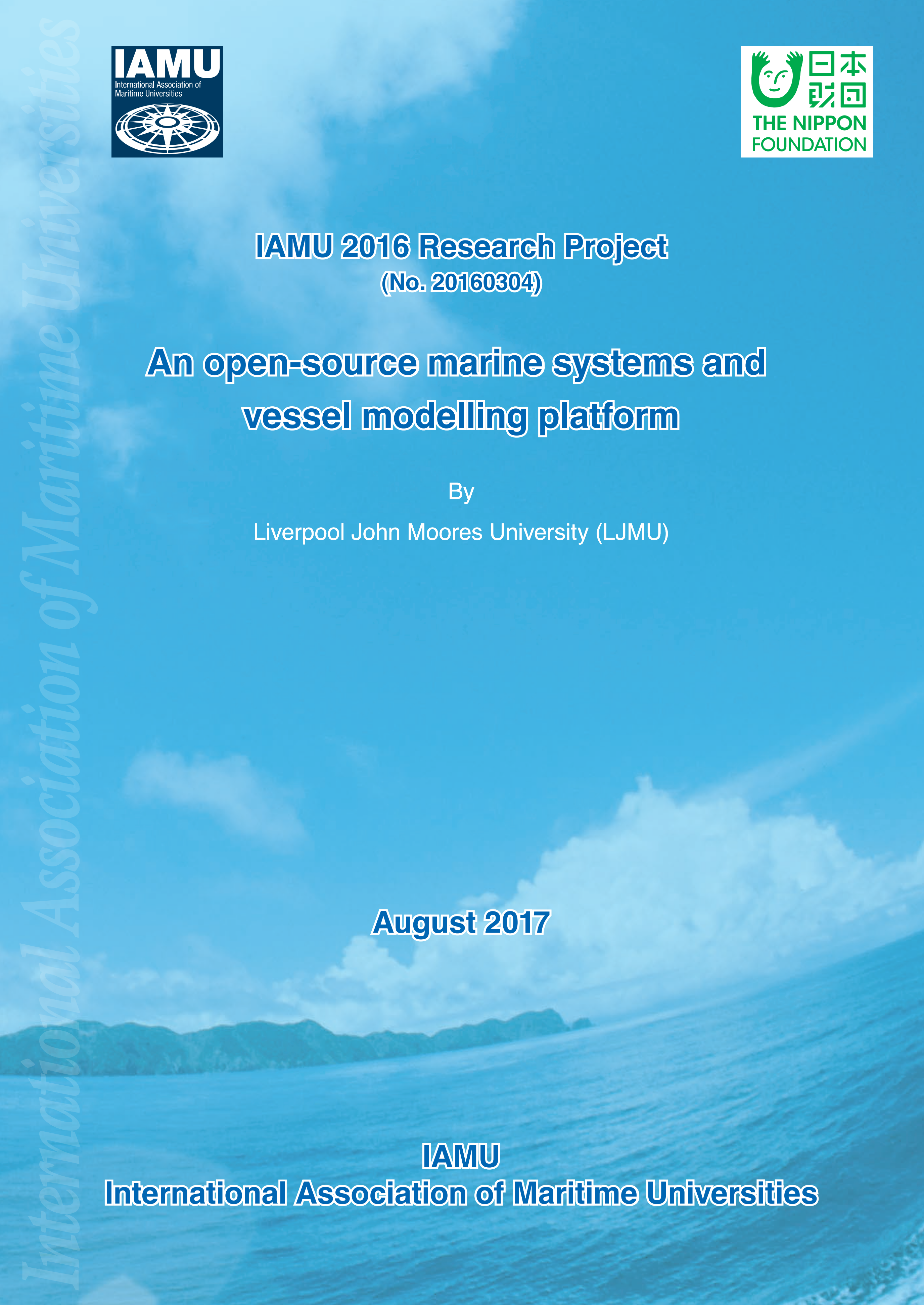


IAMU 2016 Research Project

(No. 20160304)

An open-source marine systems and vessel modelling platform

By

Liverpool John Moores University (LJMU)

August 2017

IAMU
International Association of Maritime Universities

This report is published as part of the 2016 Research Project in the 2016 Capacity Building Project of International Association of Maritime Universities, which is fully supported by The Nippon Foundation.

The text of the paper in this volume was set by the author. Only minor corrections to the text pertaining to style and/or formatting may have been carried out by the editors.

Published by the International Association of Maritime Universities (IAMU) Secretariat
Meiwa Building 8F, 1-15-10 Toranomon, Minato-ku,
Tokyo 105-0001, JAPAN
TEL : 81-3-6257-1812 E-mail : info@iamu-edu.org URL : http://www.iamu-edu.org

ISBN978-4-907408-19-0

IAMU
International Association of Maritime Universities

日本財団
THE NIPPON FOUNDATION

IAMU 2016 Research Project

(No. 20160304)

An open-source marine systems and vessel modelling platform

By

Liverpool John Moores University (LJMU)

Contractor : Ian Jenkinson, LJMU
Research Coordinator : Christian Matthews, LJMU
Research Partner : James Wharton, LJMU

Table of Contents

An open-source marine systems and vessel modelling platform

Theme: Emerging Technologies in the Maritime Sector

Liverpool John Moores University

Prof. Ian Jenkinson
Professor, Liverpool John Moores University, i.d.jenkinson@ljmu.ac.uk

Research Coordinator

Dr. Christian Matthews
Principal Lecturer, Liverpool John Moores University, c.matthews@ljmu.ac.uk

Research Student

Mr. James Wharton
PhD Researcher, Liverpool John Moores University

This paper presents the work which has been carried out as part of an IAMU research project which was carried out in the period from May 2016 to April 2017. It describes a newly developed tool for the modelling and simulation of complex marine systems and vessels. It outlines the common structure of the library, describes how it may be used create models of complex marine systems, and discusses how the resulting models can be used for design, optimisation and verification. It also describes how the models can be used to develop interactive real-time models for use in training applications.

Keyword: *Vessel modelling, Real-time simulation, Computational fluid dynamics (CFD), Hydrodynamic derivatives, Planar motion mechanism (PMM), Linear hydrodynamic coefficients,*

Executive Summary

This report details the development of a new software library for the modelling and simulation of marine vessels. The main aims of the project, which were all successfully achieved, was to develop a common library structure (with a hierarchical approach) so that a user with limited expertise can build a model that represents the many complex and connected systems that occur on a vessel. Another aim was to incorporate accurate descriptions of the manoeuvrability and resistances of a vessel to 3 degrees of freedom, whilst through additional python-based script tools, make it easier for a user to attain accurate inputs for the vessel modelling library. The final aim of the study was to incorporate a user interactive real-time simulation interface. A feature which is particular important from a training and educational perspective.

One of the problems when describing the hydrodynamic behaviour of a hull form, in order to understand its manoeuvrability and stability, is that there is no "one-size" fits all approximation. This means that expensive testing methods are usually required, which for many, is out of reach. So, the idea behind the introduction of the high-fidelity computational fluid dynamics (CFD) approach by automated script is to determine Abkowitz-type hydrodynamic derivatives without the need of tow tank testing. Abkowitz showed that these derivatives can describe the manoeuvrability of the ship effectively and from the systems modelling perspective, can easily be implemented within the hull resistance class of the library. This means that a vessel can be simulated in real-time, using the vesselEfficiency library, with the confidence that the force and moments acting upon the body are reasonably realistic.

Before the automation script had been written, a validation exercise was performed to at provide some certainty about the cogency of the modelling approach. Experimental data from the Duisberg test case (DTC) was compared against the steady tow motion simulated. This numerical modelling was performed using the volume of fluids (VOF) solver in the C++ based OpenFOAM CFD code. A range of speeds were simulated (Froude number of 0.174 to 0.218) and the maximum deviation of total drag was found to be 5.69% when compared to experimental data. The k-omega shear stress transport (SST) turbulence model with wall function was used and the results showed that a y^+ value, in the range of 30 to 300, was maintained for the majority of the hull surface, satisfying the criteria required for correct boundary layer modelling.

Once this validation exercise had been completed, the next stage was to create an automation script which performed the necessary motions of the planar motion mechanism (PMM): oblique steady drift, pure sway and pure yaw. Using this script in conjunction with OpenFOAM allows for the determination of the hydrodynamic derivatives but also provide validation of actual PMM testing performed in a tow tank facility. The derivatives produced by the script describe the vessel in three degrees of freedom (DOF). The next task is to develop the script so it can account for roll-associated resistance, increasing the description to 4 DOF.

Finally, a key feature of the library is the interactive real-time simulator. It has been demonstrated that the user can interactively control the vessel in real-time by modifying the inputs to the steering and propulsion classes. The Simulink Real-Time software executes the model and gives output relating to its global position. This information is then received by the Unity3D simulation interface over a user datagram protocol (UDP), where a 3D model of the vessel and virtual environment is rendered in real-time.

In relation to future developments, one of the immediate tasks is to continue the resistance modelling with benchmark data for validation purposes. Then the resistance classes shall be developed further in order to incorporate roll-related resistance. Once validated, confidence in the modelling process will allow for the generation of example hulls and their descriptions which can be included within the library for ease-of-use. Future work shall be to further develop the vessel modelling library by expanding on the number of subsystems available.

1. Introduction

Modelling and simulation are important techniques when analyzing the emergent behaviour of complex systems. In general, marine engineering systems and vessels are highly heterogeneous, being composed of mechanical, electrical, hydraulic and thermal subsystems which are governed by automatic control systems and human operators. As the complexity of maritime systems continues to grow, the effective definition, modelling and simulation of individual systems, and their interaction as part of a 'system-of-systems' is increasingly important. It is also desirable that interactive simulations which are used in crew training should accurately reflect the behaviour of vessels which are equipped with advanced systems. In the department of Maritime and Mechanical Engineering at Liverpool John Moores University, research, teaching and training take place across maritime and marine disciplines at all levels. It is therefore attractive to operate with a standardized software platform which allows models to be collaboratively developed, maintained and channeled.

