Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 1 (2012) 55 – 64

Procedia Technology

INSODE-2011

Automatic generation of equations of motion for multibody system in discrete event simulation framework

Namkug Ku [a] *, Ara Jo [a], Sol Ha [a], Myung-Il Rho [c], Kyu-Yeul Lee [c]

[a]*Graduate Student, Seoul National University, Shinlim-Dong Kwanak-Gu, Seoul, Korea*
[b]*Professor, University of Ulsan, Daehak-ro, Nam-gu, Ulsan, Korea*
[c]*Professor, Seoul National University, Shinlim-Dong Kwanak-Gu, Seoul, Korea*

Abstract

In this paper, the development of a simulation program that can automatically generate equations of motion for mutibody systems in the discrete event simulation framework is presented. The need to analyze the dynamic response of mechanical systems that are under event triggered conditions is increasing. General mechanical systems can be defined as multibody systems that are collections of interconnected rigid bodies, consistent with various types of joints that limit the relative motion of pairs of bodies. For complex multibody systems, a systematic approach is required to efficiently set up the mathematical models. Therefore, a dynamics kernel was developed to automatically generate the equations of motion for multibody systems based on multibody dynamics. The developed dynamics kernel also provides the numerical solver for the dynamic analysis of multibody systems. The general multibody dynamics kernel cannot deal with discontinuous state variables, event triggered conditions, and state triggered conditions, though. To enable it to deal with multibody systems in discontinuous environments, the multibody dynamics kernel was integrated into a discrete event simulation framework, which was developed based on the discrete event system specification (DEVS) formalism. DEVS formalism is a modular and hierarchical formalism for modeling and analyzing systems under event triggered conditions, which are described by discontinuous state variables. To verify the developed program, it was applied to an block-lifting and transport simulation, and dynamic analysis of the system is carried out.

Keywords: Multibody dynamics, DEVS;

---

1. Introduction

Requests for accurate dynamic response analysis using a simulation tool have been increasing in many engineering fields, including ship building industry. Especially in the ship building industry, it is very important to predict the delivery day of the ship to its owner. Therefore, for process planning designers in ship yards, accurate dynamic response analysis is becoming more important during the shipbuilding process. For this reason, the ship yards designers use commercial programs when they receive requests for dynamic response analysis. However, these methods have some limitations. The commercial programs for dynamic analysis are usually developed for general purposes, so that they may not be suitable for various requirements of process planning in ship yards.

For instance, a dead weight 300,000-ton VLCC (very large crude carrier), which can carry 300,000 tons of crude

---

* Namkug Ku

doi:10.1016/j.protcy.2012.02.012

oil, can be delivered to a ship owner after a total design and production period of about 14 months. To build the VLCC, the ship is divided into about 200 blocks, as shown in Fig 1-(a), and the blocks are erected in a dock. For the erection of the blocks, block-lifting and transport design is required, and is a part of the process planning. Block-lifting and transport are performed using various types of cranes, and the cranes are defined as multibody systems, which are collections of interconnected rigid bodies, consistent with various types of joints that limit the relative motion of pairs of bodies. Fig 1-(b) shows a goliath crane, which can also be regarded as a multibody system.

Therefore, the process planning designers need to analyze the dynamic response of multibody systems. The block-lifting and transport procedure is composed, however, of several discontinuous stages, such as hoist-up, hoist-down, and turn-over. Meanwhile most of commercial programs for multibody dynamic analysis cannot deal with discontinuous state variables, event triggered conditions, and state triggered conditions.

