Modelling and simulation of marine rudder system in a unified M&S platform

Chang Chen*, Guojin Chen, Shaohui Su, Haiqiang Liu

School of Mechanical Engineering, Hangzhou Dianzi University, 310018, Hangzhou, China

Received 18 June 2014, www.tsi.lv

Abstract

For modelling and simulating of marine rudder system, there are lots of along with their model libraries, such as AMEsim could be used. But the models in these tools lack of flexibility and are not open to the end-user. And these tools could not model the whole marine rudder system consisted of mechanical, hydraulic and control sub-system in a unified form. In order to solve those problems, a flexible and extensible marine rudder system library was constructed, based on the Modelica, by the object-oriented strategy. It supports the reuse of knowledge on different granularities: physical phenomenon, component model and system model. A conventional model of marine rudder system was built and calculated using the library, and the results shows that the object-oriented modelling strategy is effective; the framework of the library is reasonable.

Keywords: Marine Rudder System, Modelica, M&S Unified Platform, Object-oriented modelling strategy

1 Introduction

Marine rudder system, which is the actuator of operation control system, is important part of ship. Its basic task is accurately turning the rudder according to the given rudder angle. Electronic hydraulic position valve system colligating the advantages of electric and hydraulic is the typical marine rudder system. It has advantages of high control accuracy, fast response speed, flexible signal processing, etc. Up to present, lots of modelling platform could be used to design marine rudder system and analyse its performance in single domain. Performance analysis of steering control system could be realized with Matlab/Simulink. The AMEsim could be used to design and analyse the hydraulic sub-system of marine rudder system [1]. However, the whole marine rudder system is difficult to model and simulate because of the shortcoming as follows.

(1) The model is commonly described as a black box which users just know how to use but don't know the details of. So it is very difficult to modify an existing model and introduce a new one in these simulators. In another word, the models lack of flexibility and are not open to the end-user.

(2) And the worse is the models developed in one tool cannot be easily used in another one because of the different model description, translation, data organization etc. among these simulators.

(3) The model is usually expressed in an explicit state-space form and iterative process of model solution has to be given out if the explicit expression could not be found. Consequently, the topology of the system gets lost and any future extension and reuse of the mode is also tedious and error-prone.

Fortunately, among the last dozen years' research in modelling and simulation, there are two concepts that closely related to these problems: object oriented modelling language and non-causal modelling. Modelica [2-4], developed in an international effort, is such a kind of physical system modelling language. It supports encapsulation, composition and inheritance facilitating model development and update. Elementary models of physical elements are defined in the declarative expression by their constitutive physical principles, and their interface with the outer world is described by physical connectors without any implied causality, rather than by writing assignments relating inputs to outputs. This makes the description of physical systems much more flexible and natural than it is possible with causal or block-oriented modelling languages, or by directly writing simulation code using procedural languages such as C or FORTRAN. Complex models can then be built by connecting elementary models through their ports. The ports are non-causal, so any connection which is physically meaningful is allowed without restrictions. The Modelica is applied to the modelling and simulation of vehicle [5-6], craft [7], machine [8], fluid system [9-10], control system [11] and so on.

The Modelica language includes graphical annotations, which allow to use graphical user interfaces (such as the one provided by the tool MWorks [12]) to select components from a library, drag them into a diagram, connect them, and set their parameters, thus making the process of model development highly intuitive for end users. In order to minimize the

* ***Corresponding author*** e-mail: chenchang@hdu.edu.cn

development time, Modelica allows the definition and reuse of intermediate elements, common to more models of the same component. The key feature to extensive model reusability and flexibility is given by the extensive exploitation of two Modelica constructs: extend and redeclare, and their associated keywords: partial and replaceable. The partial/extends binomial permits the extension of partial (i.e., partial-complete) models into complete, fully detailed models.

In this article, a Marine Rudder System library was built based on the novel multidisciplinary physical modelling language Modelica and the Modelica-supported tool MWorks, aiming at providing a framework of Marine Rudder System model library with the object oriented technology and equation-based model description.

2 Structuring idea of library

2.1 OBJECT-ORIENTED MODELLING STRATEGY

Traditionally, a physical system model development has followed the top-down design approach, which applies the method of functional decomposition to establish the model structure. This method has been successfully applied in many physical engineering domains, but it fails to reflect the real world. As a result many attempts have been made to tackle this problem by applying object-oriented technology.