Modelica is an open-source modelling language designed to describe and simulate multi-domain cyber-physical systems. It is an object-orientated language which enables systems to be described hierarchically in the same way that real systems are constructed, from standard component and sub-system models (e.g. motors, pumps, shafts, valves etc.). Models of standard components are typically packaged in model libraries, while systems and sub-systems are typically constructed using a graphical model editor. The documentation for the model is written in HTML and embedded as part of the library. This modelling approach has been selected for this work because it offers, in the view of the authors, significant long and short term benefits over causal block diagram methods. The principle benefit being that systems are described in terms of the connections between components not in the form of differential equations which must be decomposed by an expert user.

The intended spectrum of end-users for this modelling platform is broad, ranging from technical specialists who wish to model and simulate novel concepts, through to non-expert users who wish to explore vessel behaviour through interactive simulation. The educational application of the software would be in teaching marine engineering at undergraduate and postgraduate degree levels and for the development of real-time simulation systems for hardware-in-the-loop validation of control systems, and for interactive simulation.

The work carried out within the scope of this project was intended as a starting point for a modelling platform, based upon accessible software technology, which is technically capable of representing vessels, on-board systems and subsystems. It is does not present a complete or comprehensive new model of a vessel, nor does it seek to, instead it offers a basis for the further development of an open and freely available resource which it is hoped will be used to enhance education and research in the maritime and marine sector.

The structure of the report is as follows. Firstly, the structure of the modelling library is described in terms of a simple example. This demonstrates the hierarchical nature of the models which are created using this method. In the following section, work is presented on the description of hydrodynamic forces acting upon the hull in 3 degrees-of-freedom, and a newly developed automated numerical method for calculating resistance coefficients in an open-source CFD code (OpenFOAM). In the third section, a test system is presented, demonstrating how the models which are created can be used to run interactive real-time simulations.

2. Model Structure

The common modelling library structure is hierarchical. This enables non-expert users to configure and run a simulation by defining a set of five top-level components. These are currently:

- Bridge & Navigation
- Vessel
- Motion Constraints
- Sea-State
- Environment and Atmosphere

The top level of an example model is shown in Fig. 1. This illustrates the top level view of a typical example model. The vessel is subject to motion constraints which govern its degrees of freedom within the global (world) coordinate system. The environment is defined in terms of weather, atmospheric conditions, and the sea state. The bridge provides a standard collection of data which is made available to the real or simulated vessel operators.

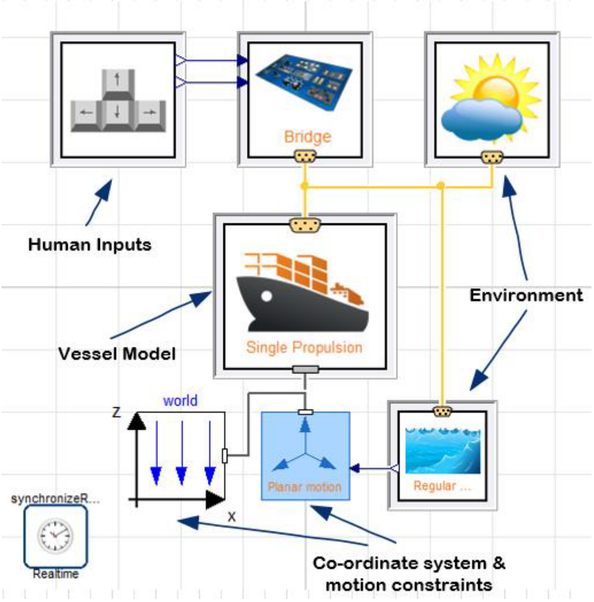


Fig. 1. Top-level view of an example, showing the main systems comprising a typical simulation model.

Each of these components in turn consists of a system of sub-components (a subsystem) which can either be selected from a pre-existing library, or can be defined from first principles if desired.

For example, a simple vessel component might consist of subsystems representing the hull, a single propulsion source, steering and a resistance model. However, because the library uses class inheritance, it is possible to quickly and easily create new templates which would enable new vessel variants to be created by (for example) adding or modifying propulsion components. The image in Fig. 2 shows the vessel model structure, one step down in the hierarchy.

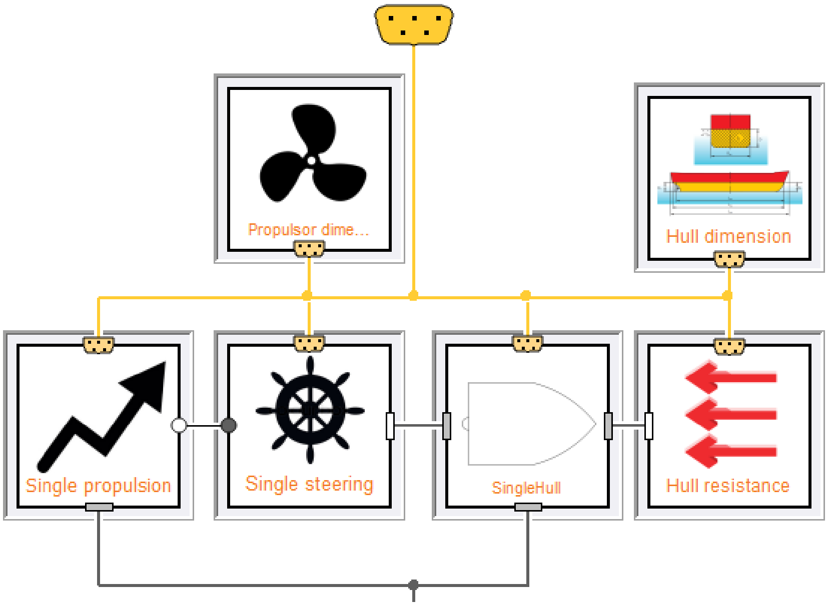


Fig. 2. Simple Vessel Model configuration

The development of this common library structure is one of the most important outcomes of this project. A significant benefit is that users can operate at the level of detail which is appropriate to them and their requirements.

2.1 Class Structures

In the example vessel model Fig. 2 the propulsion model consists of three subsystems. These are an Engine, the transmission and a driveline. In each case it is possible to replace the subsystem model with any compatible model. A compatible model is one which is designed with the same interface connections.

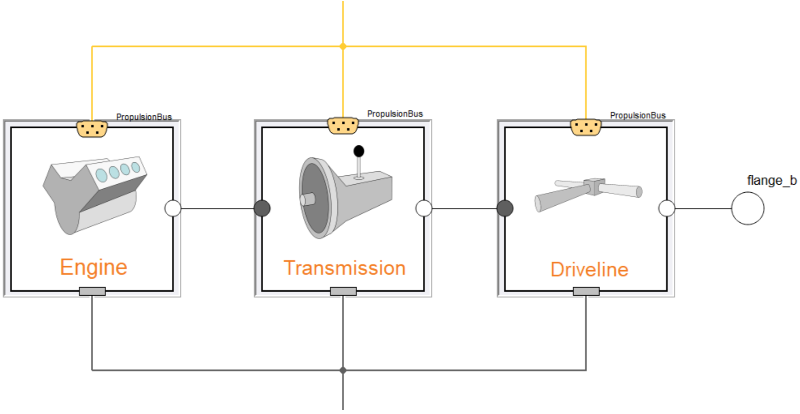


Figure 3 - Single propulsion model subsystem

The library is designed using model classes following an object-oriented design principle. In the library presented here, this is achieved by defining interface classes, templates and examples.

In the case of the engine, all classes derive from a base class (Figure 4) with three connections. These are:

- Rotational Mechanical Connection (White Circle)
- Multi-Body Connection (Grey Rectangle)
- Signal Bus Connection (Yellow Connector Shape)

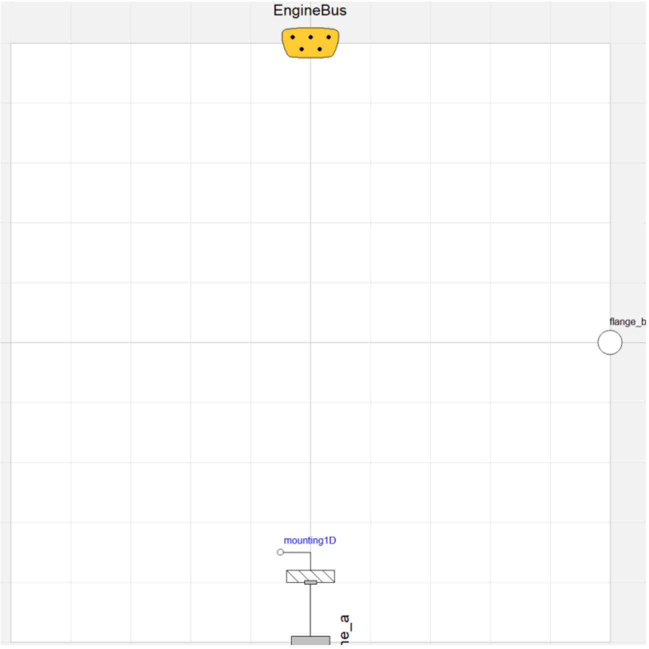


Figure 4 - Engine Base Class

For a simple range of engine models, these connections are sufficient to provide the mechanical connection between the engine and the transmission, reaction forces between the engine and the hull, and may send and receive data about desired and actual engine operation over the bus connection with the rest of the model.

