



Pasi Korkealaakso

**REAL-TIME SIMULATION OF MOBILE AND
INDUSTRIAL MACHINES USING THE
MULTIBODY SIMULATION APPROACH**

*Thesis for the degree of Doctor of Science (Technology) to
be presented with due permission for public examination and
criticism in the Auditorium 1382 at Lappeenranta University
of Technology, Lappeenranta, Finland on the 18th of September,
2009, at noon.*

Acta Universitatis
Lappeenrantaensis
347

- Supervisor Professor Aki Mikkola
Faculty of Technology
Department of Mechanical Engineering
Lappeenranta University of Technology
Finland
- Reviewers Professor Javier Cuadrado
Mechanical Engineering Laboratory of the Department of Industrial
Engineering
University of La Coruña
Spain
- Professor Evtim Venets Zahariev
Department of Dynamics and Optimization of Controlled Mechanical Systems
Bulgarian Academy of Sciences
Bulgaria
- Opponents Professor Javier Cuadrado
Mechanical Engineering Laboratory of the Department of Industrial
Engineering
University of La Coruña
Spain
- Professor José Luis Escalona Franco
Group of Mechanical Engineering of the Department of Mechanical and
Materials Engineering
University of Seville
Spain

ISBN 978-952-214-791-2

ISBN 978-952-214-792-9 (PDF)

ISSN 1456-4491

Lappeenrannan teknillinen yliopisto

Digipaino 2009

ABSTRACT

Pasi Korkealaakso

Real-Time Simulation of Mobile and Industrial Machines Using the Multibody Simulation Approach

Lappeenranta, 2009

58 p.

Acta Universitatis Lappeenrantaensis 347

Diss. Lappeenranta University of Technology

ISBN 978-952-214-791-2, ISBN 978-952-214-792-9 (PDF)

ISSN 1456-4491

This thesis introduces a real-time simulation environment based on the multibody simulation approach. The environment consists of components that are used in conventional product development, including computer aided drawing, visualization, dynamic simulation and finite element software architecture, data transfer and haptics. These components are combined to perform as a coupled system on one platform. The environment is used to simulate mobile and industrial machines at different stages of a product life time. Consequently, the demands of the simulated scenarios vary. In this thesis, a real-time simulation environment based on the multibody approach is used to study a reel mechanism of a paper machine and a gantry crane. These case systems are used to demonstrate the usability of the real-time simulation environment for fault detection purposes and in the context of a training simulator.

In order to describe the dynamical performance of a mobile or industrial machine, the nonlinear equations of motion must be defined. In this thesis, the dynamical behaviour of machines is modelled using the multibody simulation approach. A multibody system may consist of rigid and flexible bodies which are joined using kinematic joint constraints while force components are used to describe the actuators. The strength of multibody dynamics relies upon its ability to describe nonlinearities arising from wearing of the components,

friction, large rotations or contact forces in a systematic manner. For this reason, the interfaces between subsystems such as mechanics, hydraulics and control systems of the mechatronic machine can be defined and analyzed in a straightforward manner.

Keywords: flexible multibody systems, real-time simulation, fault detection, kinematic joints, floating frame of reference.

UDC 531.36 : 004.94 : 62-231

PREFACE

This thesis was prepared during 2003-2009 in the Institute of Mechatronics and Virtual Engineering at the Department of Mechanical Engineering of Lappeenranta University of Technology. The work was done as a part of several research projects financed by the Academy of Finland and the National Technology Agency of Finland (TEKES).

A number of people who have influenced this work deserve a special acknowledgement for their help in the preparation of this thesis. First of all, I want to thank my supervisors: Professor Aki Mikkola for your patience, guidance and advice during this long journey, and Professor Heikki Handroos for providing a number of the research projects in which to prepare this thesis.

Many thanks to Dr. Asko Rouvinen who I would like to call my third supervisor. Asko has given me valuable advice and had interesting discussions with me throughout this process. I am very grateful to Tero Eskola, Sami Moisio and Jani Peusaari for their cooperation in several projects. I would also like to thank all of the people in the Virtual Engineering laboratory and personnel of MeVEA Ltd.

I would like to thank my reviewers, Professor Evtim Zahariev from the Bulgarian Academy of Sciences, Bulgaria, and Professor Javier Cuadrado from the University of La Coruna, Spain, for their valuable comments.

I greatly appreciate the financial support provided by the Finnish Cultural Foundation, Walter Ahlström Foundation, Jenny and Antti Wihuri Foundation, Lauri ja Lahja Hotisen rahasto, Kaupallisten ja teknillisten tieteiden tukisäätiö – KAUTE, and the Ford Foundation.

The people who have supported me for my whole live deserve special thanks – my parents, sister and brother. I wish to express my gratitude to Annina, my wife, for her continued support and encouragement. And finally, Moona, my daughter, thank you for understanding that I had to work long days to complete this thesis.

Lappeenranta, August 2009

Pasi Korkealaakso

CONTENTS

LIST OF PUBLICATIONS

NOMENCLATURE

1	INTRODUCTION.....	13
1.1	Overview of multibody system dynamics	13
1.2	Real-time simulation environment	18
1.3	Contribution of the thesis	21
2	MODELING OF MULTIBODY SYSTEMS USING THE REFERENCE	
	FRAME APPROACH.....	23
2.1	Spatial kinematics of a flexible body	23
2.2	Virtual work.....	26
2.2.1	Integration of the equations of motion.....	28
2.3	Description of multibody equations of motion	29
2.3.1	Method of Lagrange multipliers.....	29
2.3.2	Augmented Lagrangian method	30
2.3.3	Method based on projection matrix	31
2.4	Kinematic joint description.....	33
2.4.1	Basic Constraints	33
2.4.2	Modeling of Joints Based on Basic Constraints	37
3	A DESCRIPTION OF A REAL-TIME SIMULATION ENVIRONMENT	40
3.1	Dynamics module.....	41
3.1.1	Description of external force components	42
3.1.2	Collision detection library.....	43
3.2	Control interface.....	44
3.3	Visualization	45
3.3.1	Visualization library	45
3.3.2	Physical visualization system.....	46
3.4	Server library	48
3.5	Model definition.....	50

3.5.1	Using an XML parser in model definition	50
4	CONCLUSIONS	52
5	REFERENCES	54
6	APPENDICES	58

LIST OF PUBLICATIONS

The thesis consists of following scientific journal articles.

- I Korkealaakso P., Mikkola, A., Rantalainen, T., Rouvinen, A., 2009, "Description of Joint Constraints in the Floating Frame of Reference Formulation", *Journal of Multi-body Dynamics*, **223**(2), pp. 133-145.
- II Korkealaakso, P., Rouvinen, A., Moisio, S. and Peusaari, J., 2007, "Development of a Real-Time Simulation Environment" *Multibody System Dynamics*, **17**(2-3), pp. 177-194.
- III Korkealaakso, P., Rouvinen, A., Mikkola, A., 2006, "Multibody Approach for Model-Based Fault Detection of a Reel", *Journal of Computational and Nonlinear Dynamics*, **1**(2), pp. 116-122.
- IV Korkealaakso, P., Mikkola, A., Rouvinen, A., 2006, "Multi-Body Simulation Approach for Fault Diagnosis of a Reel", *Journal of Multi-body Dynamics*, **220**(1), pp. 9-19.
- V Rouvinen, A., Lehtinen, T., Korkealaakso, P., 2005, "Container Gantry Crane Simulator for Operator Training", *Journal of Multi-body Dynamics*, **219**(4), pp. 325-335.

NOMENCLATURE

Symbols

A	rotation matrix
\mathbf{A}_f^P	rotation matrix describing orientation due to deformation at the location of particle P
B	transformation matrix from velocities of generalized coordinates to velocities of independent generalized coordinates
C	vector of kinematic constraint equations
\mathbf{C}_q	constraint Jacobian matrix
\mathbf{d}^{ij}	vector from P^i to P^j defined in a global coordinate system
\mathbf{F}^P	external force per unit mass
F_j	j -th force component acting on body
$\overline{\mathbf{G}}$	velocity transformation matrix between angular velocities and first time derivative of Euler parameters
I	(3×3) identity matrix
K	modal stiffness matrix
\mathbf{q}	vector of generalized coordinates
\mathbf{q}_d	vector of dependent generalized coordinates
\mathbf{q}_i	vector of independent generalized coordinates
\mathbf{q}_f	vector of elastic coordinates
m	number of constraint equations
M	mass matrix
n	number of generalized coordinates
P	particle in body
\mathbf{p}	vector of integrable generalized coordinates
\mathbf{Q}^c	vector of velocity dependent terms due to differentiation of constraint equations
\mathbf{Q}^e	vector of generalized forces
\mathbf{Q}^f	vector of elastic forces

\mathbf{Q}^v	vector of quadratic velocity inertia terms
\mathbf{r}^P	position vector of particle P in a global coordinate system
\mathbf{R}	position vector of the frame of reference
\mathbf{R}	velocity transformation matrix
t	time
$\bar{\mathbf{u}}_j$	position vector of j -th force component within the frame of reference
$\bar{\mathbf{u}}^P$	position vector of particle P within the frame of reference
$\bar{\mathbf{u}}_0^P$	position vector of particle P within the frame of reference in undeformed state
$\bar{\mathbf{u}}_f^P$	displacement of particle P within the frame of reference due to the deformation
$\bar{\mathbf{v}}$	vector within the frame of reference of body in undeformed state
$\bar{\mathbf{v}}_f$	vector within the frame of reference of body in deformed state
\mathbf{v}_f	vector within the frame of reference of body in deformed state in a global coordinate system
\mathbf{v}_f^1	vector within the frame of reference of body in deformed state in a global coordinate system used in definition of revolute, cylindrical and translational joints
\mathbf{v}_f^2	vector within the frame of reference of body in deformed state in a global coordinate system used in definition of revolute, cylindrical and translational joints
V	volume of body
W^i	work of inertial forces
W^e	work of externally applied forces
W^s	work of elastic forces

Greek letters

$\boldsymbol{\alpha}$	matrix of penalty terms
$\boldsymbol{\gamma}$	acceleration of generalized coordinates with zero acceleration for independent generalized coordinates
δ	partial differential operator of calculus
$\boldsymbol{\varepsilon}^P$	vector of small rotations due to deformation
$\boldsymbol{\theta}$	general rotation vector

θ^E	vector of Euler parameters
θ_j^E	j -th component of the vector of Euler parameters
λ	vector of Lagrange multipliers and vector of penalty forces
λ^*	vector of penalty forces
μ	matrix of fictitious damping ratios
ρ	density of body
Ψ_R^P	modal matrix whose columns describes translation of particle P in assumed deformation modes
Ψ_θ^P	modal transformation matrix whose columns describes rotation coordinates of point P in assumed deformation modes
Ψ_j	modal matrix associated with the node to which the j -th force component applies
$\bar{\omega}$	vector of local angular velocities
Ω	matrix of fictitious natural frequencies

Superscripts

i	Index of the body
j	Index of the body
T	Transpose of vector or matrix

1 INTRODUCTION

Simulation is an abstract theme which can be used to describe an imitative action of a real system. In this study, simulation is comprised of a computer-aided approach to analyze complex mechanical systems such as mobile and industrial machines. A common feature of these machines is that they include mechanical components as well as various actuators and control schemes. In order to simulate a mechanical system using computers, a mathematical description of the system – a simulation model – needs to be formulated. The simulation model may include sub-models such as hydraulic, pneumatic or electrical drives. These actuators are usually important in terms of the dynamic performance of the machines. In order to provide activation commands to the functions of a simulation model, a user interface and control system need to be implemented. Further, by adding a visualization system and motion platform and taking care of computing the simulation model in real time, the entire system can be defined as a real-time simulation environment, as depicted in Fig. 1.

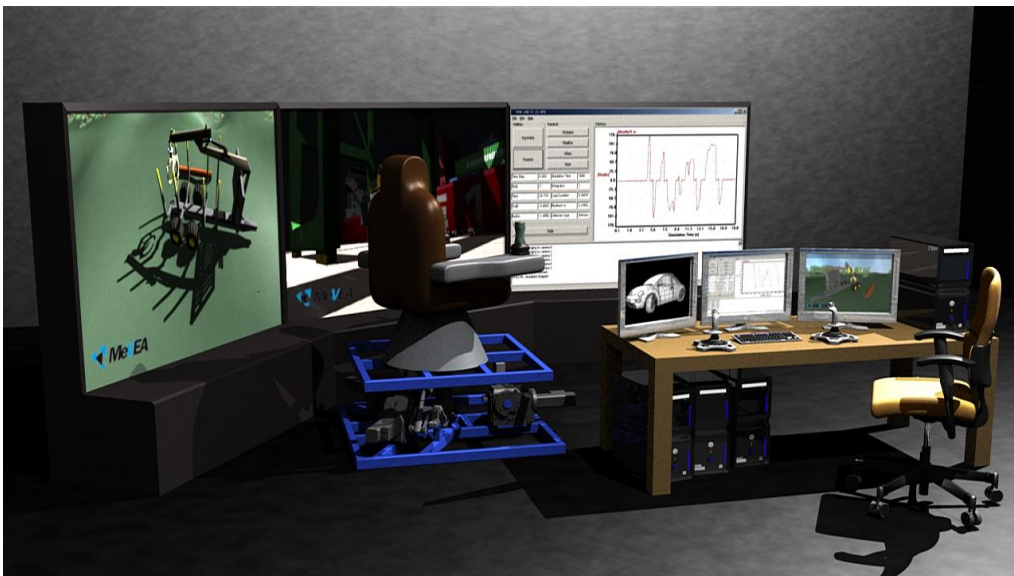


Figure 1. The coupled real-time simulation environment.

1.1 Overview of multibody system dynamics

A multibody system consists of rigid and flexible bodies, joint constraints that couple the bodies, and power components describing dampers, springs and actuators. Depending on the components needed for the multibody model, the dynamic behavior of the system can be

described by a system of equations consisting of differential and nonlinear algebraic equations. In a historical timeline, multibody system analysis has been developed based on the achievements of classical mechanics, which is generally divided into two branches. In the first branch, which can be referred to as the direct approach to dynamics, force and momentum are considered as the primary parameters in differential equations of motion. This form of dynamical equation can be directly derived by employing the approach of Newton and Euler. The second branch is called the indirect or variational approach where forces that perform no work can be neglected. D'Alembert studied a set of rigid bodies introducing the concept of virtual work. In order to make the concept mathematically consistent, Lagrange utilized the results of d'Alembert, making possible the systematic analysis of a constrained particle system. Subsequently, the invention of digital computers made it possible to reformulate these achievements, leading to multibody formalisms in the 1960s [1]. Probably the best-known method in the field of multibody dynamics is the method of Lagrange multipliers, which can be derived from the variational approach. When Newton-Euler equations are used, the linear and angular momentum principles can be utilized directly in formulating equations of motion, whereas the free body principle can be used to solve the reaction forces due to the constraints. However, the use of free body diagrams in large systems is laborious, making the approach vulnerable to human error. Fortunately, the Newton-Euler equations can be derived from the Lagrange equation using the variational approach and the centroidal body reference frame. Accordingly, constraints can be taken into account by applying the Lagrange multiplier theorem to the variational form of Newton-Euler equations [1].

Flexible multibody dynamics

Multibody dynamics analyses frequently require that structural flexibility is accounted for in order to reliably predict the dynamic behavior of slender structures under a heavy load. It is noteworthy that even though the topological structure of models remains unchanged in the case of rigid and flexible bodies, the modeling of systems with flexible bodies is remarkably challenging regardless of the method used for describing the flexibility [2].

Common techniques to describe the elasticity of the bodies are the lumped mass technique and the floating frame of reference formulation. In the lumped mass technique, the body is

divided into rigid segments which are interconnected by force elements. The method is easy to implement in simulation software based on the multibody approach due to the fact that each segment can be treated as a rigid body. However, after segmentation, each flexible body contains several rigid bodies increasing the degrees of the freedom of the system. In practice, the method can be used to describe beam type bodies. In this thesis, structural flexibility is accounted for by using the floating frame of reference formulation. In the method, the generalized coordinates that define the configuration of the flexible body can be divided into ones that describe the position and orientation of the reference coordinate system and ones that describe deformations with respect to the reference coordinate system. In the floating frame of reference formulation, deformations are usually described using methods based on the finite element approach. The first general purpose implementation of the floating frame of reference formulation applicable to large flexible multibody systems in planar cases was introduced by Song and Haug [3]. They used nodal coordinates from finite element discretization to describe deformations. Nevertheless, in that study, the implementation was cumbersome especially for geometrically complex bodies, leading to computationally expensive equations of motion due to a need for a large number of nodal coordinates. To reduce the number of coordinates related to flexibility, Shabana [4] extended the floating frame of reference formulation to three-dimensional mechanisms, and proposed the use of component mode synthesis to extract the structural vibration modes. In this way, the set of nodal coordinates from the finite element method can be replaced by a lower number of modal coordinates, making the numerical solution of the equations of motion more efficient. However, the general purpose application of the approach was impeded because elements used in the modeling of flexible bodies were included in the solution algorithm leading to element-specific volume integrals to be solved. Yoo and Haug [5, 6] introduced the use of static correction modes in order to account for local deformations due to joint constraints and force components. The advantage of the method is that it allows vibration and static correction modes to be solved directly using commercial finite element software.

Real-time multibody dynamics

Real-time simulation can be defined as a special case of conventional simulation. In the case of real-time simulation, the software modules must be able to process all actions according to predetermined time requirements. In order to influence the simulation, the software modules

need to have a time synchronous connection to the real world. The connection can be accomplished, for example, using data from external devices (Hardware-in-the-Loop simulation – HIL) or by visual observation (Man-in-the-Loop simulation – MIL).

Generally, the equations of motion can be formulated using either the topology based approach or the global approach. The approaches differ in the choice of generalized coordinates used in the description of the system configuration. The topology based approach employs relative coordinates, which allows the kinematic analysis to be accomplished recursively by studying one body at a time in a kinematic chain. The number of generalized coordinates required in the approach is equal to the number of degrees of freedom in open kinematic chains of the system. Impeding the approach is the fact that closed kinematic chains must be opened before kinematic analysis by removing the necessary number of joint constraints. Removed joint constraints must be taken into consideration in the solution of dynamic responses. The method leads to strongly nonlinear equations of motion that may be difficult to represent in a general form. On the other hand, the matrices to be solved remain small, which often makes the method computationally efficient. A general purpose algorithm for solving rigid body systems using the topological approach was introduced by Kim [7]. He used global coordinates to describe the system, while the solution itself was achieved using coordinates that describe the degrees of freedom of joints. This was accomplished by mapping global variables into joint variables using the velocity transformation matrix [8]. A similar approach for natural coordinates was introduced by García de Jalón and Bayo [9]. Chang and Shabana [10, 11] derived the recursive velocity transformation equations to flexible multibody systems, but they did not demonstrate a systematic approach to execute the velocity transformation. A systematic approach to obtain the velocity transformation matrix for flexible multibody systems was proposed by Lee [12]. In global methods, generalized coordinates are used to describe the position, orientation and state of deformation of each body. In order to couple the bodies together, the kinematic joints are defined in terms of constraint equations that are functions of the generalized coordinates. Consequently, the equations formulated for each body are of the same form, leading to the systematic assembly of equations of motion for the entire system. The disadvantage of this method is that it leads to large systems of equations due to a large number of generalized coordinates, and for this reason the method may be computationally inefficient. However, it has been perceived that

global methods may be more efficient than topological methods in the solution of systems consisting of less than 50 generalized coordinates [13].

Constraint modeling

Creating a general-purpose multibody algorithm that takes structural flexibility into account is a challenging endeavor. One of the most difficult tasks in the implementation is to create a component library, which is needed for taking kinematic joint constraints into consideration. References [14, 15] introduce an approach which models joint constraints by using virtual bodies. In this approach, the constraint equations are developed between massless rigid bodies. The advantage of this approach is its applicability to be used in different descriptions of flexibility. On the other hand, adding virtual bodies increases the computation time compared to methods which derive joint constraints individually for each approach to describing flexibility. The formulation of kinematic joints composed of simple basic constraints in the case of systems of rigid bodies has been discussed in References [16, 17]. The basic constraint equations for modeling spherical, universal and revolute joints between flexible bodies have been presented in Reference [5]. Shabana [18, 19] has introduced an approach based on intermediate body fixed joint coordinate systems which are rigidly attached to joint definition points. In this approach, the joint coordinate systems are used to derive basic constraint equations including sliding joints with the assumption that the joint axis can be described as a rigid line. Cardona [20] has introduced the finite element approach for mechanical joints, which can be integrated into finite element software. In Reference [21], the basic joint constraints were used in the context of topological multibody formulation. Hwang [22] has presented basic constraint types used with translational joint models which account for the deformation of the axis line. Hwang used the floating frame of reference approach accounting for multiple contact points, whereas the numerical results are only shown in the case of a single contact point.

In order to be able to employ traditional solvers for the Ordinary Differential Equation (ODE) within the system of equations, the constraint equations must be differentiated twice with respect to time. It is important to note that in previous literature, the terms of the Jacobian matrix and terms that are related to second time differentials of basic constraint equations are

not explicitly presented. In order to alleviate the development of modular simulation, the components that are required to take constraints into account need to be obtained.

1.2 Real-time simulation environment

In most cases, the traditional simulation methods used in the product development processes are free from solution time restrictions. Accordingly, the simulation of a few seconds is allowed to take several hours of real time. In these systems, the control signals of the simulated system must be pre-defined and, for this reason, user interaction is described more or less experimentally based on measured data. When the simulation is executed in synchrony with real-time, the operator can produce a control signal during simulation. Real-time solution requirements often force to simplify the simulation model. In practise, the real-time model can be considered as a trade-off between efficiency and accuracy.

Real-time simulation environments are complicated systems consisting of several different engineering disciplines. Developing a real-time simulation environment requires considering various aspects, including modeling, numerical methods, computer science and programming, and control and automation engineering. The components needed in real-time simulation environments can be categorized into three main fields: functional model components, immersion related components and simulator description components. The functional model includes the dynamical model of a machine in its operational environment. Immersion related components contain, for example, visualization and audio system haptic devices and a motion platform. Finally, simulator description components are used to connect different areas and to define the simulator environment. Each above-mentioned set consists of several submodules with well-defined data-transfer interfaces.

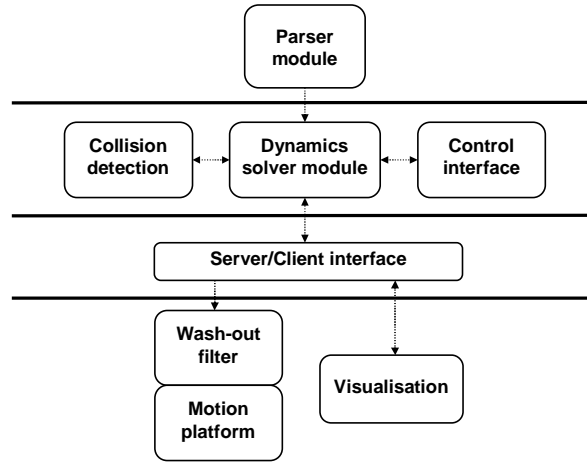


Figure 2. The modular structure of the real-time simulation environment.

This work introduces a modular real-time simulation environment that is depicted in Fig. 2. In the real-time simulation environment introduced, the separate modules are categorized into four levels according to Fig. 2. The first level is the model definition level. The model is defined utilizing XML-based files which can be read using the parser module. The second level comprises the dynamics solver module containing the control interface to enable operator interaction and the separate collision detection module. The modular design has been applied throughout the entire software architecture. The core of the real-time dynamics solver consists of two static libraries: the solver library of numerical algorithms and the modeling library of the formulations of dynamics equations. At the third level, the interface between the solver and the visualization modules as well as the motion platform module at the fourth level are defined. The visualization module provides a Graphical User Interface (GUI) to control model parameters during simulation via a client-server interface. Due to the well defined interfaces between the modules, the real-time simulation environment enables the rapid and straightforward implementation of new components by updating individual function libraries.

The modular structure used in the real-time simulation environment enables the use of distributed computing, allowing the allocation of independent modules to separate computers. The use of distributed computing provides computational resources for the solver, as it is executed on an independent processor. At the moment, the environment consists of two

computers, one for the graphics engine and the other for the solver core. The graphics engine is implemented on a Windows operating system, while the solver core is portable to Windows and Linux operating systems. The communication between these components is implemented through standard network sockets. The same approach may be taken to distribute more subsystems of the simulator into additional nodes. This approach requires that the subsystems are independent of each other. In addition, the processing power required by the subsystems dictates whether multiple nodes can be used effectively.

The solver module is implemented using ANSI C. This enables portability to different operating systems and computer architectures. The modeling library includes several combinations of multibody formulations for rigid and flexible bodies including both the Newton-Euler and Euler parameter forms of constrained equations of motion. The Newton-Euler equations are, however, preferable in order to reduce the number of velocities and accelerations of the generalized coordinates. In addition, the equations of motion may be simplified and a constant mass matrix may be obtained, resulting in a more efficient solution. Kinematic joint constraints have been taken into consideration in the differential equations using either Lagrange multipliers or penalty functions. In order to describe the system with the minimum set of differential equations, the projection matrix from the independent generalized coordinates to the dependent ones can be solved by partitioning the generalized coordinates.

The graphics engine for the real-time simulation environment is implemented using C++, which allows the flexible addition of features. The engine is based on the OpenGL library for graphics environments and the OpenAL library for audio environments. The OpenGL Utility Toolkit (GLUT) simplifies the use of projection matrices and the positioning of the camera point. The optional stereoscopic view uses the OpenGL quad buffer feature. The graphics are imported using the 3ds file format, which enables the efficient pre-processing of the graphics objects in external software. Another important feature of the 3ds file format is its structure, which consists of object related meshes that are based on triangle polygons. A single object may have several meshes, and consequently, the structure may be used in collision detection. An efficient collision detection tree can be obtained by considering the collision detection

during the design of the 3ds graphics. Moreover, some of the non-colliding features can be eliminated already at the trunk level of the collision detection tree.

1.3 Contribution of the thesis

This thesis introduces a design engineering approach to the implementation of a real-time simulation environment of machines systems. In the real-time simulation environment, mechanical structures including flexibility as well as hydraulic actuators with associated control schemes can be described. The simulation environment introduced in this study includes a user interface with appropriate visualization and a motion platform. The real-time simulation environment introduced can be utilized in the following stages of a product life cycle: operator training, product development and failure analysis. The main contribution of the thesis can be further divided into the following sub-studies:

Fault detection of a reel using the multibody simulation approach

Using the multibody simulation approach for fault detection purposes, the supervision of a machine can be focused on the functionality of the entire process instead of an individual component. In this thesis, the multibody system simulation approach is applied to the fault detection of the reel mechanism of a paper machine. Due to the requirements of real-time computing, different multibody formulations are compared, and the most appropriate one can be chosen for each case. This original scientific contribution has been published in the following journal papers:

- Korkealaakso, P., Mikkola, A., Rouvinen, A., 2006, "Multi-Body Simulation Approach for Fault Diagnosis of a Reel", *Journal of Multi-body Dynamics*, **220**(1), pp. 9-19.
- Korkealaakso, P., Rouvinen, A., Mikkola, A., 2006, "Multibody Approach for Model-Based Fault Detection of a Reel", *Journal of Computational and Nonlinear Dynamics*, **1**(2), pp. 116-122.

Joint constraint modeling of rigid-flexible mechanisms

In order to model rigid-flexible mechanisms, a general approach is needed to define joint constraints. This study provides detailed derivations of constraint equations that can be applied with the floating frame of reference formulation. The derivation is accomplished with

the help of three basic constraints, which can be further utilized in the modeling of joints. The basic components derived can also be used in methods based on the system topology when joint constraints are removed in order to open closed chains. In this sub-study, generalized Newton-Euler equations of motion have been derived according to the principle of virtual work, while the local angular accelerations of the frame of reference are variables to be integrated ahead of time. This original scientific contribution has been published in the following journal paper:

- Korkealaakso P., Mikkola, A., Rantalainen, T., Rouvinen, A., 2009, “Description of Joint Constraints in the Floating Frame of Reference Formulation”, *Journal of Multi-body Dynamics*, **223**(2), pp. 133-145.

The structure of the real-time simulation environment in the framework of a gantry crane training simulator

This thesis introduces a real-time simulation environment which can be used in the application of mobile as well as industrial machines. The environment introduced is modular and easily expandable, which systematically and efficiently facilitates studying and testing different modeling approaches and modules, such as motion platforms, visualization environments and additional computational nodes. In these simulation models, the mechanical dynamic model including rigid and flexible bodies, hydraulic subsystems, electric motors, a visualization system and motion platform with control devices can all be coupled together. The original scientific contribution associated with the real-time simulation environment has been published in the following journal papers:

- Korkealaakso, P., Rouvinen, A., Moisio, S. and Peusaari, J., 2007, “Development of a Real-Time Simulation Environment” *Multibody System Dynamics*, **17**(2-3), pp. 177-194.
- Rouvinen, A., Lehtinen, T., Korkealaakso, P., 2005, “Container Gantry Crane Simulator for Operator Training”, *Journal of Multi-body Dynamics*, **219**(4), pp. 325-335.

2 MODELING OF MULTIBODY SYSTEMS USING THE REFERENCE FRAME APPROACH

The method of the floating frame of reference is the method most frequently applied to describe linear deformations in multibody applications. This is due to the computational efficiency of the method and the possibility to utilize commercial finite element software to define properties of flexible bodies. In this chapter, the floating frame of reference approach with three different descriptions of equations of motion is briefly introduced.

2.1 Spatial kinematics of a flexible body

The floating frame of reference formulation can be applied to bodies that experience large rigid body translations and rotations as well as elastic deformations. The method is based on describing deformations of a flexible body with respect to a frame of reference. The frame of reference, in turn, is employed to describe large translations and rotations. The deformations of a flexible body with respect to its frame of reference can be described with a number of methods, whereas in this study, deformation is described using linear deformation modes of the body. Deformation modes can be defined using a finite element model of the body. Fig. 3 illustrates the position of particle P^i in a deformed body i .

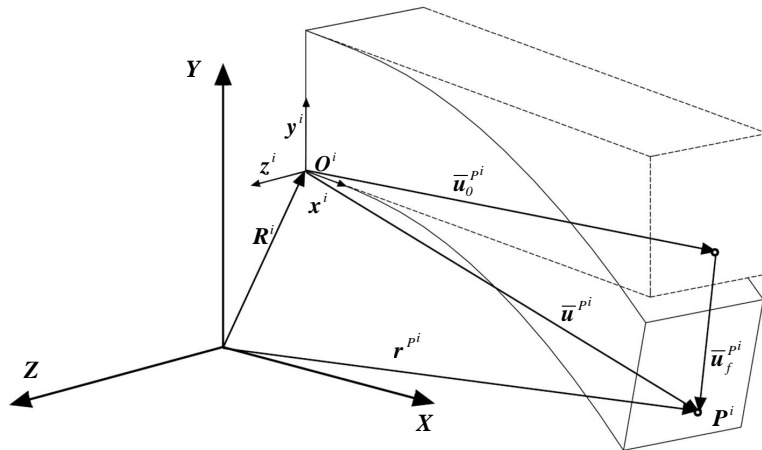


Figure 3. The position of the particle P^i in global coordinate system.

The position of particle P^i of the flexible body i can be described in a global coordinate system using the vector \mathbf{r}^{P^i} as follows:

$$\mathbf{r}^{P^i} = \mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}^{P^i} = \mathbf{R}^i + \mathbf{A}^i (\bar{\mathbf{u}}_0^{P^i} + \bar{\mathbf{u}}_f^{P^i}), \quad (1)$$

where \mathbf{R}^i is the position vector of the frame of reference, \mathbf{A}^i is the rotation matrix of body i , $\bar{\mathbf{u}}^{P^i}$ is the position vector of particle P^i within the frame of reference, $\bar{\mathbf{u}}_0^{P^i}$ is the undeformed position vector of the particle within the frame of reference, and $\bar{\mathbf{u}}_f^{P^i}$ is the displacement of particle P^i within the frame of reference due to the deformation of body i . In this study, the rotation matrix \mathbf{A}^i is expressed using Euler parameters $\boldsymbol{\theta}^{E^i T} = [\theta_0^{E^i} \ \theta_1^{E^i} \ \theta_2^{E^i} \ \theta_3^{E^i}]^T$ in order to avoid singular conditions which are a problem when three rotational parameters are used, such as in the cases of Euler and Bryant angles [23]. The rotation matrix can be written using Euler parameters as follows:

$$\mathbf{A}^i = 2 \begin{bmatrix} \frac{1}{2} - (\theta_2^{E^i})^2 - (\theta_3^{E^i})^2 & \theta_1^{E^i} \theta_2^{E^i} - \theta_0^{E^i} \theta_3^{E^i} & \theta_1^{E^i} \theta_3^{E^i} + \theta_0^{E^i} \theta_2^{E^i} \\ \theta_1^{E^i} \theta_2^{E^i} + \theta_0^{E^i} \theta_3^{E^i} & \frac{1}{2} - (\theta_1^{E^i})^2 - (\theta_3^{E^i})^2 & \theta_2^{E^i} \theta_3^{E^i} - \theta_0^{E^i} \theta_1^{E^i} \\ \theta_1^{E^i} \theta_3^{E^i} - \theta_0^{E^i} \theta_2^{E^i} & \theta_2^{E^i} \theta_3^{E^i} + \theta_0^{E^i} \theta_1^{E^i} & \frac{1}{2} - (\theta_1^{E^i})^2 - (\theta_2^{E^i})^2 \end{bmatrix}. \quad (2)$$

The following mathematical constraint must be taken into consideration when Euler parameters are applied:

$$(\theta_0^{E^i})^2 + (\theta_1^{E^i})^2 + (\theta_2^{E^i})^2 + (\theta_3^{E^i})^2 = 1. \quad (3)$$

The deformation vector $\bar{\mathbf{u}}_f^{P^i}$ can be described using a linear combination of the deformation modes as follows:

$$\bar{\mathbf{u}}_f^{P^i} = \boldsymbol{\Psi}_R^{P^i} \mathbf{q}_f^i, \quad (4)$$

where $\boldsymbol{\Psi}_R^{P^i}$ is the modal matrix whose columns describe the translation of particle P^i within the assumed deformation modes of the flexible body i [18], and \mathbf{q}_f^i is the vector of elastic coordinates. Consequently, the generalized coordinates that uniquely define the position of point P^i can be represented with vector \mathbf{p}^i as follows:

$$\mathbf{p}^{i T} = \left[\mathbf{R}^{i T} \quad \boldsymbol{\theta}^{E^i T} \quad \mathbf{q}_f^{i T} \right]^T. \quad (5)$$

The velocity of particle P^i can be obtained by differentiating the position description (Eq. 1) with respect to time as follows:

$$\dot{\mathbf{r}}^{P^i} = \dot{\mathbf{R}}^i - \mathbf{A}^i \left(\tilde{\omega}^{P^i} + \tilde{\Psi}_R^{P^i} \mathbf{q}_f^i \right) \bar{\omega}^i + \mathbf{A}^i \Psi_R^{P^i} \dot{\mathbf{q}}_f^i, \quad (6)$$

where $\bar{\omega}^i$ is the vector of local angular velocities. In Eq. 6, the generalized velocity vector can be defined as follows:

$$\dot{\mathbf{q}}^{i^T} = \left[\dot{\mathbf{R}}^{i^T} \quad \bar{\omega}^{i^T} \quad \dot{\mathbf{q}}_f^{i^T} \right]^T. \quad (7)$$

By differentiating Eq. (6) with respect to time, the following formulation for the acceleration of particle P^i can be obtained:

$$\ddot{\mathbf{r}}^{P^i} = \ddot{\mathbf{R}}^i + \mathbf{A}^i \tilde{\omega}^i \tilde{\omega}^i \bar{\mathbf{u}}^{P^i} + \mathbf{A}^i \tilde{\omega}^i \tilde{\omega}^i \bar{\mathbf{u}}^{P^i} + 2\mathbf{A}^i \tilde{\omega}^i \dot{\mathbf{q}}_f^{P^i} + \mathbf{A}^i \ddot{\mathbf{q}}_f^{P^i}, \quad (8)$$

where $\tilde{\omega}^i$ is a skew-symmetric representation of the angular velocity of the body in the frame of reference, $\ddot{\mathbf{R}}^i$ is the vector that defines the translational acceleration of the frame of reference, $\mathbf{A}^i \tilde{\omega}^i \tilde{\omega}^i \bar{\mathbf{u}}^{P^i}$ is the normal component of acceleration, $\mathbf{A}^i \tilde{\omega}^i \dot{\mathbf{q}}_f^{P^i}$ is the tangential component of acceleration, $2\mathbf{A}^i \tilde{\omega}^i \dot{\mathbf{q}}_f^{P^i}$ is the Coriolis component of acceleration and $\mathbf{A}^i \ddot{\mathbf{q}}_f^{P^i}$ is the acceleration of particle P^i due to the deformation of body i .

When deformation modes are used with the floating frame of reference, rotations due to body deformation are usually ignored. However, in order to compose all of the basic constraints, rotation due to body deformation must be accounted for. The vector $\bar{\mathbf{v}}_f^i$ due to deformation at the location of particle P^i within the frame of reference can be expressed as follows:

$$\bar{\mathbf{v}}_f^i = \mathbf{A}_f^{P^i} \bar{\mathbf{v}}^i, \quad (9)$$

where $\bar{\mathbf{v}}^i$ is defined in the undeformed state at the location of particle P^i , and $\mathbf{A}_f^{P^i}$ is a rotation matrix that describes the orientation due to deformation at the location of particle P^i with respect to the reference frame. Note that all components in Eq. 9 are expressed in the reference frame. The rotation matrix $\mathbf{A}_f^{P^i}$ can be expressed as follows:

$$\mathbf{A}_f^{P^i} = \mathbf{I} + \tilde{\boldsymbol{\varepsilon}}^{P^i}, \quad (10)$$

where \mathbf{I} is a (3×3) identity matrix and $\tilde{\boldsymbol{\varepsilon}}^{P^i}$ is a skew symmetric form of the rotation change caused by deformation. Rotation changes due to deformation can be represented in the following way:

$$\boldsymbol{\varepsilon}^{P^i} = \boldsymbol{\Psi}_\theta^{P^i} \mathbf{q}_f^i, \quad (11)$$

where $\boldsymbol{\Psi}_\theta^{P^i}$ is the modal transformation matrix whose columns describe rotation coordinates of point P^i within the assumed deformation modes of the flexible body i [18], and \mathbf{q}_f^i is the vector of elastic coordinates.