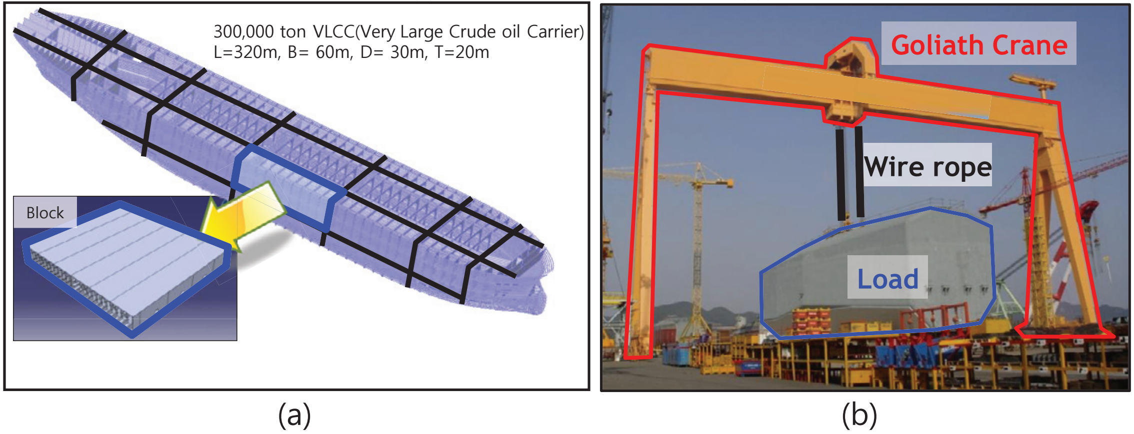


Fig. 1. (a) 300,000 ton VLCC(Very Large Crude oil Carrier) divided into about 200 blocks; (b) Load lifting using a goliath crane in shipyards.

Therefore, dynamics kernel was developed, which can automatically generate the equations of motion of multibody systems for accurate analysis of dynamic systems. To deal with multibody system in a discontinuous environment, the multibody dynamics kernel is integrated into the discrete event simulation framework, which is developed based on the discrete event system specification (DEVS) formalism. DEVS formalism is a modular and hierarchical formalism for modeling and analyzing systems under event triggered conditions, which are described by discontinuous state variables.

2. Related Works

ADAMS (Automatic Dynamic Analysis of Mechanical Systems) is a software system that consists of a number of integrated programs that help an engineer in performing three-dimensional kinematic and dynamic analysis of mechanical systems (Orlandea et al. 1977, Schiehlen 1990). ADAMS generates equations of motion for multibody systems using augmented formulation. The user can define any multibody system composed of several bodies that are interconnected by joints. ADAMS supplies various types of joints, such as fixed, revolute, and spherical joints. Various external forces can also be applied to multibody systems, but ADAMS cannot handle discontinuous state variables, event triggered conditions, and state triggered conditions.

ODE (Open Dynamics Engine) is an open-source, library for simulating multibody dynamics (Smith 2006). Similar to ADAMS, ODE derives equations of motion for multibody systems using augmented formulation. ODE can treat only rigid bodies, though, not flexible bodies. Moreover, it cannot handle discontinuous state variables, event triggered conditions, and state triggered conditions.

RecurDyn is the three-dimensional simulation software that combines dynamic response analysis and finite element analysis tools for multibody systems. It is from 2 to 20 times faster than other dynamic solutions because of its advanced fully recursive formulation. Various joints and external forces can also be applied to the multibody systems, but RecurDyn cannot handle discontinuous state variables, event triggered conditions, and state triggered conditions.

On the other hand, Praehofer, Zeigler et al. (1990, 2000) proposed a modeling and simulation method that can handle simulation models of discrete event and discrete time. They also developed a simulation framework based on the proposed method. In the case of discrete event simulation, the operation of a simulation system is represented as a chronological sequence of events. Process or material flow simulation systems and the like are included in the category of discrete event simulation. On the other hand, in the case of discrete time simulation, the operation of a simulation system is represented as the progress of time. State changes only occur at discrete time instants. Dynamic simulation systems and the like are included in the category of discrete time simulation. However, the developed simulation frame work focuses only on the material flow simulation system of a workshop. Thus, it was difficult for it to be applied to a large factory such as a ship yard, and it was hard to use existing design and production information for the simulation.

Many researches related to mutibody dynamic analysis and discrete event simulation have been conducted, but they had some limitations in their application to process planning in ship yards, as mentioned. To overcome these limitations, a dynamics kernel was developed, which can automatically generate the equations of motion of multibody systems, and is integrated into the discrete event simulation framework. The remainder of this paper is as follows. Section 3 describes the developed dynamics kernel. Section 4 presents the integration of the dynamics kernel and DEVS framework. Section 5 presents a sample block-lifting and transport application. Finally, Section 6 summarize this study.

3. Dynamics Kernel for Automatic Generation of Equations of Motion using Graph Method

In this section, the dynamics kernel for the automatic generation of equations of motion is presented. Let's consider the four-link arm in Fig 2-(a) as a simple example of multibody system.