In Figure 5 a simple engine model is shown. It inherits its connections from the base engine class. The simple model in this case uses a closed loop controller to regulate engine speed by modify fuelling. The engine's torque generation is modelled using a 2D lookup table. The engine inertia is lumped at the output and no friction model is included.

Throughout the model library, this class inheritance structure is used in order to minimise the need to replicate standard modelling elements and interfaces. It makes compatible subsystem models interchangeable and makes maintenance easier since modifications which are made to base classes are inherited by all higher-level classes which use them.

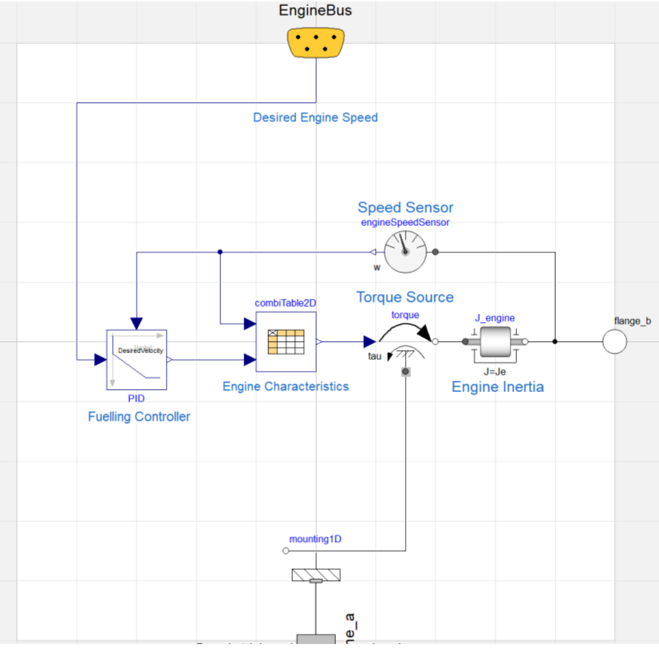


Figure 5 - Simple engine model, inheriting connectors from the engine base class

2.2 Environment

The environmental model incorporates both atmospheric conditions and the sea-state. The atmospheric model is currently a simple implementation which allows relevant background variables or parameters to be set. These include the air pressure, temperature & density, wind direction and wind velocity. These atmospheric factors do not interact directly with the seas state but are used to determine aerodynamic drag-force vector acting upon the hull. The air density is used globally by any subsystem which requires ambient air conditions, including those incorporating thermodynamic processes. While the current implementation is static, it would be easy to extend this class to include time or position dependent values for some or all of these variables.

2.3 Vessel

The typical configuration for a vessel is based around the mass and geometry of a hull-form to produce a rigid body representation, connecting lumped masses in 3D space. The steering and propulsion systems generate forces which are centred at propellers, thrusters, rudders or other control surfaces. The representation of the hydrodynamic forces which act upon the hull were identified early in the project as a priority area for development. In order to maintain computational efficiency, the approach presented by Abkowitz (1964) [8] has been adopted. The model class which applies these forces upon the hull is shown in Figure 6. The x and y velocity and acceleration components are taken from the ‘ResistanceBus’ which distributes states regarding hull motion to subsystems which determine resistance and hydrodynamic forces. The resulting forces and movements are imposed on the hull through the ‘frame_hydrodynamic’ connector.

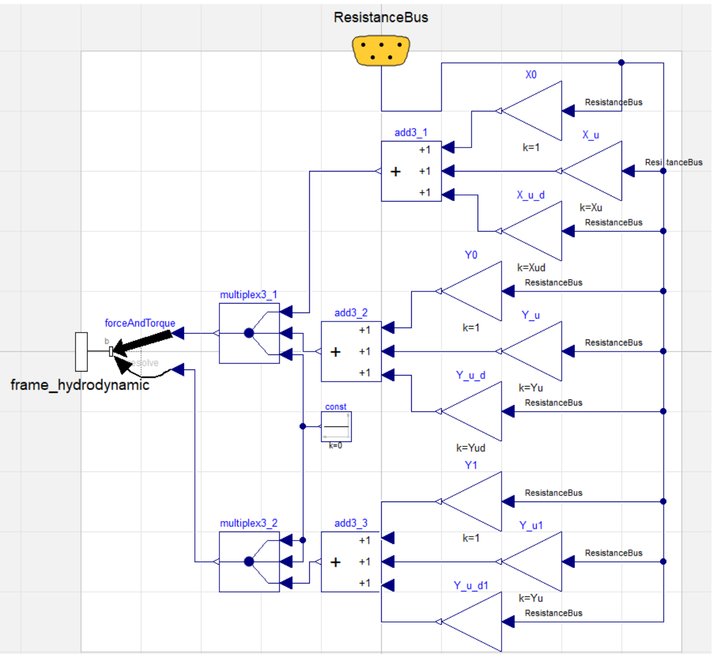


Figure 6 - Model class for imposing hydrodynamic forces on the hull model in three degrees of freedom using Abkowitz derivatives.

In an educational environment, this approach has the distinct benefit that students may parameterise the model using parameters that they have derived from steady tow experiments or, as will be seen, simulation. In the following section, a methodology for generating the Abkowitz parameters using an open-source CFD code, supported by an automated scripting process, is presented.

3. Numerical Simulation of Steady Tow

3.1 Introduction to Hydrodynamic Analysis

When designing a vessel, there are few questions more important than those related to the hydrodynamics of the ship but unfortunately, they are always the hardest to answer. There is not a "one-size" fits all approximation and usually, ship designers will often turn to experimentation using scaled models. This approach is time-consuming, expensive and depending on the facility, limitations on size could inflict on the exploration of steering capabilities/manoeuvrability. Better predictions of a ship's manoeuvrability is becoming increasingly important within hydrodynamics as modern vessels increase in size in order to meet global demands [1]. As this happens, concerns relating to the safe navigation of such sizeable vessels comes in to question and the recent publication of the International Maritime Organization (IMO) [2] standard, relating to ship manoeuvrability, is surely a reflection of this. An alternative to experimental investigation is numerical modelling using computational fluid dynamics (CFD) tools. The SIMMAN 2008 workshop was one of the first maritime events to showcase a range of publications relating to the numerical predication of hydrodynamic derivatives for a small collection of "benchmark" hull forms. Although these studies were relatively simple due to a limited number of forced motions being simulated, it has provided the basis for which many publications have explored with further detail [3-6]. In addition to this, the International Towing Tank Conference (ITTC) 2011 has since provided guidance on using CFD tools for 4 degrees-of-freedom (DOF) manoeuvring predictions [7]. Even though there are clear benefits of using CFD tools for this purpose, this method still has its downfalls. The first being the complexity of setting up such simulations and for that reason, experienced staff are required to "drive" the code. The second is the upfront costs associated with license fees. Commercial codes are extremely expensive and for many companies, this cost can outweigh the potential benefits.

For the *vesselEffiency* library to be effective in describing the maneuverability of a vessel, it needs to make use of Abkowitz (1964) [8] type hydrodynamic derivatives and the prediction of these coefficients shall be based upon the method seen in the study by Kim et. al. (2015) [4]. This study performs the planar motion mechanism (PMM) tests (steady drift, pure sway and pure yaw motions) on a bare hull within a CFD tool in order to determine three DOF based hydrodynamic derivatives. PMM tests were originally designed for facilities where large tanks were not available to perform rotating arm and zig-zag type maneuvers that are descibed by Abkowitz. PMM tests can be performed within standard towing tanks where the breadth of the channel is limited. Performing the PMM tests in CFD allows for comparison against derivatives obtained by experimentation. For the benefit of the community, it was decided to create a python script to automate the process of generating these derivatives using the open-source CFD tool, OpenFOAM. By doing this, some of the aforementioned downfalls are removed and it means that any user of the library can implement a description of their own particular hull form provided that a reasonable workstation is available for computation. Before the automation was implemented, a test case was run within the code in order to validate the modelling approach.

3.2 Geometry

To validate the modelling, the Duisberg test case (DTC) by Moctar, Shigunov and Zorn (2012) [9] shall be used (see Fig. 3 for hull shape and Tab. 1 for dimensions of model). The DTC study is a proposed "modern 14000 TEU post-panamax container carrier" hull form which can be used as benchmark study.

Tab. 1. Model and full scale dimensions [9].

Dimension	Model	Full Scale
L_{pp} (m)	5.976	355.0
B_{wl} (m)	0.859	51.0
T_m (m)	0.244	15.5
V (m^3)	0.827	173,467.0
C_B	0.661	0.661
S_w (m^2)	6.243	22,032.0
v_d (knots)	3.244	25.0
v_d – Fr. No.	0.218	0.218
Re No.	9.145×10^6	4.189×10^9

Fig. 3. Shape of DTC hull.