2.2 Virtual work

The equations of motion can be developed using the principle of virtual work, which can be written for inertia forces as follows:

$$\delta W^{i^i} = \int_{V^i} \rho^i \delta \mathbf{r}^{P^i T} \boldsymbol{\mathbb{R}}^{P^i} dV^i, \quad (12)$$

where $\delta \mathbf{r}^{P^i}$ is the virtual displacement of the position vector of a particle, $\boldsymbol{\mathbb{R}}^{P^i}$ is the acceleration vector of the particle defined in Eq 8, ρ^i is density of body i , and V^i is volume of body i . Accordingly, the virtual displacement of the position vector can be expressed in terms of virtual displacement of generalized coordinates as follows:

$$\delta \mathbf{r}^{P^i T} = \begin{bmatrix} \delta \mathbf{R}^{i T} & \delta \boldsymbol{\theta}^{i T} & \delta \mathbf{q}_f^{i T} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ -\tilde{\mathbf{u}}^{P^i T} \mathbf{A}^{i T} \\ \boldsymbol{\Psi}_R^{P^i T} \mathbf{A}^{i T} \end{bmatrix}, \quad (13)$$

where $\delta \boldsymbol{\theta}^i$ is virtual rotation. By substituting the virtual displacement of the position vector (13) into the equation of virtual work of the inertial forces (12) and by separating the terms related to acceleration from the terms related quadratically to velocities, the following equation for the virtual work of inertial forces can be obtained:

$$\delta W^{i^i} = \delta \mathbf{q}^i \left[\mathbf{M}^i \boldsymbol{\mathbb{R}}^i + \mathbf{Q}^{v^i} \right], \quad (14)$$

where \mathbf{M}^i is the mass matrix and \mathbf{Q}^{v^i} is the quadratic velocity vector. The mass matrix can be expressed as follows:

$$\mathbf{M}^i = \int_{V^i} \rho^i \begin{bmatrix} \mathbf{I} & -\mathbf{A}^i \tilde{\mathbf{u}}^{P^i} & \mathbf{A}^i \Psi_R^{P^i} \\ \tilde{\mathbf{u}}^{P^i T} \tilde{\mathbf{u}}^{P^i} & -\tilde{\mathbf{u}}^{P^i T} \Psi_R^{P^i} & \Psi_R^{P^i} \\ sym & & \Psi_R^{P^i T} \Psi_R^{P^i} \end{bmatrix} dV^i. \quad (15)$$

And, correspondingly, the quadratic velocity vector takes the form

$$\mathbf{Q}^{v^i} = \int_{V^i} \rho^i \begin{bmatrix} \mathbf{A}^i \tilde{\omega}^i \tilde{\omega}^i \bar{\mathbf{u}}^{P^i} + 2\mathbf{A}^i \tilde{\omega}^i \Psi_R^{P^i} \mathbf{q}_f^i \\ -\tilde{\mathbf{u}}^{P^i T} \tilde{\omega}^i \tilde{\omega}^i \bar{\mathbf{u}}^{P^i} - 2\tilde{\mathbf{u}}^{P^i T} \tilde{\omega}^i \Psi_R^{P^i} \mathbf{q}_f^i \\ \Psi_R^{P^i T} \tilde{\omega}^i \tilde{\omega}^i \bar{\mathbf{u}}^{P^i} + 2\Psi_R^{P^i T} \tilde{\omega}^i \Psi_R^{P^i} \mathbf{q}_f^i \end{bmatrix} dV^i \quad (16)$$

The virtual work of the externally applied forces can be written as:

$$\delta W^{e^i} = \int_{V^i} \delta \mathbf{r}^{P^i T} \mathbf{F}^{P^i} dV^i = \delta \mathbf{q}^{i T} \mathbf{Q}^{e^i}, \quad (17)$$

where \mathbf{F}^{P^i} is external force per unit mass and \mathbf{Q}^{e^i} is the vector of generalized forces which can be expressed as follows:

$$\mathbf{Q}^{e^i} = \begin{bmatrix} \sum_{j=1}^{n_F} \mathbf{F}_j^i \\ \sum_{j=1}^{n_F} \tilde{\mathbf{u}}_j^i \mathbf{A}^{i T} \mathbf{F}_j^i \\ \sum_{j=1}^{n_F} \Psi_j^{i T} \mathbf{A}^{i T} \mathbf{F}_j^i \end{bmatrix} \quad (18)$$

where \mathbf{F}_j^i is the j -th force component acting on body i , $\tilde{\mathbf{u}}_j^i$ is a skew symmetric matrix of the location vector of the j -th force components, and Ψ_j^i is the terms of the modal matrix associated with the node to which the j -th force component applies.

The elastic forces can be defined using the modal stiffness matrix \mathbf{K}^i and modal coordinates. The modal stiffness matrix is associated with the modal coordinates and the matrix can be obtained from the conventional finite element approach using the component mode synthesis technique [18]. The virtual work of elastic forces can be written as follows:

$$\delta W^{s^i} = \delta \mathbf{q}_f^{i T} \mathbf{K}^i \mathbf{q}_f^i. \quad (19)$$

Accordingly, the vector of elastic forces can be represented as follows:

$$\mathbf{Q}^{f^i} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{K}^i \mathbf{q}_f^i \end{bmatrix}. \quad (20)$$

Using Eqs. 14, 17 and 19, the equation of virtual work including inertial, external and internal force components can be written as follows:

$$\delta \mathbf{q}^i \left[\mathbf{M}^i \ddot{\mathbf{q}}^i + \mathbf{Q}^{y^i} + \mathbf{Q}^{f^i} - \mathbf{Q}^{e^i} \right] = 0. \quad (21)$$

The terms inside the brackets can be used to form unconstrained Newton-Euler equations as follows:

$$\begin{bmatrix} \int_{V^i} \rho^i \mathbf{I} dV^i & - \int_{V^i} \rho^i \mathbf{A}^i \tilde{\mathbf{u}}^{p^i} dV^i & \int_{V^i} \rho^i \mathbf{A}^i \boldsymbol{\Psi}_R^{p^i} dV^i \\ & \int_{V^i} \rho^i \tilde{\mathbf{u}}^{p^i T} \tilde{\mathbf{u}}^{p^i} dV^i & - \int_{V^i} \rho^i \tilde{\mathbf{u}}^{p^i T} \boldsymbol{\Psi}_R^{p^i} dV^i \\ \text{sym} & & \int_{V^i} \rho^i \boldsymbol{\Psi}_R^{p^i T} \boldsymbol{\Psi}_R^{p^i} dV^i \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{R}}^i \\ \ddot{\boldsymbol{\phi}}^i \\ \ddot{\mathbf{q}}_f^i \end{bmatrix} \quad (22)$$

$$= \begin{bmatrix} \int_{V^i} \mathbf{F}^{p^i} dV^i \\ \int_{V^i} \tilde{\mathbf{u}}^{p^i} \mathbf{A}^{i T} \mathbf{F}^{p^i} dV^i \\ \int_{V^i} \boldsymbol{\Psi}_R^{p^i T} \mathbf{A}^{i T} \mathbf{F}^{p^i} dV^i \end{bmatrix} - \begin{bmatrix} \int_{V^i} \rho^i \left(\mathbf{A}^i \tilde{\boldsymbol{\omega}}^i \tilde{\boldsymbol{\omega}}^i \tilde{\mathbf{u}}^{p^i} + 2 \mathbf{A}^i \tilde{\boldsymbol{\omega}}^i \boldsymbol{\Psi}_R^{p^i} \ddot{\mathbf{q}}_f^i \right) dV^i \\ \int_{V^i} \rho^i \left(- \tilde{\mathbf{u}}^{p^i T} \tilde{\boldsymbol{\omega}}^i \tilde{\boldsymbol{\omega}}^i \tilde{\mathbf{u}}^{p^i} - 2 \tilde{\mathbf{u}}^{p^i T} \tilde{\boldsymbol{\omega}}^i \boldsymbol{\Psi}_R^{p^i} \ddot{\mathbf{q}}_f^i \right) dV^i \\ \int_{V^i} \rho^i \left(\boldsymbol{\Psi}_R^{p^i T} \tilde{\boldsymbol{\omega}}^i \tilde{\boldsymbol{\omega}}^i \tilde{\mathbf{u}}^{p^i} + 2 \boldsymbol{\Psi}_R^{p^i T} \tilde{\boldsymbol{\omega}}^i \boldsymbol{\Psi}_R^{p^i} \ddot{\mathbf{q}}_f^i \right) dV^i \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \mathbf{K}^i \mathbf{q}_f^i \end{bmatrix}$$

Equations of motion in this form are referred to as Generalized Newton-Euler equations in Reference [18], where Newton-Euler equations of rigid bodies are extended to flexible bodies.

2.2.1 Integration of the equations of motion

Due to the use of Generalized Newton-Euler equations as a description of dynamics, the equations of motion are expressed using the angular velocity and angular acceleration vectors. Eq. 22 can be solved for angular accelerations in the body frame which can be integrated with angular velocities. However, the problem arises when the coordinates describing the orientation of the body have to be solved. This is due to the fact that angular velocities cannot be directly integrated with the parameters which uniquely describe the orientation of the body. For this reason, a new set of variables \mathbf{p} is defined, containing the orientation coordinates of the body reference frame. In order to integrate the position level coordinates, the first time

derivative of Euler parameters and the vector of angular velocities defined in the body reference frame can be related through the following linear expression:

$$\dot{\boldsymbol{\theta}}^{E^i} = \frac{1}{2} \overline{\mathbf{G}}^{i^T} \overline{\boldsymbol{\omega}}^i, \quad (23)$$

where the velocity transformation matrix $\overline{\mathbf{G}}^i$ can be written as follows:

$$\overline{\mathbf{G}}^i = \begin{bmatrix} -\theta_1^{E^i} & \theta_0^{E^i} & \theta_3^{E^i} & -\theta_2^{E^i} \\ -\theta_2^{E^i} & -\theta_3^{E^i} & \theta_0^{E^i} & \theta_1^{E^i} \\ -\theta_3^{E^i} & \theta_2^{E^i} & -\theta_1^{E^i} & \theta_0^{E^i} \end{bmatrix}. \quad (24)$$

The time derivatives of the body variables to be intergrated can be stated using vector $\dot{\boldsymbol{p}}$ as follows:

$$\dot{\boldsymbol{p}}^{i^T} = \left[\dot{\mathbf{R}}^{i^T} \quad \dot{\boldsymbol{\theta}}^{E^i^T} \quad \dot{\boldsymbol{\phi}}_f^{i^T} \right]^T, \quad (25)$$

which can be integrated to obtain position level generalized coordinates \boldsymbol{p} .

2.3 Description of multibody equations of motion

In this section, the three multibody formalisms used in this work are briefly described. The formalisms discussed here are the method based on Lagrange multipliers, which is also referred to as the descriptor form [24, 18], the penalty and augmented Lagrangian methods [25, 9] and the method based on projection matrix [26, 27, 28].

2.3.1 Method of Lagrange multipliers

When constraint equations are augmented to equations of motion using the Lagrange multiplier technique, the result can be written as:

$$\mathbf{M} \ddot{\boldsymbol{q}} + \mathbf{C}_q^T \boldsymbol{\lambda} = \mathbf{Q}^e - \mathbf{Q}^v - \mathbf{Q}^f, \quad (26)$$

where \boldsymbol{q} is the vector of n generalized coordinates that define the position and orientation of each body in the system, \mathbf{M} is the mass matrix, \mathbf{Q}^e is the vector of generalized forces, \mathbf{Q}^v is the quadratic velocity vector that includes velocity dependent inertia forces, \mathbf{C}_q is the Jacobian matrix of the constraint equations, \mathbf{Q}^f is the vector of elastic forces and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. To satisfy a set of m constraint equations related to generalized coordinates, the following equation must be fulfilled:

$$\mathbf{C}(\mathbf{q}, t) = \mathbf{0}, \quad (27)$$

where \mathbf{C} is a vector of constraints of the system and t is time. Eqs. (26) and (27) comprise a system of differential algebraic equations (DAE) which describe the dynamical behavior of the mechanics. In order to solve the set of equations using ordinary integration methods for differential equation (ODE), the equations must be transformed to the second order ODE. For this reason, Eq. (27) is differentiated twice with respect to time:

$$\ddot{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = \mathbf{C}_q \ddot{\mathbf{q}} + \mathbf{Q}^c = \mathbf{0}, \quad (28)$$

where \mathbf{Q}^c includes velocity dependent terms due to differentiation. By combining Eqs. (26) and (28), the matrix representation of equations of motion can be obtained as follows:

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}_q^T \\ \mathbf{C}_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^e - \mathbf{Q}^v - \mathbf{Q}^f \\ -\mathbf{Q}^c \end{bmatrix}, \quad (29)$$

where the invertable matrix is of the size $(m+n) \times (m+n)$. The equation of motion can be integrated using the standard ODE solver [18]. However, equations of motion cannot guarantee that constraint equations in Eq. (27) are satisfied. This is due to the fact that during the differentiation of a constraint equation, constant terms disappear, and consequently, Eq. (29) fulfils the constraints at acceleration level only. Therefore, numerical integration causes errors to accumulate in the kinematic joint constraints. To overcome this problem, a stabilization method must be used. Another possibility to solve this problem is to use methods which produce a general solution to differential algebraic equations [9, 29].

2.3.2 Augmented Lagrangian method

In the penalty method, Lagrange multipliers are eliminated from the equations of motion by employing penalty terms. This procedure leads to a set of n differential equations as follows:

$$(\mathbf{M} + \mathbf{C}_q^T \boldsymbol{\alpha} \mathbf{C}_q) \ddot{\mathbf{q}} = \mathbf{Q}^e - \mathbf{Q}^v - \mathbf{Q}^f - \mathbf{C}_q^T \boldsymbol{\alpha} (\mathbf{Q}^c + 2\boldsymbol{\Omega} \boldsymbol{\mu} \dot{\mathbf{C}} + \boldsymbol{\Omega}^2 \mathbf{C}), \quad (30)$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\Omega}$ and $\boldsymbol{\mu}$ are $m \times m$ diagonal matrices which contain penalty terms, natural frequencies and damping ratios for constraints, respectively. If the penalty terms are equivalent to each constraint, the matrices are identity matrices multiplied with a constant penalty factor.

A drawback associated with the penalty method is that large numerical values for penalty factors must be used, which may lead to numerical ill-conditioning and round-off errors. However, the method can be improved by adding penalty terms or correction terms which are zero when constraint equations are fulfilled. Using this approach, equations of motion can be written as follows:

$$(\mathbf{M} + \mathbf{C}_q^T \boldsymbol{\alpha} \mathbf{C}_q) \ddot{\boldsymbol{\phi}} = \mathbf{Q}^e - \mathbf{Q}^v - \mathbf{Q}^f - \mathbf{C}_q^T \boldsymbol{\alpha} (\mathbf{Q}^c + 2\boldsymbol{\Omega} \boldsymbol{\mu} \dot{\boldsymbol{\phi}} + \boldsymbol{\Omega}^2 \mathbf{C}) + \mathbf{C}_q^T \boldsymbol{\lambda}^*, \quad (31)$$

where $\boldsymbol{\lambda}^*$ is the vector of penalty forces. By comparing Eqs. (1) and (31), it can be concluded that

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}^* - \boldsymbol{\alpha} (\mathbf{C}_q \ddot{\boldsymbol{\phi}} + \mathbf{Q}^c + 2\boldsymbol{\Omega} \boldsymbol{\mu} \dot{\boldsymbol{\phi}} + \boldsymbol{\Omega}^2 \mathbf{C}). \quad (32)$$

Since the exact values of $\boldsymbol{\lambda}^*$ are not known in advance, an iterative procedure should be used as follows:

$$\boldsymbol{\lambda}_{i+1}^* = \boldsymbol{\lambda}_i^* - \boldsymbol{\alpha} (\mathbf{C}_q \ddot{\boldsymbol{\phi}}_i + \mathbf{Q}^c + 2\boldsymbol{\Omega} \boldsymbol{\mu} \dot{\boldsymbol{\phi}}_i + \boldsymbol{\Omega}^2 \mathbf{C}), \quad (33)$$

where $\boldsymbol{\lambda}_0^* = \mathbf{0}$ is used for the first iteration. Using this equation, the forces caused by errors in constraint equations at iteration $i+1$ can be defined and compensated. In this case, the penalty terms do not need to have large numerical values. An iterative procedure can be applied directly to Eq (31), which leads to the following expression:

$$(\mathbf{M} + \mathbf{C}_q^T \boldsymbol{\alpha} \mathbf{C}_q) \ddot{\boldsymbol{\phi}}_{i+1} = \mathbf{M} \ddot{\boldsymbol{\phi}}_i - \mathbf{C}_q^T \boldsymbol{\alpha} (\mathbf{Q}^c + 2\boldsymbol{\Omega} \boldsymbol{\mu} \dot{\boldsymbol{\phi}}_i + \boldsymbol{\Omega}^2 \mathbf{C}). \quad (34)$$

In the case of the first iteration, $\mathbf{M} \ddot{\boldsymbol{\phi}}_0 = \mathbf{Q}^e - \mathbf{Q}^v - \mathbf{Q}^f$. The leading matrix of Eq. (34) is a symmetric and positive definite, which makes the solution of the equation efficient. This formulation behaves satisfactorily also in singular configurations of a mechanical system.

2.3.3 Method based on projection matrix

The two previously introduced formulations define the equations of motion using a complete set of generalized coordinates. However, the number of the equations can be reduced to the minimum number of differential equations using a set of independent generalized coordinates. Independent generalized velocities $\dot{\boldsymbol{\phi}}_i$ can be defined as a projection of velocities of generalized coordinates $\dot{\boldsymbol{\phi}}$ using matrix \mathbf{B} as follows:

$$\dot{\boldsymbol{\phi}}_i = \mathbf{B} \dot{\boldsymbol{\phi}}. \quad (35)$$

It is noteworthy that the rows of matrix \mathbf{B} are linearly independent. For skleronomous systems, a solution to describe the transformation from independent generalized coordinates

to a complete set of generalized coordinates is available and can be defined using transformation matrix \mathbf{R} as follows:

$$\dot{\mathbf{q}} = \mathbf{R} \dot{\mathbf{q}}_i. \quad (36)$$

Using coordinate partitioning to dependent \mathbf{q}_d and independent \mathbf{q}_i generalized coordinates, vector \mathbf{q} can be written in the partitioned form $\mathbf{q} = [\mathbf{q}_d^T \quad \mathbf{q}_i^T]^T$. The virtual change of generalized coordinates with respect to constraint equations can be expressed as follows:

$$\mathbf{C}_{q_d} \delta \mathbf{q}_d + \mathbf{C}_{q_i} \delta \mathbf{q}_i = \mathbf{0}, \quad (37)$$

where \mathbf{C}_{q_d} and \mathbf{C}_{q_i} are partitioned Jacobian matrices. \mathbf{C}_{q_d} is a $m \times m$ matrix where m is the number of constraint equations. Using Eq. (37), the virtual change of dependent generalized coordinates can be defined as:

$$\delta \mathbf{q}_d = -\mathbf{C}_{q_d}^{-1} \mathbf{C}_{q_i} \delta \mathbf{q}_i. \quad (38)$$

The virtual change of generalized coordinates can now be expressed using independent generalized coordinates as follows:

$$\delta \mathbf{q} = \begin{bmatrix} \delta \mathbf{q}_d \\ \delta \mathbf{q}_i \end{bmatrix} = \begin{bmatrix} -\mathbf{C}_{q_d}^{-1} \mathbf{C}_{q_i} \\ \mathbf{I} \end{bmatrix} \delta \mathbf{q}_i. \quad (39)$$

Correspondingly, the transformation matrix \mathbf{R} can be expressed as follows:

$$\mathbf{R} = \begin{bmatrix} -\mathbf{C}_{q_d}^{-1} \mathbf{C}_{q_i} \\ \mathbf{I} \end{bmatrix}. \quad (40)$$

Using coordinate partitioning, accelerations of generalized coordinates can be written as follows:

$$\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\mathbf{q}}_d \\ \ddot{\mathbf{q}}_i \end{bmatrix} = \begin{bmatrix} -\mathbf{C}_{q_d}^{-1} \mathbf{C}_{q_i} \\ \mathbf{I} \end{bmatrix} \ddot{\mathbf{q}}_i + \boldsymbol{\gamma} \quad (41)$$

with the definition

$$\boldsymbol{\gamma} = \begin{bmatrix} -\mathbf{C}_{q_d}^{-1} \left[(\mathbf{C}_{q_d} \ddot{\mathbf{q}})_q \ddot{\mathbf{q}} + 2\mathbf{C}_{q_i} \dot{\mathbf{q}} \dot{\mathbf{q}} + \mathbf{C}_{ii} \right] \\ 0 \end{bmatrix}. \quad (42)$$

It can be seen that vector $\boldsymbol{\gamma}$ consist of the accelerations of generalized coordinates when the accelerations of independent coordinates are equal to zero. Using Eq. (42), Eq. (41) can be written as follows:

$$\ddot{\mathbf{q}} = \mathbf{R} \ddot{\mathbf{q}}_i + \boldsymbol{\gamma}. \quad (43)$$

Substitution of the result into Eq. (26) leads to:

$$\mathbf{R}^T \mathbf{M} \mathbf{R} \ddot{\mathbf{q}}_i + \mathbf{R}^T \mathbf{M} \boldsymbol{\gamma} - \mathbf{R}^T (\mathbf{Q}^e - \mathbf{Q}^v - \mathbf{Q}^f) = \mathbf{0}. \quad (44)$$

This equation of motion can be solved for independent accelerations which can be integrated to solve the new independent velocities and positions for the next time step. This form of equation of motion is complicated and highly nonlinear and the set of independent generalized coordinates must be changed every time when the pivot of \mathbf{C}_{q_d} approaches zero.

2.4 Kinematic joint description

In this section, geometric constraint equations are derived for three basic constraint components, which can be further applied to the modeling of spherical joints, revolute joints, cylindrical joints and translational joints. The terms within the equations of motion that are related to the constraints are formulated so that they can easily be incorporated into multibody dynamics codes.

2.4.1 Basic Constraints

Joints in multibody systems can be described as combinations of three basic constraints. These basic constraints are the spherical constraint and two different perpendicularity constraint conditions. The basic constraints for rigid bodies have been presented e.g. in References [17] and [16]. For flexible bodies, however, there is no comprehensive analytic representation which could describe all of the components in Eq. (24) that are related to the constraints.

Spherical Constraint on Two Points

The spherical constraint on two points, which is depicted in Fig. 4, is a simple basic constraint that prevents translational movement between two bodies. The constraint equation can be defined at given points P^i and P^j . This basic constraint removes three degrees of freedom from the system.

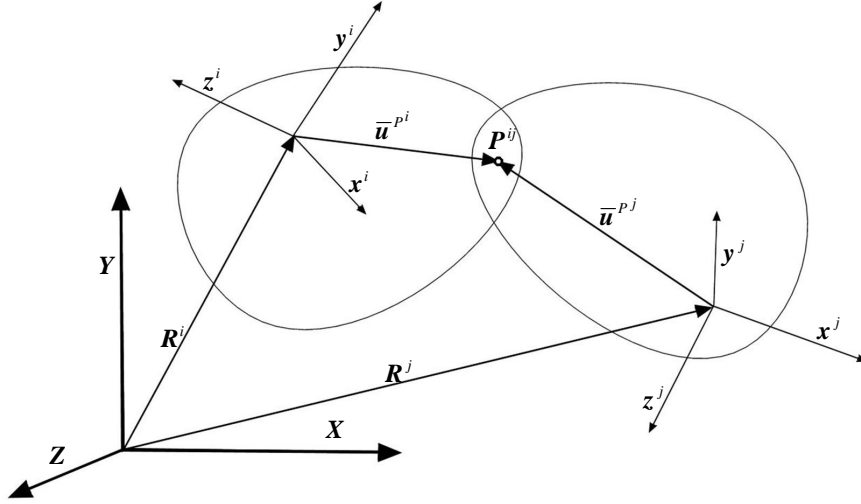


Figure 4. Spherical constraint on two points.

The constraint equation associated with points P^i and P^j can be written as follows:

$$C^s = R^j + A^j \bar{u}^{P^j} - R^i - A^i \bar{u}^{P^i} = 0. \quad (45)$$

By differentiating Eq. (45) twice with respect to time, the following equation can be obtained:

$$\begin{aligned} \ddot{C}^s &= C_q^s \ddot{q} + (C_q^s \dot{q})_q \dot{q} \\ &= \ddot{R}^j - A^j \ddot{u}^{P^j} \dot{\phi}^j + A^j \ddot{\omega}^j \ddot{\omega}^j \bar{u}^{P^j} + 2A^j \ddot{\omega}^j \dot{\phi}^{P^j} + A^j \psi_R^{P^j} \dot{\phi}_f^j \\ &\quad - \ddot{R}^i + A^i \ddot{u}^{P^i} \dot{\phi}^i - A^i \ddot{\omega}^i \ddot{\omega}^i \bar{u}^{P^i} - 2A^i \ddot{\omega}^i \dot{\phi}^{P^i} - A^i \psi_R^{P^i} \dot{\phi}_f^i. \end{aligned} \quad (46)$$

Based on Eq. (46), the following terms can be obtained for generalized coordinates related to the translation, orientation and flexibility of the Jacobian matrix:

$$C_q^s = \begin{bmatrix} -\mathbf{I} & A^i \ddot{u}^{P^i} & -A^i \psi_R^{P^i} & \mathbf{I} & -A^j \ddot{u}^{P^j} & A^j \psi_R^{P^j} \end{bmatrix}. \quad (47)$$

Similarly, a vector that includes quadratic velocity terms can be obtained as follows:

$$Q^{c^s} = -(C_q^s \dot{q})_q \dot{q} = -A^j \ddot{\omega}^j (\ddot{\omega}^j \bar{u}^{P^j} + 2\dot{\phi}^{P^j}) + A^i \ddot{\omega}^i (\ddot{\omega}^i \bar{u}^{P^i} + 2\dot{\phi}^{P^i}). \quad (48)$$

Perpendicular Constraint C^{dl}

The perpendicular constraint (type 1) preventing the rotation of vectors with respect to each other on levels which are not perpendicular to each other. The perpendicularity constraint is illustrated in Fig. 5. This basic constraint can be described with one constraint equation, which removes one degree of freedom from the system.

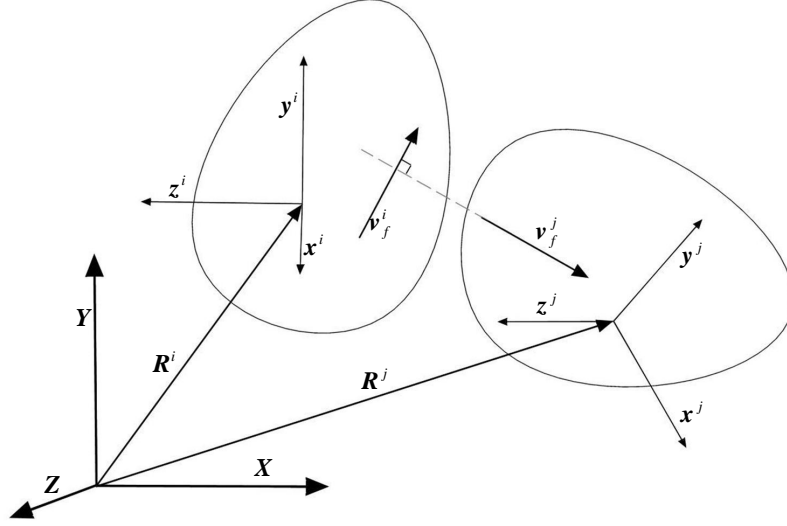


Figure 5. Type 1 perpendicular constraint.

The constraint equation for a perpendicular constraint of vectors can be written as

$$C^{d1} = \mathbf{v}_f^{iT} \mathbf{v}_f^j = \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \bar{\mathbf{v}}_f^j = \bar{\mathbf{v}}_f^{iT} \mathbf{A}_f^{PiT} \mathbf{A}^{iT} \mathbf{A}^j \mathbf{A}_f^{Pj} \bar{\mathbf{v}}_f^j = 0. \quad (49)$$

By differentiating the equation twice with respect to time, the following equation can be obtained:

$$\begin{aligned} \ddot{C}^{d1} &= C_q^{d1} \ddot{\mathbf{q}} + (C_q^{d1} \dot{\mathbf{q}})_q \dot{\mathbf{q}} = \ddot{\mathbf{v}}_f^{iT} \mathbf{v}_f^j + \mathbf{v}_f^{iT} \ddot{\mathbf{v}}_f^j + 2 \dot{\mathbf{v}}_f^{iT} \dot{\mathbf{v}}_f^j \\ &= -\bar{\mathbf{v}}_f^{jT} \mathbf{A}^{jT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i \tilde{\omega}^i - \bar{\mathbf{v}}_f^{jT} \mathbf{A}^{jT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i \Psi_\theta^{Pi} \dot{\mathbf{q}}_f^i - \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\mathbf{v}}_f^j \tilde{\omega}^j \\ &\quad - \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\mathbf{v}}_f^j \Psi_\theta^{Pj} \dot{\mathbf{q}}_f^j + \bar{\mathbf{v}}_f^{jT} \mathbf{A}^{jT} \mathbf{A}^i \tilde{\omega}^i (\tilde{\omega}^i \bar{\mathbf{v}}_f^i + 2 \dot{\mathbf{q}}_f^i) \\ &\quad + \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\omega}^j (\tilde{\omega}^j \bar{\mathbf{v}}_f^j + 2 \dot{\mathbf{q}}_f^j) + 2 (\mathbf{A}^i \tilde{\omega}^i \bar{\mathbf{v}}_f^i + \mathbf{A}^i \dot{\mathbf{q}}_f^i)^T (\mathbf{A}^j \tilde{\omega}^j \bar{\mathbf{v}}_f^j + \mathbf{A}^j \dot{\mathbf{q}}_f^j). \end{aligned} \quad (50)$$

Based on Eq. (50), the following terms can be obtained for generalized coordinates related to the translation, orientation and flexibility of the Jacobian matrix:

$$C_q^{d1} = \begin{bmatrix} 0 & -\bar{\mathbf{v}}_f^{jT} \mathbf{A}^{jT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i & -\bar{\mathbf{v}}_f^{jT} \mathbf{A}^{jT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i \Psi_\theta^{Pi} & \mathbf{L} \\ \mathbf{L} & 0 & -\bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\mathbf{v}}_f^j & -\bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\mathbf{v}}_f^j \Psi_\theta^{Pj} \end{bmatrix}. \quad (51)$$

Correspondingly, the term that includes quadratic velocity terms can be represented as

$$Q^{c^{d1}} = -(\mathbf{C}_q \dot{\mathbf{q}})_q \dot{\mathbf{q}} = -\bar{\mathbf{v}}_f^{jT} \mathbf{A}^{jT} \mathbf{A}^i \tilde{\boldsymbol{\omega}}^i (\tilde{\boldsymbol{\omega}}^i \bar{\mathbf{v}}_f^i + 2\dot{\mathbf{p}}_f^i) - \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\boldsymbol{\omega}}^j (\tilde{\boldsymbol{\omega}}^j \bar{\mathbf{v}}_f^j + 2\dot{\mathbf{p}}_f^j) - 2(\mathbf{A}^i \tilde{\boldsymbol{\omega}}^i \bar{\mathbf{v}}_f^i + \mathbf{A}^i \dot{\mathbf{p}}_f^i)^T (\mathbf{A}^j \tilde{\boldsymbol{\omega}}^j \bar{\mathbf{v}}_f^j + \mathbf{A}^j \dot{\mathbf{p}}_f^j). \quad (52)$$

Perpendicular Constraint C^{d2}

Perpendicular constraint type 2 differs from type 1 in that one of the vectors is defined as constant with respect to body i , whereas the other is defined between the bodies as shown in Fig. 6. This constraint is also known as the point on plane since it contains one constraint equation eliminating one degree of freedom.

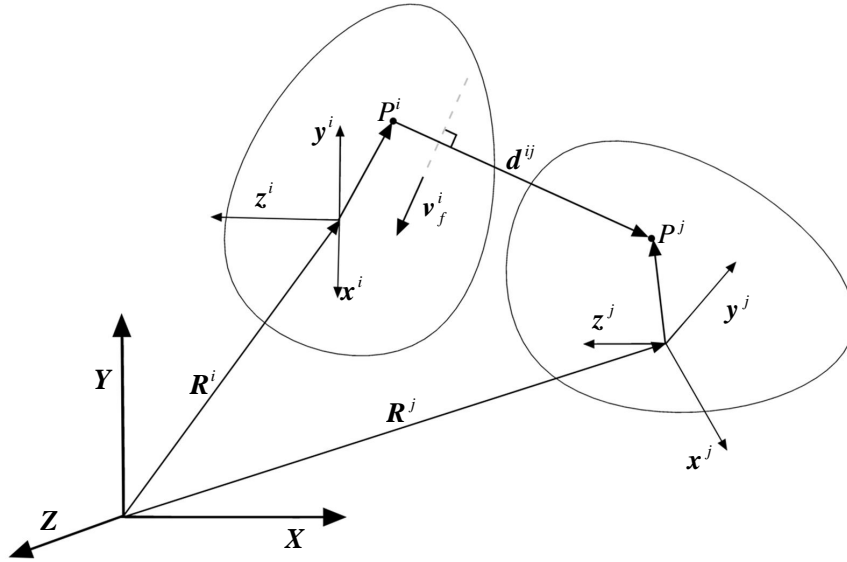


Figure 6. Type 2 perpendicularity constraint.

The constraint equation for a type 2 perpendicularity constraint can be represented as

$$\begin{aligned} C^{d2} &= \mathbf{v}_f^{iT} \mathbf{d}^{ij} = \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} (\mathbf{R}^j + \mathbf{A}^j \bar{\mathbf{u}}^{P^j} - \mathbf{R}^i - \mathbf{A}^i \bar{\mathbf{u}}^{P^i}) \\ &= \bar{\mathbf{v}}_f^{iT} \mathbf{A}_f^{P^i T} \mathbf{A}^{iT} (\mathbf{R}^j + \mathbf{A}^j \bar{\mathbf{u}}^{P^j} - \mathbf{R}^i - \mathbf{A}^i \bar{\mathbf{u}}^{P^i}) = 0, \end{aligned} \quad (53)$$

where \mathbf{d}^{ij} is vector from P^i to P^j defined in the global coordinate system. By differentiating the equation twice with respect to time, the following equation can be obtained:

$$\begin{aligned}
\mathbb{C}^{d2} &= C_q^{d2} \mathbb{C} + (C_q^{d2} \mathbb{C})_q \mathbb{C} = \mathbb{R}_f^{iT} \mathbf{d}^{ij} + \mathbf{v}_f^{iT} \mathbb{R}^{ij} + 2 \mathbb{R}_f^{iT} \mathbf{d}^{ij} \\
&= -\bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbb{R}^i + (\bar{\mathbf{v}}_f^{iT} \tilde{\mathbf{u}}^{pi} - \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i) \mathbb{C}^i - (\bar{\mathbf{v}}_f^{iT} \Psi_R^{pi} + \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i \Psi_\theta^{pi}) \mathbb{C}_f^i \\
&\quad + \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbb{R}^j - \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\mathbf{u}}^{pj} \mathbb{C}^j + \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \Psi_R^{pj} \mathbb{C}_f^j \\
&\quad + \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\omega}^i (\tilde{\omega}^i \bar{\mathbf{v}}_f^i + 2 \mathbb{R}_f^i) + \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\omega}^j (\tilde{\omega}^j \bar{\mathbf{u}}^{pj} + 2 \mathbb{R}^{pj}) \\
&\quad + 2 (\mathbf{A}^i \tilde{\omega}^i \bar{\mathbf{v}}_f^i + \mathbf{A}^i \mathbb{R}_f^{pi})^T (\mathbb{R}^j + \mathbf{A}^j \tilde{\omega}^j \bar{\mathbf{u}}^{pj} + \mathbf{A}^j \mathbb{R}^{pj} - \mathbb{R}^i) \\
&\quad + \bar{\mathbf{v}}_f^{iT} \tilde{\omega}^i \tilde{\omega}^i \bar{\mathbf{u}}^{pi} + 2 \bar{\mathbf{v}}_f^{iT} \mathbb{R}_f^{pi} \tilde{\omega}^i \bar{\mathbf{u}}^{pi} + 2 \bar{\mathbf{v}}_f^{iT} \mathbb{R}_f^{pi} \mathbb{R}^{pi}.
\end{aligned} \tag{54}$$

Based on Eq. (54), the following terms are obtained for generalized coordinates related to the translation, orientation and flexibility of the Jacobian matrix:

$$C_q^{d2} = \begin{bmatrix} \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} & \bar{\mathbf{v}}_f^{iT} \tilde{\mathbf{u}}^{pi} - \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i & -\bar{\mathbf{v}}_f^{iT} \Psi_R^{pi} - \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i \Psi_\theta^{pi} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} & -\bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\mathbf{u}}^{pj} & \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \Psi_R^{pj} \end{bmatrix}. \tag{55}$$

Correspondingly, the term that contains quadratic velocity terms can be expressed as follows:

$$\begin{aligned}
Q^{c d2} &= -(C_q \mathbb{C})_q \mathbb{C} = -\mathbf{d}^{ijT} \mathbf{A}^i \tilde{\omega}^i (\tilde{\omega}^i \bar{\mathbf{v}}_f^i + 2 \mathbb{R}_f^i) - \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\omega}^j (\tilde{\omega}^j \bar{\mathbf{u}}^{pj} + 2 \mathbb{R}^{pj}) \\
&\quad - 2 (\mathbf{A}^i \tilde{\omega}^i \bar{\mathbf{v}}_f^i + \mathbf{A}^i \mathbb{R}_f^{pi})^T (\mathbb{R}^j + \mathbf{A}^j \tilde{\omega}^j \bar{\mathbf{u}}^{pj} + \mathbf{A}^j \mathbb{R}^{pj} - \mathbb{R}^i) \\
&\quad - \bar{\mathbf{v}}_f^{iT} \tilde{\omega}^i \tilde{\omega}^i \bar{\mathbf{u}}^{pi} + 2 \mathbb{R}_f^{iT} \tilde{\omega}^i \bar{\mathbf{u}}^{pi} + 2 \mathbb{R}_f^{iT} \mathbb{R}^{pi}.
\end{aligned} \tag{56}$$

2.4.2 Modeling of Joints Based on Basic Constraints

In this section, the basic joint types used in multibody dynamics modeling applying the basic constraints presented above are briefly introduced. With different combinations of basic constraints, it is possible to model any joint. Table 1 summarizes partial derivatives with regard to generalized coordinates for each basic constraint.

Table 1. Partial derivatives for basic constraints.

	C^s	C^{d1}	C^{d2}
C_{R_i}	$-\mathbf{I}$	$\mathbf{0}$	$-\bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT}$
C_{θ_i}	$\mathbf{A}^i \tilde{\mathbf{u}}^{pi}$	$-\bar{\mathbf{v}}_f^{jT} \mathbf{A}^{jT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i$	$\bar{\mathbf{v}}_f^{iT} \tilde{\mathbf{u}}^{pi} - \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i$
$C_{q_f^i}$	$-\mathbf{A}^i \Psi_R^{pi}$	$-\bar{\mathbf{v}}_f^{jT} \mathbf{A}^{jT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i \Psi_\theta^{pi}$	$-\bar{\mathbf{v}}_f^{iT} \Psi_R^{pi} - \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i \Psi_\theta^{pi}$
C_{R_j}	\mathbf{I}	$\mathbf{0}$	$\bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT}$

$\mathbf{C}_{\bar{\theta}^j}$	$-\mathbf{A}^j \tilde{\mathbf{u}}^{Pj}$	$-\bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\mathbf{v}}_f^j$	$-\bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\mathbf{u}}^{Pj}$
$\mathbf{C}_{q_f^i}$	$\mathbf{A}^j \Psi_R^{Pj}$	$-\bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\mathbf{v}}_f^j \Psi_\theta^{Pj}$	$\bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \Psi_R^{Pj}$

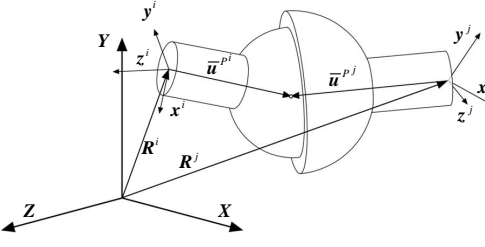
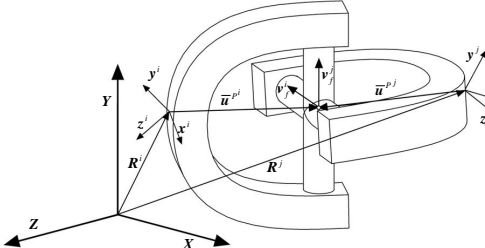
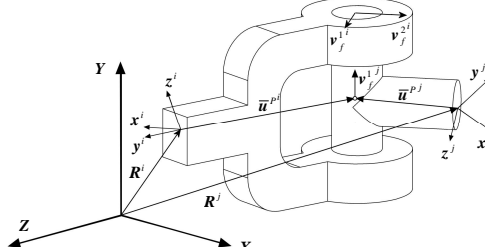
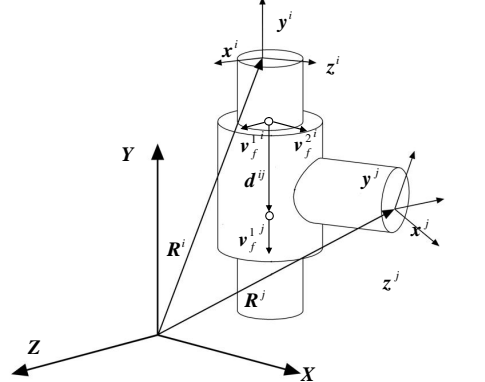
Table 2 presents the components of the constraint force vector related to the basic constraints.

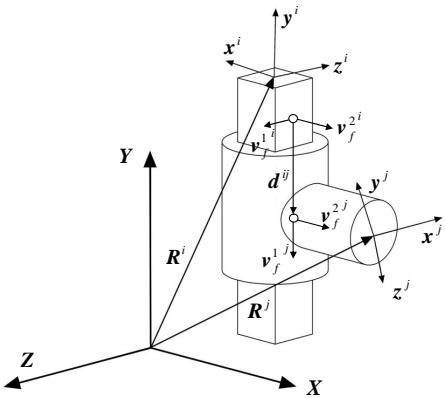
Table 2. Components of the constraint force vector related to basic constraints.