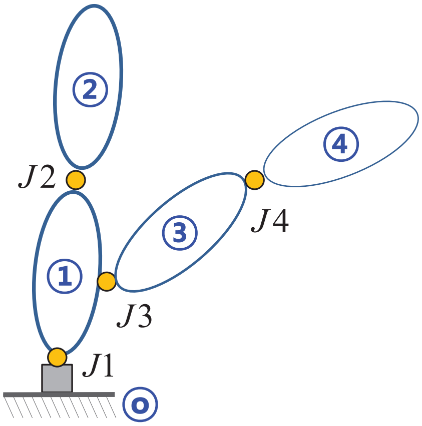


Fig. 2. Four-link arm as a simple example of multibody system

In Fig 2, ⓞ represents the base, which does not move, and ①-④ represent the four-links, which is considered as rigid bodies. J1-J4 represent the joints, which limit the relative motion of pairs of bodies. For dynamic response analysis, four equations of motion for each body are required, since the four-link arm consists of four rigid bodies. If the four-links are moving in the two-dimensional plane, the equations of motion for this multibody system are as follows:

$$m_1\ddot{\mathbf{r}}_{O_1/E} = \mathbf{F}_{O_1} + \mathbf{F}_{constraint\_1} - \mathbf{F}_{constraint\_2} - \mathbf{F}_{constraint\_3} \quad -(1)$$

$$m_2\ddot{\mathbf{r}}_{O_2/E} = \mathbf{F}_{O_2} + \mathbf{F}_{constraint\_2} \quad -(2)$$

$$m_3\ddot{\mathbf{r}}_{O_3/E} = \mathbf{F}_{O_3} + \mathbf{F}_{constraint\_3} - \mathbf{F}_{constraint\_4} \quad -(3)$$

$$m_4\ddot{\mathbf{r}}_{O_4/E} = \mathbf{F}_{O_4} + \mathbf{F}_{constraint\_4} \quad -(4)$$

, where $m_1 \sim m_4$ are the masses, $\mathbf{r}_{O1/E} \sim \mathbf{r}_{O4/E}$ are the position vectors of the center of mass, $\mathbf{F}_{O1} \sim \mathbf{F}_{O4}$ are the external forces, and $\mathbf{F}_{constraint\_i}$ is the constraint force exerted on each body from body i. In the forward dynamics problem, the mass and the external forces are given, and the second derivatives of the position vectors are found. To solve the forward dynamics problem, the constraint forces $\mathbf{F}_{constraint\_i}$ should be calculated or suppressed from the equations of motion. There are several methods of doing this, such as embedding formulation, augmented formulation, and recursive formulation (Featherstone 2008, Haug 1992, Shabana 2005). The recursive formulation was used to solve the equations of motion in this paper. Before the equations of motion were solved, though, they should be generated. Generating them is not so difficult, and does not take a long time for the four-link arm. When the number of the bodies increases, though, and the structure of the multibody system becomes complicated, the generation of the equations of motion will become more difficult. Therefore, the graph theory is used to automatically generate equations of motion.

The structure of the four-link arm can be represented using a graph as shown in Fig. 3.

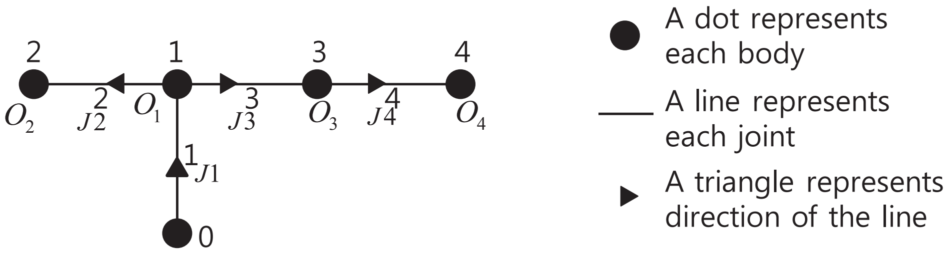


Fig. 3. Representation of the multibody system using a graph

The dots, lines, and triangles represent the bodies, joints, and directions of the line, respectively. From the graph, the path matrix T, eq. (5), can be uniquely determined (Wittenberg, 2006).

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \text{ - (5)}$$

This path matrix T has all the information on the structure of the multibody system. Therefore, the equations of motion of the multibody system can be automatically generated using the path matrix. For example, Eq. (1) and Eq. (4) can be generated with the following sequence as shown in Fig. 4 and Fig. 5.