3.3 Model Setup

The volume of fluids (VOF), finite volume based InterFOAM solver was used from the OpenFOAM code. This is a solver which makes use of the pressure implicit with splitting of operator (PISO) method. The pressure solver also has a geometric algebraic multi-grid (GAMG) preconditioner applied. It uses a pseudo transient, first order, implicit Euler approach for time marching to a steady state solution. Second order, upwinding was used for the discretization of the convective terms all other terms use the second order linear (central differencing) method. The boundary condition (BC) applied to the hull was a non-slip BC (i.e. velocities at the surface are equal to zero). Symmetry BCs (velocity in the face normal direction is equal to zero) was applied to the front and back of the domain, where faces which have normal aligned with the y direction. The inlet of the domain has a uniform velocity applied and the outlet contained a zero-gradient pressure BC. Slip BCs were applied to the top and bottom of the domain. The mesh was generated using the blockMesh hexahedral, structured mesh tool. This background mesh was then refined in the areas specified in Fig. 4. The final stage of the meshing process was using the SnappyHexMesh tool. Using the STL and eMesh surface mesh file, cells are removed from the background mesh where the geometry is positioned and a "cell snapping" procedure moves the vertices of the intersecting cells to the surface profile. The last procedure refines the boundary layer mesh to meet the required y^+ of the first cell layers. The turbulence model selected for the study is the shear stress transport (SST) k-ω model. As the model implements a wall function, the first cell centre should be in the region of 30-300 (log-law region) for y^+ and the growth ratio was limited to 1.2. In order to grade the mesh, the first task was to calculate a first cell centre distance of $y^+ = 150$ based on boundary layer theory (in order to model the boundary layer correctly). The length of development of the boundary layer was set to be equal to the L_{pp} (length between perpendiculars) value specified by the user. This is assuming that no separation exists along the vessel wall. The grading of the background mesh is based on the calculated wall distance. For each direction, the cell size is halved for each refinement stage. So to maintain an aspect ratio of one in the final mesh sizing, the original mesh size is the calculated wall distance multiplied by 2^6. For this half model setup, the mesh size was approximately equal to 1.2 million cells. The study was run for 4,000 timesteps/iterations and convergence of the drag forces was checked afterwards. The simulation was repeated six times in the range of 1.335-1.668ms^{-1}.

Tab. 2. Mesh refinement Dimensions for Fig. 2.

Refine Level	Dimensions in Multiples of Lpp		
	x-min	x-max	y-min
1	-1.673	1.673	-1.004
2	-0.873	1.506	-0.502
3	-0.502	1.339	-0.251
4	-0.335	1.171	-0.167
5	-0.167	1.088	-0.100
6	-0.084	1.046	-0.092

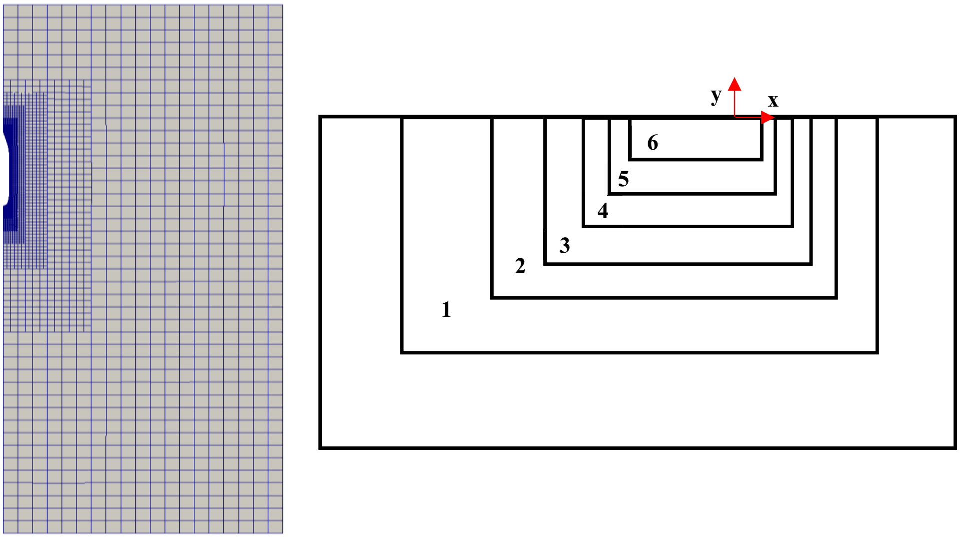


Fig. 4. Mesh refinement areas in x and y direction with origin specified (see Tab. 2 for dimensions).

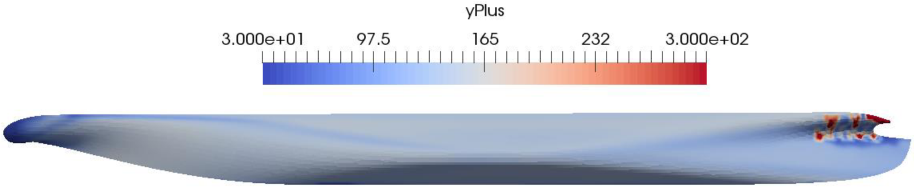


Fig. 5. Verification of y^+ values for the range 30 to 300.

In order to verify that the calculation of y^+ was correct for the mesh grading, a contour plot of this parameter was generated across the hull surface (Fig. 5). Apart from some spurious areas at the very front and back, the mesh conforms to the wall function requirements and is deemed acceptable.
To compare the results of the study to that of the DTC study, the total reistance coefficient, C_T, was calculated:

$$C_T = \frac{R_T}{0.5\rho V^2 S} \tag{1}$$

Where, R_T, is the total drag resistance, ρ, is water density, V, is the resultant velocity, S, is the wetted area. To non-dimensionalise the velocity, the Froude number, Fr, was also calculated:

$$Fr = \frac{V}{\sqrt{gL_{pp}}} \tag{2}$$

Where, g, is gravitional acceleration. Looking at Fig. 6, a reasonable comparison can be made with a maximum of 5.69% error for CFD attained R_T against experimentally measured data.

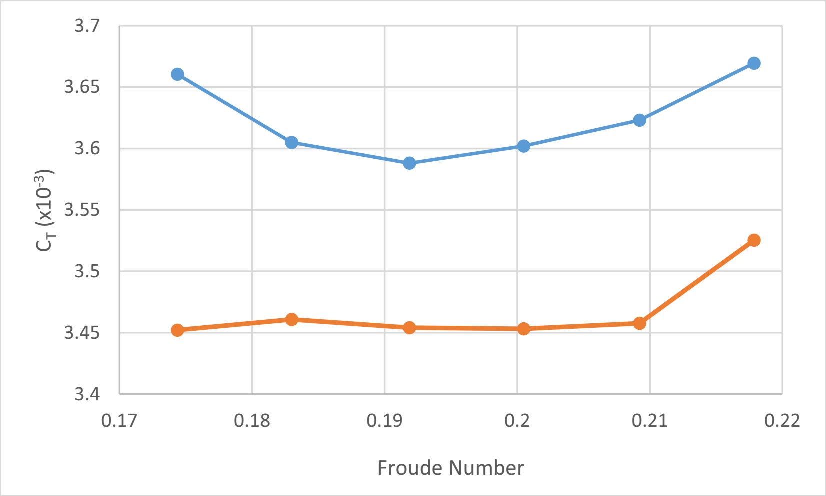


Fig. 6. Total drag coefficient for CFD (orange) and DTC data (blue).

For further validation, the bouyancy (vertical pressure based) and hydrostatic (horizontal pressure based) forces were calculated. To calculate the hydrostatic force, F_H:

$$F_H = \rho g z_G A \tag{3}$$

Where, z_G, is the vertical distance from the water level down to the centroid point, A, is the projected area of the wetted face. The centroid of the wetted hull face was calculated using the STL file in ParaView and was found to be equal to 123.338mm. The projected area of the wetted face was found to be 1.432m^2, resulting in a hydrostatic force of 1.731kN.
To calculate the buoyancy force, F_B:

$$F_B = \rho g V_d \tag{4}$$

Where, V_d, is the volume of water displaced by the hull at that particular draft. The buoyancy force was calculated to be equivalent to 3.977kN for the half model. Looking at Fig. 7 and 8, reasonable comparisons between the theoretical predictions of these forces against CFD can be made.

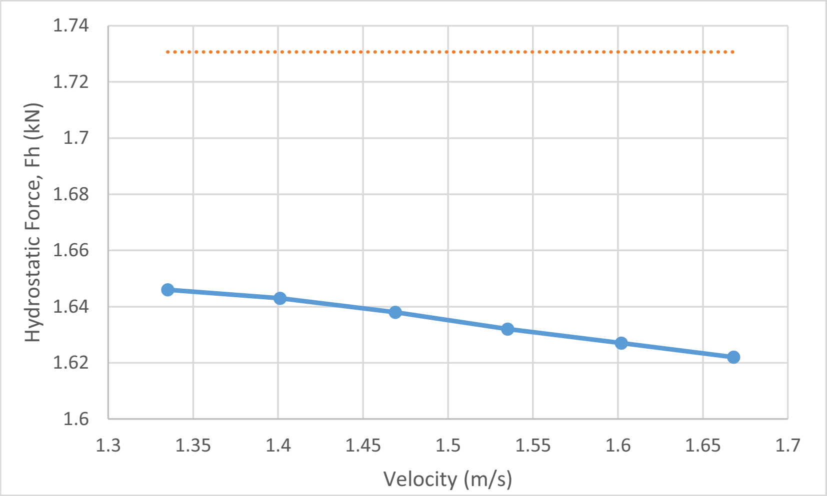


Fig. 7. Predicted bouyancy force (orange) against CFD value (blue)

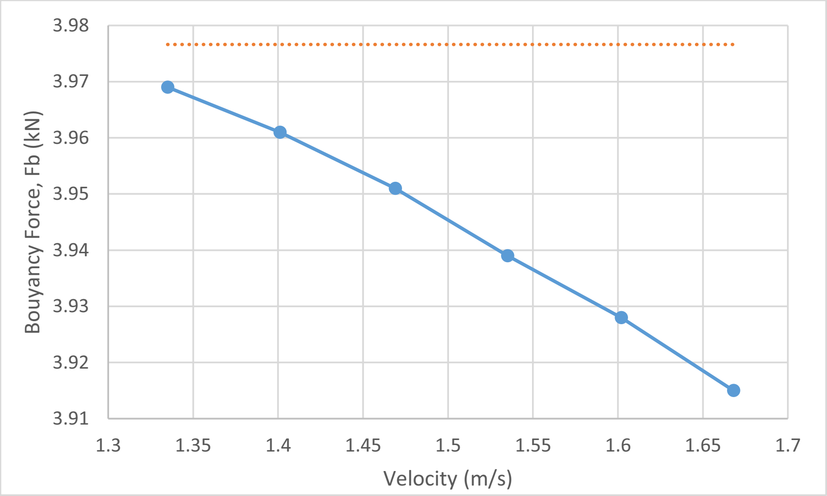


Fig. 8. Predicted hydrostatic force (orange) against CFD value (blue).