	\mathcal{Q}_c
\mathcal{Q}^{c^s}	$-\mathbf{A}^j \tilde{\boldsymbol{\omega}}^j (\tilde{\boldsymbol{\omega}}^j \bar{\mathbf{u}}^{Pj} + 2\dot{\mathbf{h}}^{Pj}) + \mathbf{A}^i \tilde{\boldsymbol{\omega}}^i (\tilde{\boldsymbol{\omega}}^i \bar{\mathbf{u}}^{Pi} + 2\dot{\mathbf{h}}^{Pi})$
$\mathcal{Q}^{c^{d1}}$	$-\bar{\mathbf{v}}_f^{jT} \mathbf{A}^{jT} \mathbf{A}^i \tilde{\boldsymbol{\omega}}^i (\tilde{\boldsymbol{\omega}}^i \bar{\mathbf{v}}_f^i + 2\dot{\mathbf{h}}_f^i) - \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\boldsymbol{\omega}}^j (\tilde{\boldsymbol{\omega}}^j \bar{\mathbf{v}}_f^j + 2\dot{\mathbf{h}}_f^j)$ $-2(\mathbf{A}^i \tilde{\boldsymbol{\omega}}^i \bar{\mathbf{v}}_f^i + \mathbf{A}^i \dot{\mathbf{h}}_f^i)^T (\mathbf{A}^j \tilde{\boldsymbol{\omega}}^j \bar{\mathbf{v}}_f^j + \mathbf{A}^j \dot{\mathbf{h}}_f^j)$
$\mathcal{Q}^{c^{d2}}$	$-\mathbf{d}^{ijT} \mathbf{A}^i \tilde{\boldsymbol{\omega}}^i (\tilde{\boldsymbol{\omega}}^i \bar{\mathbf{v}}_f^i + 2\dot{\mathbf{h}}_f^i) - \bar{\mathbf{v}}_f^{iT} \mathbf{A}^{iT} \mathbf{A}^j \tilde{\boldsymbol{\omega}}^j (\tilde{\boldsymbol{\omega}}^j \bar{\mathbf{u}}^{Pj} + 2\dot{\mathbf{h}}^{Pj})$ $-2(\mathbf{A}^i \tilde{\boldsymbol{\omega}}^i \bar{\mathbf{v}}_f^i + \mathbf{A}^i \dot{\mathbf{h}}_f^i)^T (\dot{\mathbf{R}}^j + \mathbf{A}^j \tilde{\boldsymbol{\omega}}^j \bar{\mathbf{u}}^{Pj} + \mathbf{A}^j \dot{\mathbf{h}}^{Pj} - \dot{\mathbf{R}}^i)$ $-\bar{\mathbf{v}}_f^{iT} \tilde{\boldsymbol{\omega}}^i \tilde{\boldsymbol{\omega}}^i \bar{\mathbf{u}}^{Pi} + 2\dot{\mathbf{h}}_f^{iT} \tilde{\boldsymbol{\omega}}^i \bar{\mathbf{u}}^{Pi} + 2\dot{\mathbf{h}}_f^{iT} \dot{\mathbf{h}}^{Pi}$

In the case of spherical joints, universal joints and revolute joints, the constraint location remains in place and the joints can be modeled by changing the constraints of the rotations. Joints such as cylinder and translational joints that enable relative translational movement between bodies are challenging to model due to their varying location, to which the constraint is applied. For flexible bodies, the location of the constraint must be solved for each time step. The location can be found for instance by applying interpolation between the nodes of the joint. Table 3 summarizes the descriptions of the joints and constraint equations with which they can be modeled.

Table 3. Descriptions of joints and basic constraint equations applied to them.

Joint	Illustration	Constraint equations
<p>Spherical joint</p> <p>The spherical joint is the simplest to model and can be described using one of the basic constraints. In the point constraint, the global coordinates of the points must be located overlappingly. The spherical joint constrains three degrees of freedom from the system.</p>		C^s
<p>Universal joint</p> <p>Universal joints can be modeled with a spherical constraint and a type 1 perpendicular constraint. The universal joint removes four degrees of freedom from the system.</p>		C^s C^{d1}
<p>Revolute joint</p> <p>A revolute joint can be adapted from the universal joint by adding another type 1 perpendicularity constraint. The revolute joint removes five degrees of freedom from the system.</p>		C^s C^{d1} C^{d1}
<p>Cylindrical joint</p> <p>The modeling of a cylindrical joint requires two type 1 perpendicular constraints to prevent the relative rotation of the bodies at point P^j, and two type 2 perpendicular constraints to maintain point P^i on the translational axis. The cylindrical joint removes four degrees of freedom from the system.</p>		C^{d1} C^{d1} C^{d2} C^{d2}

<p>Translational joint</p> <p>A translational joint can be derived directly from the cylindrical joints by adding a type 1 perpendicularity constraint. A translational joint removes five degrees of freedom from the system.</p>		C^{d1} C^{d1} C^{d1} C^{d2} C^{d2}
---	--	--

It is important to note that basic constraints can be combined in various other ways than the ones described in Table 3. Due this fact, also more unconventional joints can be described.

3 A DESCRIPTION OF A REAL-TIME SIMULATION

ENVIRONMENT

In real-time simulation, the main challenge is to give an accurate and realistic response from the simulated system while maintaining sufficient number of details in the model. The larger the number of details, the more applicable the simulator is for different purposes. The detailed simulator can be applied to several areas of applications, such as product development, testing and diagnostics, training and research.

In the automotive industry, simulators are widely used in product development. Daimler-Benz tested concepts of all-wheel steering in a simulator [30]. The company found a set of parameters which allow drivers to perform optimally. Later, the set of parameters was verified in a real car. Currently, the most advanced driving simulator is the National Advanced Driving Simulator – NADS [31] – which includes engineering level detailed vehicle dynamics. The NADS simulator is used to enhance product development and analysis with the help of human-in-the-loop prototyping [32]. Generally, using a simulator in product development has many benefits. Several design concepts can be tested due to rapid design changes in the early stages of development. Simulation data can be recorded from the real-time simulator and can be used as input data in non-real-time analysis. Simulation in the early

stages of development is often implemented by omitting the motion platform because in simulators for product development purposes the numerical results are usually the most interesting. A motion platform is needed when the “handling qualities” are under development and for example the vibrations induced by the dynamics of the vehicle are studied.

Training simulation is an important application area for simulators. A survey of driving simulators can be found from Reference [33]. A training simulator may consist of different levels of details. Training on basic controls can be arranged using a procedural simulator without the need for sensing a real operation environment. Advanced handling skills with different operation scenarios can be trained using a high end simulator with a motion platform. When the simulator is used for research purposes, the usage of a motion platform becomes increasingly important. The simulator can be used to study the interaction between machine and operator since a variety of problems related to the human factor can be addressed in a straightforward manner by the simulator. Also several other advantages can be obtained using training simulators. They are more economical, versatile, ecological and safe than the use of an actual machine.

In this chapter, the individual modules of the real-time simulation environment are introduced. The modules of the environment are the dynamics solver, visualization, parser, and server modules. Besides the description of mechanical as well as actuator components and constraints in the dynamics function library, the dynamics solver module comprises the numerical integrator and solution algorithms. The dynamics solver module interacts with the visualization module via the server module. The client-server module enables the interaction between multiple solver and visualization modules while managing run-time simulation.

3.1 Dynamics module

The dynamics solver module, presented in Fig. 7, is used to form and solve equations of motion related to different multibody formulations. The solver library contains numerical integration algorithms and subroutines for solving sets of linear and nonlinear algebraic equations. The collision detection library is optional and may be used to detect possible collisions between chosen graphical objects. The components required for dynamic

formulations are defined in the dynamics function library, and they can be formed using the same functions for each of the multibody formulations.

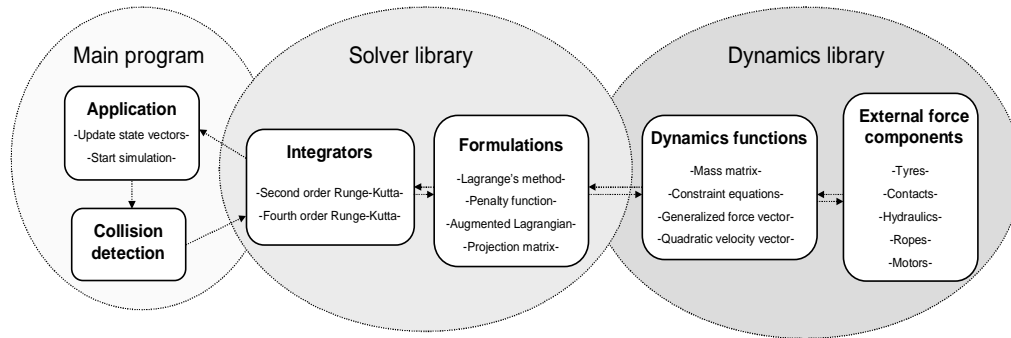


Figure 7. The structure of the dynamics solver module

3.1.1 Description of external force components

In order to apply translational and rotational forces between two bodies, the basic force primitives [16, 17] are defined in the dynamics function library. The forces descriptions from external force components such as tyres, motors, ropes, hydraulics or contacts are passed to these basic primitives and, consequently, force components are applied to the multibody system through the generalized force vector. Accordingly, the generalized forces may be solved using the same approach for different types of external force components. The simulation algorithm may also be extended with minor modifications to new force components.

In order to clarify the description of contact forces, a separate contact module is defined. This module contains information on contacting bodies detected by a collision detection algorithm. Traditionally, three different approaches can be used in the modeling of contacts: the analytical [34], penalty [35] and impulse methods [36]. In this study, contacts are modeled using the penalty method for interpenetration so that the direction of the contact force is perpendicular to the contact plane at the contact point. A virtual spring damper element is temporarily placed at the contact point in order to describe the contact force. Spring and damping constants are defined based on the elasticity of the bodies. Friction forces are

calculated using the friction coefficient and the normal force applied to the contact points of the bodies.

The lumped fluid theory [37] is used in the modeling of the hydraulic circuit. The lumped fluid theory divides a hydraulic circuit into volumes wherein the pressure is assumed to be equally distributed. The valves are modeled employing a semi-empirical approach [38] which allows obtaining the parameters used in the flow equations through the orifices, in many cases from manufacturer catalogues. Usually, the hydraulic system has high nominal frequencies, and for this reason, the time step must be short in order to produce reliable results. Since one time integration algorithm is used for the machine system, the time integration in hydraulics forces the time step in the integration of mechanical components to be small. It is possible to use separate integration routines for mechanics and hydraulics, whereas in this case problems may arise in the synchronization of the submodels.

3.1.2 Collision detection library

In simulations of mechatronic machines, the bodies will inevitably collide. Therefore, it is important that the simulation software has an algorithm for collision detection [39]. The collision detection algorithm is usually straightforward in terms of the mathematical procedure. In the algorithm, the triangularized geometries are tested in the case of contact. Usually, the description of a geometric shape consists of a vast amount of data which makes the collision detection algorithm a time-consuming procedure. Due to this fact, it is often impossible to perform primitive collision testing for all triangles in real time. Therefore, it is necessary to use other methods to exclude bodies and body parts that for certain are not in contact with each other from the primitive level collision testing. Various collision detection hierarchies, such as trees, are normally used for this purpose. In this study, the collision detection tree is based on the structure of the 3ds graphics format. A single 3ds graphic may contain several different meshes. This enables the use of the entire graphic as the main trunk of the collision detection tree and the separate meshes as the branches. For instance, a lifting boom may be divided into two meshes: the front end and back end. Since the back end of the boom does not usually collide with other objects, it will be excluded from testing at the branch level. The front end, however, may come into contact with the cargo and must be

accounted for in the collision test. The collision detection tree used in the simulation environment proceeds in three steps, as follows:

- 1° Collision detection between different shapes (outer limits)
- 2° Collision detection between the different meshes of the colliding shapes
- 3° Primitive collision testing between the points and triangles of the colliding meshes

The use of in-house collision detection algorithms provides full control over the collision detection. For example, the use of the 3ds format allows the utilization of the graphics file in order to determine and control the collision tree shape. Another important feature of in-house collision detection algorithms is that the algorithm can be changed to suit the collision model. At the moment, force descriptions are used to solve the collision response in such a way that several collision points can be used. It is also possible to use the data to calculate a specific impact area or other features required by the collision response.

3.2 Control interface

In this study, a general control interface is used to connect the user to the dynamics model. A certain amount of different primitives which can be chained to each other are employed in the control interface. In real systems, user input may be simplified as a reference value for an actuator. Based on this observation, a chain of primitives can be created in order to connect the user to the dynamics model in a systematic manner, as depicted in Fig. 8.

The physical control devices may consist of either a USB game controller or the control existing in the machine under investigation. The I/O library is used as a driver interface to enable similar use of game controllers and A/D boards. Input primitives define which channel of a given controller is connected to a given force primitive. Each channel can also be connected to several primitives. Primitives also establish the type of the connection and define which input variable of a primitive is connected to the control.

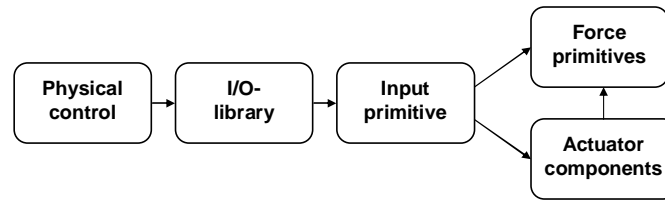


Figure 8. Primitives used in user input definition.

3.3 Visualization

The visualization environment consists of the visualization module, which is based on the object-oriented visualization library, and the physical visualization environment. The visualization library includes separate objects, such as the dialog, display and plotting window objects. The physical simulation environment consists of two different types of displays and an audio system.

3.3.1 Visualization library

In the real-time simulation environment, the visualization library is programmed using C/C++ with OpenGL graphics and OpenAL audio libraries. The visualization library consists of both the graphics engine, used for the actual visualization, and the Graphical User Interface. The object-oriented structure of the visualization library allows full customization of the graphics engine as well as the Graphical User Interface. For example, while utilizing the display object, any number of display windows may be added, for example in a vehicle in which the front window is one display and the side and rear view mirrors are separate displays. The object-oriented structure of the system is illustrated in Fig. 9.

The core of the visualization library is the dialog object. It includes the functions needed in loading the model, creating the virtual world and opening the displays and the Graphical User Interface. The dialog object contains two separate threads. One is used to update the rigid body coordinates, while the other is utilized in the visualization the virtual world. The body coordinates can be acquired utilizing the client-server interface between the visualization and dynamics solver applications. Using separate threads to acquire the body coordinates and draw the scenes enables these actions to be performed in parallel and independently.

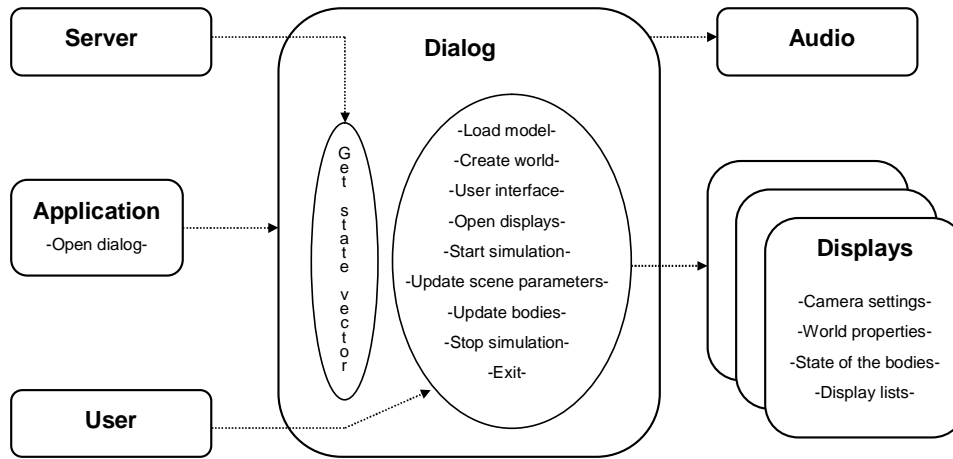


Figure 9. The structure of the visualization library

The graphics engine is capable of reading 3ds format input files, which enables the efficient development of graphics by employing external graphics software. The graphics engine may also be used to draw standard OpenGL environmental options, such as lights and fog. The model definition includes an option for several pre-programmed cameras and other environmental properties needed in a real-time simulation environment. The future development of the graphics engine includes the utilization of the latest technologies from the gaming industry, for example pixel and vertex shaders.

The OpenAL audio library enables the modeling of a 3D audio with a reasonable amount of effort. Separate sounds may be attached to moving objects and played simultaneously. Most of the audio properties, such as the frequency and volume of a sound, can be controlled. For example, when a motor is accelerating, a single sound file can be used just by adjusting the frequency according to the angular velocity of the motor.

3.3.2 Physical visualization system

Different visualization systems can be used with the developed visualization library. In this section, two separate visual projection system are introduced: a three screen 120° system, Fig. 10a, and a stereoscopic system with one screen, Fig. 10b.



Figure 10. a) 120° visualization environment. b) The stereoscopic visualization environment.

The choices regarding the different visualization systems are made based on the different approaches to visual immersion. In the three screen system, the screens surround the operator, thus immersing the operator in the virtual world, whereas the stereoscopic system utilizes depth vision and sensory of movement. To maximize the advantages of these effects, they should be optimized to certain applications. For instance, in a gantry crane it is necessary to evaluate the container distance from the ship hull, and for this reason, depth vision plays an important role. In this case, the stereoscopic view is the most effective. Another example might be revolving machinery, such as an excavator, for which the peripheral vision is vital and the three screen option is a more appropriate choice.

In the three screen version, the projectors are located under the screens pointing backwards. The images are reflected back to the screen by mirrors. This mirrored back projection method cuts down the space required for the visualization system to half of the space needed for a conventional non-mirrored system. Moreover, mirrored back projection ensures that the operator will not create a shadow on the image. The three screens are placed in order to create a 120° viewing angle. This provides the user with peripheral vision.

The stereoscopic visualization system was implemented with two projectors in which polarization is opposite with each other. Two alternative methods can be used to produce the stereoscopic image. The first is to embed the required software into the graphics engine. This is the preferred method, as it leads to extensive control over the image quality. The second

method is to use an external application. For example, this can be accomplished with the Cyviz stereo image splitter, which transforms the normal OpenGL or DirectX signal into a stereo image using external hardware.

3.4 Server library

The server library provides a means for communication between the different modules of the system. The functionality needed for each component is inherited from the base classes of the server library. Due to the vast variety of operations, the server library has been divided into specialized classes, i.e. the Client (User Interface), Server (Solver) and Manager classes. The appropriate class is chosen based on the requirements of the module and, particularly, based on the topology of the computing environment. The server library has been designed so that each model can be described in different kinds of environments in which the number of computers and type of modules may vary.

The responsibilities of each class are defined as follows. Depending on the topology of the computing environment, the Client class decides whether it needs to create the Manager (single computer environment) or connect to an existing one (multiple node environment), and then communicates with it. The Manager creates the needed Server objects, i.e. the actual solvers. The solvers connect back to the Manager, and also form a communication network between each other. The Manager functions as a controlling unit in the environment: it routes messages from the Solver to the User Interface, Motion Platform Controller, and Logging Facility. An example of the whole environment can be seen in Fig. 11, where each box describes a separate instance of a given module, and the dotted arrows depict the most relevant communication paths.

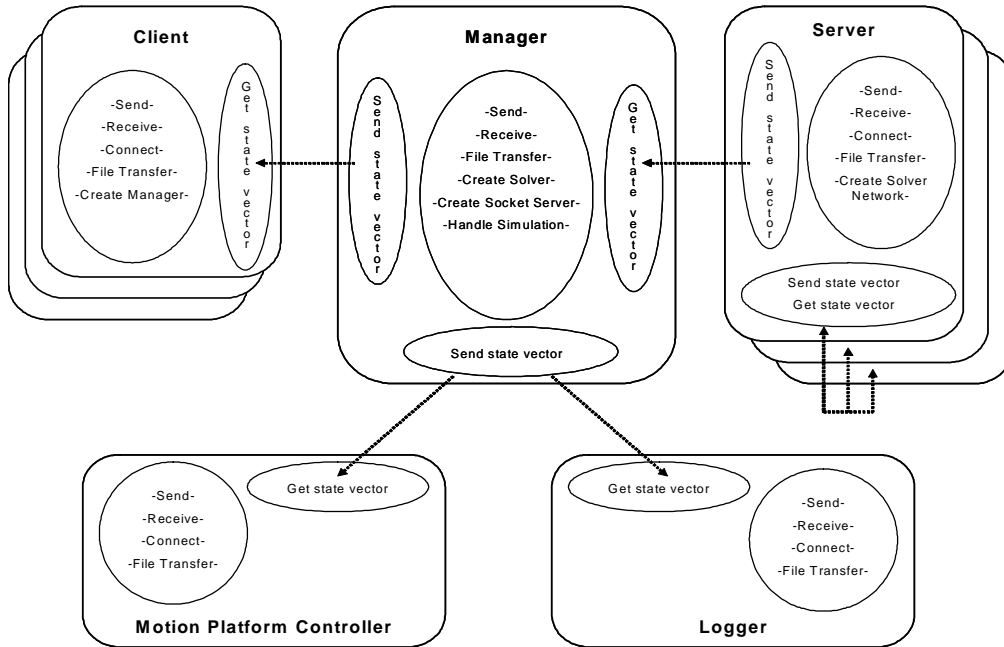


Figure 11. Server library classes and their connections

The functionalities of each class may be divided into start-up, runtime, and finalizing operations. Start-up operations consist of creating actual solver objects, forming communication networks, and transferring model specific files. All communication is treated through standard network socket application programming interfaces. The Manager implements a transmission control protocol socket server, while the other components need only functionality to connect to it. For this purpose, an internet protocol address and network port are needed. This approach removes all restrictions regarding the place of execution of each individual component. Every component may be defined within a single computer or by using separate computers.

During runtime, the rigid body positions and other relevant data from the solver must be continuously conveyed to the User Interface, Logger and Motion Platform Controller. Parameters relevant to the model may be updated to the Solver simultaneously. Thus, during simulation, the Manager needs to listen for incoming messages from all instances in the environment, and react accordingly. The Manager is capable of handling an arbitrary number of connections and can differentiate between each instance.

3.5 Model definition

The modeling of the mechanism dynamics in the real-time simulation environment does not differ considerably from the modeling of non-real-time applications. The main difference is associated with the connection of the user to the system due to the fact that in non-real-time applications it is ignored.

3.5.1 Using an XML parser in model definition

In this study, the real-time simulation model is defined utilizing an XML-based input file. XML was chosen since it provides functions for reading and writing different types of primitives – a tree-structure visual interface with reasonable expandability. The input file described in Fig. 12 includes the definition of simulation parameters, rigid and flexible bodies, constraints, force primitives, user inputs, graphics and textures related to separate bodies and the definition of the virtual world.

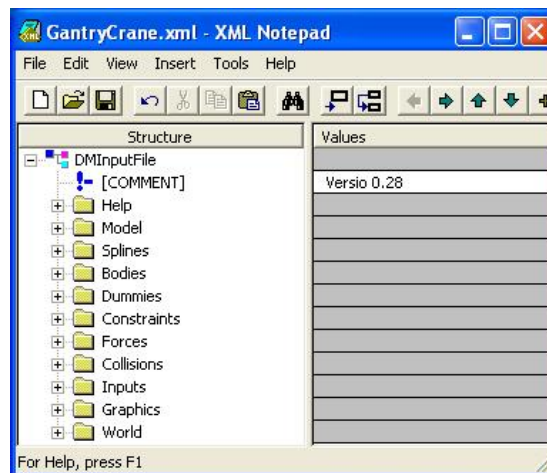


Figure 12. Example view of an input file.

A real-time simulation environment is often used in the simulation of mobile machines. Frequently used force descriptions in mobile machines include tyre and suspension forces, steering, drive-line and brake forces. The input file enables the modeling of tyres based on Pacejka's equations, and straightforward motors in which the maximum torque is limited by

spline-curves. Mobile machine applications, such as cranes, have ropes to lift cargo. Thus, a simple rope model can be defined utilizing the input file.

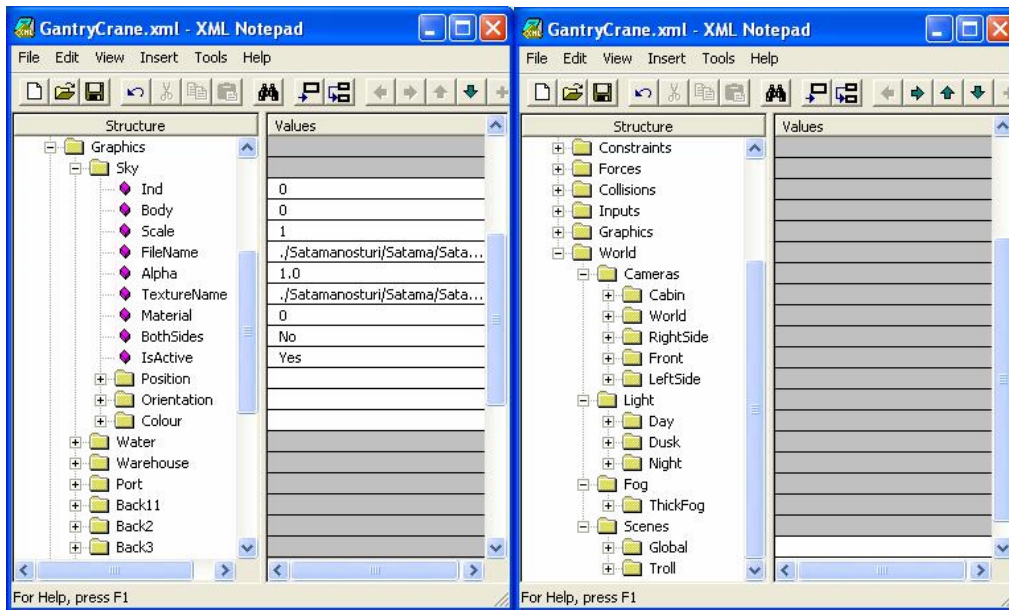


Figure 13. a) Example view of graphics definition. b) Example view of virtual world definition.

Graphics in the 3ds format are connected to reference bodies, as shown in Fig. 13a. The graphics definition includes the position, orientation and scale of a single graphic. Every graphic also has the transparency value and the information of the texture attached to it. A single graphic may be used with several reference bodies without reloading in order to shorten the start-up time and to minimize the amount of memory usage. Accordingly, one texture may be attached to several graphics.

The virtual world depicted in Fig. 13b consists of a definition for the pre-programmed cameras which may be connected to the scenes and the definition of lights and fog. In practice, each scene is a new visualization window. Pre-programmed cameras may be

changed within a scene via mouse commands. The virtual world has one general light source and up to seven spotlights, which may be attached to the reference bodies.

4 CONCLUSIONS

During the past two decades, several real-time simulator approaches for training and product development have been introduced. However, the simulators are mostly case specific and usually tailored to the needs of the vehicle or airplane industries, such as pilot or driver training simulators. Consequently, there is no general purpose software which is implemented in a real-time simulation environment with a three dimensional visualization system, audio system, several control devices of different machines and motion platform.

This thesis introduces a real-time simulator approach that does not depend on the application considered. In this thesis, the dynamic behavior of the mechanical devices is described using multibody simulation approaches. A multibody system may consist of rigid and flexible bodies which are coupled using kinematic joint constraints, while force components are used to describe wide varieties of actuators. The strength of multibody dynamics relies on its ability to describe nonlinearities arising from wearing of the components, friction, large rotations or contact forces in a systematic manner. For this reason, the interfaces between different subsystems such as mechanics, hydraulics and control systems of the industrial and mobile machine can be defined and analyzed as coupled systems. In the real-time simulation environment introduced, the user interfaces, visualization systems and motion platform can be connected to the simulation model. The environment can be utilized in different stages of the product life cycle, such as product development, operator training and failure analysis. In this thesis, the real-time simulation environment based on the multibody approach is used to study the reel mechanism of a paper machine and a gantry crane. These case systems are used to demonstrate the usability of the real-time simulation environment for fault detection purposes and in the context of a training simulator.

One objective of the thesis was to improve constraint modeling methods used in existing multibody simulation algorithms. In the study, structural flexibility was taken into account by extracting deformation modes of the structure with the finite element approach. Particular

attention was paid to the modeling of different kinds of sliding and telescopic joints between flexible bodies, which is a cumbersome or impossible task in several commercial multibody simulation software programs. The method used to describe joints was based on the three primitive constraints that can be combined in a various ways.

The second objective of the thesis was to study the usability of the multibody simulation approach for fault detection purposes. The real-time simulation environment was used to construct a simulation model for the reel mechanism of a paper machine. The model includes the most essential mechanical components, hydraulics circuits, electrical motor models and nip contact models. The simulation model was used to simulate failure scenarios and to study the suitability to detect these scenarios. It can be concluded that using the approach introduced, the supervision of the machine can be focused on the functionality of the entire process instead of the functionality of an individual component.

Furthermore, the real-time simulation environment introduced was used as a platform for a training simulator. In addition to being a platform for vehicle simulators, the environment can be applied to the simulation of several types of mobile and industrial machines. The environment introduced is modular and easily expandable, which systematically and efficiently facilitates studying and testing different modeling approaches and modules, such as motion platforms, visualization environments and additional computational nodes. In this thesis, the real-time simulation environment was used to implement a gantry crane training simulator.

Currently, the development of the real-time simulation environment is extended by more up-to-date multibody simulation approaches. In many cases, the semi-recursive multibody algorithm utilizing the topological structure of the mechanism simulated is proved to be a computationally efficient alternative to its global counterparts. In addition, the environment is used in several projects in the field of product development and operator training. Further development of the real-time simulation environment aims to enhance the properties of the graphical user interface in order to make the modeling procedure more efficient.

5 REFERENCES

- [1] Schiehlen, W., 1997, "Multibody System Dynamics: Roots and Perspectives", *Multibody System Dynamics*, **1**(2), pp. 149-188.
- [2] Shabana A. A., 1997, "Flexible Multibody Dynamics – Review of Past and Recent developments", *Multibody System Dynamics*, **1**(2), pp. 189-222.
- [3] Song J. O., Haug E. J., 1980, "Dynamic Analysis of Planar Flexible Mechanisms", *Computer Methods in Applied Mechanics and Engineering*, **24**(3), pp. 359-381.
- [4] Shabana, A. A., Wehage, R. A., 1983, "Coordinate Reduction Technique for Transient Analysis of Spatial Substructures with Large Angular Rotations", *Journal of Structural Mechanics*, **11**(3), pp. 401-431.
- [5] Yoo, W. S., Haug, E. J., 1986, "Dynamics of Flexible Mechanical Systems Using Vibration and Static Correction Modes", *Journal of Mechanisms, Transmissions and Automation in Design*, **108**(3), pp. 315-322.
- [6] Yoo, W. S., Haug, E. J., 1986, "Dynamics of Articulated Structures Part I. Theory", *Journal of Structural Mechanics*, **14**(1), pp. 105-126.
- [7] Kim, S. S., Vanderploeg, M. J., 1986, "A General and Efficient Method for Dynamic Analysis of Mechanical Systems Using Velocity Transformations", *ASME Journal of Mechanisms, Transmissions and Automation in Design*, **108**(2), pp. 176-182.
- [8] Jerkovsky, W., 1978, "The Structure of Multibody Dynamics Equations", *Journal of Guidance and Control*, **1**(3), pp. 173-182.
- [9] García de Jalón, J., Bayo, E., 1994, *Kinematic and Dynamic Simulation of Multibody Systems – the Real-Time Challenge*, Springer-Verlag, New York.
- [10] Chang, C. W., Shabana, A. A., 1990, "Spatial Dynamics of Deformable Multibody Systems With Variable Kinematic Structure: Part 1 – Dynamic Model", *Journal of Mechanical Design*, **112**(2), pp. 153-159.
- [11] Chang, C. W., Shabana, A. A., 1990, "Spatial Dynamics of Deformable Multibody Systems With Variable Kinematic Structure: Part 2 – Velocity Transformation", *Journal of Mechanical Design*, **112**(2), pp. 1160-1167.
- [12] Lee, B. H., Yoo, W. S., Kwak, B. M., 1993, "Systematic Formulation for Dynamics of Flexible Multi-Body Systems Using the Velocity Transformation Technique",

Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, **207**(4), pp. 231-238.

- [13] Cuadrado, J., Dopico, D., Naya, M. A., Gonzales, M., 2004, "Penalty, Semi-Recursive and Hybrid Methods for MBS Real-Time Dynamics in the Context of Structural Integrators", *Multibody System Dynamics*, **12**(2), pp. 117-132.
- [14] Bae, D. S., Han, J. M., 2000, "An Implementation Method for Constrained Flexible Multibody Dynamics Using a Virtual Body and Joint", *Multibody System Dynamics*, **4**(4), pp. 297-315.
- [15] Ambrósio, J., 2003, "Efficient Kinematic Joint Descriptions for Flexible Multibody Systems Experiencing Linear and Non-Linear Deformations", *International Journal for Numerical Methods in Engineering*, **56**(12), pp. 1771-1793.
- [16] Haug, E. J., 1989, *Computer-Aided Kinematics and Dynamics of Mechanical Systems, Volume I: Basic Methods*, Allyn and Bacon, Massachusetts.
- [17] Nikravesh, P. E., 1988, *Computer-Aided Analysis of Mechanical Systems*, Prentice Hall, New Jersey.
- [18] Shabana, A. A., 2005, *Dynamics of Multibody Systems, (third edition)*, Cambridge University Press.
- [19] Shabana, A. A., 1991, "Constrained Motion of Deformable Bodies", *International Journal for Numerical Methods in Engineering*, **32**(8), pp. 1813-1831.
- [20] Cardona, A., Geradin, M., Doan, J. B., 1991, "Rigid and Flexible Joint Modeling in Multibody Dynamics Using Finite Elements", *Computer Methods in Applied Mechanics and Engineering*, **89**(1-3), pp. 395-418.
- [21] Shabana, A. A., Hwang, Y. L., 1992, "Projection Methods in Flexible Multibody Dynamics. Part I: Kinematics", *International Journal for Numerical Methods in Engineering*, **35**(10), pp. 1927-1939.
- [22] Hwang, R. S., Haug, E. J., 1990, "Translational Joints in Flexible Multibody Dynamics", *Mechanics of Structures and Machines*, **18**(4), pp. 543-564.
- [23] Nikravesh, P. E., Chung, I. S., 1982, "Application of Euler Parameters to the Dynamic Analysis of Three-Dimensional Constrained Mechanical Systems", *Journal of Mechanical Design*, **104**(4), pp. 785-791.
- [24] Serban, R., Haug, E. J., 1998, "Kinematic and Kinetic Derivatives in Multibody System Analysis", *Mechanics of Structures and Machines*, **26**(2), pp. 145-173.

- [25] Bayo, E., García de Jalón, J., Serna, M. A., 1988, "A Modified Lagrangian Formulation for the Dynamic Analysis of Constrained Mechanical Systems", *Computer Methods in Applied Mechanics and Engineering*, **71**(2), pp. 183-195.
- [26] Serna, M. A., Rafael, A., García de Jalón, J., 1982, "Dynamic Analysis of Plane Mechanisms with Lower Pairs in Basic Coordinates", *Mechanism and Machine Theory*, **17**(6), pp. 397-403.
- [27] Wehage, R. A., Haug, E. J., 1982, "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems", *Journal of Mechanical Design*, **104**(1), pp. 247-255.
- [28] Shabana, A. A., 2001, *Computational Dynamics*, John Wiley & Sons, New York.
- [29] Haug, E. J., Yen, J., 1990, "Generalized Coordinate Partitioning Methods for Numerical Integration of Differential-Algebraic Equations of Dynamics", in *Real-Time Integration Methods for Mechanical System Simulation, NATO ASI Series*, **F69**, pp. 97-114.
- [30] Käding, W., Hoffmeyer, F., 1995, "The Advanced Daimler-Benz Simulator", *SAE Paper*, **950175**, pp. 91-98.
- [31] Freeman, J. S., Watson, G., Papelis, Y. E., Lin, T. C., Tayyab, A., Romano, R. A., Kuhl, J. G., 1995, "The Iowa Driving Simulator: An Implementation and Application Overview", *SAE Paper*, **950174**, pp. 113-122.
- [32] Salaani, M. K., Heydinger, G. J., Grygier, P. A., 2003, "Heavy Tractor-Trailer Vehicle Dynamics Modeling for the National Advanced Driving Simulator", *SAE Paper*, **2003-01-0965**, pp. 465-478.
- [33] Weir, D. H., Clark, A. J., 1995, "A Survey of Mid-Level Driving Simulators", *SAE Paper*, **950172**, pp. 86-106.
- [34] Baraff, D., 1989, "Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies", *Computer Graphics*, **23**(3), pp. 223-232.
- [35] Moore, M., Wilhelms, J., 1988, "Collision Detection and Response for Computer Animation", *Computer Graphics*, **22**(4), pp. 289-298.
- [36] Kraus, P. R., Kumar, V., 1997, "Compliant Contact Models for Rigid Body Collisions", *Proceedings – IEEE International Conference on Robotics and Automation*, pp. 1382-1387.
- [37] Watton, J., 1989, *Fluid Power Systems*, Prentice Hall International (UK) Ltd.

- [38] Handroos, H. M., Vilenius, M. J., 1991, "Flexible Semi-Empirical Models for Hydraulic Flow Control Valves", *Journal of Mechanical Design*, **113**(3), pp. 232-238.
- [39] Ericson, C., 2005, *Real-time Collision Detection*, Morgan Kaufmann Publishers, San Francisco CA.

6 APPENDICES

Scientific journal articles included in the thesis.

I

“Description of Joint Constraints in the Floating Frame of Reference Formulation”. Reprinted from the Journal of Multi-Body Dynamics, with kind permission of the Professional Engineering Publishing.

Description of joint constraints in the floating frame of reference formulation

P Korkealaakso*, A Mikkola, T Rantalainen, and A Rouvinen

Department of Mechanical Engineering, Lappeenranta University of Technology, Skinnarilankatu, Lappeenranta, Finland

The manuscript was received on 18 June 2008 and was accepted after revision for publication on 17 February 2009.

DOI: 10.1243/14644193JMBD170

Abstract: This article presents the modelling principles of the joint constraints for flexible multibody systems. The joints are composed using three basic constraint primitives which are derived including their first and second time derivatives as well as the components of the Jacobian matrix. The description of the derived components of constraint primitives can be used to develop a library of kinematic joints to use in multibody codes. In this study, the equations of motion are defined using generalized Newton–Euler equations where the deformations are accounted for by using the floating frame of reference formulation with modal coordinates. The deformation modes used in the floating frame of reference formulation are obtained from the finite-element analysis by employing the lumped mass matrix. Dynamic analysis of a mechanism consisting of rigid and flexible bodies is used to illustrate the validity of the constraint formulation.

Keywords: flexible multibody system, multibody simulation, joint constraints

1 INTRODUCTION

A multibody system consists of rigid and flexible bodies, joint constraints that couple the bodies, and power components describing dampers, springs, and actuators. Depending on the components needed for the multibody model, the dynamic behaviour of the system is described by a system of equations consisting of differential and nonlinear algebraic equations. Multibody dynamics analyses frequently require that structural flexibility is accounted for to reliably predict the behaviour of slender structures under a heavy load. It is noteworthy that even though the topological structure of models remains unchanged in the case of rigid and flexible bodies, modelling of systems with flexible bodies is remarkably challenging regardless of the method used for describing the flexibility. In the literature, the development of multibody system formulations is reviewed in reference [1] for rigid bodies and in references [2] and [3] for flexible bodies.

In multibody applications, the structural flexibility can be accounted for using the floating frame of reference formulation. In the method, the generalized coordinates that define the configuration of the body can be divided into ones that describe the position and orientation of the reference coordinate system and ones that describe deformations with respect to the reference coordinate system. In the floating frame of reference formulation, deformations are usually described using methods based on the finite-element approach. The first general purpose implementation of the floating frame of reference formulation applicable to large flexible multibody systems in planar cases was introduced by Song and Haug [4]. Nevertheless, in that study, the implementation was cumbersome especially for geometrically complex bodies leading to computationally expensive equations of motion due to a large number of nodal coordinates used in the description of flexibility. To reduce the number of coordinates related to flexibility, Shabana [5] extended the floating frame of reference formulation to a three-dimensional mechanism and proposed the use of component mode synthesis to extract the structural vibration modes. This way the set of nodal coordinates from the finite-element method can be replaced by a much lower set of modal coordinates making the numerical solution of the equations of motion more

*Corresponding author: Department of Mechanical Engineering, Lappeenranta University of Technology, Skinnarilankatu 34, P.O. Box 20, Lappeenranta, FI-53851, Finland.
email: pasi.korkealaakso@lut.fi

efficient. However, the general purpose application of the approach was impeded because elements used in the modelling of flexible bodies were included in the solution algorithm leading to the element-specific volume integrals to be solved. Yoo and Haug [6, 7] introduced the use of static correction modes to account for local deformations due to joint constraints and force components. They also derived the equations of motion using the lumped mass finite-element approach. The advantage of the method is that it allows vibration and static correction modes to be solved directly using commercial finite-element software. García de Jalón [8, 9] has developed a method in which a multibody system composed of rigid bodies can be modelled using fully Cartesian natural coordinates. The method accounts for joint constraints using points and vectors, which are also employed in the description of the inertia properties of the bodies. For this reason, the equations of motion can be formulated in computationally efficient ways. Vukasovic [10] extended the natural coordinate approach to the analysis of systems of flexible bodies by using the floating frame of reference formulation.