So the library of Marine Rudder System is designed as an object-oriented work, which is a set of classes that embodies an abstract design for solution to a family of related problems. The set of classes define "partial-complete" application that captures the common characteristic of object structures and functionality. Specific functionality in new applications is realized by inheriting from, or composing with, framework components. In this paper, new object-oriented modelling language Modelica and a Modelica-supported tool MWorks are used to develop the model library which can be later used to assemble system-level Marine Rudder System model.

During the simulation, the system's mathematical model is mapped to collections of interacting objects, rather than decomposed into segments of different functions that can implement certain algorithms. Each object mimics the behavioural and structural characteristic of a physical or conceptual entity models. And it represents an instance of a software class, while the classes are united into a hierarchy via inheritance relationships.

According to the modelling idea mentioned above, the Marine Rudder System is decomposed into the units according to the physical reality and further the process units are decomposed into the basic physical phenomenon, such as conservation of mass or of energy, as the left branch shows in Fig. 1. Afterwards, models of basic physical phenomenon, unit and system are built level by level using language Modelica and tool MWorks, as the right branch shows in Fig.1. As a result, the different levels' models: from basic physical phenomenon models, over physical unit modes, up to system models could be reused flexibly in the process of constructing a new model. So the professional engineering knowledge can be solidified, propagated and reused in varied granularity.

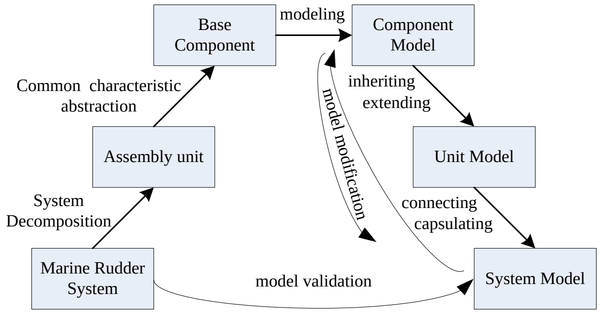


FIGURE 1 System decomposing and modelling with Modelica and Mworks

2.2 MODEL LIBRARY FOR MARINE RUDDER SYSTEM

Following the level progressive modelling strategy mentioned above and considering the maximum reuse of codes in the library, the Marine Rudder System could be divided into several sub-systems. Each sub-system consists of various components that were created through inheriting and expanding the basic physical phenomenon models. Fig.2 presents the architecture of the library. The library consisted of fourteen parts.

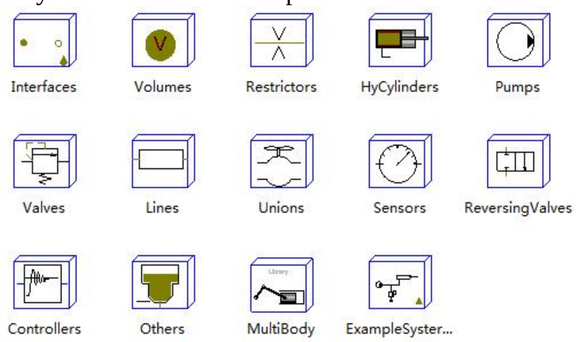


FIGURE 2 Structure of model library for marine rudder system

Interfaces: the connectors among the components are defined here.

Volumes: package of tank and boundary in hydraulic system.

Restrictors: package of restrictors.

HyCylinders: package of hydraulic cylinders.

Pumps: package of pumps, such as centrifugal pump, constant pressure pump and so on.

Valves: package of valves, like check valve, dropping valve, etc.

Lines: the pipes with lumped parameter and distributed parameter method.

Unions: connection like bend, buffer valve.

Sensors: sensors like pressure sensor, flow sensor.

ReversingValves: multi-channel directional control valve consist of models in package Valves.

Controllers: controller model like PID, PI.

Others: filter.

MultiBody: models of mechanical part,nsuch as rudder blade.

ExampleSystems: demos of marine rudder system.

3 Common Models

3.1 INTERFACE

A special-purpose class connector as an interface defines the variables of the model shared with other models, without implied causality, rather than by writing assignments relating inputs to outputs [13]. In this way the connections can be, besides inheritance concepts, thought of as one of the key features of object oriented modelling, enabling effective model reuse. There are two types of built-in variables: potential variable and flow variable with no prefix, the prefixes flow respectively.