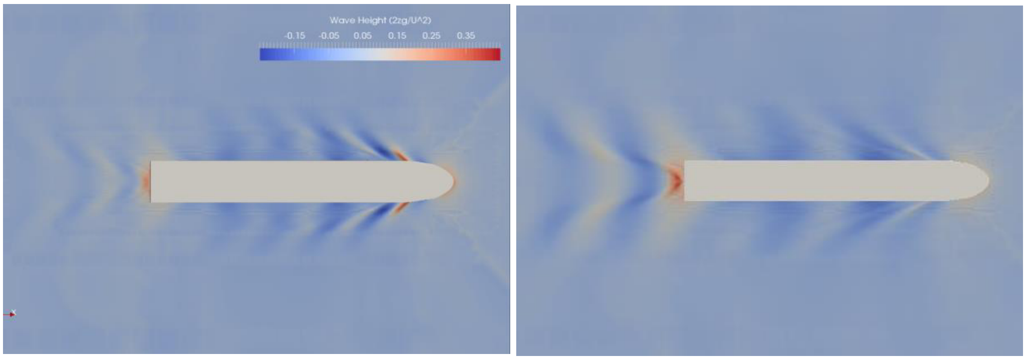


Fig. 9. Dimensionless wave height ($2\zeta g/U^2$) comparison (Fr = 0.174, left and 0.218, right). ζ is the wave change in height from the static water line.

4. Simulation of Planar Motion Mechanism (PMM)

4.1 Introduction

Before an explanation of the hydrodynamic derivatives is given, it should be noted that all derivatives for this study will be non-dimensionalised using the following equations for the benefit of scaling:

$$\text{Force}' = \frac{\text{Force}}{\rho V^2 L^2} \tag{5}$$

$$\text{Moment}' = \frac{\text{Moment}}{\rho V^2 L^3} \tag{6}$$

$$\text{Time, } T' = T\left(\frac{V_o}{L}\right) \tag{7}$$

$$\text{Acceleration, a'} = a\left(\frac{1}{V_o^2}\right) \tag{8}$$

$$\text{Velocity, v'} = \text{v}\left(\frac{1}{v_0}\right) \tag{9}$$

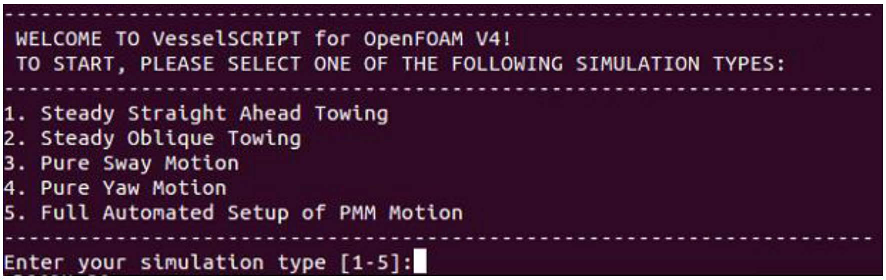
Abkowitz (1964) [8] derived a description of the hydrodynamic forces acting on a vessel through Taylor series expansion and it resulted in the following linear expressions of surge force, sway force and yaw moment:

$$\begin{gathered} X = m(\dot{u} - vr - x_G r^2) = X_o + X_u(u - U) + X_v v + X_r r + X_{\dot{u}}\dot{u} + X_{\dot{v}}\dot{v} + X_{\dot{r}}\dot{r} \\ Y = m(\dot{v} + ur + x_G\dot{r}) = Y_o + Y_u(u - U) + Y_v v + Y_r r + Y_{\dot{u}}\dot{u} + Y_{\dot{v}}\dot{v} + Y_{\dot{r}}\dot{r} \\ N = I_z\dot{r} + m x_G(\dot{v} + ur) = N_o + N_u(u - U) + N_v v + N_r r + N_{\dot{u}}\dot{u} + N_{\dot{r}}\dot{r} \end{gathered} \tag{10}$$

Due to the ship symmetryY_u, $Y_{\dot{u}}$, $N_{\dot{u}}$, N_u, X_v, $X_{\dot{v}}$, $X_{\dot{r}}$ and X_r are all equal to zero. Plus, for this study we are assuming no control-based derivatives. Once these terms are removed, the forces can be calculated using eqn. 11. The value assigned to each of the remaining derivatives for the DTC bare hull shall be determined from the following PMM motions.

4.2 Automation by Python Scripting

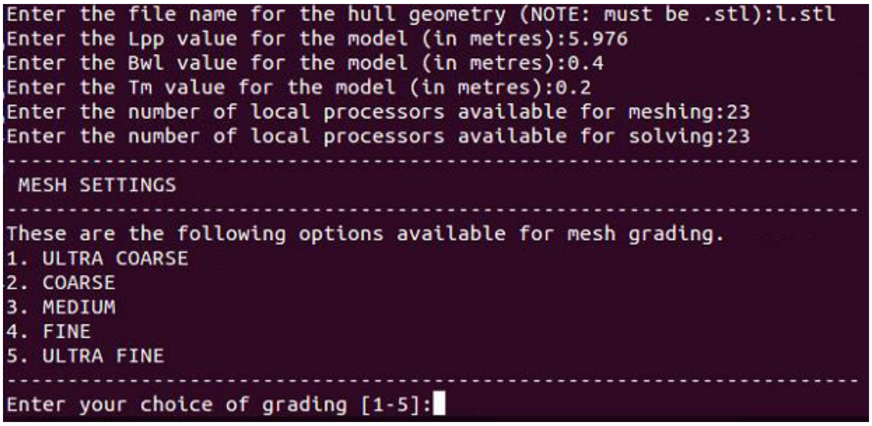
OpenFOAM is a collection of solvers written in C++ and the settings for each solver are dictated by a collection of dictionary text files which must be setup in the working directory. The idea behind the automation of each simulation is to have a collection of dictionary files filled with placeholders. The calculations for each setting, based on the user input, will be used to overwrite placeholders before the script automatically calls up the functions of the OpenFOAM library. When the user starts vesselScript.py, the first thing that will be asked is what type of simulation is to be performed (Snip. 1.). A choice of five is currently available, including a full automated run of PMM motion.



```
--------------------------------------------------------------------
 WELCOME TO VesselSCRIPT for OpenFOAM V4!
 TO START, PLEASE SELECT ONE OF THE FOLLOWING SIMULATION TYPES:
--------------------------------------------------------------------
1. Steady Straight Ahead Towing
2. Steady Oblique Towing
3. Pure Sway Motion
4. Pure Yaw Motion
5. Full Automated Setup of PMM Motion
--------------------------------------------------------------------
Enter your simulation type [1-5]:
```

Snip. 1. User request on startup.

After the user has made a choice between 1 and 5, the user will then be required to enter the name of the stereolithography (STL) file containing the information about the hull form/shape. The remaining parameters are sizes of the hull model, needed for setting up the computational domain. The user at this point can also specify the amount of physical cores available for the simulation to be run in parallel on. The user will then be asked for inputs relating to the chosen simulation.



```
Enter the file name for the hull geometry (NOTE: must be .stl):l.stl
Enter the Lpp value for the model (in metres):5.976
Enter the Bwl value for the model (in metres):0.4
Enter the Tm value for the model (in metres):0.2
Enter the number of local processors available for meshing:23
Enter the number of local processors available for solving:23
------------------------------------------------------------------
 MESH SETTINGS
------------------------------------------------------------------
These are the following options available for mesh grading.
1. ULTRA COARSE
2. COARSE
3. MEDIUM
4. FINE
5. ULTRA FINE
------------------------------------------------------------------
Enter your choice of grading [1-5]:
```

Snip. 2. Additional details requested before case setup.

4.3 Static Drift Test

In this particular PMM motion the forward speed is kept constant, the main parameter that changes for the static drift test is the drift angle. An increased drift angle will give the vessel an increased magnitude of v velocity. When this program is chosen, the user is asked for forward speed; the minimum and maximum drift angles; how many cases (the range is divided equally)? The static drift test is used to determine the parameters Yv and Nv. This is a steady state solution, so the solver setup is identical to that described in the validation study.

One will have noticed looking at Fig. 10, that the mesh is now a full model and the refinement areas detailed in the validation study have now changed by some angle. An easier implementation of this study would have been to use the circular partition seen in the pure yaw motion and rotate this region in accordance to the drift angle. But it was decided to pursue with this approach in order to minimize the effects of numerical diffusion. This method will keep the majority of the cell faces aligned with the main flow direction.

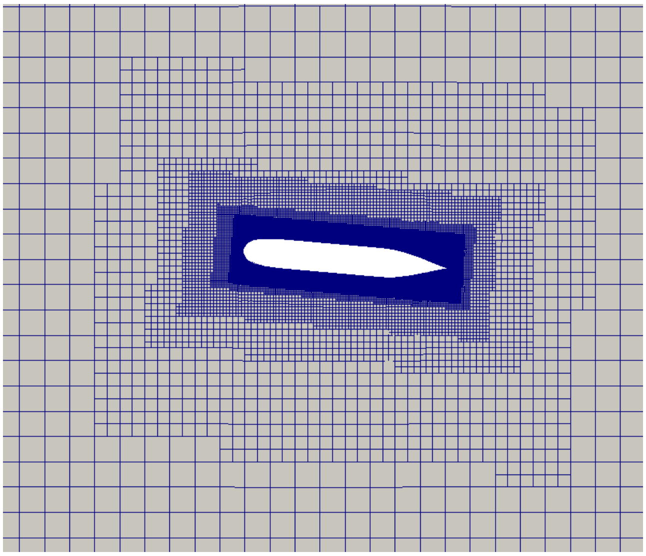


Fig. 10. Drift study mesh setup.