There are two main approaches to formulate equations of motion for multibody systems: the approach that takes the system topology into account, and the global approach. They differ in the choice of generalized coordinates used in the description of the system configuration. The approach that takes topology into account employs relative coordinates, which allows the kinematic analysis to be accomplished recursively by studying one body at a time in a kinematic chain. The number of generalized coordinates required in the approach is the same as the number of degrees of freedom in open kinematic chains of the system. It is worth mentioning that closed kinematic chains must be opened before kinematic analysis by removing the necessary number of joint constraints. Removed joint constraints must be taken into consideration in the solution of dynamic responses. The method leads to strongly nonlinear equations of motion, which may be difficult to represent in a general form. On the other hand, the matrices to be solved remain small, which makes the method computationally efficient. A general approach to solving rigid body systems based on a velocity transformation matrix [11] was introduced by Kim [12]. He uses global coordinates to describe the system, but the solution itself is obtained with coordinates that describe degrees of freedom of joints. A similar approach for natural coordinates was introduced by García de Jalón and Bayo [9]. Chang and Shabana [13, 14] derived the recursive velocity transformation equations to flexible multibody systems, but they did not demonstrate a systematic approach to execute the velocity transformation. A systematic approach to obtain the velocity transformation matrix for flexible multibody systems was proposed by Lee [15]. In global methods, generalized coordinates

are used to describe the position, orientation and state of deformation of each body. To couple the bodies together, the joint constraints are defined in terms of constraint equations that are a function of the generalized coordinates. Consequently, the equations formulated for each body are of the same form, leading to the systematic assembly of equations of motion for the entire system. The disadvantage of this method is that it leads to extensive systems of equations due to a large number of generalized coordinates, and for this reason the method may be computationally inefficient. However, it has been perceived that global methods may be more efficient than topological methods in the solution of systems consisting of less than 50 generalized coordinates [16]. Commercial multibody dynamics software has traditionally been based on global formulations because of their universal applicability. Examples of such software include LMS Virtual Lab [17, 18] and MSC.ADAMS [19, 20].

Creating a general-purpose multibody algorithm that takes structural flexibility into account is a challenging endeavour. One of the most difficult tasks in the implementation is to create a component library needed for taking kinematic joint constraints into consideration. References [21] and [22] introduce an approach that models joint constraints by using virtual bodies. In this approach, the constraint equations are developed between the flexible and massless rigid body. The advantage of this approach is its applicability to different approaches to describe flexibility. On the other hand, adding virtual bodies increases the computation time compared with methods which derive joint constraints individually for each approach to describe flexibility. The formulation of kinematic joints composed of simple basic constraints in the case of systems of rigid bodies has been discussed in references [17] and [23]. The basic constraint equations for modelling spherical, universal, and revolute joints between flexible bodies have been presented in reference [6]. Shabana [24, 25] used intermediate body fixed joint coordinate systems which are rigidly attached to joint definition points. The joint coordinate systems can be used to derive basic constraint equations including sliding joints with the assumption that the joint axis can be described as a rigid line. Cardona [26] introduced a finite-element formulation for mechanical joints, which can be integrated into finite-element software. In reference [27], the basic joint constraints were used in the context of topological multibody formulation. Hwang [28] presented basic constraint types used with translational joint models, which accounts for the deformation of the axis line. However, to be able to employ traditional solvers of ordinary differential equation (ODE) within the system of equations, the constraint equations must be differentiated twice with respect to time.

It is noteworthy that previous literature does not explicitly present the terms of the Jacobian matrix and

the terms that are related to the second time differentials of basic constraint equations. The objective of this study is to derive explicit expressions for constraint equations that can be applied with the floating frame of reference formulation. In this study, the derivation is accomplished with the help of three basic constraints, which are further utilized in the modelling of mechanical joints. The approach based on three basic constraints provides a general procedure for a wide variety of joints, including spherical, universal, revolute, cylindrical, and translational joints. Joint clearance and damping as well as friction forces associated with joints are not considered in this study. Explicit expressions of constraint equations introduced in this study provide an important instrument for the development for modular multibody simulation software. The basic components derived can also be used in methods based on the system topology when joint constraints are removed to open closed chains. In this study, generalized Newton–Euler equations of motion [29] have been derived according to the principle of virtual work while the local angular accelerations of the frame of reference are integrable variables.

2 FLOATING FRAME OF REFERENCE FORMULATION

The floating frame of reference formulation can be applied to bodies that experience large rigid body displacements and rotations and elastic deformations. The method is based on describing deformations of a flexible body with respect to a frame of reference. The frame of reference, in turn, is employed to describe large displacements and rotations. The deformations of a flexible body with respect to the frame of reference can be described with a number of methods, while in this study, deformation is described using linear deformation modes of the body. Deformation modes can be formulated using a finite-element model of the body. Figure 1 illustrates the position of particle P^i in a deformed body i .

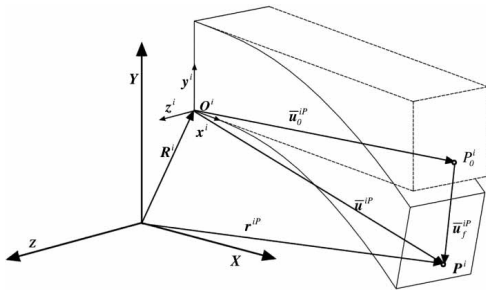


Fig. 1 Position of particle P^i in a global coordinate system

The position of particle P^i of the flexible body i can be described in a global coordinate system with the vector r^{ip} as follows

$$r^{ip} = R^i + A^i \bar{u}^{ip} = R^i + A^i (\bar{u}_0^{ip} + \bar{u}_f^{ip}) \quad (1)$$

where R^i is the position vector of the frame of reference, A^i is the rotation matrix of body i , \bar{u}^{ip} is the position vector of particle P^i within the frame of reference, \bar{u}_0^{ip} is the undeformed position vector of the particle, and \bar{u}_f^{ip} is the displacement of particle P^i within the frame of reference due to the flexibility of body i . In the present study, the rotation matrix A^i is expressed using the Euler parameters $\theta^{Ei} = [\theta_0^{Ei} \theta_1^{Ei} \theta_2^{Ei} \theta_3^{Ei}]^T$ to avoid singular conditions, which are a problem when three rotational parameters are used, such as in the cases of Euler and Bryant angles [30]. The rotation matrix can be written using the Euler parameters as follows

$$A^i = 2 \begin{bmatrix} \frac{1}{2} - (\theta_2^{Ei})^2 - (\theta_3^{Ei})^2 & \theta_1^{Ei} \theta_2^{Ei} - \theta_0^{Ei} \theta_3^{Ei} \\ \theta_1^{Ei} \theta_2^{Ei} + \theta_0^{Ei} \theta_3^{Ei} & \frac{1}{2} - (\theta_1^{Ei})^2 - (\theta_3^{Ei})^2 \\ \theta_1^{Ei} \theta_3^{Ei} - \theta_0^{Ei} \theta_2^{Ei} & \theta_2^{Ei} \theta_3^{Ei} + \theta_0^{Ei} \theta_1^{Ei} \\ \theta_1^{Ei} \theta_3^{Ei} + \theta_0^{Ei} \theta_2^{Ei} & \theta_2^{Ei} \theta_3^{Ei} - \theta_0^{Ei} \theta_1^{Ei} \\ \frac{1}{2} - (\theta_1^{Ei})^2 - (\theta_2^{Ei})^2 & \end{bmatrix} \quad (2)$$

The following mathematical constraint must be taken into consideration when the Euler parameters are applied

$$(\theta_0^{Ei})^2 + (\theta_1^{Ei})^2 + (\theta_2^{Ei})^2 + (\theta_3^{Ei})^2 = 1 \quad (3)$$

The deformation vector \bar{u}_f^{ip} can be described using a linear combination of the deformation modes as follows

$$\bar{u}_f^{ip} = \Psi_R^{ip} q_f^i \quad (4)$$

where Ψ_R^{ip} is the modal transformation matrix whose columns describes translation of particle P^i in assumed deformation modes of the flexible body i [24], and q_f^i is the vector of elastic coordinates. Consequently, the generalized coordinates that uniquely define the position of point P^i can be represented with vector p^i as follows

$$p^{iT} = [R^{iT} \theta^{EiT} q_f^{iT}]^T \quad (5)$$

The velocity of particle P^i can be obtained by differentiating the position description (equation (1)) with respect to time as follows

$$\dot{r}^{ip} = \dot{R}^i - A^i (\dot{\bar{u}}_0^i + \dot{\Psi}_R^{ip} q_f^i) \bar{\omega}^i + A^i \Psi_R^{ip} \dot{q}_f^i \quad (6)$$

where $\bar{\omega}^i$ is the vector of local angular velocities. In equation (6), the generalized velocity vector can be

defined as follows

$$\dot{\mathbf{q}}^{iT} = [\dot{\mathbf{R}}^{iT} \quad \dot{\tilde{\omega}}^{iT} \quad \dot{\mathbf{q}}_f^{iT}]^T \quad (7)$$

By differentiating equation (6) with respect to time, the following formulation for the acceleration of particle P^i can be obtained

$$\ddot{\mathbf{r}}^{iP} = \ddot{\mathbf{R}}^i + \mathbf{A}^i \tilde{\omega}^i \tilde{\omega}^i \tilde{\mathbf{u}}^{iP} + \mathbf{A}^i \dot{\tilde{\omega}}^i \tilde{\mathbf{u}}^{iP} + 2\mathbf{A}^i \tilde{\omega}^i \dot{\tilde{\mathbf{u}}}^{iP} + \mathbf{A}^i \ddot{\tilde{\mathbf{u}}}^{iP} \quad (8)$$

where $\tilde{\omega}^i$ is a skew-symmetric representation of the angular velocity of the body in the frame of reference, $\ddot{\mathbf{R}}^i$ is the vector that defines the translational acceleration of the frame of reference, $\mathbf{A}^i \tilde{\omega}^i \tilde{\omega}^i \tilde{\mathbf{u}}^{iP}$ is the normal component of acceleration, $\mathbf{A}^i \dot{\tilde{\omega}}^i \tilde{\mathbf{u}}^{iP}$ is the tangential component of acceleration, $2\mathbf{A}^i \tilde{\omega}^i \dot{\tilde{\mathbf{u}}}^{iP}$ is the Coriolis component of acceleration and $\mathbf{A}^i \ddot{\tilde{\mathbf{u}}}^{iP}$ is the acceleration of particle P^i due to the deformation of body i .

When deformation modes are used with the floating frame of reference, rotation coordinates due to body deformation are usually ignored. However, to compose all of the basic constraints, rotation coordinates due to body deformation must be accounted for. The orientation at the location of particle P^i within the frame of reference can be expressed using the vector $\tilde{\mathbf{v}}_f^i$ as follows

$$\tilde{\mathbf{v}}_f^i = \mathbf{A}_f^{iP} \tilde{\mathbf{v}}^{iP} \quad (9)$$

where $\tilde{\mathbf{v}}^{iP}$ is a vector that defines the undeformed orientation at the location of particle P^i and \mathbf{A}_f^{iP} is a rotation matrix that describes orientation due to deformation at the location of particle P^i with respect to the reference frame. Note that components in equation (9) are expressed in the reference frame. The rotation matrix \mathbf{A}_f^{iP} can be expressed as follows

$$\mathbf{A}_f^{iP} = \mathbf{I} + \tilde{\boldsymbol{\varepsilon}}^{iP} \quad (10)$$

where \mathbf{I} is a (3×3) unit matrix and $\tilde{\boldsymbol{\varepsilon}}^{iP}$ is a skew symmetric form of the orientation change caused by deformation. Orientation changes due to deformation can be represented in the following way

$$\boldsymbol{\varepsilon}^{iP} = \Psi_\theta^{iP} \mathbf{q}_f^i \quad (11)$$

where Ψ_θ^{iP} is the modal transformation matrix whose columns describes rotation coordinates of point P^i in assumed deformation modes of the flexible body i [24], and \mathbf{q}_f^i is the vector of elastic coordinates.

It is noteworthy that the velocity vector of generalized coordinates contains the translational and angular velocities of the frame of reference of the body. As the angular velocity vector is used in equation (7),

vector $\dot{\mathbf{q}}^i$ cannot be integrated into the position level coordinates to obtain the coordinates expressed in equation (5). For this reason, the new vector which includes angular variables used to describe the orientation of the body is defined

$$\dot{\mathbf{p}}^{iT} = [\dot{\mathbf{R}}^{iT} \quad \dot{\boldsymbol{\theta}}^{EiT} \quad \dot{\mathbf{q}}_f^{iT}]^T \quad (12)$$

The first time derivative of the Euler parameters and the angular velocity vector with respect to the frame of reference of the body has the following linear relation

$$\dot{\boldsymbol{\theta}}^{Ei} = \frac{1}{2} \bar{\mathbf{G}}^i \dot{\boldsymbol{\omega}}^i \quad (13)$$

where the transformation matrix $\bar{\mathbf{G}}^i$ can be expressed as

$$\bar{\mathbf{G}}^i = \begin{bmatrix} -\theta_1^{Ei} & \theta_0^{Ei} & \theta_3^{Ei} & -\theta_2^{Ei} \\ -\theta_2^{Ei} & -\theta_3^{Ei} & \theta_0^{Ei} & \theta_1^{Ei} \\ -\theta_3^{Ei} & \theta_2^{Ei} & -\theta_1^{Ei} & \theta_0^{Ei} \end{bmatrix} \quad (14)$$

Using the transformation matrix $\bar{\mathbf{G}}^i$, the velocity vector can be integrated to obtain the vector \mathbf{p} .

2.1 Inertia of a flexible body

The virtual work of the inertial forces can be represented as follows

$$\delta W^i = \int_{V^i} \rho^i \delta \mathbf{r}^{iP^T} \ddot{\mathbf{r}}^{iP} dV^i \quad (15)$$

where $\delta \mathbf{r}^{iP}$ is the virtual displacement of the position vector of the particle, and $\ddot{\mathbf{r}}^{iP}$ is the acceleration vector of the particle. The virtual displacement of the position vector can be expressed as follows

$$\delta \mathbf{r}^{iP^T} = [\delta \mathbf{R}^i \quad \delta \boldsymbol{\theta}^i \quad \delta \mathbf{q}_f^i] \begin{bmatrix} \mathbf{I} \\ -(\mathbf{A}^i \tilde{\boldsymbol{\omega}}^{iP})^T \\ (\mathbf{A}^i \Psi_R^{iP})^T \end{bmatrix} \quad (16)$$

By substituting the virtual displacement of the position vector (16) into the equation of virtual work of the inertial forces (15) and separating the terms related to acceleration from the terms related quadratically to velocities, the following equation for the virtual work of inertial forces can be obtained

$$\delta W^i = \int_{V^i} \rho^i \delta \mathbf{r}^{iP^T} \ddot{\mathbf{r}}^{iP} dV^i = \delta \mathbf{q} [\mathbf{M}^i \ddot{\mathbf{q}}^i + \mathbf{Q}^{vi}] \quad (17)$$

where \mathbf{M}^i is the mass matrix and \mathbf{Q}^{vi} is the quadratic velocity vector. The mass matrix can be expressed as

follows

$$\mathbf{M}^i = \int_{V^i} \rho^i \begin{bmatrix} \mathbf{I} & -\mathbf{A}^i \tilde{\mathbf{u}}_f^{iP} & \mathbf{A}^i \Psi_R^{iP} \\ \text{sym} & \tilde{\mathbf{u}}^{iP\top} \tilde{\mathbf{u}}^{iP} & -\tilde{\mathbf{u}}^{iP\top} \Psi_R^{iP} \\ & & \Psi_R^{iP\top} \Psi_R^{iP} \end{bmatrix} dV^i \quad (18)$$

and the quadratic velocity vector as follows

$$\mathbf{Q}^i = \int_{V^i} \rho^i \begin{bmatrix} \mathbf{A}^i \tilde{\omega}^i \tilde{\omega}^i \tilde{\mathbf{u}}^{iP} + 2\mathbf{A}^i \tilde{\omega}^i \Psi_R^{iP} \dot{\mathbf{q}}_f^i \\ -\tilde{\mathbf{u}}^{iP\top} \tilde{\omega}^i \tilde{\omega}^i \tilde{\mathbf{u}}^{iP} - 2\tilde{\mathbf{u}}^{iP\top} \tilde{\omega}^i \Psi_R^{iP} \dot{\mathbf{q}}_f^i \\ \Psi_R^{iP\top} \tilde{\omega}^i \tilde{\omega}^i \tilde{\mathbf{u}}^{iP} + 2\Psi_R^{iP\top} \tilde{\omega}^i \Psi_R^{iP} \dot{\mathbf{q}}_f^i \end{bmatrix} dV^i \quad (19)$$

2.2 Generalized forces

Generalized forces express external forces applying in a certain particle point of the body in terms of the generalized coordinates of the body. The vector of generalized forces for body i (\mathbf{Q}^i) consists of the following components

$$\mathbf{Q}_R^i = \sum_{j=1}^{n_F} \mathbf{F}_j^i \quad (20a)$$

$$\mathbf{Q}_\theta^i = \sum_{j=1}^{n_F} \tilde{\mathbf{u}}_j^i \mathbf{A}^{i\top} \mathbf{F}_j^i \quad (20b)$$

$$\mathbf{Q}_{q_f}^i = \sum_{j=1}^{n_F} \Psi_j^{i\top} \mathbf{A}^{i\top} \mathbf{F}_j^i \quad (20c)$$

where \mathbf{F}_j^i is the j th force acting on body i , $\tilde{\mathbf{u}}_j^i$ is a skew symmetric matrix of the location vector of the j th force components, and Ψ_j^i is the terms of the modal matrix associated with the acting node of the j th force. The vector of elastic forces can be represented as follows

$$\mathbf{Q}^i = \begin{bmatrix} 0 \\ 0 \\ \mathbf{K}^i \mathbf{q}_f^i \end{bmatrix} \quad (21)$$

where \mathbf{K}^i is the stiffness matrix for body i . The stiffness matrix \mathbf{K}^i is associated with the modal coordinates and it can be obtained from a standard finite-element stiffness matrix using the modal transformation matrix [24].

3 FORMULATION OF EQUATIONS OF MOTION

Equations of motion can be formulated using the Lagrange approach, in which kinematic constraints are accounted for as augmented algebraic equations with the help of Lagrange multipliers [31]. The method falls into the category of global formulation since it does not make a difference between open and closed kinematic chains, as topological methods do. By using

the same generalized coordinates as in section 2 for the position and orientation of the body, the equation of motion can be written as

$$\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C}_q^\top \boldsymbol{\lambda} = \mathbf{Q}^e + \mathbf{Q}^v - \mathbf{Q}^f \quad (22)$$

where $\ddot{\mathbf{q}}$ is the vector of accelerations of generalized coordinates, \mathbf{M} is the mass matrix, \mathbf{Q}^e is the vector of generalized forces, \mathbf{Q}^v is the quadratic velocity vector, which includes the Coriolis and centrifugal forces that depend quadratically on velocities, \mathbf{C}_q is the Jacobian matrix of constraint equations, \mathbf{Q}^f is the vector of elastic forces, and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. Provided that the origin of the frame of reference is located at the centre of the mass of the rigid body, the translational terms of vector \mathbf{Q}^v are zero. In the case of flexible bodies, however, a corresponding location of the reference frame does not lead to the zero translational terms of vector \mathbf{Q}^v .

In multibody dynamics, different types of joints between bodies are accounted for by kinematic constraints that are functions of generalized coordinates. The vector including the constraint equations can be written as

$$\mathbf{C}(\mathbf{p}, t) = 0 \quad (23)$$

Equations (22) and (23) together compose a system of differential algebraic equations (DAE) that describes the dynamic behaviour of a mechanism [32, 33]. To be able to apply traditional ODE solvers to the system of equations, the constraint equations must be differentiated twice with respect to time

$$\ddot{\mathbf{C}}(\mathbf{p}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = \mathbf{C}_q \ddot{\mathbf{q}} + (\mathbf{C}_q \dot{\mathbf{q}})_q \dot{\mathbf{q}} + 2\mathbf{C}_{q_t} \dot{\mathbf{q}} + \mathbf{C}_{tt} = 0 \quad (24)$$

When considering holonomic systems only, time-dependent constraints are eliminated from the equation. As a result, the final matrix equation describing the system dynamics can be written as

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}_q^\top \\ \mathbf{C}_q & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^e + \mathbf{Q}^v - \mathbf{Q}^f \\ \mathbf{Q}^c \end{bmatrix} \quad (25)$$

where $\mathbf{Q}^c = -(\mathbf{C}_q \dot{\mathbf{q}})_q \dot{\mathbf{q}}$ is the constraint force vector. Due to the differentiation of the constraint equations, the constant terms associated with the constraint equations are eliminated. It is noteworthy that equation (25) satisfies the constraint equations only at the acceleration level. This may lead to the accumulation of errors in kinematic constraints when equations of motion are integrated as forward of time. The problem is usually solved with a constraint stabilization method, which enhances the fulfilment of the constraint equations. Alternatively, the integration may be focused directly on equations (22) and (23) by applying integration methods that are suitable for DAE systems.

4 MODELLING OF CONSTRAINTS IN THE FLOATING FRAME OF REFERENCE FORMULATION

In this section, geometric constraint equations are derived for three basic constraint components, which can be further applied in the modelling of spherical joints, revolute joints, cylindrical joints and translational joints. The terms within the equations of motion that are related to the constraints are formulated so that they can be easily coupled to multibody dynamics codes.

4.1 Basic constraints

Joints in multibody systems can be described as combinations of three basic constraints. These basic constraints are the spherical constraint and two different perpendicularity conditions. The basic constraints for rigid bodies have been presented, e.g. in references [17] and [23]. For flexible bodies, however, there is no comprehensive analytic representation that could describe all of the components in equation (24) that are related to the constraints.

4.1.1 Spherical constraint on two points

The spherical constraint on two points, depicted in Fig. 2, is a simple basic constraint that prevents translational movement between two bodies. The constraint equation can be defined at given points P^i and P^j . This basic constraint removes three degrees of freedom from the system.

The constraint equation associated with points P^i and P^j can be written as follows

$$C^s = R^j + A^j \bar{u}^{jp} - R^i - A^i \bar{u}^{ip} = 0 \tag{26}$$

By differentiating equation (27) with respect to time, the following equation can be obtained

$$\begin{aligned} \dot{C}^s &= C_q^s \dot{q} = \dot{R}^j + A^j \dot{\bar{\omega}}^j \bar{u}^{jp} + A^j \dot{\bar{u}}^{jp} - \dot{R}^i \\ &\quad - A^i \dot{\bar{\omega}}^i \bar{u}^{ip} - A^i \dot{\bar{u}}^{ip} \end{aligned} \tag{27}$$

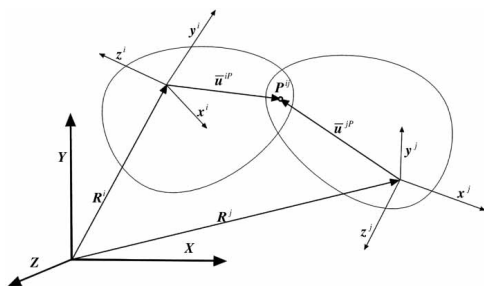


Fig. 2 Spherical constraint on two points

By repeating the differentiation with respect to time, the following equation can be obtained

$$\begin{aligned} \ddot{C}^s &= C_q^s \ddot{q} + (C_q^s \dot{q})_q \dot{q} \\ &= \ddot{R}^j - A^j \ddot{\bar{u}}^j \dot{\bar{\omega}}^j + A^j \dot{\bar{\omega}}^j \dot{\bar{\omega}}^j \bar{u}^{jp} + 2A^j \dot{\bar{\omega}}^j \dot{\bar{u}}^{jp} \\ &\quad + A^j \Psi_R^{jp} \ddot{q}_f^j - \ddot{R}^i + A^i \ddot{\bar{u}}^{ip} \dot{\bar{\omega}}^i - A^i \dot{\bar{\omega}}^i \dot{\bar{\omega}}^i \bar{u}^{ip} \\ &\quad - 2A^i \dot{\bar{\omega}}^i \dot{\bar{u}}^{ip} - A^i \Psi_R^{ip} \ddot{q}_f^i \end{aligned} \tag{28}$$

Based on equation (28), the following terms can be obtained for generalized coordinates related to the translation, orientation and flexibility of the Jacobian matrix

$$C_q^s = [-I \ A^j \bar{u}^{jp} \ -A^i \Psi_R^{ip} \ I \ -A^j \dot{\bar{u}}^{jp} \ A^j \Psi_R^{jp}] \tag{29}$$

Similarly, a vector that includes the quadratic velocity terms can be obtained as follows

$$\begin{aligned} Q^{cs} &= -(C_q \dot{q})_q \dot{q} = -A^j \dot{\bar{\omega}}^j (\dot{\bar{\omega}}^j \bar{u}^{jp} + 2\dot{\bar{u}}^{jp}) \\ &\quad + A^i \dot{\bar{\omega}}^i (\dot{\bar{\omega}}^i \bar{u}^{ip} + 2\dot{\bar{u}}^{ip}) \end{aligned} \tag{30}$$

4.1.2 Perpendicular constraint C^{d1}

The perpendicular constraint (type 1) preventing the rotation of vectors with regard to each other on levels which are not perpendicular to each other. The perpendicularity constraint is illustrated in Fig. 3. This basic constraint can be described with one constraint equation, which removes one degree of freedom from the system.

The constraint equation for a perpendicular constraint of vectors can be written as

$$C^{d1} = v_f^{i^T} v_f^j = \bar{v}_f^{i^T} A^{i^T} A^j \bar{v}_f^j = \bar{v}_f^{i^T} A_f^{ip^T} A^{i^T} A^j A_f^{jp} \bar{v}_f^j = 0 \tag{31}$$

By differentiating the equation with respect to time, the following equation can be obtained

$$\begin{aligned} \dot{C}^{d1} &= C_q^{d1} \dot{q} = \dot{v}_f^{i^T} v_f^j + v_f^{i^T} \dot{v}_f^j \\ &= -\bar{v}_f^{i^T} A^{i^T} A^j \dot{\bar{\omega}}^j \bar{v}_f^i - \bar{v}_f^{j^T} A^{j^T} A^i \dot{\bar{\omega}}^i \bar{v}_f^j + A^{i^T} \Psi_R^{ip} \dot{q}_f^i \\ &\quad - \bar{v}_f^{i^T} A^{i^T} A^j \dot{\bar{\omega}}^j \bar{v}_f^j - \bar{v}_f^{j^T} A^{j^T} A^i \dot{\bar{\omega}}^i \bar{v}_f^i + A^{j^T} \Psi_R^{jp} \dot{q}_f^j \end{aligned} \tag{32}$$

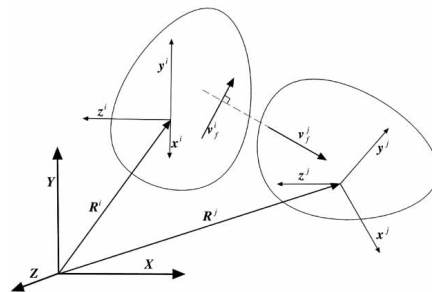


Fig. 3 Type 1 perpendicular constraint

Repeating the differentiation of the equation with respect to time yields

$$\begin{aligned}\dot{C}^{d1} &= C_q^{d1} \dot{\mathbf{q}} + (C_q^{d1})_q \dot{\mathbf{q}} = \ddot{\mathbf{v}}_f^i \mathbf{v}_f^j + \mathbf{v}_f^i \ddot{\mathbf{v}}_f^j + 2\dot{\mathbf{v}}_f^i \dot{\mathbf{v}}_f^j \\ &= -\ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\mathbf{v}}_f^i \dot{\omega}^i - \ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\Psi}_R^i \dot{\mathbf{q}}_f^i - \ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\mathbf{v}}_f^j \dot{\omega}^j \\ &\quad - \ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\mathbf{v}}_f^j \dot{\Psi}_R^j \dot{\mathbf{q}}_f^j + \ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\omega}^i \tilde{\omega}^i \tilde{\mathbf{v}}_f^i + 2\dot{\mathbf{v}}_f^i \dot{\omega}^i \\ &\quad + \ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\omega}^j (\tilde{\omega}^j \tilde{\mathbf{v}}_f^j + 2\dot{\tilde{\mathbf{v}}}_f^j) + 2(\mathbf{A}^i \tilde{\omega}^i \tilde{\mathbf{v}}_f^i + \mathbf{A}^i \tilde{\mathbf{v}}_f^i)^T \\ &\quad \times (\mathbf{A}^j \tilde{\omega}^j \tilde{\mathbf{v}}_f^j + \mathbf{A}^j \dot{\tilde{\mathbf{v}}}_f^j)\end{aligned}\quad (33)$$

Based on equation (33), the following terms can be obtained for generalized coordinates related to the translation, orientation and flexibility of the Jacobian matrix

$$C_q^{d1} = \begin{bmatrix} 0 & -\ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\mathbf{v}}_f^i & -\ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\Psi}_R^i & \dots \\ \dots & 0 & -\ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\mathbf{v}}_f^j & -\ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\Psi}_R^j \end{bmatrix}\quad (34)$$

Correspondingly, the term that includes quadratic velocity terms can be represented as

$$\begin{aligned}Q_c^{d1} &= -(C_q \dot{\mathbf{q}})_q \dot{\mathbf{q}} = -\ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\omega}^j (\tilde{\omega}^j \tilde{\mathbf{v}}_f^i + 2\dot{\tilde{\mathbf{v}}}_f^i) \\ &\quad - \ddot{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\omega}^j (\tilde{\omega}^j \tilde{\mathbf{v}}_f^j + 2\dot{\tilde{\mathbf{v}}}_f^j) \\ &\quad - 2(\mathbf{A}^i \tilde{\omega}^i \tilde{\mathbf{v}}_f^i + \mathbf{A}^i \dot{\tilde{\mathbf{v}}}_f^i)^T (\mathbf{A}^j \tilde{\omega}^j \tilde{\mathbf{v}}_f^j + \mathbf{A}^j \dot{\tilde{\mathbf{v}}}_f^j)\end{aligned}\quad (35)$$

4.1.3 Perpendicular constraint C^{d2}

This perpendicular constraint (type 2) differs from the type 1 in that one of the vectors is defined as constant with respect to body i , whereas the other is defined between the bodies, as shown in Fig. 4. This constraint is also known as the point on plane since it contains one constraint equation eliminating one degree of freedom.

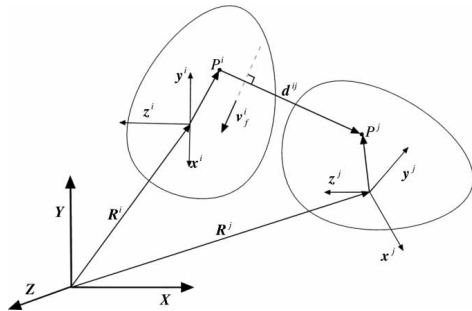


Fig. 4 Type 2 perpendicularity constraint

The constraint equation for a type 2 perpendicularity constraint can be represented as

$$\begin{aligned}C^{d2} &= \mathbf{v}_f^i \mathbf{d}^{ij} = \tilde{\mathbf{v}}_f^i \mathbf{A}^i (\mathbf{R}^j + \mathbf{A}^j \tilde{\mathbf{u}}^{jP} - \mathbf{R}^i - \mathbf{A}^i \tilde{\mathbf{u}}^{iP}) \\ &= \tilde{\mathbf{v}}_f^i \mathbf{A}_f^{iP} \mathbf{A}^i (\mathbf{R}^j + \mathbf{A}^j \tilde{\mathbf{u}}^{jP} - \mathbf{R}^i - \mathbf{A}^i \tilde{\mathbf{u}}^{iP}) = 0\end{aligned}\quad (36)$$

By differentiating the equation with respect to time, the following equation can be obtained

$$\begin{aligned}\dot{C}^{d2} &= C_q^{d2} \dot{\mathbf{q}} = \dot{\mathbf{v}}_f^i \mathbf{d}^{ij} + \mathbf{v}_f^i \dot{\mathbf{d}}^{ij} \\ &= -\mathbf{d}^{ijT} \mathbf{A}^i \tilde{\Psi}_\theta^i \dot{\mathbf{q}}_f^i + \tilde{\mathbf{v}}_f^i \mathbf{A}^i \dot{\mathbf{R}}^j - \tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\mathbf{u}}^{jP} \dot{\omega}^j \\ &\quad + \tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^j \tilde{\Psi}_R^j \dot{\mathbf{q}}_f^j - \tilde{\mathbf{v}}_f^i \mathbf{A}^i \dot{\mathbf{R}}^i + (\tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\mathbf{u}}^{iP}) \\ &\quad - \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i \dot{\omega}^i - \tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\Psi}_R^i \dot{\mathbf{q}}_f^i\end{aligned}\quad (37)$$

Repeating the differentiation with respect to time leads to

$$\begin{aligned}\ddot{C}^{d2} &= C_q^{d2} \ddot{\mathbf{q}} + (C_q^{d2})_q \dot{\mathbf{q}} = \ddot{\mathbf{v}}_f^i \mathbf{d}^{ij} + \mathbf{v}_f^i \ddot{\mathbf{d}}^{ij} + 2\dot{\mathbf{v}}_f^i \dot{\mathbf{d}}^{ij} \\ &= -\ddot{\mathbf{v}}_f^i \mathbf{A}^i \dot{\mathbf{R}}^i + (\tilde{\mathbf{v}}_f^i \tilde{\mathbf{u}}^{iP} - \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i) \dot{\omega}^i - (\tilde{\mathbf{v}}_f^i \tilde{\Psi}_R^i \\ &\quad + \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\Psi}_\theta^i) \dot{\mathbf{q}}_f^i + \tilde{\mathbf{v}}_f^i \mathbf{A}^i \dot{\mathbf{R}}^j - \tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^j \tilde{\mathbf{u}}^{jP} \dot{\omega}^j \\ &\quad + \tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^j \tilde{\Psi}_R^j \dot{\mathbf{q}}_f^j + \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\omega}^i (\tilde{\omega}^i \tilde{\mathbf{v}}_f^i + 2\dot{\tilde{\mathbf{v}}}_f^i) \\ &\quad + \tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^j \tilde{\omega}^j (\tilde{\omega}^j \tilde{\mathbf{u}}^{jP} + 2\dot{\tilde{\mathbf{u}}}_f^{jP}) + 2(\mathbf{A}^i \tilde{\omega}^i \tilde{\mathbf{v}}_f^i \\ &\quad + \mathbf{A}^i \tilde{\mathbf{v}}_f^i)^T (\mathbf{R}^j + \mathbf{A}^j \tilde{\omega}^j \tilde{\mathbf{u}}^{jP} + \mathbf{A}^j \dot{\tilde{\mathbf{u}}}_f^{jP} - \dot{\mathbf{R}}^i) \\ &\quad + \tilde{\mathbf{v}}_f^i \tilde{\omega}^i \tilde{\mathbf{u}}^{iP} + 2\tilde{\mathbf{v}}_f^i \tilde{\mathbf{v}}_f^i \tilde{\omega}^i \tilde{\mathbf{u}}^{iP} + 2\tilde{\mathbf{v}}_f^i \tilde{\mathbf{v}}_f^i \tilde{\mathbf{u}}^{iP}\end{aligned}\quad (38)$$

Table 1 Partial derivatives for basic constraints

	C^s	C^{d1}	C^{d2}
C_{R^i}	$-\mathbf{I}$	0	$-\tilde{\mathbf{v}}_f^i \mathbf{A}^i$
$C_{\dot{\omega}^i}$	$\mathbf{A}^i \tilde{\mathbf{u}}^{iP}$	$-\tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\mathbf{v}}_f^i$	$\tilde{\mathbf{v}}_f^i \tilde{\mathbf{u}}^i - \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i$
$C_{\dot{\mathbf{q}}_f^i}$	$-\mathbf{A}^i \tilde{\Psi}_R^i$	$-\tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\Psi}_R^i$	$-\tilde{\mathbf{v}}_f^i \tilde{\Psi}_R^i - \mathbf{d}^{ijT} \mathbf{A}^i \tilde{\mathbf{v}}_f^i \tilde{\Psi}_\theta^i$
C_{R^j}	\mathbf{I}	0	$\tilde{\mathbf{v}}_f^i \mathbf{A}^i$
$C_{\dot{\omega}^j}$	$-\mathbf{A}^j \tilde{\mathbf{u}}^{jP}$	$-\tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^j \tilde{\mathbf{v}}_f^j$	$-\tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^j \tilde{\mathbf{u}}^{jP}$
$C_{\dot{\mathbf{q}}_f^j}$	$\mathbf{A}^j \tilde{\Psi}_R^j$	$-\tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^j \tilde{\Psi}_R^j$	$\tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^j \tilde{\Psi}_R^j$

Table 2 Components of the constraint force vector related to basic constraints

	Q_c
Q_c^s	$-\mathbf{A}^j \tilde{\omega}^j (\tilde{\omega}^j \tilde{\mathbf{u}}^{jP} + 2\dot{\tilde{\mathbf{u}}}_f^{jP}) + \mathbf{A}^i \tilde{\omega}^i (\tilde{\omega}^i \tilde{\mathbf{u}}^{iP} + 2\dot{\tilde{\mathbf{u}}}_f^{iP})$
Q_c^{d1}	$-\tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^i \tilde{\omega}^j (\tilde{\omega}^j \tilde{\mathbf{v}}_f^i + 2\dot{\tilde{\mathbf{v}}}_f^i) - \tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^j \tilde{\omega}^j (\tilde{\omega}^j \tilde{\mathbf{v}}_f^j + 2\dot{\tilde{\mathbf{v}}}_f^j) \\ - 2(\mathbf{A}^i \tilde{\omega}^i \tilde{\mathbf{v}}_f^i + \mathbf{A}^i \dot{\tilde{\mathbf{v}}}_f^i)^T (\mathbf{A}^j \tilde{\omega}^j \tilde{\mathbf{v}}_f^j + \mathbf{A}^j \dot{\tilde{\mathbf{v}}}_f^j)$
Q_c^{d2}	$-\mathbf{d}^{ijT} \mathbf{A}^i \tilde{\omega}^i (\tilde{\omega}^i \tilde{\mathbf{v}}_f^i + 2\dot{\tilde{\mathbf{v}}}_f^i) - \tilde{\mathbf{v}}_f^i \mathbf{A}^i \mathbf{A}^j \tilde{\omega}^j (\tilde{\omega}^j \tilde{\mathbf{u}}^{jP} + 2\dot{\tilde{\mathbf{u}}}_f^{jP}) \\ - 2(\mathbf{A}^i \tilde{\omega}^i \tilde{\mathbf{v}}_f^i + \mathbf{A}^i \dot{\tilde{\mathbf{v}}}_f^i)^T (\mathbf{R}^j + \mathbf{A}^j \tilde{\omega}^j \tilde{\mathbf{u}}^{jP} + \mathbf{A}^j \dot{\tilde{\mathbf{u}}}_f^{jP} - \dot{\mathbf{R}}^i) \\ - \tilde{\mathbf{v}}_f^i \tilde{\omega}^i \tilde{\mathbf{u}}^{iP} + 2\tilde{\mathbf{v}}_f^i \tilde{\omega}^i \tilde{\mathbf{u}}^{iP} + 2\tilde{\mathbf{v}}_f^i \tilde{\mathbf{u}}^{iP}$

Table 3 Descriptions of joints and basic constraint equations applied to them

Joint	Illustration	Constraint equations
<p><i>Spherical joint</i> The spherical joint is the simplest to model and can be described using one of the basic constraints. In the point constraint, the global coordinates of the points must be located overlapping. The spherical joint constrains three degrees of freedom from the system.</p>		C^S
<p><i>Universal joint</i> Universal joints can be modelled with a spherical constraint and a type 1 perpendicular constraint. The universal joint removes four degrees of freedom from the system.</p>		C^S C^{d1}
<p><i>Revolute joint</i> A revolute joint can be adapted from the universal joint by adding another type 1 perpendicularity constraint. The revolute joint removes five degrees of freedom from the system.</p>		C^S C^{d1} C^{d1}
<p><i>Cylindrical joint</i> The modelling of a cylindrical joint requires two type 1 perpendicular constraints to prevent the relative rotation of the bodies at point P^j, and two type 2 perpendicular constraints to maintain point P^i on the translational axis. The cylindrical joint removes four degrees of freedom from the system.</p>		C^{d1} C^{d1} C^{d2} C^{d2}
<p><i>Translational joint</i> A translational joint can be derived directly from the cylindrical joints by adding a type 1 perpendicularity constraint. A translational joint removes five degrees of freedom from the system.</p>		C^{d1} C^{d1} C^{d1} C^{d2} C^{d2}

Based on equation (38), the following terms are obtained for generalized coordinates related to the translation, orientation and flexibility of the Jacobian matrix

$$C_q^{d2} = \begin{bmatrix} -\bar{\mathbf{v}}_f^i \mathbf{A}^i \bar{\mathbf{v}}_f^i \bar{\mathbf{u}}_f^i - \mathbf{d}^{ij} \mathbf{A}^i \bar{\mathbf{v}}_f^i \\ -\bar{\mathbf{v}}_f^i \Psi_R^{ip} - \mathbf{d}^{ij} \mathbf{A}^i \bar{\mathbf{v}}_f^i \Psi_\theta^{ip} \dots \dots \bar{\mathbf{v}}_f^i \mathbf{A}^i \\ -\bar{\mathbf{v}}_f^i \mathbf{A}^i \bar{\mathbf{u}}_f^i \bar{\mathbf{u}}_f^i \bar{\mathbf{v}}_f^i \mathbf{A}^i \bar{\mathbf{u}}_f^i \Psi_R^{ip} \end{bmatrix} \quad (39)$$

Correspondingly, the term that contains quadratic velocity terms can be expressed as follows

$$\begin{aligned} \mathbf{Q}^{cd2} = & -(\mathbf{C}_q \dot{\mathbf{q}})_q \dot{\mathbf{q}} = -\mathbf{d}^{ij} \mathbf{A}^i \bar{\omega}^i (\bar{\omega}^i \bar{\mathbf{v}}_f^i + 2\dot{\bar{\mathbf{v}}}_f^i) \\ & - \bar{\mathbf{v}}_f^i \mathbf{A}^i \bar{\omega}^i \bar{\omega}^i (\bar{\omega}^i \bar{\mathbf{u}}_f^i + 2\dot{\bar{\mathbf{u}}}_f^i) - 2(\mathbf{A}^i \bar{\omega}^i \bar{\mathbf{v}}_f^i \\ & + \mathbf{A}^i \bar{\mathbf{v}}_f^i)^T (\dot{\mathbf{R}}^j + \mathbf{A}^j \bar{\omega}^j \bar{\mathbf{u}}_f^j + \mathbf{A}^j \dot{\bar{\mathbf{u}}}_f^j - \dot{\mathbf{R}}^j) \\ & - \bar{\mathbf{v}}_f^i \bar{\omega}^i \bar{\omega}^i \bar{\mathbf{u}}_f^i + 2\dot{\bar{\mathbf{v}}}_f^i \bar{\omega}^i \bar{\mathbf{u}}_f^i + 2\dot{\bar{\mathbf{v}}}_f^i \dot{\bar{\mathbf{u}}}_f^i \end{aligned} \quad (40)$$

4.2 Modelling of joints based on basic constraints

In this section, the basic joint types used in multi-body dynamics modelling applying the basic constraints presented above are introduced. With different combinations of basic constraints, it is possible to model any joint. Table 1 summarizes partial derivatives with regard to generalized coordinates for each basic constraint.