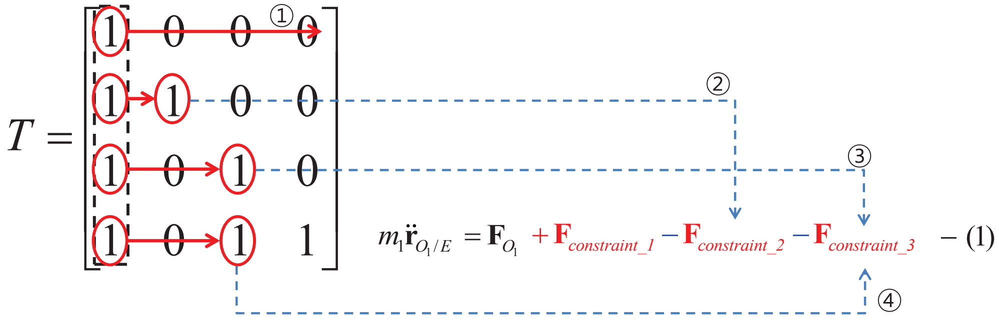


Fig. 4. Example of the generation of the equation of motion for the link 1 using the graph theory

1) Generation of Eq. (1)

a. The left hand side of Eq. (1) is the production of the mass and the second derivative of the position vector.

b. The first term on the right hand side of Eq. (1) is the external force exerted on the link 1, which should always be included.

c. The second term on the right hand side of Eq. (1) is the constraint force from the joint 1, which should always be included.

d. To determine the rest of the constraint forces of Eq. (1), the first column is selected.

e. On the right side of the (1, 1) element that is 1, there is no the element that is 1, so no constraint force must be included (Fig. 4-①).

f. On the right side of the (2, 1) element that is 1, there is an element that is 1 in the second column so the constraint force from the joint 2 should be included on the right side of Eq. (1)(Fig. 4-②).

g. On the right side of the (3, 1) element that is 1, there is an element that is 1 in the third column so the constraint force from the joint 3 should be included on the right side of Eq. (1) (Fig. 4-③).

h. On the right side of the (4, 1) element that is 1, there is an element that is 1 in the third column so the constraint force from the joint 3 should be included on the right side of Eq. (1) (Fig. 4-④).

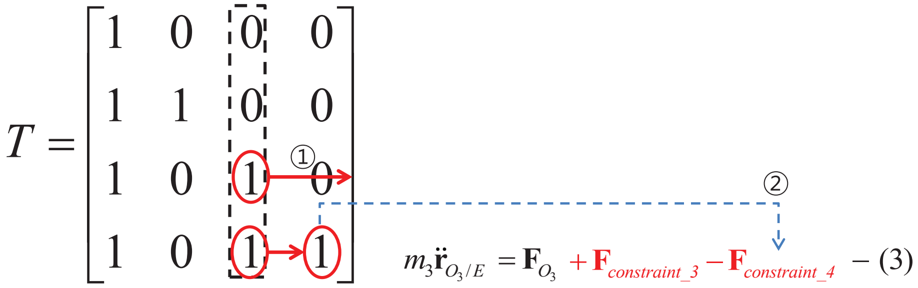


Fig. 5. An example of generation of the equation of motion for the link 3 using graph theory

2) Generation of Eq. (3)

a. The left hand side of Eq. (3) is the production of the mass and the second derivative of the position vector.

b. The first term on the right hand side of Eq. (3) is the external force exerted on the link 3, which should always be included.

c. The second term on the right hand side of Eq. (3) is the constraint force from the joint 3, which should always be included.

d. To determine the rest of the constraint forces of Eq. (3), the third column is selected.

e. On the right side of the (3, 3) element that is 1, there is no element that is 1, so no constraint force must be included (Fig. 5-①).

f. On the right side of the (4, 3) element that is 1, there is an element that is 1 in the fourth column so the constraint force from the joint 4 should be included on the right side of Eq. (3)(Fig. 5-②).

The actual equations of motion that were formulated using the recursive formulation for each body are shown in Fig. 6.

$$\mathbf{v}_i = \mathbf{v}_{i-1} + \mathbf{S}_i \dot{q}_i \quad \cdots\cdots \text{(a)}$$

$$\mathbf{a}_i = \mathbf{a}_{i-1} + \mathbf{S}_i \ddot{q}_i + \dot{\mathbf{S}}_i \dot{q}_i \quad \cdots\cdots \text{(b)}$$

$$\mathbf{f}_i^B = \mathbf{I}_i \mathbf{a}_i + \mathbf{v}_i \times^* \mathbf{I}_i \mathbf{v}_i \quad \cdots\cdots \text{(c)}$$

$$\mathbf{f}_i^c = \mathbf{f}_i^B - \mathbf{f}_i^e - \sum \mathbf{f}_j^c \quad \cdots\cdots \text{(d)}$$

$$\boldsymbol{\tau}_i = \mathbf{S}_i^T \mathbf{f}_i^c \quad \cdots\cdots \text{(e)}$$