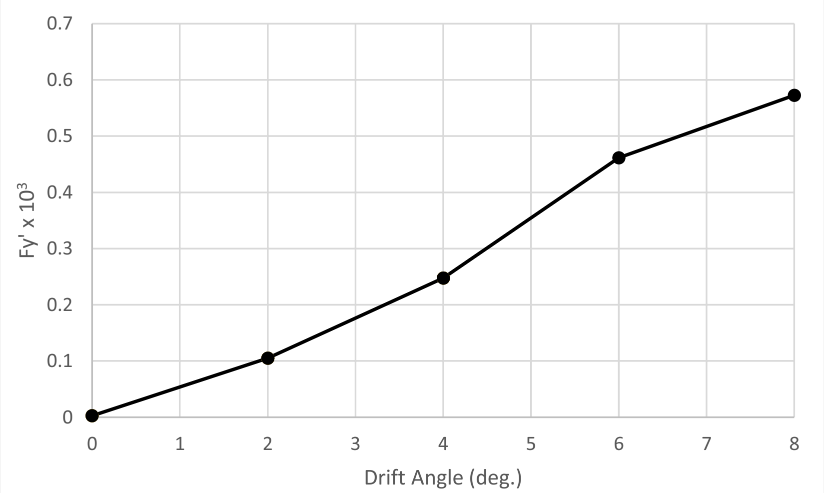


Fig. 11. Determination of Yv.

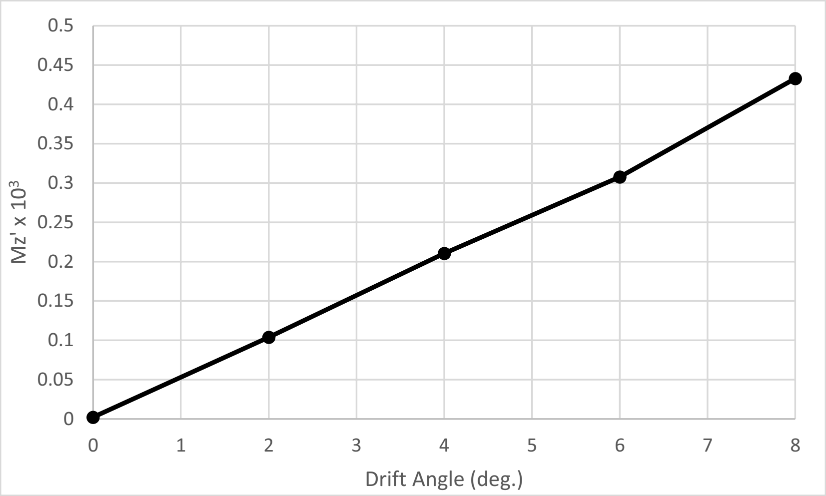


Fig. 12. Determination of Nv.

4.4 Pure Sway Test

For the pure sway PMM motion, the mesh described in the validation study is retained, except that it is a full model setup (as the flow over time is no longer symmetrical). The swaying motion is achieved by deforming the grid using the Laplacian displacement solver. An oscillatingDisplacement BC is applied to the hull surfaces, with a set amplitude and frequency (calculated by the script for each case). A diffusivity of 1 is applied to the hull mesh to retain the correct sizing of the boundary layer mesh, whereas everywhere else in the domain is allowed to change size. The main aim of this study is to determine $Y_{\dot{v}}$. This is achieved by inducing a swaying motion with increasing amplitude for the same time period, hence increasing the maximum acceleration for each case. When the user selects this study, the forward motion velocity will need to be specified; the time period for each complete motion and then the user will need to specify the range of amplitude (maximum sway distance) for the study.

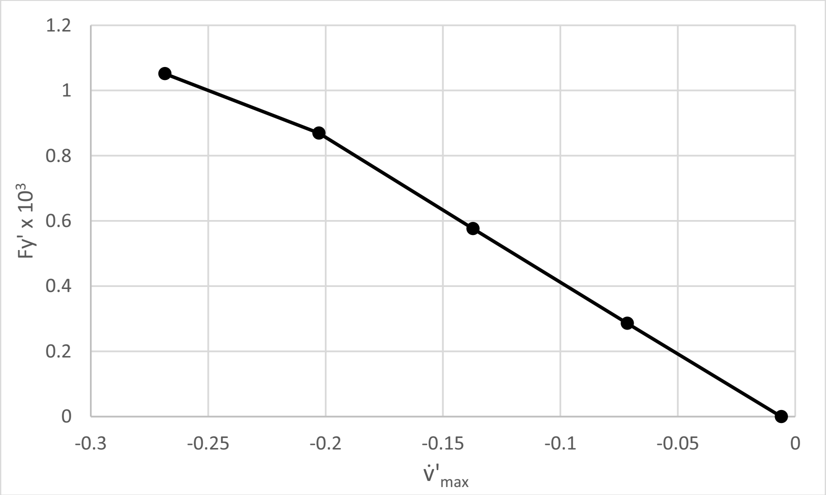


Fig. 13. Determination of $Y_{\dot{v}}$.

4.5 Pure Yaw Test

The pure yaw test is a little more difficult to setup. The idea is to retain the sway motion seen in the previous case (kept at constant amplitude) but now force the vessel to change its yaw angle throughout the motion. The user must specify the range of maximum yaw angle for the study. Parameters which are kept constant in this simulation are forward velocity, lateral amplitude and the time period for a complete cycle of yaw motion. Some changes were made to the refinement areas in order to facilitate the rotational region. To create the rotational region, a unit size cylinder geometry is supplied to the snappyHexMesh utility in order to split the region and create a separate partition (by the “snapping” process mentioned earlier”). The unit sized cylinder is scaled to the correct dimensions by the python script beforehand. Essentially, the main difference in implementing this simulation over the pure sway study, is that a multi-motion BC is specified retaining the oscillatingDisplacement BC but also specifying an additional oscillatingRotating BC to the partition. The partition acts as what is referred to as a “sliding mesh” technique. A cyclicAMI BC is applied to the two faces between the stationary and rotating region. Pressure and velocity values are interpolated to the nearest neighbour cells.

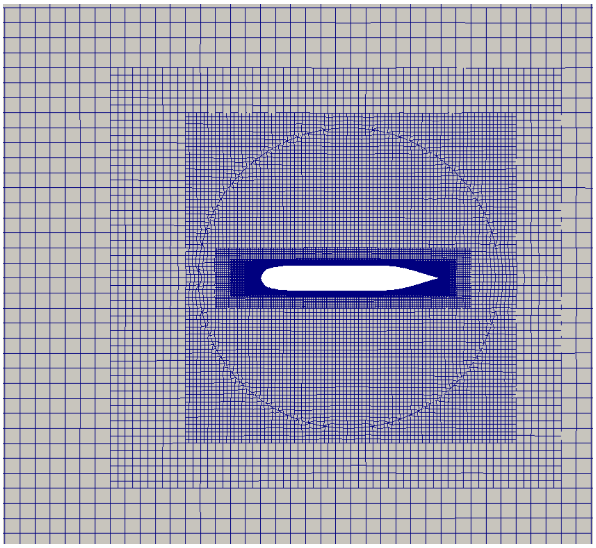


Fig. 14. Mesh used for pure yaw study.

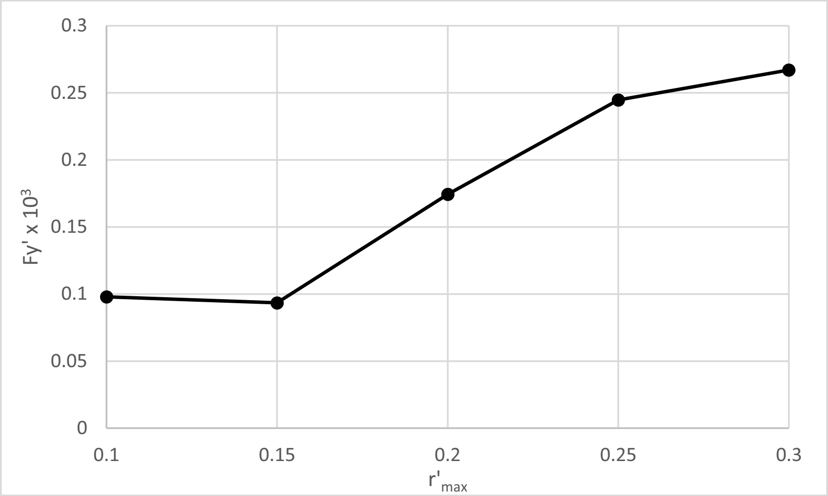


Fig. 15. Determination of Y_r.

.