Table 2 presents the components of the constraint force vector related to basic constraints.

In the case of spherical joints, universal joints and revolute joints, the constraint location remains in place and the joints can be modelled by changing the constraints of the rotations. Joints such as cylinder and translational joints that enable the relative translational movement between bodies are challenging to model due to their varying location to which the constraint is applied. For flexible bodies, the location of the constraint must be solved for each time step. The location can be found for instance by applying interpolation between the nodes of the joint. Table 3 displays descriptions of the joints and constraint equations with which they can be modelled.

It is important to note that basic constraints can be combined in various other ways than the ones described in Table 3. This allows to express joints that are out of the line of conventional joints. This feature of basic constraints will be further explained in the section 5.

5 NUMERICAL EXAMPLES

To evaluate the functionality of the kinematic constraints, two numerical examples, namely the crank mechanism and the slider mechanism, are studied.

The validation of the kinematic constraints is examined on the basis of the energy balances. In addition, the errors in the kinematic constraints of the crank mechanism are plotted during simulation. It is noteworthy that possible constraint violations or implementation flaws can be easily observed as a fluctuation of the total energy of a multibody system.

5.1 Computational algorithm

For a system of n_b bodies and n_c kinematic constraint equations, the computational algorithm can be outlined as follows.

1. Given an initial condition for $\mathbf{p} = [\mathbf{p}^1 \ \mathbf{p}^2 \ \dots \ \mathbf{p}^{n_b}]^T$ and $\dot{\mathbf{q}} = [\dot{\mathbf{q}}^1 \ \dot{\mathbf{q}}^2 \ \dots \ \dot{\mathbf{q}}^{n_b}]^T$.
2. Start numerical integration routine:
 - (a) Compute $\dot{\mathbf{p}} = [\dot{\mathbf{p}}^1 \ \dot{\mathbf{p}}^2 \ \dots \ \dot{\mathbf{p}}^{n_b}]^T$ using equation (12).
 - (b) Compute $\mathbf{M} = \text{diag}(\mathbf{M}^1 \ \mathbf{M}^2 \ \dots \ \mathbf{M}^{n_b})$ and $\mathbf{Q}^p = [\mathbf{Q}^{p1} \ \mathbf{Q}^{p2} \ \dots \ \mathbf{Q}^{p n_b}]^T$ using equations (18) and (19) and substitute results to equation (25).
 - (c) Compute terms of $\mathbf{C}_q = \begin{bmatrix} \frac{\partial C_1}{\partial \mathbf{q}} & \frac{\partial C_2}{\partial \mathbf{q}} & \dots & \frac{\partial C_{n_c}}{\partial \mathbf{q}} \end{bmatrix}^T$ and $\mathbf{Q}^c = [\mathbf{Q}^{c1} \ \mathbf{Q}^{c2} \ \dots \ \mathbf{Q}^{c n_c}]^T$ for the kinematic constraints using required joint dependent equations (29), (30), (34), (35), (39) and (40) and substitute results to equation (25).
 - (d) Compute external and internal force components $\mathbf{Q}^e = [\mathbf{Q}^{e1} \ \mathbf{Q}^{e2} \ \dots \ \mathbf{Q}^{e n_b}]^T$ and $\mathbf{Q}^f = [\mathbf{Q}^{f1} \ \mathbf{Q}^{f2} \ \dots \ \mathbf{Q}^{f n_b}]^T$ from equations (20) and (21) and substitute results to equation (25).
 - (e) Solve system of linear equation (25) for $\ddot{\mathbf{q}}$ and λ .
 - (f) Integrate $\ddot{\mathbf{q}}$ and $\dot{\mathbf{p}}$ to obtain $\dot{\mathbf{q}}$ and \mathbf{p} .
 - (g) Return to step (a) until desired simulation time is reached.

5.2 Crank mechanism

As a simulation example, a simple three-dimensional crank mechanism, presented in Fig. 5, will be examined. The crank mechanism consists of both flexible and rigid bodies. The flexible bodies are crank *A* and lever *B*. Four three-dimensional beam elements are used in the modelling of each flexible body. Both flexible bodies are assumed to have solid square sections of a thickness of 5 mm. The flexible bodies are 0.4 and 0.8 m long, respectively. The material properties of the crank and lever are as assumed to be the following: Young's modulus $E = 210$ MPa, Poisson's ratio $\nu = 0.3$, and density $\rho = 7801$ kg/m³. The two lowest orthonormal Craig-Bampton eigenmodes in the *y*-direction are used in the description of flexible bodies. The crank block is modelled as a rigid body with a mass of 0.1 kg, while principal moments of inertia are set to 2.0×10^{-4} kg/m². A torque of 0.5 Nm is applied on the

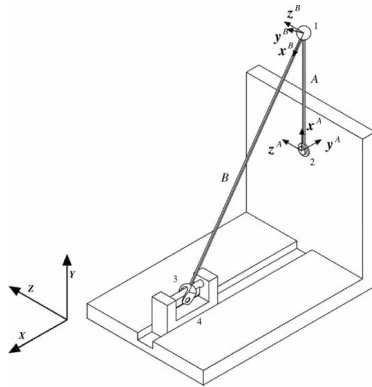


Fig. 5 Simulated crank mechanism

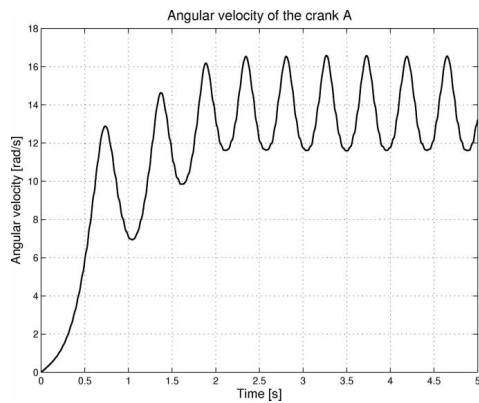


Fig. 6 Angular velocity of crank A

crank around the global x -axis for 2 s, after which the torque will be removed.

The assembled crank mechanism in Fig. 5 includes the following four joints.

1. Spherical joint between crank A and lever B.
2. Revolute joint between crank A and the framework.
3. Universal joint between lever B and the crank block.
4. Translational joint between the framework and the crank block.

Figures 6 and 7 show the angular velocity of crank A and the deformation of the middle point of lever B in the y -direction in relation to the frame of reference as a function of time.

The implementation of the joint constraints was examined through Fig. 8, which presents the errors of three constraint equations used in spherical joint between flexible bodies during simulation.

The validity of the dynamic behaviour of the system can be examined on the basis of the energy

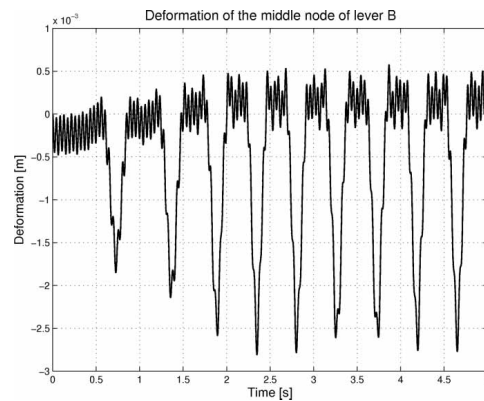


Fig. 7 Deformation of the middle point of lever B in the direction of the local y -coordinate

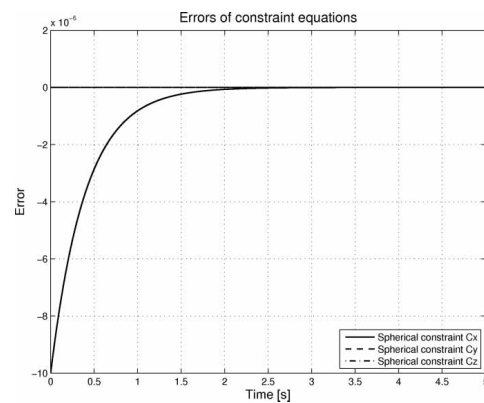


Fig. 8 Errors of constraint equations of spherical joint in x -, y - and z -direction

balance of the system. Figure 9 includes diagrams on kinetic energy, strain energy, work done by external forces and potential energy due to gravitation during the simulation of the crank mechanism. The dynamic response of the crank mechanism is calculated employing equation (25). The computation algorithm used in the dynamic analysis is described in detail in reference [34]. The fourth-order Runge–Kutta numerical integration procedure with a time step of 0.5 ms is used in the dynamic analysis. The frequently applied Baumgarte stabilization is employed in the numerical integration procedure to minimize constraint errors in the position coordinates [24, 35]. It is noteworthy that, according to Fig. 9, the total energy of the system is constant. This is due to the fact that constraints are not violated during the simulation. According to Fig. 8, the errors of the constraint equations are greatest at the beginning and decrease

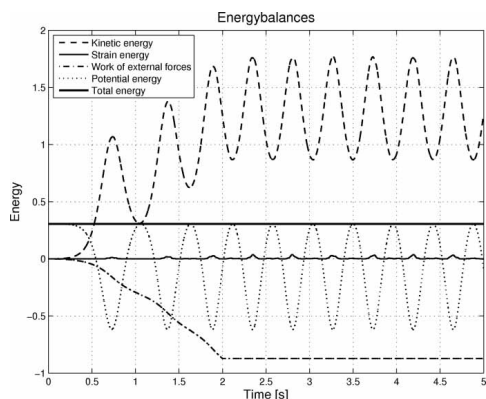


Fig. 9 Energy balances during simulation

as the simulation continues. This is due to the fact that at the beginning, the initial positioning of the bodies with regard to each other may not be accurate, and the stabilized constraint forces the bodies into place.

5.3 Slider mechanism

Sliding joints are commonly used joint types in Cranes with telescopic booms, hydraulic cylinders or mechanisms with guide bars. When the constrained bodies are flexible, it is important to note that the changing contact point must be tracked to account for the deformation of translation axes. The tracking of the contact point is implemented using interpolation polynomials which define the axes of translation. To clarify the case, the telescopic slider mechanism depicted in Fig. 10 is studied as a second numerical example.

The slider mechanism consists of two flexible beams, which are constrained using two contact points. Both the beams are modelled using the same dimensions and material properties as the body B in the former case system. Beams interact using a spring force component with a spring coefficient of 1.0 N/m

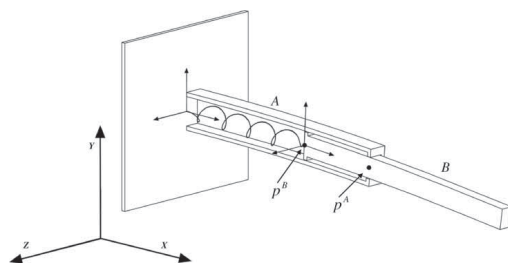


Fig. 10 Flexible slider mechanism

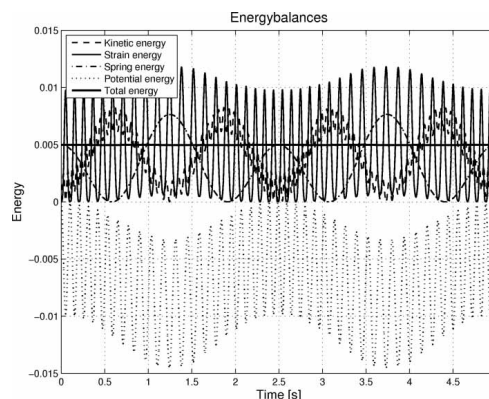


Fig. 11 Energy balances during simulation

and an initial compression of 0.1 m . The body A is constrained to the framework using a bracket joint. The beams are constrained using two type 2 perpendicular constraints at the location of nodes P^A and P^B associated with the bodies A and B , respectively. In addition, one type 1 perpendicular constraint at location P^A is used to prevent rotation around the sliding axel. Due to the fact that the locations of the nodes P^A and P^B are known, the corresponding contact point in the bodies B and A can be solved. The terms of the modal matrix in the contact point are then interpolated from the terms of the neighbouring nodes.

The dynamical behaviour is solved using the same solver settings as in the former case. The validity of the method is again examined on the basis of the energy balance of the system, depicted in Fig. 11.

According to Fig. 11, the total energy is constant throughout the simulation and is equal to the initial spring potential energy due to compression of the spring. Consequently, no excessive energy develops due to the constraint forces, which indicates that the constraints are modelled correctly.

6 CONCLUSIONS

This study presented the basic constraints and their first and second time derivatives, and the terms related to the Jacobian matrix for flexible bodies. With these components, it is possible to model basic joint types for flexible bodies. The basic components enable supplementing the modular multibody dynamics simulation code with a basic constraint library, which in turn allows the automatic formulation of the Jacobian matrix and the vector of constraint forces for the entire system for numerical solution. In addition, these components may be applied to reclosing the kinematics chain that has been opened due to the use of recursive methods.

The use of basic joint types was illustrated with a numerical example. The mechanics under consideration consists of both flexible and rigid bodies, and joints that connect them. The purpose of the example was to examine joint constraint errors and the energy balance of the system. The results indicate that the joint constraints presented in the study perform correctly.

The floating frame of reference formulation applied in this study is often used with component mode synthesis to improve computation performance. In this approach, the deformations of flexible bodies are defined using modes and corresponding modal coordinates. For this reason, attention must be paid to the selection of deformation modes. Accordingly, the selection of deformation modes for different types of joints would be a topic for future research.

REFERENCES

- 1 Schiehlen, W. Multibody system dynamics: roots and perspectives. *Multibody Syst. Dyn.*, 1997, **1**(2), 149–188.
- 2 Shabana, A. A. Flexible multibody dynamics – review of past and recent developments. *Multibody Syst. Dyn.*, 1997, **1**(2), 189–222.
- 3 Wasfy, T. M. and Noor, A. K. Computational strategies for flexible multibody systems. *Appl. Mech. Rev.*, 2003, **56**(6), 553–613.
- 4 Song, J. O. and Haug, E. J. Dynamic analysis of planar flexible mechanisms. *Comput. Methods Appl. Mech. Eng.*, 1980, **24**(3), 359–381.
- 5 Shabana, A. A. and Wehage, R. A. Coordinate reduction technique for transient analysis of spatial substructures with large angular rotations. *J. Struct. Mech.*, 1983, **11**(3), 401–431.
- 6 Yoo, W. S. and Haug, E. J. Dynamics of flexible mechanical systems using vibration and static correction modes. *J. Mech. Transm. Autom. Des.*, 1986, **108**(3), 315–322.
- 7 Yoo, W. S. and Haug, E. J. Dynamics of articulated structures part I. Theory. *J. Struct. Mech.*, 1986, **14**(1), 105–126.
- 8 García de Jalón, J., Unda, J., Avello, A., and Jiménez, J. M. Dynamic analysis of three-dimensional mechanisms in ‘natural’ coordinates. *ASME J. Mech., Transm. Autom. Des.*, 1987, **109**(4), 460–465.
- 9 García de Jalón, J. and Bayo, E. *Kinematic and dynamic simulation of multibody systems – the real-time challenge*, 1994, p. 440 (Springer-Verlag, New York).
- 10 Vukasovic, N., Celigüeta, J. T., García de Jalón, J., and Bayo, E. Flexible multibody dynamics based on a fully Cartesian system of support coordinates. *J. Mech. Des.*, 1993, **115**(2), 294–299.
- 11 Jerkovsky, W. The structure of multibody dynamics equations. *J. Guid. Control*, 1978, **1**(3), 173–182.
- 12 Kim, S. S. and Vanderploeg, M. J. A general and efficient method for dynamic analysis of mechanical systems using velocity transformations. *ASME J. Mech. Transm. Autom. Des.*, 1986, **108**(2), 176–182.
- 13 Chang, C. W. and Shabana, A. A. Spatial dynamics of deformable multibody systems with variable kinematic structure: part 1 – dynamic model. *J. Mech. Des.*, 1990, **112**(2), 153–159.
- 14 Chang, C. W. and Shabana, A. A. Spatial dynamics of deformable multibody systems with variable kinematic structure: part 2 – velocity transformation. *J. Mech. Des.*, 1990, **112**(2), 160–167.
- 15 Lee, B. H., Yoo, W. S., and Kwak, B. M. Systematic formulation for dynamics of flexible multibody systems using the velocity transformation technique. *Proc. IMechE, Part C: J. Mechanical Engineering Science*, 1993, **207**(4), 231–238. DOI: 10.1243/PIME_PROC_1993_207_123_02.
- 16 Cuadrado, J., Dopico, D., Naya, M. A. and Gonzales, M. Penalty, semi-recursive and hybrid methods for MBS real-time dynamics in the context of structural integrators. *Multibody Syst. Dyn.*, 2004, **12**(2), 117–132.
- 17 Haug, E. J. *Computer-aided kinematics and dynamics of mechanical systems, volume I: basic methods*, 1989, p. 498 (Allyn and Bacon, Massachusetts).
- 18 Smith, R. C. and Haug, E. J. DADS – dynamic analysis and design system. In *Multibody systems handbook*, 1990, pp. 161–179 (Springer-Verlag, Berlin).
- 19 Orlandea, N., Chace, M. A., and Calahan, D. A. A sparsity-oriented approach to the dynamic analysis and design of mechanical systems – part 1. *J. Eng. Ind. B*, 1977, **99**(3), 773–779.
- 20 Orlandea, N., Calahan, D. A., and Chace, M. A. A sparsity-oriented approach to the dynamic analysis and design of mechanical systems – part 2. *J. Eng. Ind. B*, 1977, **99**(3), 780–784.
- 21 Bae, D. S., Han, J. M., and Choi, J. H. An implementation method for constrained flexible multibody dynamics using a virtual body and joint. *Multibody Syst. Dyn.*, 2000, **4**(4), 297–315.
- 22 Ambrósio, J. Efficient kinematic joint descriptions for flexible multibody systems experiencing linear and non-linear deformations. *Int. J. Numer. Methods Eng.*, 2003, **56**(12), 1771–1793.
- 23 Nikravesh, P. E. *Computer-aided analysis of mechanical systems*, 1988, p. 370 (Prentice Hall, New Jersey).
- 24 Shabana, A. A. *Dynamics of multibody systems*, 3rd edition, 2005, p. 384 (Cambridge University Press, Cambridge).
- 25 Shabana, A. A. Constrained motion of deformable bodies. *Int. J. Numer. Methods Eng.*, 1991, **32**(8), 1813–1831.
- 26 Cardona, A., Geradin, M., and Doan, J. B. Rigid and flexible joint modeling in multibody dynamics using finite elements. *Comput. Methods Appl. Mech. Eng.*, 1991, **89**(1–3), 395–418.
- 27 Wehage, R. A., Shabana, A. A., and Hwang, Y. L. Projection methods in flexible multibody dynamics. Part I: kinematics. *Int. J. Numer. Methods Eng.*, 1992, **35**(10), 1927–1939.
- 28 Hwang, R. S. and Haug, E. J. Translational joints in flexible multibody dynamics. *Mech. Struct. Mach.*, 1990, **18**(4), 543–564.
- 29 Shabana, A. A. Dynamics of flexible bodies using generalized Newton–Euler equations. *J. Dyn. Syst., Meas. Control*, 1990, **112**(3), 496–503.
- 30 Nikravesh, P. E. and Chung, I. S. Application of Euler parameters to the dynamic analysis of three-

- dimensional constrained mechanical systems. *J. Mech. Des.*, 1982, **104**(4), 785–791.
- 31 Baruh, H.** *Analytical dynamics*, 1999, p. 718 (McGraw-Hill, Boston, MA).
- 32 Radisavljevic, V.** *Modeling and control of complex mechanical systems*, 2001, p. 121 (Rutgers University Press, NJ).
- 33 Radisavljevic, V. and Baruh, H.** Modeling and control of complex systems and networks described by differential-algebraic equations. *Dyn. Contin. Discrete Impuls. Syst.*, 2003, **10**(special issue), 294–299.
- 34 Shabana, A. A.** *Computational continuum mechanics*, 1st edition, 2008, p. 348 (Cambridge University Press, Cambridge).
- 35 Baumgarte, J.** Stabilization of constraints and integrals of motion in dynamical systems. *Comput. Methods Appl. Mech. Eng.*, 1972, **1**, 1–16.

II

Artikkelille II ei ole lupaa sähköiseen julkaisemiseen

Electronic publishing is not permitted for Article II

III

“Multibody Approach for Model-Based Fault Detection of a Reel”. Reprinted from the Journal of Computational and Nonlinear Dynamics, with kind permission of ASME.

Multibody Approach for Model-Based Fault Detection of a Reel

Pasi Korkealaakso
e-mail: Pasi.Korkealaakso@lut.fi

Asko Rouvinen
e-mail: Asko.Rouvinen@lut.fi

Aki Mikkola
e-mail: Aki.Mikkola@lut.fi

Department of Mechanical Engineering,
Lappeenranta University of Technology,
Skinnarilankatu 34,
P.O. Box 20,
FIN-53851 Lappeenranta,
Finland

In order to improve the recognition of faulty situations, model-based fault detection can be used together with signal processing methods. In this study, faults and abnormalities of a reel are studied by employing the multibody simulation approach. The reel under consideration consists of a number of subsystems, including hydraulics, electrical drives, and mechanical parts. These subsystems are coupled by joints, friction forces, and contact forces. Using the multibody simulation approach, the complete model of the reel can be obtained by coupling different subsystems together. Three well-known multibody formulations, a method of Lagrange multipliers, an Augmented Lagrangian method, and a method based on projection matrix \mathbf{R} , are briefly described and compared in order to find out the most efficient method for simulating the studied reel. Although this study is focused on the simulation of fault scenarios, the introduced multibody simulation approach can be utilized in real-time simulation. This makes it possible to apply the model to an existing reel. [DOI: 10.1115/1.2162865]

Introduction

Fault simulation of mechatronic machines can be used for preventing financial losses in industrial applications. Generally, due to mechanical contacts, large rotations, and hydraulic and electrical components, the dynamic analysis of machines must be carried out using a nonlinear formulation. For this reason, a model-based fault detection method using systematic multibody simulation is an attractive approach. In this study, three multibody formulations are compared and analyzed to determine suitability for fault detection of a reel. The reel under investigation consists of mechanisms, hydraulic actuators, and a control system.

Multibody formulations can be divided into global and topological formulations. Topological formulations utilize the topology of a mechanism in order to improve numerical efficiency. Global methods, however, can be programmed straightforwardly, while both open and closed kinematic loops can be solved using the same algorithms. It is also noted in Ref. [1] that global methods can be more efficient for systems of moderate size when the number of generalized coordinates is <50 , as in the case of the reel under investigation. The reel includes contacts, friction, and hydraulic actuators. The coupling of these components can be carried out straightforwardly when using global formulations. For these reasons, only global methods are considered in this study. The method of Lagrange multipliers [2,3], the Augmented Lagrangian method [4,5] and a method based on projection matrix \mathbf{R} [3,6,7] are briefly described and compared to each other. The computationally most efficient formulation is chosen and employed in the simulation of faulty situations. The implementation of algorithms is accomplished using a general-purpose ANSI C computer code. Visualization of the simulations is carried out using OPENGL, which is also used as a control interface between the user and the dynamics model during simulations.

Using traditional component-level monitoring based on signal-processing techniques, the faults in the process cannot often be recognized. In addition, a number of different reeling conditions have to be obtained. This is due to different roll types, different amounts of paper on the rolls, and different paper types. Conse-

quently, every combination of reeling conditions needs the definition of a corresponding alarm level. Using model-based methods, faults in terms of process can be detected and identified more accurately. Usually models used in fault detection are linearized models that are applied to individual machine parts. Studies and guidelines of model-based fault diagnosis can be found from Refs. [8,9]. A rotordynamic finite element model was used in Ref. [10] to predict a machine lifetime. The model was used to identify the parameters describing the deterioration of the mechanism from the measurements of an actual machine. In Ref. [10], the approach is close to the proposed multibody-simulation fault-detection approach where the supervision of a machine can be focused on the functionality of the entire process instead of an individual component. In practice, this can be carried out without additional instrumentation. In the entire fault-diagnosis procedure, the introduced approach can be used to detect the difference between normal and faulty conditions. Accordingly, the simulation model can be used to detect the residual of simulated and measured variables. In order to obtain the identification of the fault, data classification methods must be employed. When the multibody simulation approach is used, the modeling procedure can be carried out straightforwardly and in a systematic manner for both offline and online purposes.

In this study, the simulation model of a reel mechanism is used as a case system for simulating faulty situations. The reel is used at the finishing end of the papermaking process and has a considerable effect on paper quality, making it reasonable to utilize more intelligent fault-detection methods. In this study, a model-based approach is employed instead of, e.g., signal-processing techniques, which are based on signal levels. In the case of the reel, signal-processing techniques are laborious to use due to varying operation conditions. It is noteworthy that using the introduced multibody approach, the nip load (which is the prior variable in estimating roll quality) can be easily estimated. Another reason for using an approach based on physical modeling is that long-term measurements for statistical modeling or for training neural networks are not available. The dynamic model of a reel can be used to compute data of faulty situations to be used in network training for fault diagnosis purposes. The reel consists of a number of nonlinear subsystems that must be accounted for when modeling the entire reeling process. Models of mechanical, electrical, and hydraulic subsystems are introduced in the description of the simulation model. Therefore, the number of state variables in the

Contributed by the Design Engineering Division of ASME for publication in the JOURNAL OF COMPUTATIONAL AND NONLINEAR DYNAMICS. Manuscript received February 28, 2005; final manuscript received October 21, 2005. Review conducted by Dr. Kurt S. Anderson.

model is large, whereas the number of variables revealing the faults can be small. Based on the changes in responses between normal and fault-situation simulations, the required measurements or state variables for identifying the faults can be identified and utilized in the design of the fault detection system. In this study, the multibody simulation model is used only in the analysis of faulty scenarios while significant variables in terms of fault identification are obtained manually. It is noteworthy, however, that the introduced multibody approach can be extended for real-time simulation and, consequently, applied to an existing reel mechanism. To this end, control signals of existing reel are utilized as input for the simulation model. In this way, residuals of measured and simulated state variables can be formed while the values of residual can be used to detect abnormalities of the system. In this study, clearances of the mechanical joints and flexibility of structural components are assumed to be neglectable. In this study, the use of the model-based method for fault detection using a multibody approach is presented. However, the introduced approach is not yet applied to an existing reel. For this reason, the study of detection sensitivity and robustness of the approach are not discussed in this paper. Classification and reasoning methods in order to isolate the faults are outlined in this study.

Multibody Formulations

In the following chapter, three multibody formalisms used in this work are briefly described. Formalisms discussed here are the method based on Lagrange multipliers (which is referred to also as a descriptor form [11,12]), penalty, and augmented Lagrangian methods [4,5], as well as a method based on projection matrix \mathbf{R} [3,6,7].

Method Based on Lagrange Multipliers. When constraint equations are augmented to the equations of motion using the Lagrange multiplier technique, the result can be written as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}_q^T \boldsymbol{\lambda} = \mathbf{Q}^e + \mathbf{Q}^v \quad (1)$$

where \mathbf{q} is the vector of n generalized coordinates that defines the position and orientation of each body in the system, \mathbf{M} is the mass matrix, \mathbf{Q}^e is the vector of generalized forces, \mathbf{Q}^v is the quadratic velocity vector that includes velocity dependent inertia forces, \mathbf{C}_q is the Jacobian matrix of the constraint equations, and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. To satisfy a set of m constraint equations for generalized coordinates, Eq. (2) must be fulfilled

$$\mathbf{C}(\mathbf{q}, t) = \mathbf{0} \quad (2)$$

where \mathbf{C} is a vector of constraints. Equations (1) and (2) comprise a system of differential algebraic equations (DAE), which describe the dynamical behavior of the mechanism. In order to solve the set of equations using ordinary differential equation (ODE) integration methods, the equations must be transformed to the second-order ODE. For this reason, Eq. (2) is differentiated twice with respect to time,

$$\ddot{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = \mathbf{C}_q \ddot{\mathbf{q}} + (\mathbf{C}_q \dot{\mathbf{q}}) \dot{\mathbf{q}} + 2\mathbf{C}_{q\dot{q}} \dot{\mathbf{q}} + \mathbf{C}_{tt} = \mathbf{0} \quad (3)$$

where \mathbf{C}_t is a partial derivative of constraint equations with respect to time. By combining Eqs. (1) and (3), the matrix representation of equations of motion can be obtained as follows:

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}_q^T \\ \mathbf{C}_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^e + \mathbf{Q}^v \\ -(\mathbf{C}_q \dot{\mathbf{q}}) \dot{\mathbf{q}} - 2\mathbf{C}_{q\dot{q}} \dot{\mathbf{q}} - \mathbf{C}_{tt} \end{bmatrix} \quad (4)$$

where the invertible matrix is of the size $(m+n) \times (m+n)$. The equation of motion can be integrated using the standard ODE solver [12]. However, equations of motion cannot guarantee that constraint equations in Eq. (2) are satisfied. This is because during differentiation of the constraint equation, constant terms disappear and, consequently, Eq. (4) only fulfills the constraints at the acceleration level. Therefore, numerical integration causes accumulation of errors to the kinematic constraints. To overcome this

problem, a stabilization method must be used. Another possibility to solve this problem is to use methods that produce a general solution to differential algebraic equations [5,13].

Penalty Method and Augmented Lagrangian Formulation. In the penalty method, Lagrange multipliers are eliminated from the equations of motion by adding constraint equations penalized by penalty terms. This procedure leads to a set of n differential equations as follows:

$$(\mathbf{M} + \mathbf{C}_q^T \boldsymbol{\alpha} \mathbf{C}_q) \ddot{\mathbf{q}} = \mathbf{Q}^e - \mathbf{C}_q^T \boldsymbol{\alpha} (\dot{\mathbf{C}}_q \dot{\mathbf{q}} + \dot{\mathbf{C}}_t + 2\boldsymbol{\Omega} \boldsymbol{\mu} \dot{\mathbf{C}} + \boldsymbol{\Omega}^2 \mathbf{C}) \quad (5)$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\Omega}$, and $\boldsymbol{\mu}$ are $m \times m$ diagonal matrices (which contain penalty terms), natural frequencies, and damping ratios, respectively. If penalty terms are equivalent to each constraint, the matrices are identity matrices multiplied with a constant penalty factor.

A drawback associated to the penalty method is that large penalty factors must be used, which may lead to numerical ill-conditioning and round-off errors. However, the method can be improved by adding penalty terms or correction terms, which are zero when constraint equations are fulfilled. Using this approach, equations of motion can be written as follows:

$$(\mathbf{M} + \mathbf{C}_q^T \boldsymbol{\alpha} \mathbf{C}_q) \ddot{\mathbf{q}} = \mathbf{Q}^e - \mathbf{C}_q^T \boldsymbol{\alpha} (\dot{\mathbf{C}}_q \dot{\mathbf{q}} + \dot{\mathbf{C}}_t + 2\boldsymbol{\Omega} \boldsymbol{\mu} \dot{\mathbf{C}} + \boldsymbol{\Omega}^2 \mathbf{C}) + \mathbf{C}_q^T \boldsymbol{\lambda}^* \quad (6)$$

where $\boldsymbol{\lambda}^*$ is the vector of penalty forces. By comparing Eqs. (1) and (6), it can be concluded that

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}^* - \boldsymbol{\alpha} (\mathbf{C}_q \dot{\mathbf{q}} + \dot{\mathbf{C}}_t + 2\boldsymbol{\Omega} \boldsymbol{\mu} \dot{\mathbf{C}} + \boldsymbol{\Omega}^2 \mathbf{C}) \quad (7)$$

Since the exact values of $\boldsymbol{\lambda}^*$ are not known in advance, an iterative procedure should be used as follows:

$$\boldsymbol{\lambda}_{i+1}^* = \boldsymbol{\lambda}_i^* - \boldsymbol{\alpha} (\mathbf{C}_q \dot{\mathbf{q}}_i + \dot{\mathbf{C}}_t + 2\boldsymbol{\Omega} \boldsymbol{\mu} \dot{\mathbf{C}} + \boldsymbol{\Omega}^2 \mathbf{C}) \quad (8)$$

$\boldsymbol{\lambda}_0^* = \mathbf{0}$ is used for the first iteration i . Using this equation, the forces caused by errors in constraint equations at iteration $i+1$ can be defined and compensated. In this case, the penalty terms do not have to be large. An iterative procedure can be applied directly in Eq. (6), which leads to the following expression:

$$(\mathbf{M} + \mathbf{C}_q^T \boldsymbol{\alpha} \mathbf{C}_q) \ddot{\mathbf{q}}_{i+1} = \mathbf{M} \ddot{\mathbf{q}}_i - \mathbf{C}_q^T \boldsymbol{\alpha} (\dot{\mathbf{C}}_q \dot{\mathbf{q}} + \dot{\mathbf{C}}_t + 2\boldsymbol{\Omega} \boldsymbol{\mu} \dot{\mathbf{C}} + \boldsymbol{\Omega}^2 \mathbf{C}) \quad (9)$$

In the case of the first iteration $\mathbf{M} \ddot{\mathbf{q}}_0 = \mathbf{Q}^e$. The leading matrix of Eq. (9) is a symmetric and positive definite, which makes solving the equation efficient. This formulation behaves satisfactorily also in singular configurations of a mechanical system.

Method Based on Matrix \mathbf{R} . Two introduced formulations define the equations of motion using a complete set of generalized coordinates. However, the number of the equations can be reduced to the minimum number of differential equations using a set of independent generalized coordinates. Independent generalized velocities $\dot{\mathbf{q}}_i$ can be defined as a projection of velocities of generalized coordinates $\dot{\mathbf{q}}$ using constant matrix \mathbf{B} as follows:

$$\dot{\mathbf{q}}_i = \mathbf{B} \dot{\mathbf{q}} \quad (10)$$

It is noteworthy that the rows of matrix \mathbf{B} are linearly independent. For skleronomous systems, a solution to the transformation from independent generalized coordinates to complete set of generalized coordinates is available and can be defined using transformation matrix \mathbf{R} as follows:

$$\dot{\mathbf{q}} = \mathbf{R} \dot{\mathbf{q}}_i \quad (11)$$

Using the coordinate partitioning to dependent and independent generalized coordinates $\mathbf{q} = [\mathbf{q}_d^T \mathbf{q}_i^T]^T$, the virtual change of generalized coordinates on constraint equations can be expressed as follows:

$$C_{q_d} \delta q_d + C_{q_i} \delta q_i = 0 \quad (12)$$

where C_{q_d} and C_{q_i} are partitioned Jacobian matrices. C_{q_d} is a $m \times m$ nonsingular matrix where m is the number of the constraint equations. Using Eq. (12), the virtual change of dependent generalized coordinates can be defined as

$$\delta q_d = -C_{q_d}^{-1} C_{q_i} \delta q_i \quad (13)$$

The virtual change of generalized coordinates can now be expressed using independent generalized coordinates as follows:

$$\delta q = \begin{bmatrix} \delta q_d \\ \delta q_i \end{bmatrix} = \begin{bmatrix} -C_{q_d}^{-1} C_{q_i} \\ \mathbf{I} \end{bmatrix} \delta q_i \quad (14)$$

which can be rewritten using projection matrix \mathbf{R} as follows:

$$\delta q = \mathbf{R} \delta q_i \quad (15)$$

where \mathbf{R} is

$$\mathbf{R} = \begin{bmatrix} -C_{q_d}^{-1} C_{q_i} \\ \mathbf{I} \end{bmatrix} \quad (16)$$

By differentiating constraint equations once and twice with respect to time, one can have

$$C_q \dot{q} = -C_i \quad (17)$$

$$C_q \ddot{q} = -[(C_q \dot{q})_q \dot{q} + 2C_{q_i} \dot{q} + C_{q_i}] = -\dot{C}_i - \dot{C}_q \dot{q} \quad (18)$$

using coordinate partitioning, Eq. (18) can be written as follows:

$$\ddot{q}_d = -C_{q_d}^{-1} C_{q_i} \ddot{q}_i - C_{q_d}^{-1} [(C_q \dot{q})_q \dot{q} + 2C_{q_i} \dot{q} + C_{q_i}] \quad (19)$$

Accelerations of generalized coordinates take the form

$$\ddot{q} = \begin{bmatrix} \ddot{q}_d \\ \ddot{q}_i \end{bmatrix} = \begin{bmatrix} -C_{q_d}^{-1} C_{q_i} \\ \mathbf{I} \end{bmatrix} \ddot{q}_i + \begin{bmatrix} -C_{q_d}^{-1} [(C_q \dot{q})_q \dot{q} + 2C_{q_i} \dot{q} + C_{q_i}] \\ 0 \end{bmatrix} \quad (20)$$

Using definition

$$\boldsymbol{\gamma} = \begin{bmatrix} -C_{q_d}^{-1} [(C_q \dot{q})_q \dot{q} + 2C_{q_i} \dot{q} + C_{q_i}] \\ 0 \end{bmatrix} \quad (21)$$

It can be seen that vector $\boldsymbol{\gamma}$ consists of the accelerations of generalized coordinates when the accelerations of independent coordinates are equal to zero. Using Eq. (21), Eq. (20) can be written as follows:

$$\ddot{q} = \mathbf{R} \ddot{q}_i + \boldsymbol{\gamma} \quad (22)$$

Substitution the result into Eq. (1) leads to

$$\mathbf{R}^T \mathbf{M} \mathbf{R} \ddot{q}_i + \mathbf{R}^T \mathbf{M} \boldsymbol{\gamma} - \mathbf{R}^T (\mathbf{Q}^e + \mathbf{Q}^v) = 0 \quad (23)$$

This equation of motion can be solved for independent accelerations, which can be integrated to solve the new independent velocities and positions for the next time step. This form of the equation of motion is complicated and highly nonlinear, and the set of independent generalized coordinates must be changed every time the pivot of C_{q_d} approaches zero.