$\mathbf{v}_i$: Velocity vector of body i (6 components)
$\mathbf{a}_i$: Acceleration vector of body i (6 components)
q_i : Generalized coordinate (joint values)
$\mathbf{S}_i$: Velocity transformation matrix
$\mathbf{I}_i$: Mass and mass moment of inertia of body i
$\mathbf{f}_i^B$: Resultant force exerted on body i
$\mathbf{f}_i^e$: External force exerted on body i
$\mathbf{f}_i^c$: Force exerted on the joint i which is on body i
$\boldsymbol{\tau}_i$: Force generated by joint i

Fig. 6. Equations of motion formulated using recursive formulation for each body

Each Eq. (1), (2), (3), and (4) corresponds to Eq. (c) and (d). Substituting Eq. (d) into Eq. (c) gives

$$\mathbf{I}_i \mathbf{a}_i + \mathbf{v}_i \times^* \mathbf{I}_i \mathbf{v}_i = \mathbf{f}_i^e + \mathbf{f}_i^c - \sum \mathbf{f}_j^c \text{ - (6)}$$

In this Eq. (6), the last term on the left hand side is determined using the graph theory. Eq. (a) and (b) can also be automatically generated for each body using the graph theory. The dynamics kernel has been developed to automatically generate the equations of motion for multibody system based on multibody dynamics.

4. Integration of the Dynamics Kernel and the Discrete Event Simulation Framework

In the previous section, the development of the dynamics kernel was presented. However, it is hard to deal with the discontinuous state variables, event triggered conditions, and state triggered conditions with the dynamics kernel, which is for the multibody dynamic analysis. To overcome this limitation, this study adopts the DEVS (Discrete Event System Specification) formalism to development framework.

4.1. DEVS (Discrete Event System Specification) formalism

The DEVS formalism, a set-theoretic formalism, specifies discrete event systems in a hierarchical and modular form. The DEVS formalism consists of two kinds of models: an atomic model and a coupled model. The atomic model is the basic model and has specifications for the dynamics of the model. Formally, 7 components, which are state variables, input events, output events, external transition function, internal transition function, output function, and time advance function, specify the atomic model. The coupled model provides the method of assembly of several atomic and/or coupled models to build complex systems hierarchy. Each DEVS model, either atomic or coupled, has correspondence to an object in the real-world system to be modeled (Zeigler 1990, Zeigler et al. 2000).

Using the DEVS formalism, the multibody systems can be defined as a coupled model as shown in Fig. 7. The multibody systems are composed of joint coupled models and body coupled models. The joint coupled models consist of a joint atomic model and an actor, and the body coupled models consist of a body atomic model and the actor. The joint atomic model has the information of joint according to a type of the joint, the body atomic model has the properties of the body, and the actor has a function, which generates the force acts on the bodies or the joints. A commander is an atomic model that has an action list and connected to the every actor. Therefore, designer can draw up the action list, so that they can handle discontinuous events.

