
<tr>
<td>Boost</td>
<td>3.2923</td>
<td>3.2070</td>
<td>3.3465</td>
<td>0.0270</td>
</tr>
<tr>
<td>w/o library</td>
<td>0.9607</td>
<td>0.9248</td>
<td>1.2071</td>
<td>0.0241</td>
</tr>
</tbody>
</table>

Unit: ms

Copyright (c) 2010 by JSME
4.2 Vehicle dynamics simulation

As a practical example of a multibody system, the vehicle running simulation was examined. The target vehicle for this simulation is shown in Figure 8, which was developed in our laboratory for various purposes of vehicle dynamics research. The numbers of bodies, constraints, matrix size and the sparsity in DAE are shown in Table 5. The steering input and the response of the yaw rate of the body was shown in Figure 9.

In this simulation, Adams-Bashforth method was chosen as a numerical integration solver, and the real-time calculation was achieved under the condition of the step size of 2 ms. The result with Boost is not discussed in this section, for the task over run error was observed in the simulation with Boost. The results of the elapsed time are compared in Table 6 and Figure 10. It should be noted that the value of the sparsity of the coefficient matrix in this example was not as small as in three dimensional pendulums problem in the previous section. Therefore, the advantage of the sparse libraries like KLU and SuperLU is small in the meaning of the average elapsed time. However, another advantage of these sparse libraries that the distribution of the elapsed time is narrow was observed in the results of standard deviation (SD) in Table 6.
Table 6 ELAPSED TIME

<table>
<thead>
<tr>
<th></th>
<th>Ave</th>
<th>Min</th>
<th>Max</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLAPACK</td>
<td>0.7720</td>
<td>0.7405</td>
<td>0.7916</td>
<td>0.0062</td>
</tr>
<tr>
<td>CPPLAPACK</td>
<td>0.8778</td>
<td>0.8367</td>
<td>0.9723</td>
<td>0.0059</td>
</tr>
<tr>
<td>KLU</td>
<td>1.2587</td>
<td>1.1924</td>
<td>1.2767</td>
<td>0.0034</td>
</tr>
<tr>
<td>SuperLU</td>
<td>1.1721</td>
<td>1.1565</td>
<td>1.1881</td>
<td>0.0044</td>
</tr>
<tr>
<td>w/o library</td>
<td>1.8787</td>
<td>1.8446</td>
<td>1.9866</td>
<td>0.0128</td>
</tr>
</tbody>
</table>

Unit: ms

Figure 10 HISTOGRAM OF THE ELAPSED TIME (VEHICLE MODEL)

5. CONCLUSION

In this study, we evaluated the performance of several kinds of matrix libraries on a real-time analysis environment with xPC target. From the fundamental matrix calculations and some examples of multibody dynamics simulation, a tendency of each matrix libraries can be figured out. The tendency is that CLAPACK and CPPLAPACK, which are categorized in the LAPACK family libraries, have an advantage in the average elapsed time, and the sparse matrix libraries like KLU and SuperLU has another advantage that the distribution of the elapsed time is small. Both advantages are important for real-time calculation, so it is not easy to conclude which matrix library is the best for real-time multibody analysis. However, it is obvious that the use of matrix libraries enhance the performance of real-time calculation, so the application of matrix libraries should be considered when developing a real-time multibody analysis environment.

REFERENCES