Dynamic Model of the Reel

The system studied in this paper is the reel mechanism of a paper machine. The reeling sequence follows after calendering at the finishing end of the papermaking process. When paper comes from the calender, it will be wound around the reel spool. The purpose of this sequence is to produce large-diameter rolls of paper called parent reels. The weight of the roll can be from 20,000 to 120,000 kg, whereas the diameter ranges from 2 to 4 m [14].

In Fig. 1, the main parts of the reel are depicted. The reel spool is attached to the carriages by hydraulically locked arms. It is

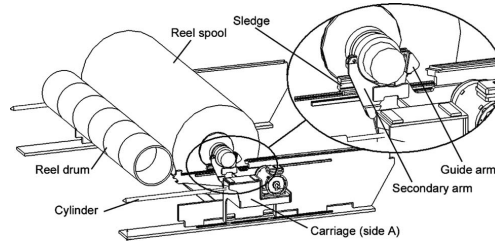


Fig. 1 Main parts of the reel mechanism

noteworthy that the carriages do not hold up the reel spool. Instead, it lies on the sledges, which move along the rails on the low-friction linear bearings. The reel spool is pressed against the reel drum by pulling the carriages using hydraulic cylinders. In the following sections, a brief introduction of subsystems of the simulation model, including a description of friction, contacts, and the hydraulic circuit, will be given.

Description of Mechanical Systems. The simulation model of the reel mechanism consists of six rigid bodies interconnected via mechanical joints, actuator forces, and contact forces. Actuators of the reel are hydraulic cylinders that produce linear motion for parent roll and electrical motors that produce rotating torque. The contact forces are produced between the reel drum and the paper roll.

Three different multibody formalisms described earlier are used in the dynamical analysis of the reel. The constraints between the bodies are modeled using the same method in all three introduced formalisms. A detailed description of modeling constraints can be found from Ref. [2]. The assembly of the mechanism is accomplished using four translational joints, one rotational joint, and five planar joints. The topology of the system is depicted in Fig. 2, where the lines refer to kinematic joints and the arrows to applied forces, correspondingly. In the figure, fixed, prismatic, revolute, and planar joints are denoted F, P, R, and PL, respectively. In the case of the planar joint, the direction of constrained motion is expressed in parenthesis. Cylinder forces, contact forces, and friction forces are denoted by the symbols F_{cyl} , F_c , and F_f , and the torque of the electric motor is denoted as T_m . The forces are solved using Eqs. (24), (30), and (32).

Description of the Friction Model. Translational velocities of the carriages and sledges are low during the work cycle. For this reason, the effect of the friction forces must be studied carefully. The friction model can be divided into three regimes: sliding, stiction, and the transition between them. Consequently, the friction coefficient is a function of the sliding velocity. When the friction force is applied to the system, it can be expressed as follows:

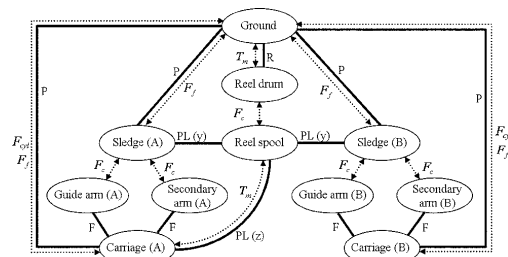


Fig. 2 Topology chart of the reel mechanism

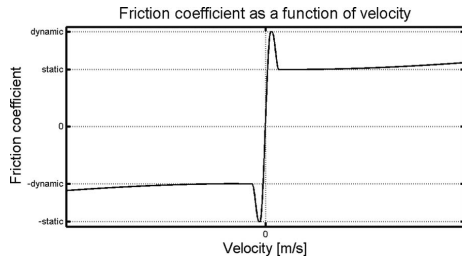


Fig. 3 Friction coefficient as a function of velocity

$$F_f = f(q, \dot{q})F_n = f(q, \dot{q})D(q)C_q^T \lambda \quad (24)$$

where F_n is the vector of joint constraint forces, $f(q, \dot{q})$ is the function of the friction coefficient, $D(q)$ is a matrix defined uniquely for each joint type and $C_q^T \lambda$ describes constraint reaction forces.

Stiction occurs when the relative velocity between sliding surfaces in the joint approaches zero. The stiction can be described as an additional constraint applied to the degrees of freedom in which sliding occurs. The stiction is valid as long as the constraint force is lower than the static friction force [15]. This approach, however, is inconvenient because the numerical integration procedure must be stopped and restarted again when the stick-slip phenomenon takes place. In this study, the friction model is simplified and defined as a combination of coulomb and viscous friction. The friction coefficient in the function of velocity is depicted in Fig. 3, in which the static and dynamic friction coefficients are used to define boundaries of stiction and sliding regimes.

Description of the Contacts. A contact model between the rolls is needed in order to describe the nip contact. One of the main features studied in reeling dynamics is subjected to the stiffness and damping characteristics of the nip contact [16]. The enlargement of the paper roll during winding transfers the contacting bodies at a constant velocity. This can be considered as an additional coupling between bodies. The diameter of the paper roll can be described as follows:

$$d(t) = \sqrt{d_0^2 + \frac{4\tau}{\pi} \int_0^t v dt} \quad (25)$$

where τ is the average web thickness, d_0 is the initial roll diameter, and v is the web speed [16]. The nip contact is modeled as a system of nonlinear spring dampers. Based on the elasticity of the bodies in contact, the spring and damping constants can be defined using the Herzian contact theory [16,17]. The penetration depth of the cylinders can be written as

$$\delta = F_c \frac{(1-\nu^2)}{\pi E} \left[2 \ln \left(\frac{2d}{a} \right) - 1 \right] \quad (26)$$

where F_c is the contact force of the nip, d is the diameter of the paper roll, E and ν are the modulus of elasticity and Poisson's ratio of the paper roll, and a is the semicontact width of the nip contact, which can be calculated as follows:

$$a = \sqrt{\frac{2d^3 F_c}{\pi E^*}} \quad (27)$$

In Eq. (27), E^* is the effective modulus of elasticity and d^* is the relative curvature of cylinders, which can be defined as follows:

$$d^* = \left(\frac{1}{d} + \frac{1}{d_1} \right)^{-1} \quad (28)$$

where d and d_1 are the diameters of the paper roll and reel drum, respectively. The effective modulus of elasticity takes the form

$$E^* \approx \frac{E}{1-\nu^2} \quad (29)$$

It is noteworthy that the use of the Herzian contact theory leads to a computationally expensive procedure. For this reason, this study employs an approach that is based on the use of approximate polynomials. The nip contact is modeled and included in the reel model using a system of springs that are distributed along the length of the contact line. The contact force in this approach is modeled using the approximated equation as follows:

$$F_c = \begin{cases} k_a(x_0 - x)^e + c_a^{\max} \dot{x}, & x_0 - x \geq d_a^{\max} \\ k_a(x_0 - x)^e + c_a^{\max} \Delta^2(3 - 2\Delta)\dot{x}, & 0 < x_0 - x < d_a^{\max} \\ 0, & x_0 - x \leq 0 \end{cases} \quad (30)$$

where $\Delta = (x_0 - x)/d_a^{\max}$. In Eq. (30), $x_0 = (d + d_1)/2$, x is the distance between cylinder centers; parameters k_a and e are approximated using Eq. (26), which describes the compression of cylinders at a certain contact force; and d_a^{\max} is the penetration depth when the maximum damping coefficient c_a^{\max} is applied. In Eq. (30), different regions are used in order to smooth the discontinuities of contact forces.

When contact occurs, the compression of cylinders in Eq. (26) is a function of the contact force, which cannot be solved explicitly. In Eq. (30), contact force is directly a function of nip deformation and can be solved without iteration. Unknown parameters in Eq. (30) will be defined so that contact force closely resamples Herzian contact theory.

Description of the Hydraulic Circuit With the Servocontrol System. In this study, the lumped fluid theory [18] is used in the modeling of the hydraulic circuit of the reel. In the lumped fluid theory, a hydraulic circuit is divided into volumes where the pressure is assumed to be equally distributed. The valves are modeled employing semi-empirical approach [19], where the parameters used in flow equations through the orifices can be obtained, in many cases, from manufacturer catalogs. In the lumped fluid theory, the differential equation for pressure in the volume can be expressed as

$$\dot{p} = \frac{B_e}{V} \left(\sum_{i=1}^{N_i} Q_{in_i} - \sum_{j=1}^{N_o} Q_{out_j} - \dot{V} \right) \quad (31)$$

where B_e is the effective bulk modulus of the volume V , Q_{in} and Q_{out} are input and output flow rates, respectively. In Eq. (31) N_i and N_o are the numbers of input and output flow rates and \dot{V} is the change of volume with respect to time. The flow rates in the hydraulic circuit are often modeled assuming the flow to be turbulent [18].

The cylinder force is defined based on pressures in cylinder chambers and the friction forces. Cylinder forces can be defined as follows:

$$F_{cyl} = p_A A_A - p_B A_B - \dot{x}_{cyl} (p_A A_A - p_B A_B) (1 - \eta_{cyl}) \quad (32)$$

where p_A and p_B are pressures in cylinder chambers at sides A and B (Fig. 4) while A_A and A_B are corresponding piston areas. In Eq. (32) \dot{x}_{cyl} is the velocity of the piston with respect to the cylinder wall and η_{cyl} is the constant describing the efficiency of the cylinder.

The quantity of the wasted paper near the reel spool can be minimized by accurate control of the most critical winding parameters such as torque, nip load and tension [14]. Most of the wasted paper results from the starts and the ends of winding the paper

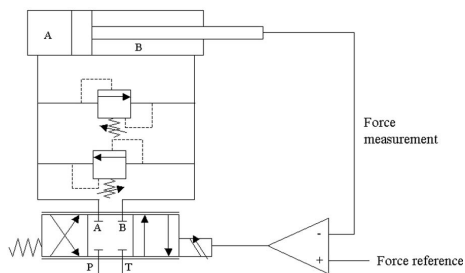


Fig. 4 The principle of the hydraulic circuit used to control the nip load

roll. In practice, the reel motion is directed using force feedback control. Accordingly, the value for the input signal for values in voltage can be written as

$$U_{\text{control}} = K_p U_{\text{err}} + K_I \int U_{\text{err}} \quad (33)$$

where K_p is the proportional gain, K_I is the integration gain, and U_{err} represents the difference between the reference and the measured signals of the cylinder forces of the reel. The servovalve is modeled using a second-order dynamic model, which can be described using transfer function $G(s)$ as follows:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (34)$$

where ω_n is the natural frequency and ζ is the damping coefficient. These coefficients can be defined using the Bode diagram. The hydraulic servocontrol system of the nip load is described in Fig. 4.

Simulated Faults and Results

In the following section, the simulation model is analyzed using introduced formulations. Considerable differences between simulation times can be detected while the results between simulations remain identical when equal step size is used. In this study, two failure situations are considered and analyzed. The model of the reel is used to simulate excessive friction faults produced by the contact of the rails and carriages. In practice, these fault types do not appear frequently. However, the fault type is harmful in the sense it affects the nip load and, consequently, decreases the roll quality. A failure situation can be considered critical when it causes a 5% change in the nip load. It is important to note that, when the stick-slip phenomenon is taken into account, the friction forces may fluctuate considerably. These forces directly increase the nip load and cannot be controlled with a force servocontroller. In the following, simulation results are presented without numerical values in order to protect the reel-related product know-how of the reel manufacturer.

Comparison of the Formulations and Integrators. All formulations are used to simulate a 500 s cycle of a reeling sequence using a step size of 0.002 s. In Fig. 5, results of the velocity of sledge A obtained using three formulations are presented. It can be seen that the results are similar. In Table 1, the simulation times are compared and the augmented Lagrangian is found the most efficient out of the three formulations. Capability for simulations in real time is obtained with a fourth-order Runge-Kutta integration procedure (RK4) when a step size of 4.5 ms is used.

The most considerable computational costs in the method of Lagrange multipliers and augmented Lagrangian method are in the calculation of the Jacobian matrix, the constraint acceleration

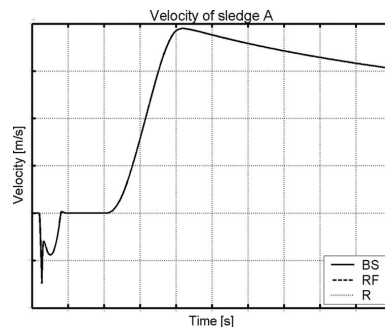


Fig. 5 Velocity of sledge A using different formulations

vector, and the inversion of the leading matrix. The augmented Lagrangian method is the fastest of the implemented algorithms because the matrix inverted is smaller and a symmetric and positive definite. For this reason, more efficient algorithms can be used to solve a set of linear equations. In the method based on projection matrix \mathbf{R} , the matrix inverted is small and can be solved efficiently. The biggest computational cost is in the use of the iterative Newton-Raphson algorithm. The code is meant to be used in real-time applications, and for this reason, the number of iterations has to be constant in order to guarantee that the solving time of each time step is the same. In offline applications, the iteration is not necessary in every time step.

It is noteworthy that the complete model is cumbersome to verify in practice. This is due to the fact that most of the variables can not be measured from the production unit. However, methods used in the modeling of submodels, including the hydraulics and contact model, are verified in previous studies [16,19]. Results of mechanical components are verified comparing simulation results to those obtained using commercial software.

Excessive Friction on the Rails of Carriage A. In the first example, the excessive friction is imposed on the rails of carriage A, shown in Fig. 1. In this example, the static and dynamic friction coefficients are increased from 0.015 to 0.025 and 0.0045 to 0.0145, respectively. Because of the force control, the friction causes an increase in the nip load at side A of the reel as can be seen from Fig. 6. This can be observed as a small error in the position of the carriages. However, the position error is only 0.2 mm, making it cumbersome to identify using measurements. To detect excessive friction, it is reasonable to simulate another stage of the working cycle. Increased friction forces can be detected directly from the force sensor of the cylinder when the paper roll is moved to storage position.

When the paper roll is completed, the secondary carriages transfer the parent reel to storage position. During the transfer sequence, the excessive friction can be perceived from the force sensors of the hydraulic actuators or the pressures of cylinder chambers as shown in Fig. 7.

Excessive Friction on the Rails of Sledge A. The friction on the rails of sledge A causes an increase in the nip load similarly to

Table 1 Comparison of simulation times

Integrator	Formulation		
	Descriptor form (s)	Matrix \mathbf{R} (s)	Augmented Lagrangian (s)
RK4	3169.8	1680.1	1103.8
RK2	1637.9	885.8	575.0

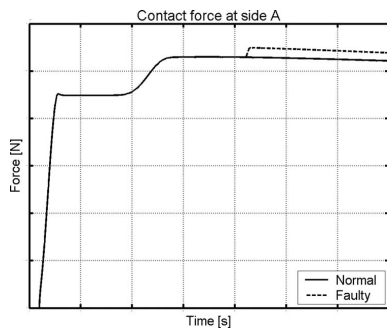


Fig. 6 Nip load when increased friction applied to rails of carriage A

excessive friction on the rails of the carriage. In this case, the friction force is a function of the mass of the paper roll and it increases the nip load during reeling. The same friction coefficients as in the first example are used in this case. In this example, a change in the friction coefficient can be perceived from the force sensors of the hydraulic actuators and, in addition, from contact forces between arms and bearing housing as in Fig. 8.

Discussion on the Results. Reeling operation is considered to be defected when a 5% difference occurs in the nip load. In practice, the nip load is cumbersome to measure and, for this reason, a simulation model can be used for estimating the load. Excessive friction on the rails directly increases the nip load. During reeling, this cannot be detected from the cylinder pressures or force sen-

sors used to control the nip load. When the paper roll is transferred to the storage position, rail failures can be perceived. Employing force measurements of the arms, the faulty rail can be identified. Using parameter estimation, the simulation model can be updated so that the simulation model functions like an existing system. Then, from the variation of parameters, possible faults can be perceived.

Conclusions

In this study, the dynamic model of a reel mechanism was utilized in fault detection. A simulation model was modeled using three different multibody formalisms. Multibody formalisms with two integrators were compared in terms of accuracy and efficiency. The results showed that the most efficient combination for the reel simulation is the augmented Lagrangian method with a fourth-order Runge-Kutta integrator. Using a second-order Runge-Kutta integrator, computation times twice as fast as those with a fourth-order Runge-Kutta can be achieved, but the results suffer a lack of accuracy. Between different formulations, the results were identical when appropriate penalty terms were defined.

Generally, multibody methodologies seem to be a practical in the simulation of fault scenarios. The studied system was modeled systematically, and different subsystems were coupled into the model directly. Based on the results of the simulations, the faults could be detected and identification could be carried out. In future studies, the simulation model will be integrated into an existing mechanism, making it necessary to study the capabilities of the simulation in field conditions.

Acknowledgment

This study was supported by the National Technology Agency in Finland (TEKES).

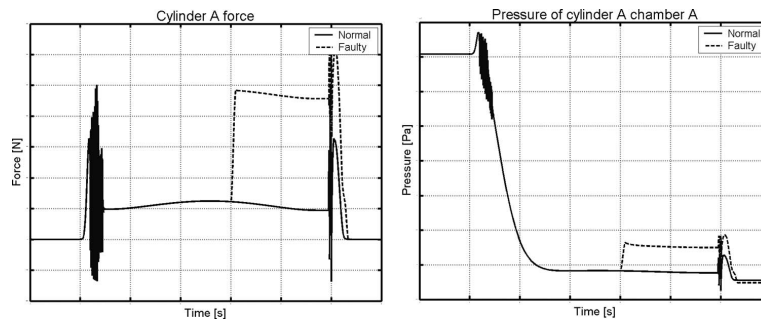


Fig. 7 Force of cylinder A and pressure of chamber A in cylinder A during transfer sequence

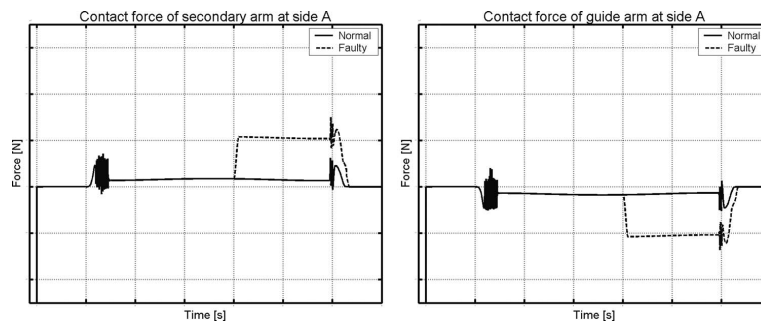


Fig. 8 Contact forces of secondary and guide arms at side A during transfer sequence

Nomenclature

A = area of the piston in the cylinder
 \mathbf{B} = transformation matrix from velocities of generalized coordinates to velocities of independent generalized coordinates
 B_e = effective bulk modulus
 c_a^{\max} = maximum damping coefficient
 \mathbf{C} = vector of kinematic constraint equations
 \mathbf{C}_q = constraint Jacobian matrix
 d = diameter of the paper roll
 d^* = relative curvature of cylinders
 d_0 = initial diameter of the paper roll
 d_1 = diameter of the reel drum
 d_a^{\max} = penetration depth when the maximum damping coefficient is achieved
 \mathbf{D} = matrix that defines direction of friction force in the joint
 e = positive real variable that specifies the exponent of the force-deformation characteristic
 E = Young's modulus
 E^* = effective modulus of elasticity
 f = function of friction coefficient
 F_c = nip contact force
 F_{cyl}^* = force of the hydraulic cylinder
 \mathbf{F}_f = vector of friction forces
 \mathbf{F}_n = vector of joint constraint forces
 $G(s)$ = transfer function of valve
 k_a = stiffness coefficient
 K_I = integration gain
 K_P = proportional gain
 p = pressure in the volume
 \mathbf{q} = vector of generalized coordinates
 \mathbf{q}_d = vector of dependent generalized coordinates
 \mathbf{q}_i = vector of independent generalized coordinates
 m = number of constraint equations
 \mathbf{M} = mass matrix
 n = number of generalized coordinates
 N_i = number of input flow rates
 N_o = number of output flow rates
 Q_{in} = input flow rate
 Q_{out} = output flow rate
 \mathbf{Q}^e = vector of generalized forces
 \mathbf{Q}^v = vector of quadratic velocity inertia terms
 \mathbf{R} = velocity transformation matrix
 t = time
 U_{control} = input control signal
 U_{err} = difference between the reference and measured signals
 v = web speed
 V = volume
 x = distance between centers of paper roll and reel drum
 \dot{x}_{cyl} = velocity of the piston with respect to the cylinder wall
 $\boldsymbol{\alpha}$ = matrix of penalty terms

$\boldsymbol{\gamma}$ = acceleration of generalized coordinates when zero acceleration for independent generalized coordinates is used
 δ = penetration depth of rolling cylinders
 Δ = constant used to define contact force
 η_{cyl} = constant describing the efficiency of the cylinder
 ζ = damping coefficient of the servovalve
 $\boldsymbol{\lambda}$ = vector of Lagrange multipliers
 $\boldsymbol{\mu}$ = matrix of fictitious damping ratios
 ν = Poisson's ratio of the paper roll
 τ = average web thickness of the paper roll
 ω_n = natural frequency of the servovalve
 $\boldsymbol{\Omega}$ = matrix of fictitious natural frequencies

References

- [1] Cuadrado, J., Dopico, D., Gonzalez, M., and Naya, M. A., 2004, "A Combined Penalty and Recursive Real-Time Formulation for Multibody Dynamics," *J. Mech. Des.*, **126**(4), pp. 602–608.
- [2] Haug, E. J., 1989, *Computer-Aided Kinematics and Dynamics of Mechanical Systems*, Allyn and Bacon, Boston.
- [3] Shabana, A. A., 2001, *Computational Dynamics*, Wiley, New York.
- [4] Bayo, E., García de Jalón, J., and Serna, M. A., 1988, "A Modified Lagrangian Formulation for the Dynamic Analysis of Constrained Mechanical Systems," *Comput. Methods Appl. Mech. Eng.*, **71**(2), pp. 183–195.
- [5] García de Jalón, J., and Bayo, E., 1994, *Kinematic and Dynamic Simulation of Multibody Systems—The Real-Time Challenge*, Springer-Verlag, Berlin.
- [6] Serna, M. A., Rafael, A., and García de Jalón, J., 1982, "Dynamic Analysis of Plane Mechanisms With Lower Pairs in Basic Coordinates," *Mech. Mach. Theory*, **17**(6), pp. 397–403.
- [7] Wehage, R. A., and Haug, E. J., 1982, "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems," *ASME J. Mech. Des.*, **104**(1), pp. 247–255.
- [8] Patton, R. J., Chen, J., and Nielsen, S. B., 1995, "Model-Based Methods for Fault Diagnosis: Some Guide-Lines," *Trans. Inst. Meas. Control (London)*, **17**(2), pp. 73–83.
- [9] Isermann, R., and Ballé, P., 1997, "Trends in the Application of Model-Based Fault Detection and Diagnosis of Technical Processes," *Control Eng. Pract.*, **5**(5), pp. 709–719.
- [10] Betti, B. P., and Han, R. P. S., 1998, "Predictive Maintenance Using the Rotordynamic Model of a Hydraulic Turbine-Generator Rotor," *ASME J. Vib. Acoust.*, **120**(2), pp. 441–448.
- [11] Serban, R., and Haug, E. J., 1998, "Kinematic and Kinetic Derivatives in Multibody System Analysis," *Mech. Struct. Mach.*, **26**(2), pp. 145–173.
- [12] Shabana, A. A., 1998, *Dynamics of Multibody Systems*, Cambridge University Press, Cambridge, England.
- [13] Haug, E. J., and Yen, J., 1990, "Generalized Coordinate Partitioning Methods for Numerical Integration of Differential-Algebraic Equations of Dynamics," *Real-Time Integration Methods for Mechanical System Simulation*, NATO Advanced Studies Institute, Series F, Vol. 69, Springer-Verlag, Berlin Heidelberg, Germany, pp. 97–114.
- [14] Jokio, M., 1999, *Papermaking Part 3. Finishing*, Gummerus Printing, Jyväskylä.
- [15] Haug, E. J., Wu, S. M., and Yang, S. M., 1986, "Dynamics of Mechanical Systems With Coulomb Friction, Stiction, Impact and Constraint Addition-Deletion—I, Theory," *Mech. Mach. Theory*, **21**(5), pp. 401–406.
- [16] Jorkama, M., 1998, "The Role of Analytical Winding Dynamics in Winder Design," *Tappi J.*, **81**(1), pp. 202–207.
- [17] Johnson, K. L., 1985, *Contact Mechanics*, Cambridge University Press, Cambridge, England.
- [18] Watton, J., 1989, *Fluid Power Systems*, Prentice-Hall, Englewood Cliff, NJ.
- [19] Handroos, H. M., and Vilenius, M. J., 1991, "Flexible Semi-Empirical Models for Hydraulic Flow Control Valves," *J. Mech. Des.*, **113**(3), pp. 232–238.

IV

“Multi-Body Simulation Approach for Fault Diagnosis of a Reel”. Reprinted from the Journal of Multi-Body Dynamics, with kind permission of the Professional Engineering Publishing.

Multi-body simulation approach for fault diagnosis of a reel

P Korkealaakso*, A Mikkola, and A Rouvinen

Department of Mechanical Engineering, Lappeenranta University of Technology, Lappeenranta, Finland

The manuscript was received on 5 August 2004 and was accepted after revision for publication on 10 October 2005.

DOI: 10.1243/146441906X78406

Abstract: This study introduces a simulation approach that can be utilized in the detection of abnormalities in the behaviour of machines. The approach is based on the utilization of a detailed multi-body simulation model. In this study, the approach to detect abnormalities is applied to the study of a reel. The reel under investigation is used at the finishing end of the papermaking process and it consists of a number of subsystems including hydraulics, electrical drives, and mechanical parts. In the reel, mechanical parts are coupled by joints, contacts, and friction.

The symptoms of the faults are generated using the variables which can be measured from the reel. When using a traditional component level monitoring method, the faults in the process often cannot be recognized. However, in the proposed multi-body simulation approach, the supervision of a machine can be focused on the functionality of the entire process, whereas detection of faults can be carried out without the use of additional instrumentation. When the multi-body simulation approach is used, the modelling of a machine can be carried out in a straightforward and systematic manner including the dynamic coupling between the subsystems of a machine.

Keywords: multi-body simulation, fault detection, a reel, coupled simulation

1 INTRODUCTION

In process industry, failures that may appear in machines can lead to severe accidents and considerable financial losses. Accordingly, the development of fault diagnosis for the process machines is highly prioritized. Methods based on signal processing techniques such as the frequency spectrum analysis, spectrograms, and wavelets are often utilized in fault detection. It is noteworthy, however, that these methods do not always provide the information which is needed in order to identify the cause of faults. Hardware redundancy can be used in the fault diagnosis of safety critical systems such as in airplane control systems. It is possible to isolate the faults using hardware redundancy, but it offers only a moderate rate of false alarms. One

clear disadvantage of this method is the need for a large number of sensors. This is due to the fact that several identical hardware systems are used in parallel [1]. Statistical models for fault detection and isolation (FDI) systems can be formulated employing data obtained from earlier measurements. A statistical model-based approach is used in reference [2] to isolate the features of the faults. This is accomplished by developing signal processing techniques using a non-linear filter approach. When the data obtained by measurements are not available, the analytical redundancy approach can be used. The method uses the signals generated by the mathematical model of the system in such a manner that redundant sensors are not needed. The studies of model-based (analytical redundancy) techniques can be divided into two main categories: quantitative and qualitative modelling. In the qualitative models, the infinitesimal close numerical changes in the system (quantitative) can be represented using a few qualitative values that describe the behaviour of the system [3]. On the other hand,

*Corresponding author: Department of Mechanical Engineering, Lappeenranta University of Technology, FIN-53851 Lappeenranta, Finland.

quantitative models are based on differential equations that describe the dynamical behaviour of the system. In these modelling approaches, linear and nonlinear models can be used, whereas in recent studies, the number of nonlinear models applied to failure diagnosis has been continuously increasing [4]. Faults in the system generate symptoms that can be identified as residual functions from the analytical model of the normal operation and behaviour of the diagnosed system. This can be achieved by comparing a measurement of a variable that is indicated with the symptom to the comparable variable from the simulation model. Residuals can be generated using different approaches. The most common are observer-based, parity relation, and parameter estimation methods [4]. More details on model-based techniques in FDI are given in survey papers by Willsky [1], Isermann [5], Frank [6], Basseville [7], and Patton *et al.* [8]. A finite element model of a rotor was used in reference [9] to predict a machine lifetime. The model was used to identify the parameters describing the deterioration of the mechanism from the measurement of an actual machine. In that study, the supervision of the machine is focused on the functionality of the entire process instead of an individual component. Accordingly, the approach introduced in reference [9] resembles the approach that is used in this study. The most distinctive difference between this study and reference [9] is, however, the coupling of different subsystems.

In this study, quantitative modelling techniques are studied using a non-linear dynamical model. Generally speaking, the forming of an accurate mathematical model for the purposes of a model-based fault diagnosis system is time-consuming. This is due to the fact that most of the modelling methodologies suffer from case dependencies. This study is focused on a general modelling approach that can be used to describe the dynamical behaviour of a mechatronic machine with non-linearities. Particularly, the dynamic analysis of multi-body systems is used for the purpose of fault detection and observation.

The multi-body system simulation approach is applied to the fault diagnosis of the reel. The simulation model consists of mechanical, electrical, and hydraulic subsystems that are coupled together. On the basis of the response differences between normal and faulty simulations, the required measurements for identifying the faults can be defined and utilized in the design of the fault diagnosis system. In the simulation model of a reel, clearances of the mechanical joints and structural flexibility are assumed to be neglectable. However, the flexibility of a hydraulic system is accounted to the reel model. Classification and reasoning methods in order to isolate the faults are outlined in this study.

2 USE OF MULTI-BODY DYNAMIC APPROACH IN FAULT DIAGNOSIS

The non-linearity of mechanical systems can lead to numerical difficulties that usually force to use computerized techniques when analysing the dynamics of the system. The multi-body dynamics simulation approach can be used to analyse a wide variety of mechanical systems including automobiles, mobile cranes, satellites, and robots. Large rotations and large variations of geometric configurations under operation conditions are the common features in these applications [10, 11]. Methods used in multi-body simulations can be divided into two wide categories: global and topological formulations. Several commercial software products, such as ADAMS [12], use constrained Cartesian coordinates and are based on global formulations. In the global methods, systems consisting of open and closed loops are considered in the same manner. By taking advantage of system topology by employing recursive methods with joint coordinates, it is possible to improve the numerical efficiency of the dynamic simulations. Formulations that use both Cartesian and relative coordinates with velocity transformation between these coordinate systems are often called semi-recursive methods. In these formulations, the equations of motion may take a simpler form and, in consequence, they may be suitable for online fault diagnosis. It is important to note, however, that semi-recursive methods are involved and cumbersome to use in a general case [13, 14]. In this study, the global multi-body simulation approach is used in the modelling of a reel mechanism with actuators and control systems. The fault diagnosis system can be used in online or offline simulation, and based on the selected approach, the formulation of the multi-body system and the accuracy of the subsystems can be selected, respectively.

A multi-body simulation approach can be used to take the coupling of separate subsystems into account. By coupling subsystems together, the functionality of the entire machine can be examined. Several symptoms in single components can often be isolated using classification or reasoning techniques. A problem related to component level condition monitoring is the omission of faults that have an effect on the operation of the process. Utilizing multi-body simulation, it is possible to capture the performance of the system with a simple sensor configuration. It is also important to note that simulation models make it possible to obtain variables which cannot be measured. For these reasons, expenses due to additional sensors can be avoided when model-based techniques are used in fault detection.

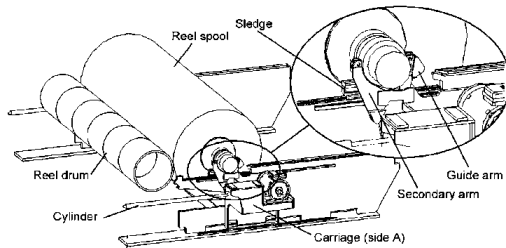


Fig. 1 The main parts of the reel

In the case of the reel, the operational cycles are normally repeated once an hour. When an offline condition monitoring approach is used, complex and detailed dynamical models can be utilized in fault detection. When faults appear seldom, as it is in the case of the reel, the simulation model can also be utilized to generate data that describe faulty scenarios.

3 DYNAMICAL MODEL OF THE REEL

In the following sections, the simulation model of the reel is introduced. A brief description of the system studied as well as the modelling method of mechanical components with constraints, friction, and contacts are described. The hydraulic system with servo control will also be introduced in the following sections.

The system under investigation is the reel mechanism of a paper machine. The reeling sequence follows calendering at the end of the papermaking process. When paper comes from the calender, it will be wound around the reel spool. The purpose of this sequence is to produce large-diameter rolls of paper called parent reels. The weight of the roll can be from 20 000 to 120 000 kg, whereas the diameter ranges from 2 to 4 m [15]. The main parts of the reel are depicted in Fig. 1. The reel spool is attached to the carriages by hydraulically locked arms. Note that the carriages do not hold up the reel spool. Instead, it lies on the sledges which move along the rails on the low friction linear bearings. The reel spool is pressed against the reel drum by pulling the carriages using hydraulic cylinders.

The wearing of the parts can lead to excessive friction forces between the components. The friction forces are a remarkable source of instability when considering the control of the reel. In this study, the hydraulic components are modelled in order to obtain realistic cylinder forces that are needed in the interaction of mechanical parts. The nip load caused by cylinders attached to the carriages is controlled using the servo control system of the cylinder forces.

The operation of the reeling sequence is shown in Fig. 2. At the first stage (Fig. 2(a)), the reel spool is pressed against the reel drum while it is rotated using constant angular velocity in such a way that appropriate web tension is obtained. When the

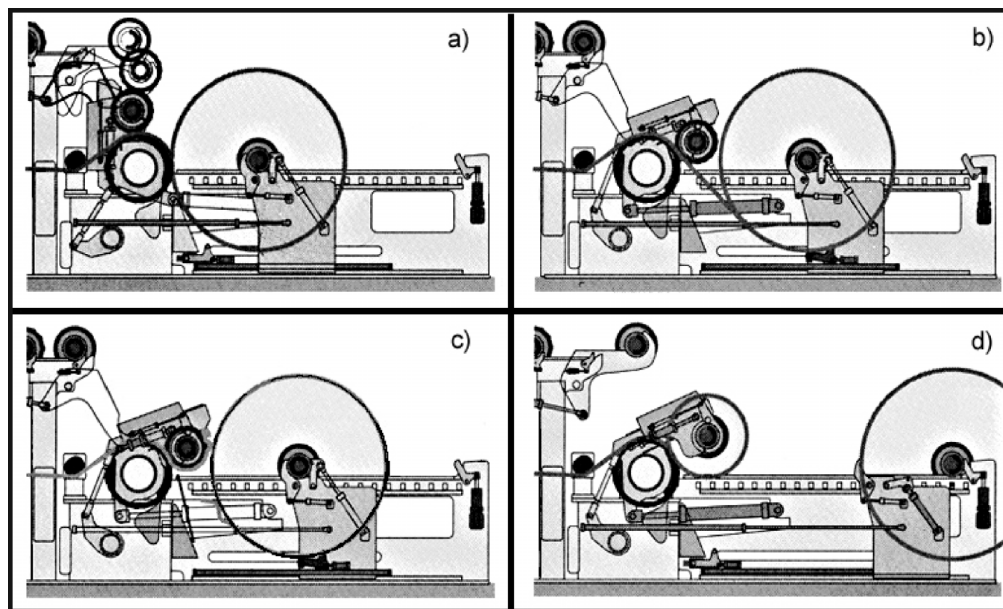


Fig. 2 The reel change sequence [15]

changing sequence begins (Fig. 2(b)), the primary reeling carriage takes the empty reel spool from the storage rack. The empty reel spool is accelerated to the appropriate web speed and switched to turn-up position. At the third stage (Fig. 2(c)), the primary nip is closed, web cut, and guided onto the new spool. At the end of the work cycle (Fig. 2(d)), the parent reel is stopped and the secondary carriages transfer the parent reel to storage position.

In the reeling machine, a good quality of paper rolls must be obtained because defects in the rolls often cause problems in unwinding at the cutting section where the winders are also used. The defects can take place if the roll is loose or the nip load is not equally distributed. The reels and winders in the paper machines contain a number of mechanical, hydraulic, and electrical components that often are inexpensive. For this reason, the requirement of condition monitoring may seem unnecessary. However, one should consider the functionality of the entire machine and try to prevent secondary damages caused by a faulty part. These critical parts should be considered carefully in fault detection systems. It is important to note that the replacement of an inexpensive part can cause losses of €10 000 [16] each hour the paper machine is not in operation.

3.1 Coupled simulation model of a reel

In this study, a dynamic model of the reel that consists of mechanical and hydraulic components with a force servo control system is utilized in fault diagnosis. The mechanism consists of 11 rigid bodies and it is constrained in such a way that it has 11 degrees of freedom. In Fig. 3, the topology of the system is described using lines to illustrate kinematic joints and arrows to illustrate applied forces. The contact model between the reel drum and the reel spool is used to describe the nip load. The contact is described using several non-linear springs between the rolls. The contact parameters are estimated using the Hertz theory of elastic contacts resulting in deformation between a long, linear-elastic, homogenous and isotropic cylinder, and a rigid cylinder, describing the paper roll and drum roll [17, 18]. The force required to produce the appropriate nip load is achieved by pressing the rolls against each other by hydraulic cylinders. The reel considered in this study is driven by motors attached to both the winding roll (reel spool) and the nipped roller (reel drum). The drives are called the secondary centre drive used in the torque control of the reel spool and the reel drum drive used in the torque control of the reel drum. This type of winder is called a centre-surface reel or winder [15].