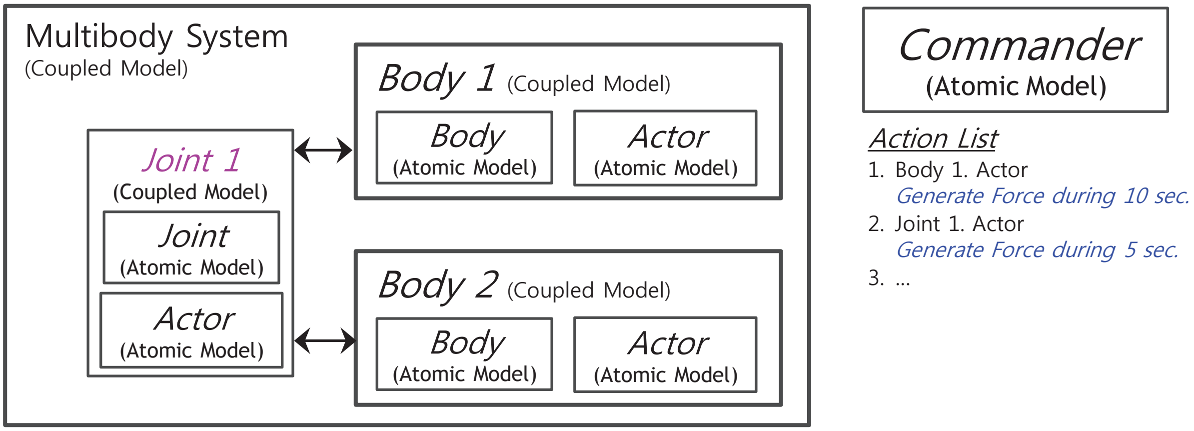


Fig. 7. Multibody system represented using DEVS formalism

4.2. Integration of DEVS framework and dynamics kernel

For the explanation, consider the four-link arm in Fig 2-(a) as a simple example of multibody system. The four-link arm is represented as a coupled model as following:

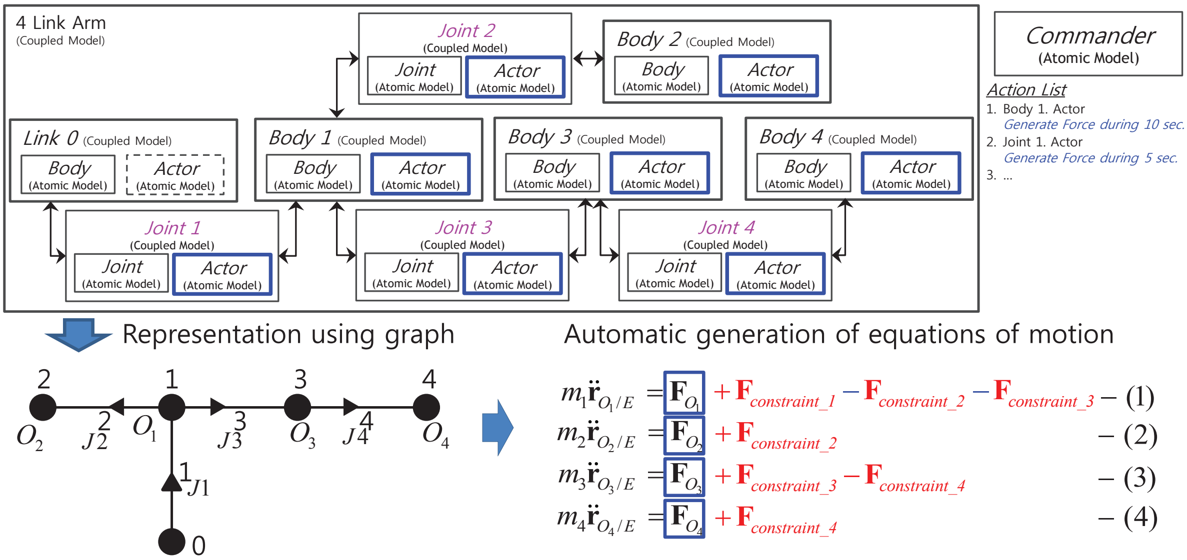
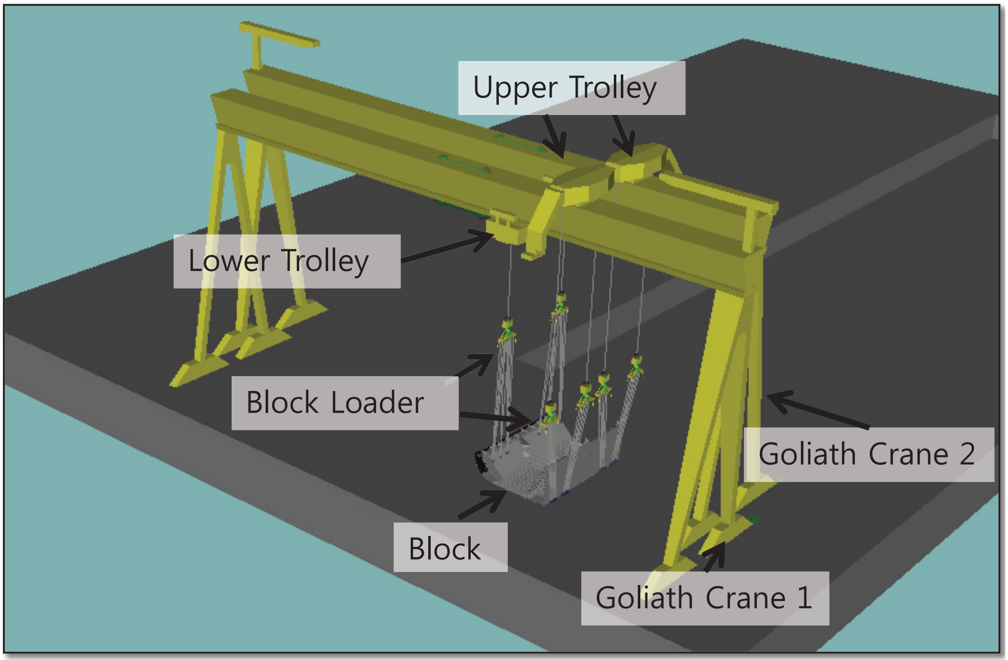