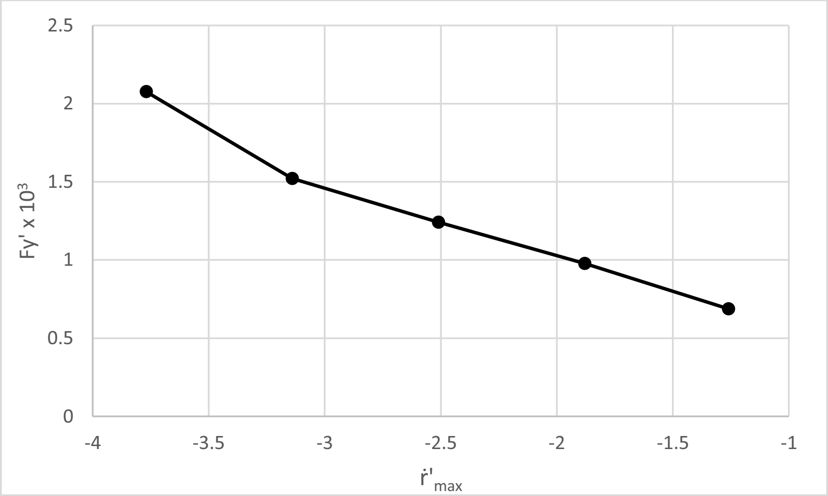


Fig. 16. Determination of $Y_{\dot{r}}$.

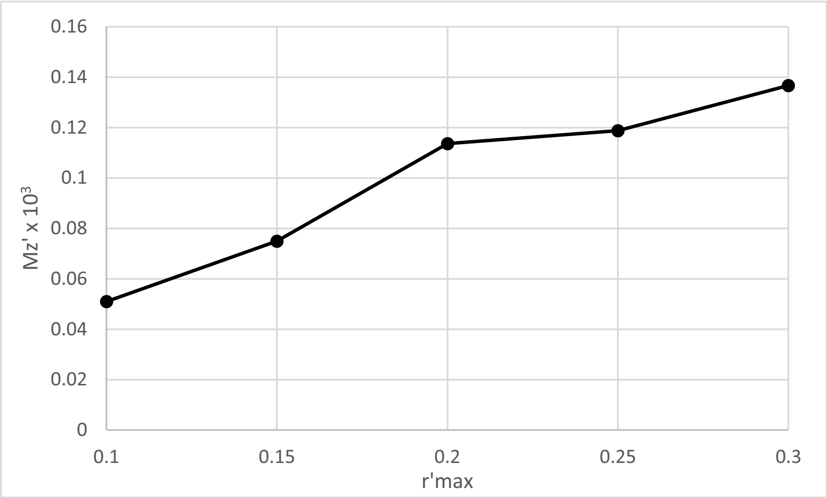


Fig. 17. Determination of Nr.

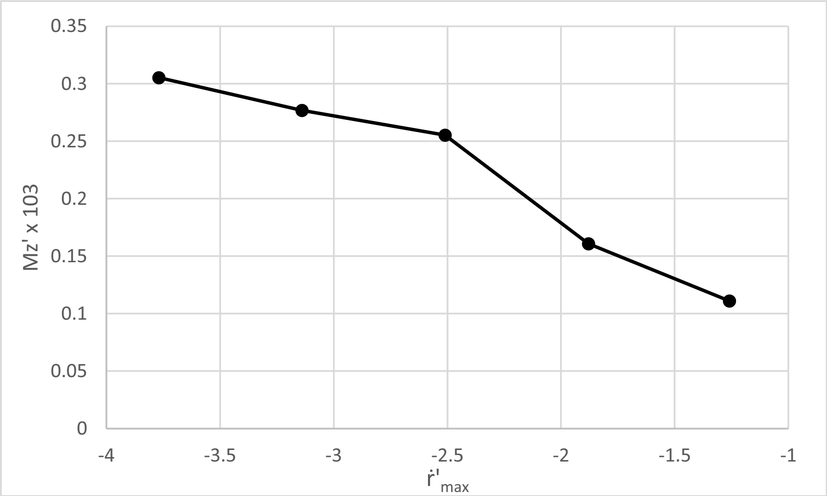


Fig. 18. Determination of $N_{\dot{r}}$.

4.6 Implementation of Hydrodynamic Derivatives

Tab. 3. Hydrodynamic Derivatives.

Derivative	Predicted Value (x10^3)
Y_v	0.0712
N_v	0.0530
$Y_{\dot{v}}$	4.0681
Y_r	0.8888
N_r	0.4880
$Y_{\dot{r}}$	0.5192
$N_{\dot{r}}$	0.0874

The resulting hydrodynamic derivatives are seen in Tab. 3 for Y (sway) forces and N moments (resulting from yaw movement). Simply put, each of the coefficients multiplied by its subscript notation will produce the required force or moment value. The surge based coefficients, which are not presented in Tab. 3, can be obtained from the approach seen in validation study. Again, from plotting the non-dimensionalised force against velocity and acceleration will allow for these coefficients to be determined. This simplistic description will be used with the determined derivatives within the hull resistance section of the vessel modelling library.

$$X = X_o + X_u u + X_{\dot{u}} \dot{u}$$

$$Y = Y_o + Y_v v + Y_r r + Y_{\dot{r}} \dot{r}$$

$$N = N_o + \ N_v v + N_r r + N_{\dot{r}} \dot{r}$$

(11)

4.7 Conclusive Remarks in Relation to the Hydrodynamic Modelling

The presented method in attaining these hydrodynamic derivatives through automation by python script with OpenFOAM has shown some success. Although these values still need to be compared against some experimental data, so a hint of caution is advised with the use this data presented. Another approach to validation is to transfer the hydrodynamic derivatives to the vesselEfficiency library and perform a set steering task whilst replicating this manoeuvre within OpenFOAM for comparison. So, further validation does still need to be performed in order to verify the accuracy of such methods. On the other hand, further testing through validation by using other benchmark hulls will allow for debugging of the script, making sure that the simulation does work for a wide variety of conditions.

From a teaching perspective, having such an interface for an open-source CFD package to predict the hydrodynamic behaviour of vessels is particularly useful. This allows staff and students to explore ideas surrounding stability and hull resistance without the large time delays of having to learn to drive the code. Then transferring this information into the vesselEfficiency library for real-time simulation and human interaction will add to the whole experience of realizing the effects of certain factors that impinge on the manoeuvrability of vessels, including that of wave induced forces.

Further to this, the script should now be developed in order to generate a set of 4 DOF hydrodynamic derivatives, taking into account roll motion which can, for medium to high speed vessels, be a hindrance on manoeuvrability. Other additional aspects of the hydrodynamics could also be investigated, such as generating propulsion/thrust curves for propellers and calculating lift off control surfaces (rudders, fins etc.).

5. Interactive Real-Time Simulator

The vessel modelling library has been developed to include data exchange over UDP (User Datagram Protocol). This enables packets of data to be transmitted and received by the model during execution. If the model is simulated in real-time, then data can be sent to, and received from, third party applications on the local machine or distributed over a network. In addition, it is possible to execute the model using a hard-real-time platform such as Simulink Real-Time, which support deterministic model execution.

The test system consists of an example vessel model which is executed in real-time on a desktop PC which is running Simulink Real-Time (the Engineering Model). The Target Simulator is connected over Ethernet to a second Desktop PC running Windows which is running an application developed using Unity3D (Simulation Interface). The Simulation Interface receives data over UDP which defines the position and orientation of the vessel. The motion of the vessel is imposed on a 3D model of the vessel which may be navigated around the virtual environment. The data exchange protocol may also be used to modify inputs to the model which makes it possible to use external hardware or software control panels to operate the vessel model. The diagram in Fig. 19 shows an overview of the test system.

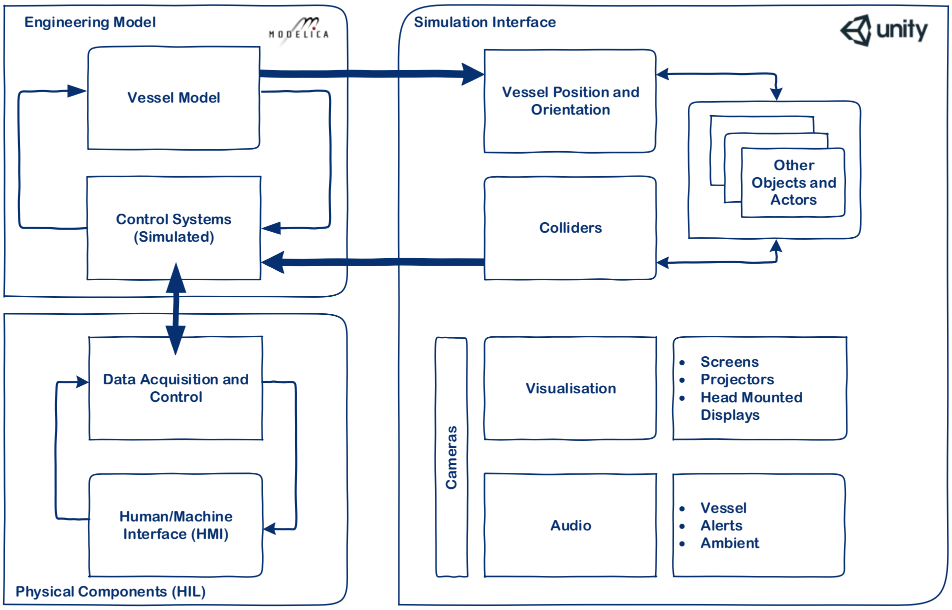


Fig. 19. Overview of the demonstration system.

The model which is executed in Simulink Real-Time is shown in Fig. 20. The position data from the model is packaged as a Byte array and is transmitted over UDP to the Simulation Interface which unpacks the Byte array, interprets its contents and renders the vessel within a 3D environment. The user may control the vessel by modifying inputs to the propulsion and steering system.