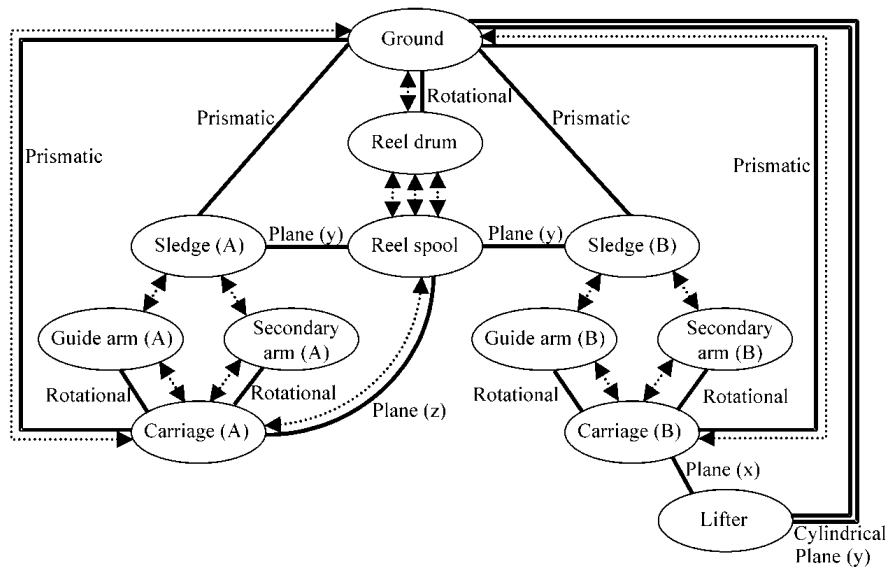


Fig. 3 The topology of the system. Arrows are applied forces and moments and lines describe kinematic joints

3.2 Description of mechanical systems

The global formulation, which examines open and closed chains the same way, is used to describe the dynamics of the multi-body mechanism. When generalized coordinates are used, the differential equations of constrained motion take the form

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}_q^T \boldsymbol{\lambda} = \mathbf{Q}^e + \mathbf{Q}^v \quad (1)$$

where \mathbf{q} is the vector of generalized coordinates that defines the position and orientation of each body in the system, \mathbf{M} the mass matrix, \mathbf{Q}^e the vector of generalized forces, \mathbf{Q}^v the quadratic velocity vector that includes velocity-dependent Coriolis and centrifugal forces, \mathbf{C}_q the Jacobian matrix of the constraint equations, and $\boldsymbol{\lambda}$ the vector of Lagrange multipliers. Note that the vector \mathbf{Q}^v contains non-zero rotational quantities because a three-dimensional space is used. Generally, in a three-dimensional case, the description of mass inertias is not constant leading to non-zero vector \mathbf{Q}^v . However, owing to a centroidal reference frame the translational components of \mathbf{Q}^v which are used are equal to zero. In order to take into account the kinematic joints connecting the separate bodies, the following vector of constraint equations for position must be satisfied

$$\mathbf{C}(\mathbf{q}, t) = \mathbf{0} \quad (2)$$

Equations (1) and (2) comprise a system of differential algebraic equations (DAEs), which determine the dynamical behaviour of the mechanism. To solve the equations using ordinary differential equation (ODE) integration methods, these equations must be transformed into the second-order ODE. Therefore, equation (2) is differentiated twice with respect to time

$$\dot{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{C}_q \ddot{\mathbf{q}} + (\mathbf{C}_q \dot{\mathbf{q}})_q \dot{\mathbf{q}} + 2\mathbf{C}_{qt} \dot{\mathbf{q}} + \mathbf{C}_{tt} = \mathbf{0} \quad (3)$$

By combining equations (1) and (3), the following matrix of linear equations can be obtained

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}_q^T \\ \mathbf{C}_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^e + \mathbf{Q}^v \\ -(\mathbf{C}_q \dot{\mathbf{q}})_q \dot{\mathbf{q}} - 2\mathbf{C}_{qt} \dot{\mathbf{q}} - \mathbf{C}_{tt} \end{bmatrix} \quad (4)$$

which can be integrated using the ODE solver [10]. However, equations of motion do not guarantee that constraint equations in equation (2) are satisfied. This is due to the fact that during differentiation constant terms of constraint equations disappear and equation (4) fulfils constraints at acceleration level only. Therefore, integration causes accumulation of errors to the kinematic constraints. Constraint equations can be fulfilled by employing a stabilization

procedure. Another possibility to solve this problem is to use methods which produce a general solution to DAEs [13, 19].

3.3 Description of the friction model

Translational velocities of the carriages and sledges are low during the work cycle. For this reason, the effect of the friction forces should be studied carefully. Generally, the friction model can be divided into three regimes: sliding, stiction, and the transition between them. Consequently, the friction coefficient is a function of velocity. When dynamic friction is applied to the system, it can be expressed as follows

$$\mathbf{F}_f = f(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{F}_n = f(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{R}(\mathbf{q}) \mathbf{C}_q^T \boldsymbol{\lambda} \quad (5)$$

where \mathbf{F}_n is a vector of joint constraint forces, $f(\mathbf{q}, \dot{\mathbf{q}})$ the function of the friction coefficient, $\mathbf{R}(\mathbf{q})$ a matrix that is defined uniquely for each joint type, and $\mathbf{C}_q^T \boldsymbol{\lambda}$ describes constraint reaction forces. Stiction occurs when the relative velocity between sliding surfaces in the joint approaches zero. Stiction can be described as an additional constraint applied to the corresponding degrees of freedom. Stiction is valid as long as the friction force is lower than the static friction force [20]. This approach, however, is inconvenient because the numerical integration procedure must be stopped and restarted again when the stick-slip phenomenon takes place. In this study, the friction model is simplified and defined as a combination of Coulomb and viscous friction. The friction coefficient in the function of velocity is depicted in Fig. 4, where the static and dynamic friction coefficients are used to define the boundaries of stiction and sliding regimes.

3.4 Description of contacts

A contact model between the rolls is needed in order to describe the nip force between the reel drum and

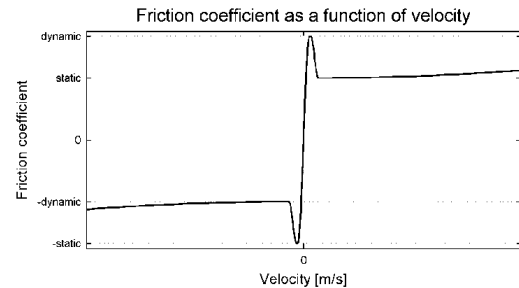


Fig. 4 Friction coefficient as a function of velocity

the reel spool. The main features studied in winder dynamics are associated to the rotation and excitation frequencies of the paper rolls, the change of the roll mass and winding geometry caused by the change in roll diameter, and stiffness and damping characteristics of the nip contact [17]. In this study, contact forces are described using an approach that resembles the procedure introduced by Jorkama [17]. It is noteworthy that the study of Jorkama contains only contacts in a plane, whereas in this study the approach is extended to spatial systems by discretizing the length of the cylinder. The enlargement of the paper roll during winding can be approximated by the compulsive movement of the reel drum which transfers the contacting bodies at constant velocity. The diameter of the paper roll can be described as

$$d(t) = \sqrt{d_0^2 + \frac{4\tau}{\pi} \int_0^t v dt} \quad (6)$$

where τ is the average web thickness, d_0 the initial roll diameter, and v the web speed [17]. In this study, the nip contact is modelled as a system of non-linear spring-dampers. On the basis of the elasticity of the bodies in contact, the spring and damping constants can be defined as will be introduced below. Applying the Herzian contact theory [17, 18], the penetration depth of the cylinders can be expressed as

$$\delta = F_c \frac{(1 - \nu^2)}{\pi E} \left[2 \ln \left(\frac{2d}{a} \right) - 1 \right] \quad (7)$$

where F_c is the contact force of the nip, d the diameter of the paper roll, E the modulus of elasticity, ν the Poisson ratio of the paper roll, and a the semicontact width which can be calculated as

$$a = \sqrt{\frac{2d^* F_c}{\pi E^*}} \quad (8)$$

In equation (8), E^* is the effective modulus of elasticity and d^* is the relative curvature of cylinders which can be defined as

$$d^* = \left(\frac{1}{d} + \frac{1}{d_1} \right)^{-1} \quad (9)$$

where d and d_1 are the diameters of the paper roll and reel drum, respectively. The effective modulus of elasticity takes the form

$$E^* \approx \frac{E}{1 - \nu^2} \quad (10)$$

It is noteworthy that the use of the Herzian contact theory leads to a computationally expensive procedure. For this reason, this study employs an approach that is based on the use of approximate polynomials. The nip contact is modelled and included in the reel model using a system of springs which are distributed along the length of the contact line. The contact force in this approach is modelled using the approximated equation as

$$F_c = \begin{cases} k_a(x_0 - x)^e + c_a^{\max} \dot{x}, & x_0 - x \geq d_a^{\max} \\ k_a(x_0 - x)^e + c_a^{\max} \Delta^2 \\ \quad \times (3 - 2\Delta) \dot{x}, & 0 < x_0 - x < d_a^{\max} \\ 0, & x_0 - x \leq 0 \end{cases} \quad (11)$$

where $\Delta = (x_0 - x)/d_a^{\max}$. In equation (11), $x_0 = (d + d_1)/2$, x is the distance between cylinder centres, parameters k_a and e are approximated using equation (7) that describes the compression of cylinders at a certain contact force, d_a^{\max} is the penetration depth when the maximum damping coefficient c_a^{\max} is applied. In equation (11), different regions are used in order to smooth discontinuities of contact forces. When contact occurs, the compression of cylinders in equation (7) is a function of the contact force which cannot be solved explicitly. In equation (11), contact force is directly a function of nip deformation and can be solved without iteration. Unknown parameters in equation (11) will be defined in such a manner that contact force closely resembles the Herzian contact theory.

3.5 Description of the hydraulic circuit with the servo control system

In this study, the lumped fluid theory [21] is used in the modelling of the hydraulic circuit of the reel. In the lumped fluid theory, a hydraulic circuit is divided into volumes where the pressure is assumed to be equally distributed. The valves are modelled employing a semi-empirical approach [22] where the parameters used in the flow equations through the orifice can be obtained, in many cases, from manufacturer catalogues. In the lumped fluid theory, the differential equation for pressure in the volume can be expressed as

$$\dot{p} = \frac{B_c}{V} \left(\sum_{i=1}^{N_i} Q_{in_i} - \sum_{j=1}^{N_o} Q_{out_j} - \dot{V} \right) \quad (12)$$

where B_c is the effective bulk modulus of the volume, V the volume, \dot{V} the change of volume with respect to time, and Q_{in} and Q_{out} are input and output flow rates, respectively. Indexes N_i and N_o are the

numbers of input and output flow rates, respectively. The flow rates in the hydraulic circuit are often modelled assuming the flow to be turbulent [21]. A leak in the direction valve can also be accounted for in the modelling approach. In this study, the friction in the cylinder is modelled as a combination of Coulomb and viscous friction.

The quantity of the wasted paper near the reel spool can be minimized using accurate control of the most critical winding parameters such as torque, nip load, and tension [15]. Most of the wasted paper results from the starts and ends of winding the paper roll. In practice, the force feedback is used as a control method for the reel motion. Accordingly, the value for the input signal in voltages can be written as

$$U_{\text{control}} = K_p U_{\text{err}} + K_i \int U_{\text{err}} \quad (13)$$

where K_p is the proportional gain, K_i the integration gain, and U_{err} represents the difference between the reference and the measured signals of the cylinder forces of the reel. The servo valve is modelled using a second-order dynamic model, which can be described using transfer function $G(s)$ as

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (14)$$

where ω_n is the natural frequency and ζ the damping. These coefficients can be defined using the Bode-diagram given to the modelled valve as

$$\omega_n = \frac{1}{\tau_2}, \quad \tau_1 = \frac{\tau_2}{|G|_{\omega=90^\circ}}, \quad \zeta = \frac{\tau_1}{2\tau_2}$$

where a -90° phase lag point is used to define ω_n and magnitude ratio $|G|_{\omega=90^\circ}$, τ_1 and τ_2 are time constants [23]. The hydraulic servo control system of the nip load is described in Fig. 5.

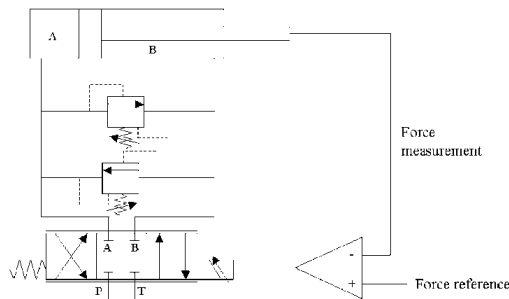


Fig. 5 The principle of the hydraulic circuit used to control the nip load.

4 SIMULATED FAULTS AND RESULTS

In the following section, three abnormal operation conditions of the reel are simulated. The simulation model introduced in the previous section is used in the analysis of the following fault conditions during the working sequence:

- excessive friction on the rails;
- leakage in the servo valve;
- noise in the force sensor signal.

In all simulated scenarios, the cylinder reference force is set to produce a nip load of 4000 N/m, while the web speed is kept constant. In the reel, it is assumed that the thickness of the paper layer on the reel is 1 m. The friction forces on the rails are defined in equation (5) accounting for both sliding and stiction effects. The numerical value 0.013 is used in the sliding area and value 0.015 is used in the stiction area. Using these parameters, the model can be tuned in such a way that the force resulting from the hydraulic servo control system and the material-dependent contact model imitate the working conditions of the real reel. It is important to note that this study is focused on finding the relations between the model state variables and the measurements in fault situations. Results of abnormal conditions are compared with the results from the normal condition. Simulated results are shown without exact numerical values in order to protect the design of the reel.

4.1 Case 1: excessive friction on the rails

In the first example, the excessive friction is imposed on the rails of carriage A, shown in Fig. 1. In this example, the static and dynamic friction coefficients are increased from 0.015 to 0.06 and from 0.013 to 0.058, respectively. Owing to the force control, the friction causes an increase in nip load at side A of the reel. This can be observed as the error in the position of the carriages as depicted in Fig. 6.

When the paper roll is completed, the secondary carriages transfer the parent reel to storage position. During the transfer sequence, the excessive friction can be perceived from the force sensors of the hydraulic actuators or the pressures of cylinder chambers as shown in Fig. 7.

4.2 Case 2: leak in the servo valve

In the second fault example, the leak in the servo valve is studied. The leakage is modelled as an increase of relative leaking from the pressure port to port A of the valve (see Fig. 5). The leaking of the valve is cumbersome to observe without the use of flow sensors. In this example, noticeable changes can be observed in the pressures in cylinder

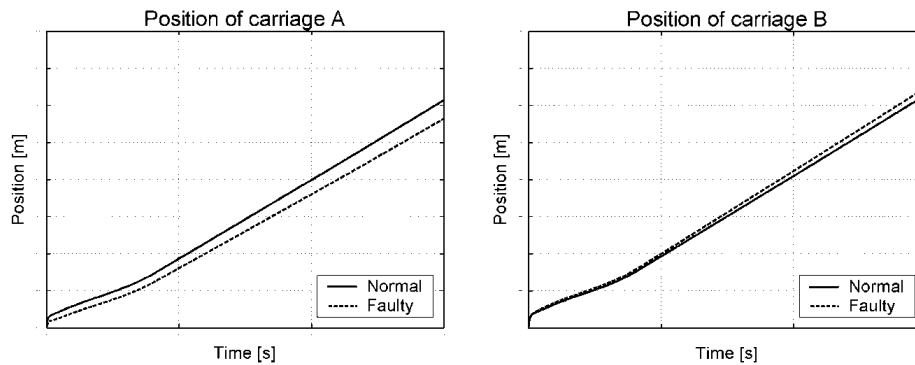


Fig. 6 Position of carriages when excessive friction is applied to the rails of carriage A

chambers, the spool displacement of the servo valve, and the control voltage. Figure 8 shows the difference between pressures and control voltages in the case of leaking and in normal operation conditions. On the basis of the simulations of this fault, the pressure in chamber A of cylinder A decreases when relative leaking is decreased as can be seen in Fig. 8. If relative leaking is increased, the force servo did not manage to achieve the desired force.

4.3 Case 3: noise signal in the force sensor

In this example, a noise signal in the force sensor is studied. When a random noise signal is added to the force sensor in the cylinder of carriage A, the effect of the signal can be seen as superimposed vibration in the system as shown in Fig. 9. Another effect is the raise of the pressure in the hydraulic system which is shown in Fig. 9, by depicting a pressure in chamber A of the cylinder at side

A. This pressure raise is caused by the effort of the control system to achieve the reference value because an increased need of reaction to varying measurement signals increases the pressure.

4.4 Discussion about the results

Three faulty situations are studied in order to show how the faults can be indicated with different state variables. Using a multi-body simulation model, the faults that have an effect on the quality of the paper roll can be isolated. Accordingly, a multi-body simulation model can be used in the design of a fault diagnosis system to define the necessary measurements. In the case of the reel, special attention must be paid to the effect of faults on the nip load. A varying distribution of the nip load may be harmful because the radius of the paper roll will be different at the ends of the roll. Consequently, the internal roll stress distribution of the parent reel is

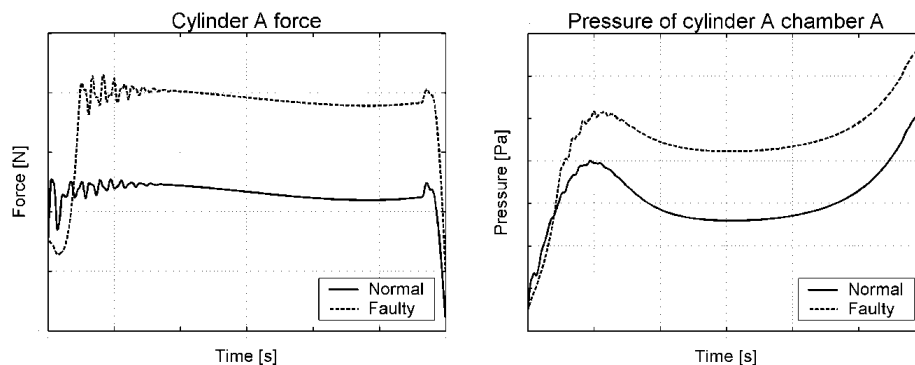


Fig. 7 The force of cylinder A and the pressure of cylinder A in chamber A when excessive friction is applied to the rails of carriage A during the transfer to the storage position

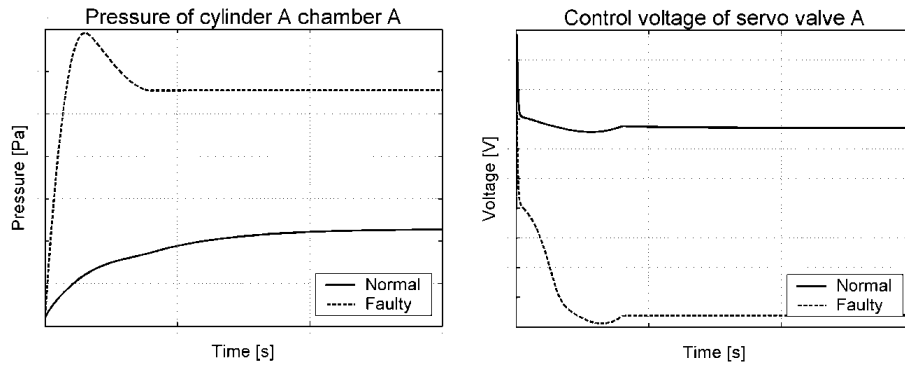


Fig. 8 The pressure difference in cylinder chamber A and the voltage difference in the control of the servo valve when the leak in the servo valve at side A is considered

not uniformly distributed, which leads to problems in roll quality. In Fig. 10, the nip forces are depicted in simulated normal and three faulty conditions, where the figure on the right is a close-up of the lower curves of the figure on the left.

Despite the faults, the force control manages to achieve its reference value with acceptable accuracy. However, in the case of excessive friction, nip force is not equally distributed as a result of which the paper quality is decreased. As can be seen from Fig. 10, excessive friction on the rails causes the raise of the nip contact force at the side of the reel where friction is applied. The noise signal in the force sensor can be detected as a varying nip load, which may cause changes in the hardness of the paper roll and also non-uniform stress inside the roll. The simulated leaking in the servo control valve does not affect the nip load directly. However, in the case of excessive leaking, a failure in operation occurs. This can

be detected before the final failure from the pressure in the cylinder chamber, as depicted in Fig. 8.

5 CONCLUSIONS

In this study, a multi-body simulation approach was used to study faults of the reel. A brief description of modelling methods of the reel was introduced in the study. The introduced simulation model was used to study three faulty operation conditions.

Typically, model-based condition monitoring relies on the mathematical system model that is constructed using measurements and applying signal-processing techniques. Using a standard-like physical modelling approach such as multi-body system dynamics, the functionality of the complete machine system, including actuators and control systems, can be used as a basis for condition

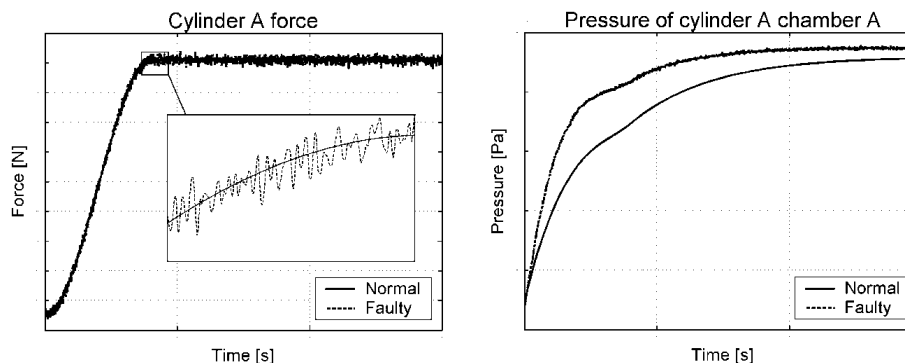


Fig. 9 Force difference of the cylinder A and pressure difference in cylinder chamber A when a noise signal is added to the force sensor signal of the cylinder

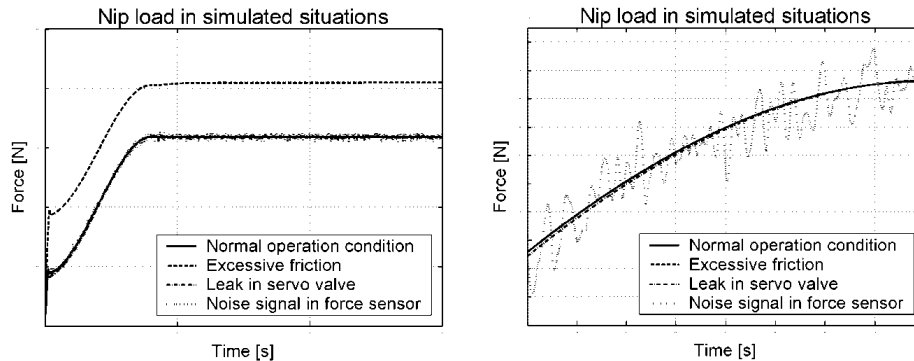


Fig. 10 The nip load in simulated conditions. The figure on the right is a close-up of the lower curves in the left figure

monitoring. In the multi-body approach, the modelling of a machine can be accomplished in a systematic manner. Usually, the models used in condition monitoring and fault detection are linearized. In the simulation approach used in this study, nonlinearities are taken into account while the functionality of a complete machine is considered. Accordingly, frictions, contacts, and different kinds of actuators can be accounted and used in the condition monitoring and fault detection. Using multi-body simulation in condition monitoring and fault diagnosis, it is possible to get information from the system with a simple sensor configuration. On the other hand, variables that cannot be measured can be defined by utilizing the simulation model. With the simulation model, the relationships between model state variables can be obtained and the failures in the system can be recognized without the need for additional sensors. In the design of a fault detection system, this knowledge can be used to observe the measurable variables that indicate a faulty situation. If there is a lack of measurements in faulty situations, the faulty situations can be reconstructed using a simulation model of the system. The model can also be utilized in the training of expert systems for fault detection.

Generally, when the multi-body approach is used, the simulation model can be integrated into the real operating environment. This method can be utilized to develop and test the fault diagnosis systems and to help understand the physical characteristics of the faulty situations. Furthermore, sensitivity to parameter changes of the modelled machine can be studied. The simulation results introduced in this study showed the applicability of the multi-body approach to fault diagnosis. Faulty behaviour can be detected even if it does not harmfully affect the

functionality of the reel as in the case of leaking in the valve.

Because a large amount of information resulted from the simulations must be analysed, some classification methods should be used to identify faults. Moreover, some parts of the model, which are cumbersome to define using the modelling tools available, can be replaced by data generated from the measurements of the real system. These are research topics for future studies.

REFERENCES

- 1 Willsky, A. S. A survey of design methods for failure detection in dynamic systems. *Automatica*, 1976, 12(6), 601–611.
- 2 Loparo, K. A., Adams, W. L., Farouk, A.-M. M., and Afshari, N. Fault detection and diagnosis of rotating machinery. *IEEE Trans. Ind. Electron.*, 2000, 47(5), 1005–1014.
- 3 Dvorak, D. and Kuipers, B. J. Model-based monitoring of dynamic systems. International Joint Conference on *Artificial intelligence* (IJCAI'89), Detroit, 20–25 August, Vol. 2, pp. 1238–1243 (Morgan-Kaufmann Publisher Inc., San Mateo, CA).
- 4 Isermann, R. and Ballé, P. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Eng. Pract.*, 1997, 5(5), 709–719.
- 5 Isermann, R. Process fault detection based on modeling and estimation methods – a survey. *Automatica*, 1984, 20(4), 387–404.
- 6 Frank, P. M. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy – a survey and some new results. *Automatica*, 1990, 26(3), 459–474.
- 7 Basseville, M. Detecting changes in signals and systems – a survey. *Automatica*, 1988, 24(3), 309–326.

- 8 Patton, R. J., Chen, J., and Nielsen, S. B.** Model-based methods for fault diagnosis: some guide-lines. *Trans. Inst. Meas. Control*, 1995, **17**(2), 73–83.
- 9 Bettig, B. P. and Han, R. P. S.** Predictive maintenance using the rotordynamic model of a hydraulic turbine-generator rotor. *ASME J. Vibration Acoust.*, 1998, **120**(2), 441–448.
- 10 Shabana A. A.** *Dynamics of multibody systems*, p. 372, ISBN 0-521-59446-4 (Cambridge University Press, Cambridge).
- 11 Schiehlen, W.** Multibody system dynamics: roots and perspectives. *Multibody Syst. Dyn.*, 1997, **1**, 149–188.
- 12 ADAMS Online Documentation**, ADAMS/SOLVER [version 2003.0.1], 2003, MSC Software Corporation.
- 13 García de Jálón, J. and Bayo, E.** *Kinematic and dynamic simulation of multibody systems – the real-time challenge*, p. 440, ISBN 0-387-94096-0 (Springer, New York).
- 14 Cuadrado, J. and Dopico, D.** Penalty, semi-recursive and hybrid methods for MBS real-time dynamics in the context of structural integrators. *Multibody Dynamics 2003 Proceedings CD*, Lisbon, Portugal, 2003.
- 15 Jokio, M.** *Papermaking part 3, Finishing*, p. 361, ISBN 952-5216-10-1 (Gummerus Printing, Jyväskylä).
- 16 Ola, K.** *Lifecycle management for improved product and system availability*. ABB Review Special Report II, Retrieved June 29, 2004. <http://www.abb.com/global/gad/gad02077.nsf/menu/us-3bce?OpenDocument>.
- 17 Jorkama, M.** The role of analytical winding dynamics in winder design. *Tappi J.*, 1998, **81**(1), 202–207.
- 18 Johnson, K. L.** *Contact mechanics*, p. 452, ISBN 0-521-34796-3 (Cambridge University Press, Cambridge).
- 19 Haug, E. J. and Yen, J.** Generalized coordinate partitioning methods for numerical integration of differential–algebraic equations of dynamics. In *Real-time integration methods for mechanical system simulation*, NATO ASI Series, Vol. F69, pp. 97–114.
- 20 Haug, E. J., Wu, S. M., and Yang, S. M.** Dynamics of mechanical systems with Coulomb friction, stiction, impact and constraint addition–deletion–I, Theory, *Mech. Mach. Theory*, 1986, **21**(5), 401–406.
- 21 Watton, J.** *Fluid power systems*, p. 490, ISBN 0-13-323197-6 (Prentice-Hall, UK).
- 22 Handroos, H. M. and Vilenius, M. J.** Flexible semi-empirical models for hydraulic flow control valves. *J. Mech. Design*, 1991, **113**(3), 232–238.
- 23 Virvalo, T.** *Modelling and design of pneumatic position servo system realized with commercial components*, p. 197 (Tampere University of Technology, Tampere).

c_a^{\max}	maximum damping coefficient
C	vector of kinematic constraint equations
C_q	constraint Jacobian matrix
d	diameter of the paper roll
d^*	relative curvature of cylinders
d_0	initial diameter of the paper roll
d_1	diameter of the reel drum
d_a^{\max}	penetration depth when the maximum damping coefficient is achieved
e	positive real variable that specifies the exponent of the force deformation characteristic
E	Young's modulus
E^*	effective modulus of elasticity
f	function of friction coefficient
F_c	nip contact force
F_f	vector of friction forces
F_n	vector of joint constraint forces
$G(s)$	transfer function of valve
$ G _{\omega=90^\circ}$	magnitude ratio when -90° phase lag point is used
k_a	stiffness coefficient
K_I	integration gain
K_P	proportional gain
M	mass matrix
N_i	number of input flow rates
N_o	number of output flow rates
p	pressure in the volume
q	vector of generalized coordinates
Q_{in}	input flow rate
Q_{out}	output flow rate
Q^e	vector of generalized forces
Q^v	vector of quadratic velocity inertia terms
R	matrix that defines direction of friction force in the joint
t	time
$U_{control}$	input control signal
U_{err}	difference between the reference and the measured signals
v	web speed
V	volume
x	distance between centres of paper roll and reel drum
δ	penetration depth of rolling cylinders
Δ	constant used to define contact force
ζ	damping coefficient of the servo valve
λ	vector of Lagrange multipliers
ν	Poisson's ratio of the paper roll
τ	average web thickness of the paper roll
τ_1, τ_2	time constants
ω_n	natural frequency of the servo valve

APPENDIX

Notation

a	semicontact width of the nip contact
B_e	effective bulk modulus

V

“Container Gantry Crane Simulator for Operator Training”. Reprinted from the Journal of Multi-Body Dynamics, with kind permission of the Professional Engineering Publishing.

Container gantry crane simulator for operator training

A Rouvinen*, T Lehtinen, and P Korkealaakso

Department of Mechanical Engineering, Lappeenranta University of Technology, Lappeenranta, Finland

The manuscript was received on 26 April 2004 and was accepted after revision for publication on 21 July 2005.

DOI: 10.1243/146441905X63322

Abstract: Real-time simulators have become more popular in the field of user training. This is due to the possibility to give basic training and knowledge of machines and their operation environment to the operator even when the machine is not actually present. The use of simulators instead of actual machines has several advantages. First of all, the available machine capacity is not tied to training and can be used in productive work. Secondly, using a simulator helps to avoid accidents that may occur using real machines. Using a simulator also enables different environmental aspects, such as lighting conditions, fog, wind, and so on, to be taken into account in the training of all operators alike. Real-time training simulators are complicated machine systems, which consist of a user interface, an I/O-system, a real-time simulation model describing the dynamics of the machine in question and its connections to the environment, a visualization of the operational environment, and a possible motion platform. The user interface is usually taken directly from the simulated machine. Consequently, the user has the possibility to become familiar with the operating interface in an early phase of training.

In this article, the development of a gantry crane operator-training simulator, including all the earlier mentioned components, is presented. The aim of this article is to present an example of methods used in the development of the separate areas of a training simulator.

Keywords: gantry crane, real-time simulation, user training, motion platform, dynamic modelling

1 INTRODUCTION

Gantry cranes are used in a harbour environment to move containers between the pier and the ships. The crane (Fig. 1) consists of three main parts that move in relation to the pier and each other: the gantry moves along the rails in the direction of the pier and the trolley moves along the rails attached to the gantry perpendicular to the pier. The spreader, used to grab the containers, is attached to the trolley via cables and carries out the hoisting movement by winding and unwinding the cables.

Traditionally, gantry crane operators are trained while operating actual cranes with real cargo in

harbours. This might cause dangerous situations, decrease work efficiency, and even become very expensive if mistakes occur. Utilization of a real-time simulator in an earlier phase of training enables the safe and controlled learning of the basic skills required in operating the crane. In a virtual environment, it is safe and possible to produce and practice the crane operations in conditions and situations that might be rare or not normally be possible at all.

While using the simulator, the operator sees the container moving in a virtual world. The virtual world combined with the motion platform, which creates movement effects, gives the operator real world-like feedback. The surrounding sounds give the final touch to the simulator environment. Besides the visual feedback, the feedback from the motion platform is very important in operator training. If the operator does not feel the accelerations of the machine system caused by control manoeuvres, the training may cause operating behaviour that is too

*Corresponding author: Department of Mechanical Engineering, Institute of Mechatronics and Virtual Engineering, Lappeenranta University of Technology, Skimmariilankatu 34, PO Box 20, FIN-53851 Lappeenranta, Finland.

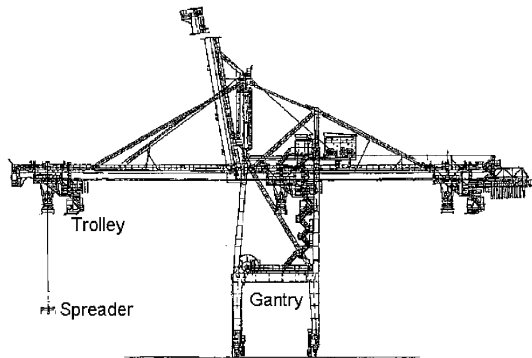


Fig. 1 Side view of a gantry crane

fast and harsh as swift operation movements seem not to cause any physical feedback. When motion platform feedback is added, the operating behaviour becomes smoother, because discontinuous control signals usually cause accelerations unpleasant to operator.

Basically real-time simulators consist of a user interface, an I/O-system, a real-time simulation model describing the dynamics of the machine in question, a visualization of the environment, and a possible motion platform producing acceleration effects for the operator. Figure 2 presents the components required to build a real-time training simulator.

Control signals given by the operator are transferred via the I/O-interface into the dynamic simulation model. In simulation models, signals can be treated in several different ways. Simple binary on/off-signals can be used to launch operation cycles such as opening or closing the container locking. Analogue signals can be used to give reference values for more complicated functions. The

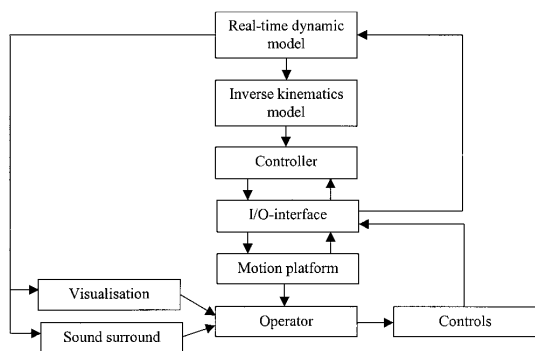


Fig. 2 Components of a real-time training simulator system

rotational velocity reference of an electric drive is a good example of the use of analogue signals. During the study, it became clear that the most natural way to treat analogue control signals is to use them as a reference for the calculation of external forces acting in the system. This way, control signals seldom cause discontinuities, as the change in force causes change in system accelerations, which are integrated twice to solve the position information of the system.

The position information of separate parts is transferred into the visualization system, where the virtual world is drawn. In this case, also the surrounding sound is controlled by the dynamic model. Sounds related to different events, such as collisions, the use of motors, and so on, are played on the basis of event information gathered from the simulation model. The information of accelerations affecting the operator is used in calculating reference values for the motion platform. An inverse kinematic model of the motion platform is required to map the operator's acceleration information from the global coordinates to actuator positions and velocities.

A control system is used to calculate the required reference values for the hydraulic cylinders that are used as actuators in the motion platform. The control method is based on the position and velocity information of the actuators. The information is calculated in the motion platform's inverse kinematic model. Control signals for valves are calculated using this information and the measured position values of the actuators. The control signals are then transferred into relief valves as voltage signals via the same I/O-interface as used in the measurement of the actuator positions.

Because the simulator was intended for operator-training purposes and might be operated by training personnel without programming level computer skills, the use of commercial software was required. Commercial real-time environments are tested in several applications and besides reliability, they offer user interfaces that enable the modification of operating parameters during simulation. The real-time environment must also offer the possibility to communicate with the real world, i.e. an I/O-interface as well as an interface for visualization. Commercial real-time systems usually also offer an interface for graphical model development environments, which allow the dynamic model to be constructed combining block diagrams of basic mathematical operations and C-language functions for special purposes. These development environments also enable the use of tested numerical solution algorithms, which is very important in order to achieve a numerically stable real-time solution.

2 SIMULATION MODEL

The dynamic model of the crane consists of several submodels. These submodels include the dynamic models of the parts of the crane, a model detecting collisions of the container and computing contact forces, an environment model describing collision planes, and electric motor models producing forces, which move separate parts. The contact forces caused by the spreader guide arms are also included in order to model the fastening of the container to the spreader more realistically. As the dynamic model itself is modelled using Lagrange's method, it can be very easily divided into several subcomponents that can be modelled separately. This modular division makes it easier to develop, administer, and test the correct operation of the required mathematical functions. Modularity also enables easier expansion of the model, if more properties or operations are required.

2.1 Dynamic model of the gantry crane

The gantry crane is divided into three rigid moving bodies, the gantry, the trolley, and the spreader, used to carry containers. The gantry moves along the rails in the direction of the pier, so it has one degree of freedom (DOF). The trolley moves along the rails attached to the gantry perpendicular to the pier, so it basically has one DOF. Nevertheless, owing to the structural flexibility of the gantry, the operator senses vertical accelerations while hoisting or lowering the container, therefore another DOF is required to model this direction of motion. The spreader is attached to the trolley via cables and has all six DOFs. The bodies of the dynamic model of the gantry crane and the coordinate systems attached to the bodies are presented in Fig. 3.

The model has a total of nine DOFs. Owing to possible singularities during numerical solutions, a more

stable behaviour can be achieved using more than three rotational coordinates. In this case, four Euler parameters were selected. As the four parameters increase the DOFs, normalization constraint of the coordinates is required [1–4]. The generalized coordinates q that fully describe the DOFs of the system can be expressed as

$$q = [Z_g \ X_t \ Y_t \ X_c \ Y_c \ Z_c \ \theta_{0c} \ \theta_{1c} \ \theta_{2c} \ \theta_{3c}]^T = [R^T \ \theta^T]^T \quad (1)$$

where

$$R = [Z_g \ X_t \ Y_t \ X_c \ Y_c \ Z_c]^T \quad \text{and} \quad \theta = [\theta_{0c} \ \theta_{1c} \ \theta_{2c} \ \theta_{3c}]^T \quad (2)$$

The normalization constraint equation of the Euler parameters is

$$C(q) = \theta^T \theta - 1 = 0 \quad (3)$$

Using Lagrange's method, it is possible to model the connecting joints between separate bodies by using constraint equations that describe joints between bodies. However, the use of constraint equations increases the size of matrices to be computed and thereby decreases computational efficiency, which is a very important aspect. Therefore, the rail contact between the gantry and the pier is implemented by giving the gantry just one DOF, which is in the direction of the rail. Similarly, the translational joint between the gantry and the trolley is implemented by giving the trolley only one DOF in the direction of the joint. The second, vertical DOF is used to model the structural flexibility of the gantry.

The solution of dynamics is based on Lagrange's equation of motion including constraints

$$M\ddot{q} + C_q^T \lambda = Q_c + Q_v \quad (4)$$

which can be presented in numerically solvable form

$$\begin{bmatrix} M & C_q^T \\ C_q & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q_c + Q_v \\ Q_c \end{bmatrix} \quad (5)$$

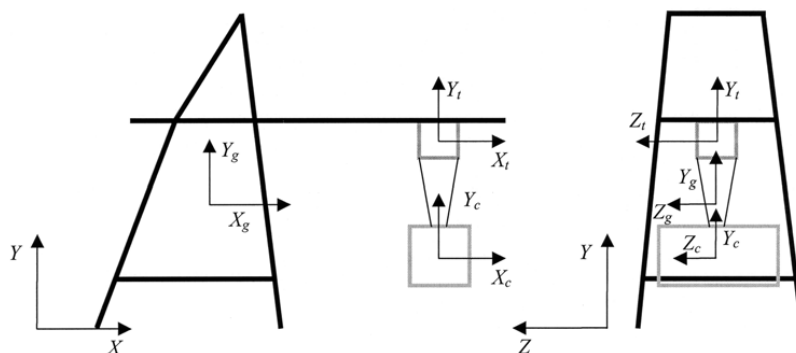


Fig. 3 Coordinate systems of the dynamic model of the gantry crane

where \mathbf{M} is the mass matrix, \mathbf{C}_q the Jacobean matrix of constraint equations, \dot{q} the vector of generalized coordinate accelerations, λ the vector of Lagrange multipliers, \mathbf{Q}_e the vector of generalized forces, \mathbf{Q}_v the quadratic inertia force vector, and \mathbf{Q}_c the vector that includes the second differentials of constraints. As can be seen, this form of equation provides a natural way to divide the solution into subcomponents, each of which includes one term of equation (5). A more detailed description of the components of equation (5) can be found from references, for example [1–4].