Fig. 8. Four-link arm represented using DEVS framework, graph, and equations of motion

After representing the four-link arm using DEVS framework, it can be easily expressed as a graph form. Using the graph, the equations of motion can be generated. The external forces of the equations of motion are interconnected with the actors, so that the external force can be exerted by input the action list. Therefore the designer can carry out the simulation of the block-lifting and transport, which is composed of several discontinuous stages, by drawing up the action list.

5. Application to Simulation of Block- Lifting and Transport

This section presents an example of block-lifting and transport and the result of the simulation.



- **Main dimension of the goliath crane**

Length: 177m, Height: 103.6m, Breadth: 51.434m

- **Max Lifting Capacity of the goliath crane: 600 tons**
- **Weight of Block: 828.5ton**

Fig. 9. The goliath cranes and block model in the simulation of the block-lifting and transport

The block-lifting and transport is carried out using two goliath cranes, six block loaders, and one block. The goliath crane is composed of a main body, upper trolley, and lower trolley. The upper trolley and lower trolley is interconnected by sliding joints with main body. The block loader consists of two bodies, interconnected by revolute joint each other. To execute the simulation of the block-lifting and transport, simulation models and dynamics models have to be made. The simulation models are shown in Fig. 10. This figure shows the DEVS models for one goliath crane, three block loaders, and the block.

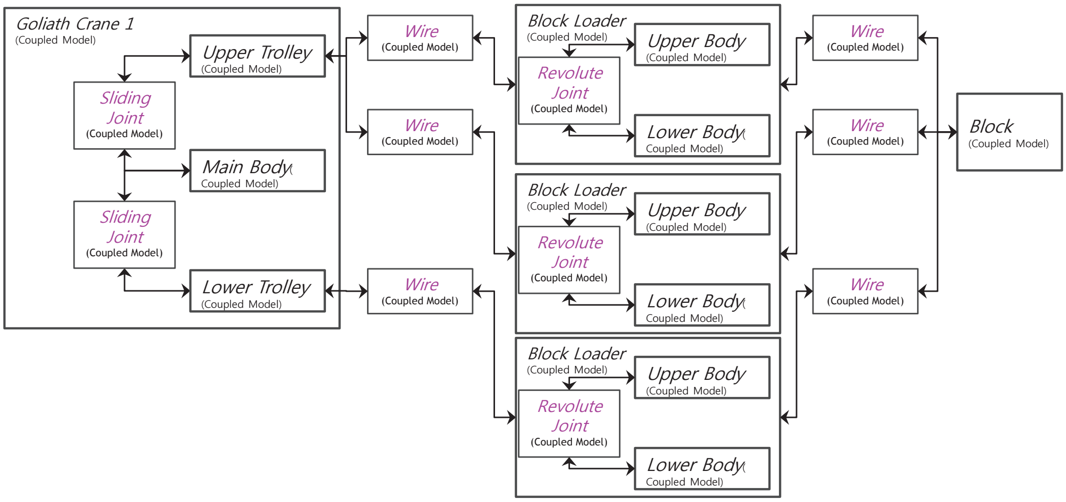


Fig. 10. The simulation models for the block-lifting and transport using two goliath cranes

As explained in section 4.2, the equations of motion, i.e. the dynamics model, are automatically generated from the DEVS models, so that the dynamic response analysis can be carried out. Discrete events of the simulation are as following;

a. Block lifting
b. Block transportation by moving the goliath crane to the dock
c. Block turn-over: the process of turning the block upside down.