The potential and flow variables follow the Generalized Kirchhoff's law. For examples, the connector Port for hydraulic oil flow in Marine Rudder System has two variables as follows: pressure p, volume flow rate q.

```
connector Port
  SI.AbsolutePressure p;
  flow SI.VolumeFlowRate q;// "positive is flowing from outside
into the component "
end Port;
```

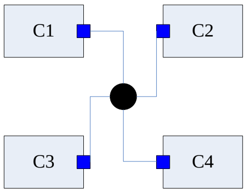


FIGURE 3 Connecting of four components

Figure 3 shows the connection of four components that has the connector Port. The connection point is treated as an infinitely small control volume. All the pressures in each connector must be equalized while the sum of all the molar flows should be zeros:

$$C_1.p = C_2.p = C_3.p = C_4.p \,, \tag{1}$$

$$C_1.q + C_2.q + C_3.q + C_4.q = 0 \,. \tag{2}$$

During the model translation, Eq. (1) and Eq. (2) originating from the connector definitions, are automatically generated and added to the other equations of the model.

3.2 PARTIAL MODEL

Partial model is a kind of semi-complete component which abstracts the common character shared by a group of models that have some common properties and behaviours or the same structure. It is a basic section of the component-model sub-library. For example, the pipe, restrictor and pump that could be considered as models include two Ports (one for inlet and another for outlet) as presented in section 3.1. They have the common properties and behaviours as follow:
mass conservation

$$q_{in} + q_{out} = 0 \,, \tag{3}$$

where, q_{in} means the mole flow rate of inlet; q_{out} the mole flow rate of outlet.

The pressure drop equation

$$p_{in} - p_{out} = \Delta p \,, \tag{4}$$

where, p_{in} means pressure of inlet; p_{out} pressure of outlet; Δp the pressure loss between two ports.

So a partial model *OneInOneOut* was built to describe the properties and physical behaviours shared by these three models and other similar models as follow:

```
partial model OneInOneOut
  Port In"inlet";
  Port out"outlet";
  SI.Pressure ploss" pressure loss between inlet and outlet";
equation
  In.q+Out.q =0; //mass balance
  In.p-Out.p =ploss; // pressure drop equation
end OneInOneOut;
```

At the beginning of the codes mentioned above, the prefix 'partial' means model *OneInOneOut* is semi-complete. When a higher level model like pipe is wanted to be introduced, *OneInOneOut* will be inherited and expanded. So the embedded engineering knowledge could be reused on physical behavioural level but not only on model level. In the "*equation*" region, the behaviours described in Eqs. (3-4) are coded in non-causal expression. When the model is needed to be simulated, MWorks will give out the calculation sequence automatically.

```
model pipe
  extends OneInOneOut;// extend from model OneInOneOut
  parameter Real m,n;
equation
  ploss = m*q^n;
end pipe;
```

3.3 COMPONENT: PIPE AS AN EXAMPLE

In this section, a pipe model will be taken as an example to illustrate how to use a partial model to build a complete component model. Besides the information included in *OneInOneOut*, an equation describes relation between the pressure loss and volume should be add.

$$\Delta p = m \cdot q^n \,, \tag{5}$$

where, m and n are parameters obtained from empirical data. The follows codes show how to build a complete pipe model through inheritance and expansion.