In the case of the gantry crane simulator, the most interesting component of equation (5) is the vector of generalized forces and more accurately, the external forces that are mapped to generalized forces. Generalized forces are external forces that affect the generalized coordinates of the system. However, forces usually do not affect the origin point of the local coordinate system of a body. Therefore, a transfer from the force attachment point to the origin of the local coordinate system is required [3]

$$\begin{aligned}\mathbf{F}_{iR} &= \mathbf{F}_{ie} \\ \mathbf{F}_{i\varphi} &= -\dot{\mathbf{u}}_{iP}^T \mathbf{F}_{ie}\end{aligned}\quad (6)$$

where \mathbf{F}_{ie} is an external force acting in point P of body i , defined in the global coordinate system. The location of point P is defined in the global coordinate system by vector \mathbf{u}_{iP} . \mathbf{F}_{iR} and $\mathbf{F}_{i\varphi}$ are forces and moments related to translational and rotational coordinates. As shown in equation (6), this mapping causes moment effect on the rotational coordinates of the body. As rotations are described using Euler parameters, a mapping from the three-dimensional space of global rotations to the four-dimensional space of Euler parameters is required

$$\mathbf{F}_{i\theta} = \mathbf{G}_i^T \mathbf{F}_{i\varphi} \quad (7)$$

where matrix \mathbf{G}_i is a mapping from Euler parameter velocities to global rotational velocities

$$\boldsymbol{\omega}_i = \mathbf{G}_i \dot{\boldsymbol{\theta}}_i \quad (8)$$

and \mathbf{G}_i is

$$\mathbf{G}_i = 2 \begin{bmatrix} -\theta_{i1} & \theta_{i0} & -\theta_{i3} & \theta_{i2} \\ -\theta_{i2} & \theta_{i3} & \theta_{i0} & -\theta_{i1} \\ -\theta_{i3} & -\theta_{i2} & \theta_{i1} & \theta_{i0} \end{bmatrix} \quad (9)$$

Generalized forces can now be expressed as

$$\mathbf{Q}_{ie} = \left[\sum_{j=1}^{n_f} \mathbf{F}_{ijR}^T \quad \sum_{j=1}^{n_f} \mathbf{F}_{ij\theta}^T \right]^T \quad (10)$$

where n_f is the number of external forces acting in body i .

The interconnection between separate bodies can be expressed using external forces. The vertical connection between the gantry and the trolley can be implemented by using a spring force that depends on the relative vertical position between trolley and gantry. Control signals given by the operator are used as reference signals for electric drives. On the basis of the signal value and current velocity of a drive, the value is calculated for an external force affecting the body to which the drive is. Similarly, gravity, rope forces, wind load forces, and collision forces affecting the container can be taken into account. The main work in constructing a dynamic model is to form equations and furthermore numerical functions that can be used to define these forces.

2.2 Collision detection and contact modelling

Handling contacts between objects while avoiding the interpenetration of two bodies is one of the most important requirements of a realistic simulation. Normally, two different types of contacts can be considered. When two bodies move towards each other at a certain velocity and at some point come into contact with each other, there is colliding contact. Another form of contact is called resting contact. This is the case when the bodies are resting on one another at some point. Interpenetration of the bodies can be prevented by using contact forces acting in the contact point. These forces can only push the objects apart, not pull them.

Traditionally, there are three categories for the methods of modelling contacts: the analytical [5], the penalty [6], and the impulse methods [7]. In this study, contacts are modelled using the penalty method for interpenetration. The contact point can be found by calculating distances between the points and the surfaces of the colliding bodies. Contact forces are obtained using a virtual spring-damper element at the contact point. This is quite a simple method, but the friction forces can also be considered.

This method is not highly accurate but it serves its purpose in this application. Because of real-time application, it is important to use a computationally efficient method. The method used does not allow the exact specification of the moment of time of collision and allows some interpenetration. The computational algorithm of contact is quite simple. Five collision planes are defined for each container position. Every hard point position on the body is checked for potential collision planes. If a collision point is found, then the data of the collision are written in the collision data structure. The collision

data includes the penetration between the hard point and the plane, the collision normal and tangent vectors, the specification of the actual contact point, and the relative velocity between the colliding bodies [8].

Contact forces are solved using results obtained from the collision detection algorithm. The direction of the force at the contact point is perpendicular to the contact plane. At the contact point, a virtual spring-damper element is temporarily placed to describe the contact force. Spring and damping constants are defined on the basis of the elasticity of the bodies. Friction forces are calculated using the friction coefficient and the normal force affecting the points of the body. The direction of the friction force is opposite to the velocity of the tangent contact front. When all forces caused by contact are calculated, the resulting moments can be accounted for. The algorithm used in handling contacts is described in Fig. 4.

Generally, a plane can be defined using two unit vectors \mathbf{u} and \mathbf{v} . The normal vector of the plane is a cross-product of these vectors

$$\mathbf{n} = \mathbf{u} \times \mathbf{v} \quad (11)$$

Distance between a point on the body P_b and any point on the plane P_{pl} is

$$\mathbf{D} = \mathbf{u}_{P_b} - \mathbf{u}_{P_{pl}} \quad (12)$$

The closest distance between the point and the plane is

$$d = \mathbf{D}^T \mathbf{n} \quad (13)$$

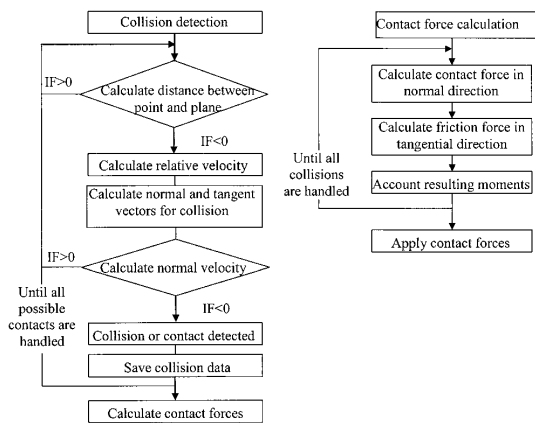


Fig. 4 Algorithms used in the computation of collision detection and contact forces

The relative velocity between two contact points is

$$\mathbf{v}_{rel} = \mathbf{v}_b + \mathbf{A}_b \tilde{\omega}_b \mathbf{u}_{P_b} - \mathbf{v}_{pl} - \mathbf{A}_{pl} \tilde{\omega}_{pl} \mathbf{u}_{P_{pl}} \quad (14)$$

where \mathbf{v}_b and \mathbf{v}_{pl} are velocities of body b and plane pl , \mathbf{A}_b and \mathbf{A}_{pl} are transformation matrices, and $\tilde{\omega}_b$ and $\tilde{\omega}_{pl}$ are skew-symmetric matrices of angular velocities. Velocity in the direction of the normal of the collision plane can be written as

$$\mathbf{v}_{rn} = \mathbf{v}_{rel}^T \mathbf{n} \quad (15)$$

To obtain the direction of the friction force, the tangent vector of the relative velocity on the plane must be described as

$$\mathbf{t} = (\mathbf{n} \times \mathbf{v}_{rel}) \times \mathbf{n} \quad (16)$$

Normalization results

$$\mathbf{t}_{norm} = \frac{\mathbf{t}}{|\mathbf{t}|} \quad (17)$$

In the case of collision, the contact force affecting the colliding bodies in the normal direction of the plane can be written as a spring-damper element

$$\mathbf{F}_n = -(kd + c\mathbf{v}_m) \mathbf{n} \quad (18)$$

where k and c are spring and damping coefficients, respectively. Friction forces are calculated in collision tangent plane as

$$\mathbf{F}_\mu = \mu \mathbf{F}_n \mathbf{t} \quad (19)$$

where μ is the friction coefficient. The total contact force at one contact point is now

$$\mathbf{F}_{cont} = \mathbf{F}_n + \mathbf{F}_\mu \quad (20)$$

2.3 Electric drive modelling

The gantry, the trolley, and the container can be moved in the directions of their DOFs using transfer forces produced by electric drives. The transfer force, F_{Tm} , is calculated on the basis of the driving torque, the total reduction ratio, and power transfer coefficient of transfer motor m . There are four motors in the gantry, the trolley is operated using four motors and the hoisting movement is implemented by using another four motors. In the simulation model, the forces affecting the gantry and the trolley are simply calculated by multiplying the transfer force of one motor by four. This is due to the fact that these motors should have the same angular velocity, based on the structure of the crane. Each

motor is described as a function of the torque of the motor and the angular velocity. The driving torque output depends on the difference between the required and achieved angular velocities. The bigger this difference, the more the torque is used to decrease it. The transfer force of a motor m can be defined as

$$F_{\tau m} = \frac{T_m}{i_{\text{totm}}} \eta_m - c_m \omega_m, \quad m = 1, \dots, 6 \quad (21)$$

where T_m is the driving torque of the motor, i_{totm} the total reduction ratio, η_m the coefficient of the driveline which includes electric and driveline losses and frictions of the motor and driveline, c_m the damping coefficient, and ω_m the angular velocity of the motor. The driving torque of the motor, T_m , can be written as

$$T_m = \min(T_m(\omega_m), T_{\text{refm}}) \quad (22)$$

where $T_m(\omega_m)$ is the driving torque of the motor as a function of the motor's angular velocity and T_{refm} is the driving torque based on the difference between the required and actual angular velocities

$$\dot{T}_{\text{refm}} = \frac{K_m(\omega_{\text{refm}} - \omega_m) - T_{\text{refm}}}{\tau_m} \quad (23)$$

where K_m is the amplification coefficient, ω_{refm} the required angular velocity of the motor, ω_m the existing angular velocity, and τ_m the time constant. The value for the amplification coefficient K_m can be described using the graph of the torque of the motor. The value for the time constant τ_m is found out by trying different values until the performance of the motor is fast enough.

2.4 Spreader guide arm modelling

Spreader guide arms, Fig. 5, are used to ease the fastening of the container onto the spreader. Because

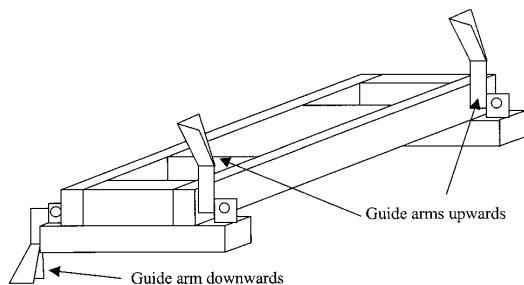


Fig. 5 Spreader guide arms

of the structure of the locking mechanism, the spreader must be positioned within the accuracy of centimetres into the correct position above the container, before the actual locking procedure is enabled. The spreader is a long distance from the operator and attached to ropes, which might cause serious problems and slow down the work. When spreader guide arms are in the reach of the container, they are lowered and used to restrict the movement of the spreader.

In this case, the spreader has six guide arms, positioned as shown in Fig. 6. The control of the arms is carried out so that arms one and two are operated simultaneously, arms three and four are separately operated simultaneously, and arms five and six separately. The upward positioning of all arms is operated simultaneously.

The guide arm actuation is modelled by the driving torque that affects the rotational acceleration of arm a

$$\dot{\theta}_a = \frac{T_{\text{tota}}}{J_a}, \quad a = 1, \dots, 6 \quad (24)$$

where J_a is the inertia of the arm and T_{tota} is the total driving torque affecting the arm. The total driving torque consists of the torque of the motor operating the guide arm and the torque caused by contact to the container or to the stop bar in both extreme positions

$$T_{\text{tota}} = T_a - T_{ca} \quad (25)$$

The driving torque of the motor can be expressed as

$$T_a = K_a(\theta_{\text{refa}} - \theta_a) - c_a \dot{\theta}_a \quad (26)$$

where K_a is the amplification coefficient of the motor driving arm a , c_a the damping coefficient, and θ_{refa} the reference value for the guide arm angle. The reference value for the guide arm angle is either $\pi/2$ or $-\pi/2$ depending on the desired position, up- or downwards. The contact torque related to the extreme positions is

$$T_{ca} = k_{ca}(\theta_a + \theta_{\text{refa}}) \quad (27)$$

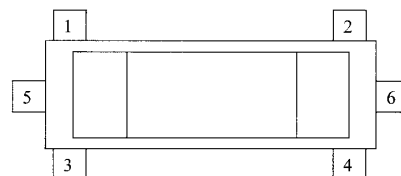


Fig. 6 Spreader guide arm positioning

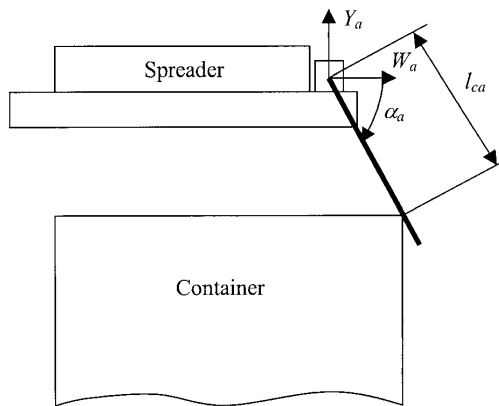


Fig. 7 Spreader guide arm contact variables

This torque is calculated only if the arm is in contact with the stop bar. In the case of contact with the container, the contact torque is

$$T_{ca} = k_{ca}(\theta_a - \alpha_a) \tag{28}$$

where α_a is the contact angle of the guide arm, presented in Fig. 7. The coordinate axis W_a is either the X - or Z -axis of the arm joint coordinate, depending on the possible contact direction of the arm. The external force affecting the spreader because of contact to the container is

$$\begin{aligned} \mathbf{F}_{arms} &= [F_X \ F_Y \ F_Z]^T \\ &= [F_1 + F_2 + F_3 + F_4 \ 0 \ F_5 + F_6]^T \end{aligned} \tag{29}$$

where the contact force F_a of each arm depends on the contact torque T_{ca} and the contact length of the arm l_{ca}

$$F_a = \frac{T_{ca}}{l_{ca}} \tag{30}$$

2.5 Implementation and numerical solution

Owing to the demand of commercial software and hardware environments, the selection of implementation environments was restricted to available real-time systems. Interfaces with modelling environments, as well as visualization capabilities, were considered during the selection. After a suitability analysis, the RT-LAB system made by Opal-RT was selected. The main reasons were the possibility to increase computing power by updating standard PC-hardware and interfaces with Matlab/Simulink and WorldUp-visualization software. The possibility

to use Matlab/Simulink also offered an interface to the I/O-system.

The use of Simulink software enables the graphic modelling of the system, using standard blocks from the Simulink library. More complicated sub-models can be written in ANSI-C and linked to the model. Matlab Real-Time Workshop enables the translation of the combined Simulink and C-language model into C-code for the RT-LAB simulation. In the solution of the equation of motion, equation (5), a fourth order Runge–Kutta solver was used with a time step of 0.005 s.

To ensure the numerical stability of the solution, a constraint stabilization method was used. As there is only one constraint, there should be no problems. However, to make sure, Baumgarte’s stabilization method [9] was used. Baumgarte’s stabilization was chosen, because it is computationally efficient, easy to implement, and works quite well, except near kinematic singularities [10]. However, the selection of suitable coefficients required in the algorithm may cause problems [11]. The coefficients required in the stabilization algorithm were selected on the basis of reference [11].

3 EVALUATION OF MODEL

To ensure the correct functionality of the real-time simulation model, the simulation results of the real-time model were compared with the results of off-line simulation results computed using commercial mechanism dynamic simulation program ADAMS. Figures 8 to 10 present comparison of trolley X -coordinate, container Y -coordinate, and container Z -coordinate, respectively. The test work cycle of off-line simulation is controlled using the control signal recorded from real-time simulation work cycle. As can be seen, there is a small difference in every coordinate. This is due to different numerical solution methods used in real-time and off-line simulation. If there is no solution inside the preset error tolerance, the off-line simulation tries new solution with smaller time step, whereas this is not an option in real-time simulation.

In the start-up of the system, there were a couple of professional gantry crane operators testing the systems. Because there were some unknown parameters, the expertise of the operators was used to set the parameter values so that the feedback of the system was correct. This is an important point as the simulator is used in training purposes and thus must give correct visual and acceleration feedback to operator.

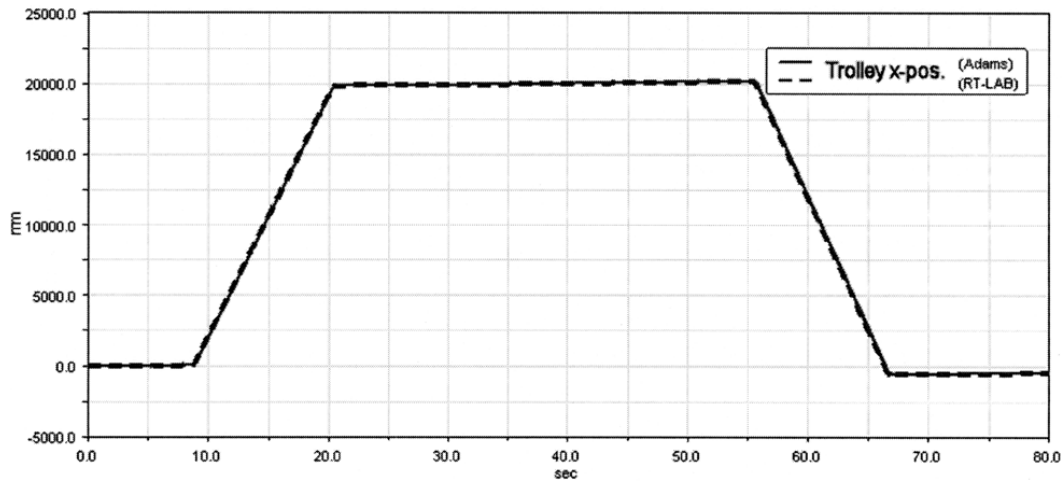


Fig. 8 Comparison of trolley X-coordinate

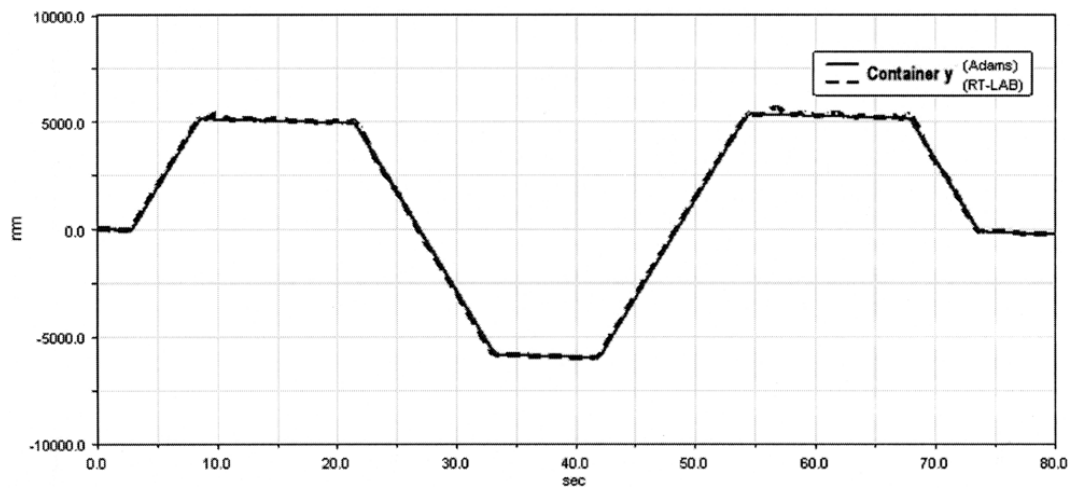


Fig. 9 Comparison of container Y-coordinate

4 IMPLEMENTATION ENVIRONMENT

As a real-time solution is required, the computer environment should enable sufficient numerical efficiency for the solution of the dynamic model. Furthermore, the control of the I/O-interface and the computation of control algorithms must be done simultaneously. These are the most important aspects related to the core of the simulator. Less important aspects, in the viewpoint of the accuracy of real-time functionality, are the control of the simulation and simulation parameters, visualization, and the control of the surrounding sound.

The visualization is carried out using WorldUp-visualization software. To achieve a realistic result, the virtual harbour environment is built using photo material from an existing harbour, crane, and ships as well as a few types of containers. The required coordinate information of the bodies is transferred from the target PC to the command station via the Ethernet connection. In WorldUp, the coordinate information required to position bodies is connected to movable objects using WorldUp BasicScript language. All the graphics were modelled separately using three-dimensional Studio Max software and then imported to the virtual

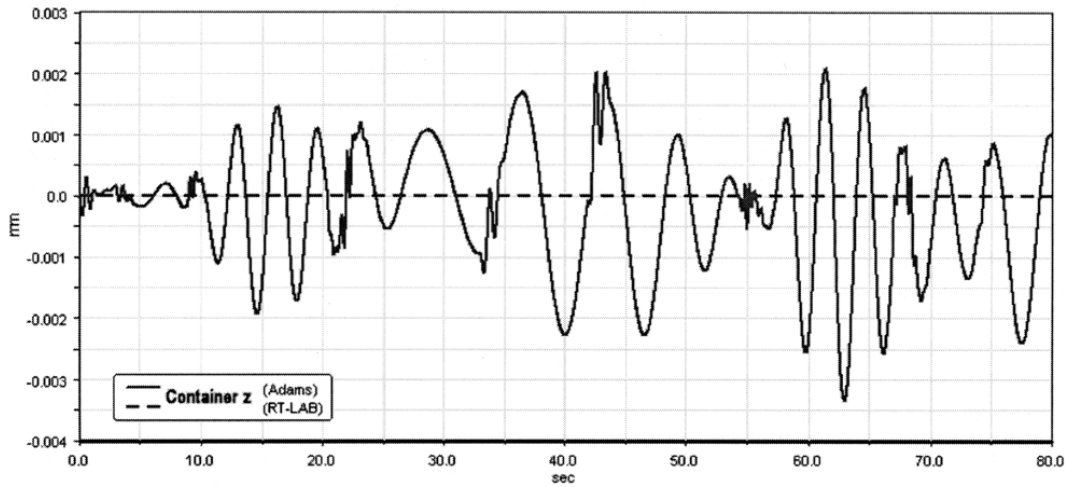


Fig. 10 Comparison of container Z-coordinate

harbour. The use of separate programs during development process is presented in Fig. 11.

As there is a natural division into two categories of real-time performance, it is logical to divide these tasks between two separate computers. This is exactly how the simulator computer environment is implemented. The basic idea of the RT-LAB real-time simulation is to distribute computational simulation tasks across a cluster of computers, Fig. 12. The RT-LAB real-time system configuration used in the study includes two PCs. One PC is equipped with the Windows 2000 operating system and is called the command station. The other computer, the target PC, solves the dynamic model in real-time. The target PC is working under the QNX real-time operating system [12]. Because the timing of I/O-functions is important, the I/O-interface is connected to the target computer. Simulation is controlled through a command station and the

trainer of the crane operator can control the simulator and modify the simulation parameters during the training using command station. The command station also does the necessary computation for visualization and surrounding sound. The communication between computers is established using Ethernet. Some examples of different views inside the virtual harbour are presented in Figs 13(c) and (d). The view from the cockpit mounted on the motion platform can be seen in Figs 13(a) and (b).

The physical simulator environment is in a ventilated room. The room is painted with non-reflective colours so that the projected view can be seen more clearly and the outside world fades from sight. The view is projected obliquely forward to the motion platform, as shown in Fig. 14. The projection is done using a mirror attached to the floor and the operator sees the view on a transparent screen. This enables the positioning of the projector so that there will be no shadows caused by the structures of the motion platform.

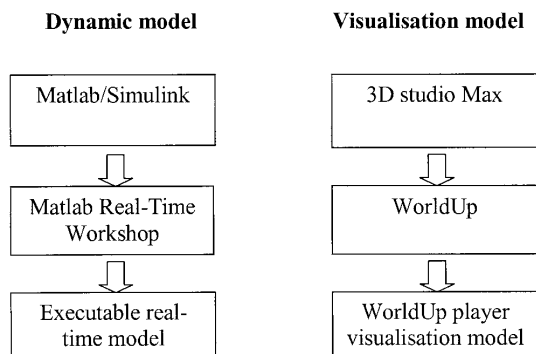


Fig. 11 Programs used during development

5 CONCLUSIONS

Real-time simulators have several advantages compared with traditional operator training. The machine capacity is not tied to training and can be used in productive work. The use of simulators enables training for situations that can cause severe damage to the operator or environment. These are the most important situations for which the operator should be trained. However, they cannot be included in traditional training procedures. Simulators also guarantee similar training conditions for all trainees and

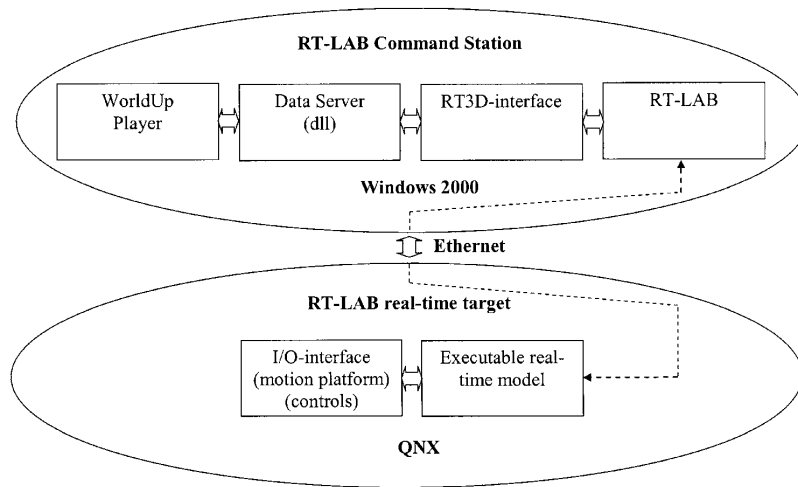


Fig. 12 Gantry crane simulator computer configuration

enable a great variety of environmental conditions to be taken into account.

Real-time training simulators consist of several separate components and require technical

knowledge of several fields. Besides computer skills, the simulator development group must have good skills in machine dynamics, hydraulics, control theory, visualization, machine automation, and so on.

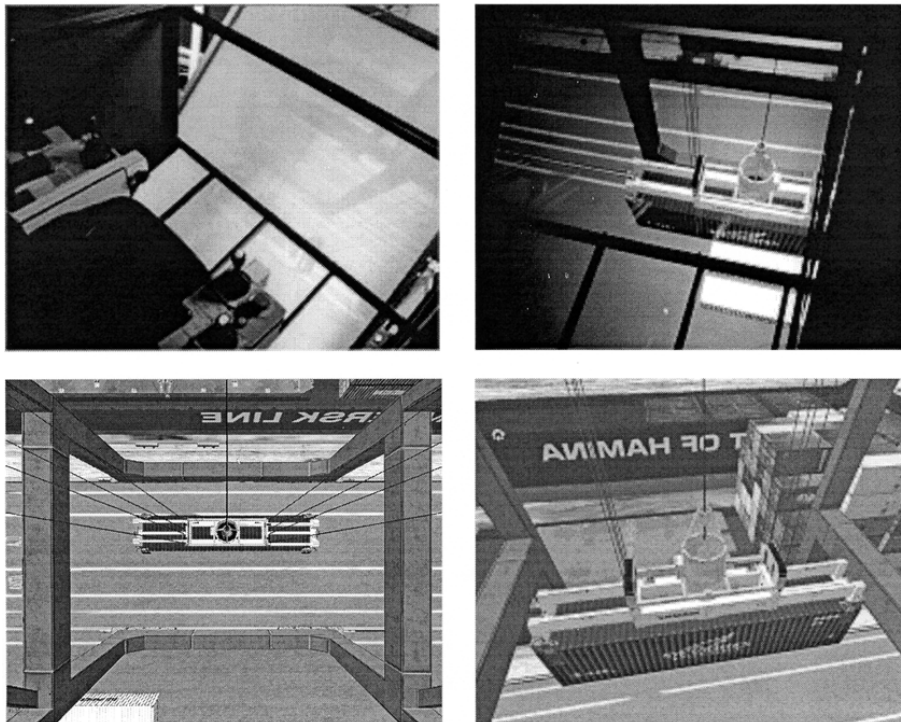


Fig. 13 (a and b) View from simulator cockpit and (c and d) some details of virtual harbour

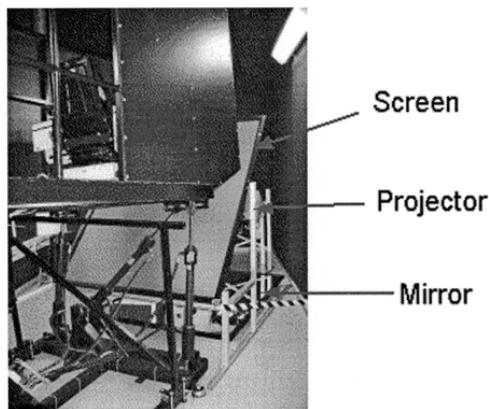


Fig. 14 Positioning of visualization equipment

One main problem in simulator development is the interfaces between these fields. For example, as simple a thing as coordinate systems or units can cause serious problems if they are not specified in an early enough phase of the project. The exact definitions require that the positions of the coordinate systems, the positive rotation directions as well as the units used, and the architecture of data transferred from one entity to another are specified. Interfaces should also be defined in a way that enables their expansion, for example, if parts or DOFs are increased, which affects the amount of motion data.

The correct user feedback from the motion platform depends greatly on the tuning of the controller parameters. This must be done using test operators that have a strong background on operating the machine in question. It is possible to tune the controllers on a way that makes the motions feel very unrealistic and the accelerations easily become very much rapid. To ensure the correct acceleration rates, it is necessary to build accurate inverse kinematic models of the motion platform. These models must also include the solution of inverse accelerations or velocities to be used in the closed control loop of the motion platform.

In more complicated simulators that have more DOFs, a more general and computationally efficient modelling method is required. Rotational DOFs can be easily omitted if bodies only move translationally relative to each other. If rotational movement is required, all six DOFs must be considered. Using recursive modelling methods that take system topology into account, more efficient and general modelling procedures, especially for open chain structures, can be achieved [13–15].

As a result of this project, a functional real-time gantry crane simulator for crane operator training was built. Time will tell how much it will help in

operator education. Reports from other simulator training sites inform that operators trained with simulators can work at nearly normal loading speed after training, whereas operators trained with conventional methods may need several years before they are able to work at normal loading speed.

ACKNOWLEDGEMENT

This simulator has been developed in co-operation with the Institute of Mechatronics and Virtual Engineering at Lappeenranta University of Technology, Kymenlaakso Polytechnic, and The Centre of Expertise in Southeast Finland.

REFERENCES

- 1 Haug, E. J. Computer aided kinematics and dynamics of mechanical systems. In *Basic methods*, 1989, Vol. I: (Allyn and Bacon, Massachusetts).
- 2 Nikravesh, P. E. *Computer-aided analysis of mechanical systems*, 1988 (Prentice Hall, New Jersey).
- 3 Shabana, A. A. *Computational dynamics*, 1994 (John Wiley & Sons, Inc., New York).
- 4 Shabana, A. A. *Dynamics of multibody systems*, 1998, 2nd edition (Cambridge University Press, New York).
- 5 Baraff, D. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *Comput. Graph.*, 1989, **23**, 223–232.
- 6 Moore, M. and Wilhelms, J. Collision detection and response for computer animation. *Comput. Graph. (Proc. SIGGRAPH)*, 1988, **22**, 289–298.
- 7 Kraus, P. R. and Kumar, V. Compliant contact models for rigid body collisions. In *Proceedings of IEEE International Conference on Robotics and automation*, Albuquerque, 1997, pp. 1382–1387.
- 8 Bourg, D. M. *Physics for game developers*, 2002 (O'Reilly & Associates, Inc., Sebastopol).
- 9 Baumgarte, J. Stabilization of constraints and integrals of motion in dynamical systems. *Comput. Methods Appl. Mech. Eng.*, 1972, **1**, 1–16.
- 10 Yoon, S., Howe, R. M., and Greenwood, D. T. Stability and accuracy analysis of Baumgarte's constraint violation stabilization method. *ASME J. Mech. Des.*, 1995, **117**, 446–453.
- 11 Lin, S.-T. and Hong, M.-C. Stabilization method for numerical integration of multibody mechanical systems. *ASME J. Mech. Des.*, 1998, **120**, 565–572.
- 12 *RT-LAB 6.0 user's manual*, 2001 (Opal-RT Technologies Inc.).
- 13 Jerkovsky, W. The structure of multibody dynamics equations. *J. Guid. Contr.*, 1978, **1**(3), 173–182.
- 14 Kim, S. S. and Vanderploeg, M. J. A general and efficient method for dynamic analysis of mechanical systems using velocity transformations. *ASME J. Mech. Trans. Automat. Des.*, 1986, **108**, 176–182.

15 **García de Jalón, J.** and **Bayo, E.** *Kinematic and dynamic simulation of multibody systems – the real-time challenge*, 1994 (Springer-Verlag, New York).

APPENDIX

Notation

A_i	transformation matrix of body i	\mathbf{n}	normal vector of a plane
c	damping coefficient	\mathbf{P}_P	position of point P
c_a	damping coefficient of guide arm a motor	\mathbf{R}	vector of translational coordinates
c_m	damping coefficient of motor m	\mathbf{q}	vector of generalized coordinates
C	constraint equation	\mathbf{Q}_c	vector of second differentials of constraints
\mathbf{C}_q	constraint Jacobian matrix	\mathbf{Q}_e	vector of generalized forces
d	closest distance between point and plane	\mathbf{Q}_v	vector of quadratic velocity inertia terms
\mathbf{D}	distance between two points	\mathbf{t}	tangent vector
F_{Tm}	transfer force of a motor	\mathbf{t}_{norm}	normalized tangent vector
F_X	force in global X -direction	T_a	driving torque of guide arm a
F_Y	force in global Y -direction	T_{ca}	contact torque of guide arm a
F_Z	force in global Z -direction	T_m	driving torque of motor m
\mathbf{F}_{arms}	guide arm force vector affecting to spreader	$T_{\text{ref}m}$	reference torque of motor m
\mathbf{F}_{cont}	total contact force	$T_{\text{total}a}$	total driving torque of guide arm a
\mathbf{F}_{ie}	external force acting in body i	\mathbf{u}	unit vector
\mathbf{F}_{iR}	translational generalized force acting in body i	\mathbf{u}_{iP}	location of point P in global coordinate system of body i
$\mathbf{F}_{i\varphi}$	rotational generalized force acting in body i	\mathbf{v}	unit vector
$\mathbf{F}_{i\theta}$	rotational generalized force related to Euler parameters acting in body i	\mathbf{v}_P	velocity of point P
\mathbf{F}_n	contact force in normal direction	\mathbf{v}_{rel}	relative velocity of two points
\mathbf{F}_μ	friction force	\mathbf{v}_{rn}	velocity in the direction of collision plane
\mathbf{G}_i	rotational velocity mapping matrix of body i	X_c	container X -coordinate
$i_{\text{tot}m}$	total reduction ratio of motor m	X_t	trolley X -coordinate
J_a	inertia of guide arm a	Y_t	trolley Y -coordinate
k	spring constant	Y_c	container Y -coordinate
k_{ca}	spring constant of guide arm a contact	Z_c	container Z -coordinate
K_a	amplification coefficient of guide arm a motor	Z_g	gantry Z -coordinate
K_m	amplification coefficient of motor m	α_a	contact angle of guide arm a
l_{ca}	contact length of guide arm a	$\boldsymbol{\lambda}$	vector of Lagrange multipliers
m	number of motors	η_m	coefficient of driveline m
\mathbf{M}	mass matrix	$\boldsymbol{\theta}$	vector of Euler parameters
		θ_{0c}	container Euler parameter
		θ_{1c}	container Euler parameter
		θ_{2c}	container Euler parameter
		θ_{3c}	container Euler parameter
		θ_a	angle of guide arm a
		$\theta_{\text{ref}a}$	reference angle of guide arm a
		τ_m	time constant of motor m
		ω_m	angular velocity of motor m
		ω_i	angular velocity vector of body

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

304. TANNINEN, KATI. Diffusion of administrative innovation: TQM implementation and effectiveness in a global organization. 2008. Diss.
305. PUISTO, ANTTI. The initial oxidation of transition metal surfaces. 2008. Diss.
306. FELLMAN, ANNA. The effects of some variables on CO₂ laser-MAG hybrid welding. 2008. Diss.
307. KALLIOINEN, MARI. Regenerated cellulose ultrafiltration membranes in the treatment of pulp and paper mill process waters. 2008. Diss.
308. PELTOLA, SATU. Capability matrix – identifying and evaluating the key capabilities of purchasing and supply management. 2008. Diss.
309. HONKAPURO, SAMULI. Performance benchmarking and incentive regulation – considerations of directing signals for electricity distribution companies. 2008. Diss.
310. KORHONEN, KIRSI. Facilitating coordination improvement efforts in cross-functional process development programs. 2008. Diss.
311. RITVANEN, VIRPI. Purchasing and supply management capabilities in Finnish medium-sized enterprises. 2008. Diss.
312. PYNNÖNEN, MIKKO. Customer driven business model – connecting customer value to firm resources in ICT value networks. 2008. Diss.
313. AL NAZER, RAMI. Flexible multibody simulation approach in the dynamic analysis of bone strains during physical activity. 2008. Diss.
314. The Proceedings of the 7th MiNEMA Workshop. Middleware for Network Eccentric and Mobile Applications. Ed. by Pekka Jäppinen, Jouni Ikonen and Jari Porras. 2008.
315. VÄÄTÄNEN, JUHA. Russian enterprise restructuring – the effect of privatisation and market liberalisation on the performance of large enterprises. 2008. Diss.
316. DABAGHMESHIN, MAHSA. Modeling the transport phenomena within the arterial wall: porous media approach. 2008. Diss.
317. HAIMALA, JUHA. Supplier's position in project marketing networks. 2008. Diss.
318. UOTILA, TUOMO. The use of future-oriented knowledge in regional innovation processes: research on knowledge generation, transfer and conversion. 2008. Diss.
319. LAPPALAINEN, TOMMI. Validation of plant dynamic model by online and laboratory measurements – a tool to predict online COD loads out of production of mechanical printing papers. 2008. Diss.
320. KOSONEN, ANTTI. Power line communication in motor cables of variable-speed electric drives – analysis and implementation. 2008. Diss.
321. HANNUKAINEN, PETRI. Non-linear journal bearing model for analysis of superharmonic vibrations of rotor systems. 2008. Diss.
322. SAASTAMOINEN, KALLE. Many valued algebraic structures as measures of comparison. 2008. Diss.
323. PEUHU, LEENA. Liiketoimintastrategisten vaatimusten syntyminen ja niiden toteutumisen arviointi keskisuurten yritysten toiminnanohjausjärjestelmähankkeissa: Tapaustutkimus kolmesta teollisuusyrityksestä ja aineistolähtöinen teoria. 2008. Diss.

324. BANZUZI, KUKKA. Trigger and data link system for CMS resistive plate chambers at the LHC accelerator. 2008. Diss.
325. HIETANEN, HERKKO. The pursuit of efficient copyright licensing – How some rights reserved attempts to solve the problems of all rights reserved. 2008. Diss.
326. SINTONEN, SANNA. Older consumers adopting information and communication technology: Evaluating opportunities for health care applications. 2008. Diss.
327. KUPARINEN, TONI. Reconstruction and analysis of surface variation using photometric stereo. 2008. Diss.
328. SEPPÄNEN, RISTO. Trust in inter-organizational relationships. 2008. Diss.
329. VISKARI, KIRSI. Drivers and barriers of collaboration in the value chain of paperboard-packed consumer goods. 2008. Diss.
330. KOLEHMAINEN, EERO. Process intensification: From optimised flow patterns to microprocess technology. 2008. Diss.
331. KUOSA, MARKKU. Modeling reaction kinetics and mass transfer in ozonation in water solutions. 2008. Diss.
332. KYRKI, ANNA. Offshore sourcing in software development: Case studies of Finnish-Russian cooperation. 2008. Diss.
333. JAFARI, AREZOU. CFD simulation of complex phenomena containing suspensions and flow through porous media. 2008. Diss.
334. KOIVUNIEMI, JOUNI. Managing the front end of innovation in a networked company environment – Combining strategy, processes and systems of innovation. 2008. Diss.
335. KOSONEN, MIIA. Knowledge sharing in virtual communities. 2008. Diss.
336. NIEMI, PETRI. Improving the effectiveness of supply chain development work – an expert role perspective. 2008. Diss.
337. LEPISTÖ-JOHANSSON, PIIA. Making sense of women managers' identities through the constructions of managerial career and gender. 2009. Diss.
338. HYRKÄS, ELINA. Osaamisen johtaminen Suomen kunnissa. 2009. Diss.
339. LAIHANEN, ANNA-LEENA. Ajopuusta asiantuntijaksi – luottamushenkilöarvioinnin merkitys kunnan johtamisessa ja päätöksenteossa. 2009. Diss.
340. KUKKURAINEN, PAAVO. Fuzzy subgroups, algebraic and topological points of view and complex analysis. 2009. Diss.
341. SÄRKIMÄKI, VILLE. Radio frequency measurement method for detecting bearing currents in induction motors. 2009. Diss.
342. SARANEN, JUHA. Enhancing the efficiency of freight transport by using simulation. 2009. Diss.
343. SALEEM, KASHIF. Essays on pricing of risk and international linkage of Russian stock market. 2009. Diss.
344. HUANG, JIEHUA. Managerial careers in the IT industry: Women in China and in Finland. 2009. Diss.
345. LAMPELA, HANNELE. Inter-organizational learning within and by innovation networks. 2009. Diss.