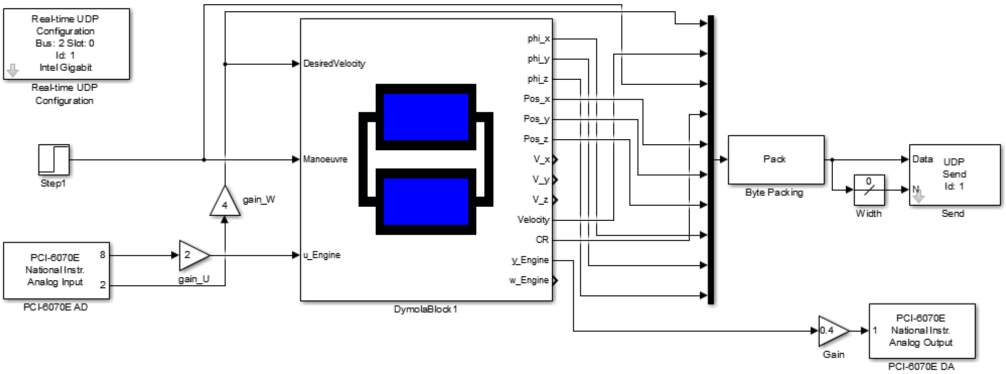
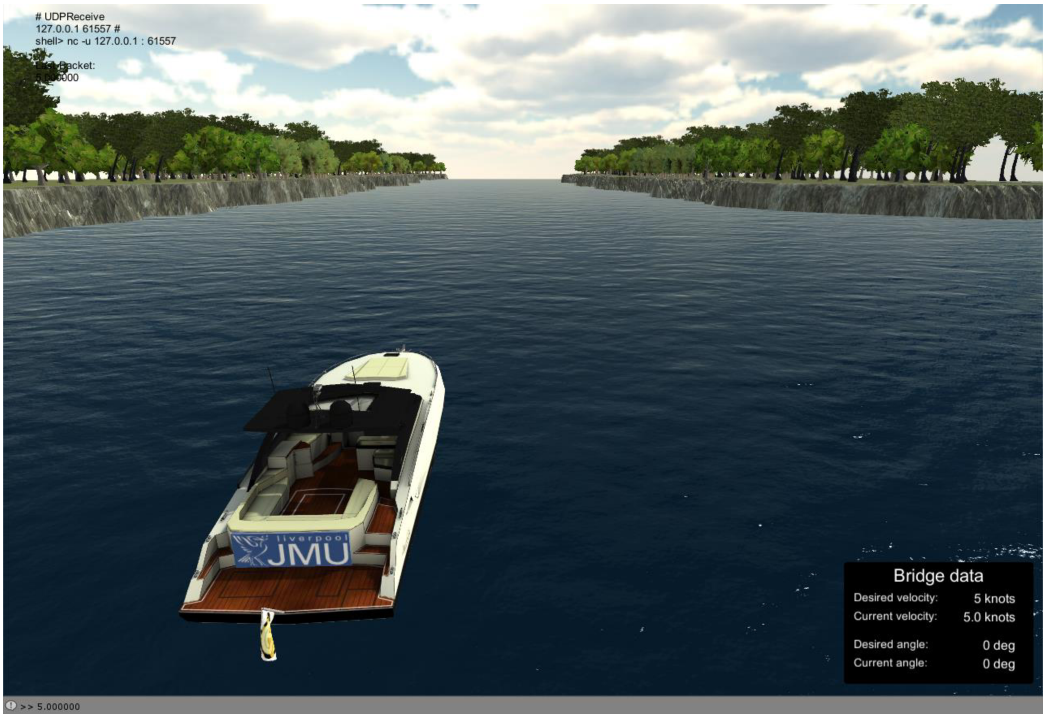


Fig. 20. Simulink Model incorporating the example vessel model



```
# UDPReceive
127.0.0.1 61557 #
shell> nc -u 127.0.0.1 : 61557

Last Packet:
5.000000
```

Bridge data

Desired velocity: 5 knots
Current velocity: 5.0 knots

Desired angle: 0 deg
Current angle: 0 deg

```
>> 5.000000
```

Fig. 21. Example of light water craft simulated in a 3D environment.

6. Applications in an Educational Environment

The model library offers a number of potential applications in the context of marine engineering education. At present the library is still in development and as such is not ready for deployment in the classroom, although it can and has been used as the basis of student projects. It is envisaged that with development and contributions from other users, the library could mature to become a useful tool in the educational environment.

In the first instance it is expected to provide a way for students to investigate the performance and operation of a vessel in simulation. This requires no technical ability in physical modelling as the vessel, manoeuvres and environment can be defined through high-level options with only basic training required. The hierarchical structure of the library, and the use of replaceable classes means that with a little more training, the user can develop their own simulation scenarios using templates for vessel and subsystems. This means that original investigations can be formulated by students without significant expertise in mathematical modelling.

If a student or teacher finds that the existing models do not offer some feature that they need, then openness of the library means that they can develop their own models which are compatible with the library using appropriate base classes and templates. This opens up the possibility for very extensive modification of vessel models with levels of detail and fidelity determined by the individual application. The library therefore offers potential benefits at all levels of a degree level programme and in postgraduate study or research projects.

The ease of use of the model, and the interactive simulation interface which it offers also extends its possible applications to the training of deck crew. The library has the potential to form the basis of a free and open-source vessel simulator. It is envisaged that with development the library could offer institutions with a highly customisable and potentially a very cost effective way of developing their own ship simulation capability. The interactive simulation techniques which are presented in Section 5 have been found to work well with standard Virtual Reality head mounted display (HMD) systems such as the Oculus Rift and the network based data exchange system would lend itself to highly scalable and geographically distributed training and simulation exercises.

7. Conclusions

This report has described the development of a new software library for modelling and simulation of marine vessels. The new library is built using an object-oriented approach, and offers hierarchical modelling. Hierarchical models can be configured for use by non-experts but continue to offer low level access to model detail. The individual modelling elements are replaceable and so new vessel configurations can be created or existing models can be maintained by swapping in or out model elements. The development version of the library is available on a public code repository (https://github.com/vesselEfficiency/vesselEfficiency-dev) and will continue to be developed internally at LJMU. It is hoped that others may choose to use and contribute to the library.

At present the library offers basic functionality and a sparse set of vessel configurations. It is expected that the library will grow as it matures and that due to its, flexibility new capabilities will be incorporated at regular intervals. In terms of model development, the primary focus during this project have been in the following areas:

- **Common Library Structure** – establishing a scalable and systematic library structure which allows for flexibility in vessel configuration, and the definition of simulation scenarios.

- **Hydrodynamic Modelling** – developing a method for implementing surrogate hydrodynamic model using Abkowitz' work. This has delivered an additional benefit in the form of an open-source tool for carrying out numerical simulations of steady towing and PMM to derive hydrodynamic coefficients. At present this is limited to 3 degrees of freedom, but work is ongoing to extend this to 4 degrees of freedom (incorporating roll effects).

- **Interactive Simulation** – the incorporation of a data exchange protocol using UDP enables the model to be operated in hard or soft real-time and for this to be used as the basis for interactive simulation. This means that in an educational context, students can develop simulation models of vessel and then operate them virtually. This opens up interesting avenues for the development of interactive learning tools in marine engineering.

Further work will include an expansion of the number of subsystems which are included in the basic library structure, and the fidelity of the example models that are included with the library. In terms of its effectiveness as an educational tool, the library is ready for use with projects involving final year Bachelors students or with Masters students but is not ready to be used in teaching (for example) an undergraduate class in Marine Engineering although it is expected that this will change as the library matures.

Acknowledgements

The authors wish to acknowledge the support which has been received from the Nippon Foundation and IAMU, without which this work would not have been possible. They would also like to acknowledge the contribution which has been made to the interactive simulation portion of this work by MSc students Tobias Marn and Ernst Klimmey from Hochschule Wismar, University of Applied Sciences, Technology, Business and Design.

References

[1] Duru, O., "The origin and consistency of the ton-mile metric in the shipping economics". *Logistics*, Vol. 1, No. 1, (2017), pp 1-8.

[2] "Resolution MSC.137(76) Standards for Ship Manoeuvrability". *International Maritime Organization (IMO)*, (2002).

[3] Hajivand, A. and S.H. Mousavizadegan, "Virtual maneuvering test in CFD media in presence of free surface". *Int. J. Nav. Archit. Ocean Eng.*, Vol. 7, (2015), pp 540-558.

[4] Kim, H., H. Akimoto, and H. Islam, "Estimation of the hydrodynamic derivatives by RaNS simulation of planar motion mechanism test". *Ocean Engineering*, Vol. 108, (2015), pp 129-139.

[5] Araki, M., et al., "Estimating maneuvering coefficients using system identification methods with experimental, system-based, and CFD free-running trial data". *Ocean Engineering*, Vol. 51, (2012), pp 63-84.

[6] He, S., et al., "Manoevering prediction based on CFD generated derivatives". *Journal of Hydrodynamics*, Vol. 28, No. 2, (2016), pp 284-292.

[7] "Guidelines on use of RANS tools for manoeuvring prediction (7.5-03-04-01)". *International Towing Tank Conference (ITTC)*, (2011).

[8] Abkowitz, M.A., "Lecture notes on ship hydrodynamics-steering and manoeuverability ". *Hydro and Aerodynamics Laboratory (Lyngby, Denmark)*, (1964).

[9] Moctar, O., V. Shigunov, and T. Zorn, "Duisburg Test Case: Post-Panamax Container Ship For Benchmarking". *Ship Technology Research*, Vol. 59, No. 3, (2012), pp 50-64.

International Association of Maritime Universities
Meiwa Building 8F, 1-15-10 Toranomon, Minato-ku, Tokyo 105-0001, Japan
Tel : 81-3-6257-1812 E-mail : info@iamu-edu.org URL : http://www.iamu-edu.org
ISBN No. 978-4-907408-19-0