3.4 ASSEMBLY UNIT

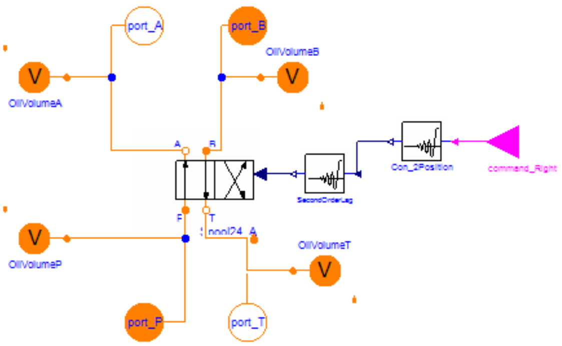
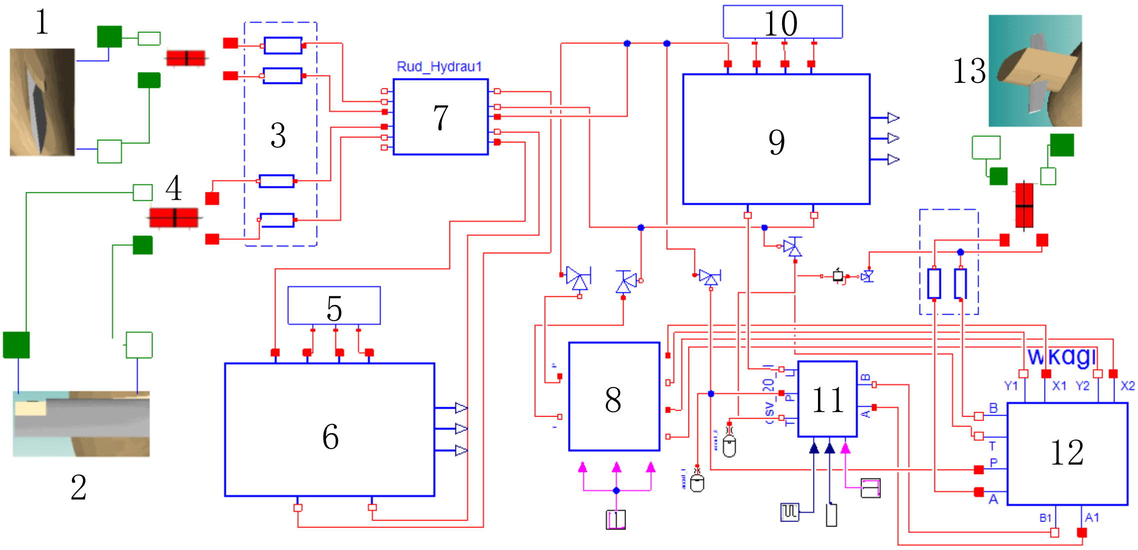


FIGURE 4 the diagram of two-position-four-channel reversing valve

Assembly unit means the assembly unit made up of component model mentioned in section 3.2. According to the practical topological relation, an assembly unit model is constructed by connecting several related component model. As shown in Figure 4, a two-position-four-channel electro-hydraulic reversing valve contains a valve body, a controller, one signal connector, four f ports, four control volumes and the connections between these components.

4 AN ILLUSTRATIVE EXAMPLE

In this section, a marine rudder system model of a typical submarine is constructed using the library graphically. As shown in Figure 5, the system model contains mechanical sub-system and electro hydraulic servo sub-system. The mechanical system is consisted of rudder, elevator and fairwater plane and the hydraulic system is consisted of pump station, energy accumulator, silencer, air system, steering gear hydraulic system, emergency manual valve group, electro-hydraulic servo valve and so on. Respectively, Figure 6 and Figure 7 show the pressure response of drainage port of DSV-20 and pressure on outlet port of pump station when the control input of DSV-20 is a pulse signal.



1-rudder, 2-elevator, 3- silencer, 4-actuator, 5-air system, 6-pump station, 7-steering gear hydraulic system, 8-emergency manual valve group, 9-pump station of ship, 10-air system, 11-electro-hydraulic servo valve (DSV-20), 12-insulated hydraulic valve group, 13- fairwater plane

FIGURE 5 Marine rudder system model of a typical submarine

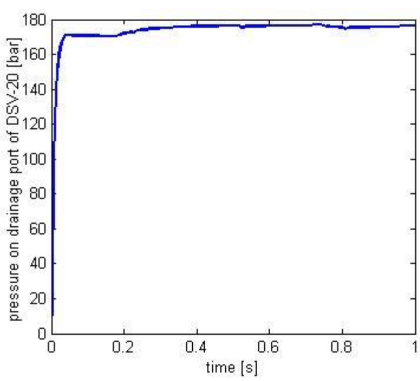


FIGURE 6 Pressure on drainage port of DSV-20

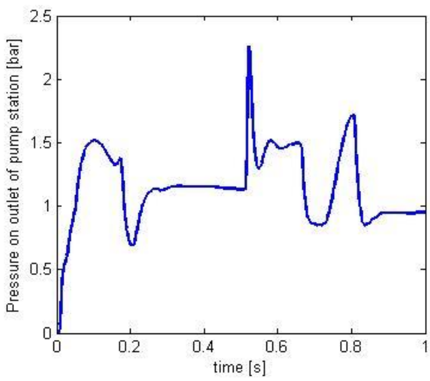


FIGURE **7** Pressure on outlet port of pump station

5 Conclusions

The whole marine rudder system is difficult to model and simulate in a unified platform because of the shortcoming as follows.

Model is described as a black box which is not open to the end-user.

Model in different simulators could not be shared with each other.

Only supporting reuse of knowledge on unit model level.