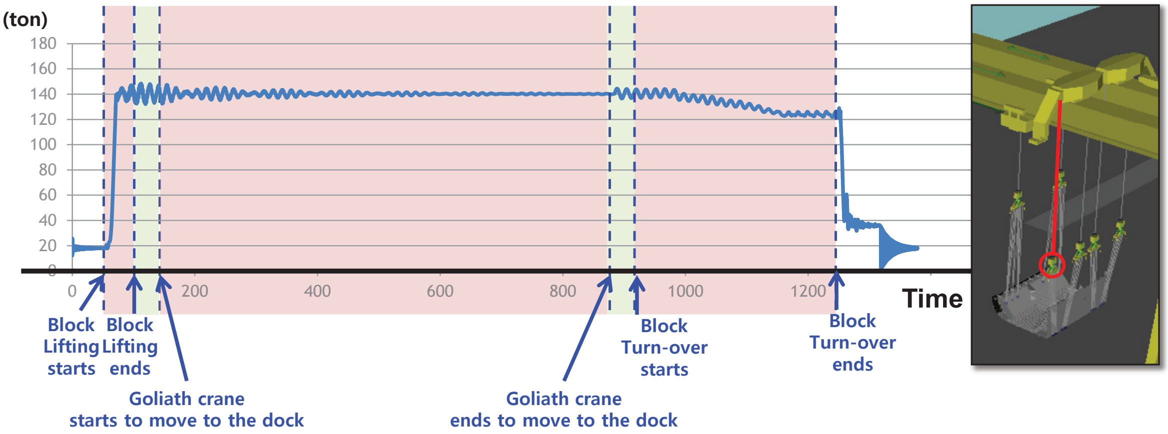


Fig. 11. Tension of block loader 1 calculated by the block-lifting and transport simulation using developed program.

Fig. 11 shows the simulation results. The graph shows that the equations of motion are automatically generated by the developed program, and the developed program can deal with the discrete events.

6. Conclusions and Future Works

A simulation framework was proposed and implemented in this study. The dynamics kernel is integrated into the DEVS framework for various simulation systems for the process planning in shipbuilding. To evaluate the efficiency of the implemented simulation program, it is applied to the simulation of the block-lifting and transport.

As future works, we will apply the developed program to various simulation systems for the process planning in shipbuilding such as a simulation the dynamic analysis of the offshore structure and the block assembly process in order to improve the efficiency and applicability.

7. Copyright

All authors must sign the Transfer of Copyright agreement before the article can be published. This transfer agreement enables Elsevier to protect the copyrighted material for the authors, but does not relinquish the authors' proprietary rights. The copyright transfer covers the exclusive rights to reproduce and distribute the article, including reprints, photographic reproductions, microfilm or any other reproductions of similar nature and translations. Authors are responsible for obtaining from the copyright holder permission to reproduce any figures for which copyright exists.

Acknowledgements

This work was supported by:

a) Industrial Strategic Technology Development Program (10035331, Simulation-based Manufacturing Technology for Ships and Offshore Plants) funded by the Ministry of Knowledge Economy (MKE, Republic of Korea);

b) Research Institute of Marine System Engineering at Seoul National University;

c) Marine Technology Education and Research Center, through the Brain Korea 21 project of Seoul National

University; and

d) SM-11: “A Study on the network-based architecture of virtual system for the simulation of underwater vehicles” of the Underwater Vehicle Research Center.

References

1. Kreuzer, E. (1994), Computerized Symbolic Manipulation in Mechanics, Springer, pp 18-35.
2. Orlandea, N., Chace, M.A., and Calahan, D.A. (1977). “A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems-Part1&2,” Journal of Engineering for Industry, Transactions of the ASME, Vol. 99, No. 3, pp 773-779.
3. Schiehlen, W. (1990), Multibody Systems Handbook, Springer, pp 361-402.
4. Smith, R. (2006). Open Dynamics Engine v0.5 User Guide, pp 15-20.
5. Zeigler BP, Object oriented simulation with modular, hierarchical models: intelligent agents and endomorphic systems, Boston: Academic Press, 1990.
6. Zeigler BP, Praehofer H, Kim TG, Theory of modelling and simulation, 2nd ed., Boston: Academic Press, 2000.
7. Featherstone, R. (2008), Rigid Body Dynamics, Springer, pp 92-100.
8. Dewitt, D., & Siraj, S. (2010). Learners perceptions of technology for design of a collaborative mLearning module. *World Journal on Educational Technology, 2*(3), 169-185.
9. Haug, E.J. (1992), Intermediate Dynamics, Prentice-hall, pp 345-346.
10. Shabana, A. A., 2005, Dynamics of multibody systems, Third edition, Cambridge University Press.
10. Wittenberg, 2006, Dynamics of Multibody Systems, Springer.