In this paper, we tried to utilize the multidisciplinary physical modelling language Modelica to build an object-oriented modular marine rudder system library in the Modelica-supported tool MWorks. The aim is to explore the modelling methods and implement of marine rudder system model based on the object-oriented technology and to provide a marine rudder system library. The library mainly includes restrictor models, hydraulic cylinder models, pump models, pipe models, valve models and son on. A model of marine rudder system of submarine shows that the object-oriented modelling strategy is effective; the framework of the library is reasonable.

Acknowledgements

The paper was supported by Science and Technology Plan Projects of Zhejiang Province, China (2010R50003), Zhejiang province education department scientific research projects (Y201327060) and the National Natural Science Foundation of China (51305113).

References

[1] Xiao D Z 2007 *Ship Science and Technology* **29**(S1) 142–5 *(In Chinese)*
[2] Elmqvist H, Mattsson S E 1997 *ESS'97 European Simulation Symposium* Passau, Germany 19-22
[3] Tiller M 2001 *Introduction to Physical Modeling with Modelica* Kluwer Academic Publishers: Boston
[4] Mattsson S E, Elmqvist H, Otter M 1998 *Control. Eng. Pract* **6**(4) 501–10
[5] Andreas D, Johannes G 2011 *Proceedings of the 8th International Modelica Conference* Linköping University Electronic Press 13-7
[6] Andreasson J 2011 *Proceedings of the 8th International Modelica Conference* Linköping University Electronic Press: Sweden 414-20
[7] Looye G 2008 *Proceedings of the 6th International Modelica Conference* Press: The Modelica Association 193-202
[8] Dressler I, Schiffer J, Robertsson A 2009 *Proceedings of the 7th International Modelica Conference* Linköping University Electronic Press: Sweden 261-9
[9] Viel A 2011 *Proceedings of the 8th International Modelica Conference* Linköping University Electronic Press: Sweden 256-65
[10] Casella F, Sielemanny M, Savoldelli L 2011 *Proceedings of the 8th International Modelica Conference* Linköping University Electronic Press: Sweden 86-96
[11] Baur M, Otter M 2009 *Proceedings of the 7th International Modelica Conference* Linköping University Electronic Press: Sweden 593-602
[12] Zhou F L, Chen L P, Wu Y Z, et al. 2006 *Proceedings of the 5th International Modelica Conference* Press: The Modelica Association 725-31
[13] Peter F 2004 *Principles of Object-Oriented Modeling and Simulation with Modelica2.1* Wiley IEEE Press: New York

Authors

Chang Chen, born on December 10, 1983, Taizhou City, Zhejiang Province, China

Current position, grades: Ph.D., lecturer of Department of Mechanical Engineering in Hangzhou Dianzi University
University studies: Huazhong University of Science and Technology
Scientific interest: Multi-domain unified modelling theory and technology
Experience: Huazhong University of Science and Technology, 2008/9-2013/1, Mechanical Design and Theory, PhD, research subjects engaged in Multi-domain unified modelling theory and technology, well known about modelling methodology and simulation strategy

Guojin Chen, born on May 2, 1961, Ningbo City, Zhejiang Province, China

Current position, grades: Ph.D., Professor of Department of Mechanical Engineering in Hangzhou Dianzi University
University studies: XiDian University
Scientific interest: Mechatronics theory and technology, Control theory and technology
Experience: XiDian University, 2004/9-2007/6, Mechanical Manufacturing and Automation PhD, research subjects engaged in the auto-focusing technology of digital image

Shaohui Su, born on September 7, 1978, Ruyang City, Henan Province, China

Current position, grades: Ph.D., Associate Professor of Mechanical Engineering in Hangzhou Dianzi University
University studies: Zhejiang University
Scientific interest: Product Digitalize Design, Manufacturing Information Engineering
Experience: Zhejiang University, 2002/9-2007/12, Mechanical Manufacturing and Automation PhD, research subjects engaged in theory and method of Product Data Management well known about PLM methodology and research integrated techniques of CAX/PDM, and focused on build the integrated product data model

Haiqiang Liu, born on April 19, 1980, JiangXi Province China

Current position, grades: Ph.D., lecturer of Department of Ocean Engineering in Hangzhou Dianzi University
University studies: Zhejiang University
Scientific interest: Intelligent design and digital product design
Experience: Zhejiang University 2005/9-2010/9 Mechanical Manufacturing and Automation PhD
research subjects engaged in theory and method of Product Data Management
well known about PLM methodology and research integrated techniques of CAX/PDM,
and focused on build the integrated product data model