# MODULAR SIMULATION OF SHIP DYNAMICS IN MANOEUVRING

Diploma Thesis, January 2010

Iason Stefanatos

Supervisor
Professor Nikolaos P. Kyrtatos

NATIONAL TECHNICAL UNIVERSITY OF ATHENS
LABORATORY OF MARINE ENGINEERING

# Summary


The subject of the present diploma thesis is to present a methology for modeling ship dynamics in manoeuvring. The model consist of small elementary subsystems which can be modeled separately and independently from the rest of the system. These subsystems are connected through well defined interfaces and can built up the whole structure.

The implementation of the model was done in the programming environment Matlab/Simulink. The model was organized hierarhicaly in building blocks, which include the separate subsystems.

The present text covers the mathematical models for each subsystem as well as the actual implemantation of the complete model.

This thesis is organized as follows:

In the first Chapter the concept of the thesis is presented. Chapter 2 includes the mathematical models used for each subsystem, which are the following:

- Ship forces block, which calculates the forces acting on ship, including
  - Resistance block
  - Hydrodynamic forces block
  - Proppeller forces block
  - Rudder forces block
- Dynamics block, which calculates the velocities and turn rates
- External forces block
- Post processing block, which converts the data to coordinates
- Miscellaneous calculations block

In Chapter 3 the actual implementation method is decribed. Every sub-block is depicted and fully explained. Each section includes an example presenting the function of each block. Chapter 4 includes the case studies used for the validation and the results of the simulations in comparison to published experimental data. Finally, Chapter 5 includes a conclusion and possible future work that could be done.

Contents

Chapter 1

# Introduction

Nowadays, the wide-spread use of computers in combination with the the increase of computer power, made simulation a very useful tool in every field of science. The benefits are simple and can be summarized in the fact that experiments can be done without wasting great amounts of resources, human labor and time.

Especially in the applied marine hydrodynamics, simulation can be used in the following aspects:

- Selection of propeller and rudder
- Ship-propeller-engine cooperation
- Manoeuvering efficiency
- Transient loads
- Shallow water effects

Manoeuvring modeling is a subject that has been addressed by many researchers in the recent years. One of the early attempts to create a complete model was done by Inoue (Inoue et. al, 1981). Several models were presented since then, some of which are by Lambropoulos (Lambropoulos, 2000),which has been developed in the Laboratory of Marine Engineering of NTUA, Woodward (M.D. Woodward et al, 2003) and Lee (T.I. Lee et al, 2003). Nowadays the main research on this field is done by marine institutes such as KRISO, MOERI and MARIN.

The model of this thesis was intended to be a state of the art model, meaning that the submodels used are up to date and complex. The model was built according to ITTC standards and procedures. The use of the overall model provides an understanding of the physical processes appearing in maneuvering of vessels. The implementation was coded in the Matlab/Simulink programming environment, which was choosen due to the simplicity of the block building concept.

Despite the complexity of the whole model, the system is decomposed in several subsystems (blocks) that include the individual submodels and are built hierarchically. Thus, in a future study, the designer can isolate the blocks of choice and add new ones, i.e. a submodel can be replaced by a much more sophisticated one. The main blocks are the following:

- the ship forces block used to model the effects that exercise forces on the ship including four main sub-blocks:
    - Resistance block

    - Hydrodynamics block
    - Propeller block
    - Rudder block
- the  dynamics block used to model the the ship's movement
- the external ship forces block used to model the external phenomena such as the wind, or ice (future development)
- some auxiliary blocks used to calculate miscellaneous parameters

Most of the sub-blocks were validated with published experimental data in order to ensure model accuracy. However, the complete validation of the complete overall model was only partly possible due to lack of large amounts of experimental data.

Chapter 2

# Theoretical Models

## 2.1 Dynamics

The ship's maneuvering motions can be described by the equations of motion. Every ship can execute 6 type of motions, according to 6 Degrees of Freedom (DOF). These motions (Fig. 2.1 & 2.2) are the following:

- Surge – linear longitudinal motion
- Sway – linear lateral motion
- Heave – linear vertical motion
- Roll – rotation about the longitudinal axis
- Pitch – rotation about the transverse axis
- Yaw – rotation about the vertical axis

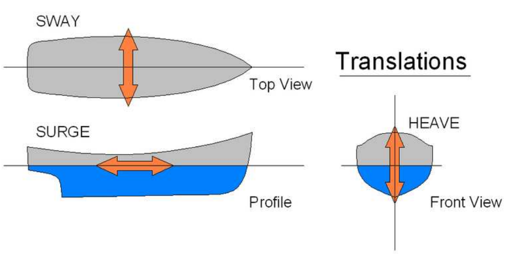


Figure 2.1- Ship's translations

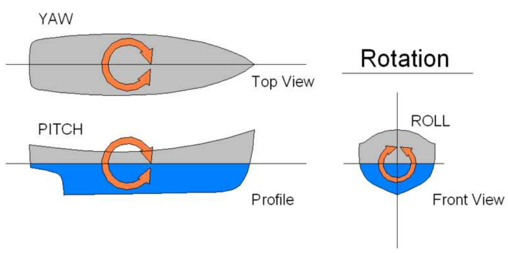


Figure 2.2 – Ship's rotations

The model described by this thesis simulates the ship motion in relatively calm water, and 4 DOF are adequate. The 4 DOF that are chosen are Surge, Sway, Roll, Pitch.

In order to analyse the system with 4 DOF, there are two systems of coordinates (Fig. 2.3). The first one is bound in the midsection of the ship and thus, moving with the ship. It is called body-fixed system and the centre of it can be located in the Centre of Gravity (COG) of the ship. The other system is earthbound and stays still. It represents the view of the ship by an observer. Its main advantage is that it gives the user the ability to observe the trajectory of the ship's motion. The conversion from the one to the other is described in paragraph 2.1.2.

At the end of each section there is a notation paragraph.

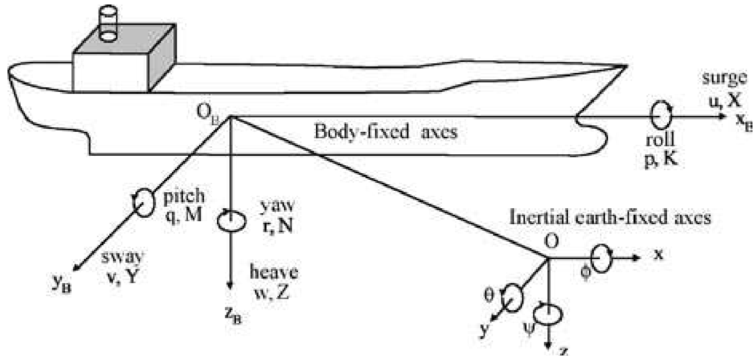


Figure 2.2 – Coordinates systems

### 2.1.1 Equations of motion

As described in the previous paragraph, the model of this thesis simulates the motion of a ship with 4 DOF. The differential equations that describe these four motions express the relationship between the forces and moments acting on ship and the velocities, acceleration, mass and moment of inertia, as following:

$$\begin{aligned}
&\text{Surge:} \quad && (m + m_x)\cdot \dot{u} - m \cdot v \cdot r = X \\
&\text{Sway:} \quad && (m + m_y)\cdot \dot{v} - m \cdot u \cdot r = Y \\
&\text{Yaw:} \quad && (I_z + J_z)\cdot \dot{r} = N \\
&\text{Roll:} \quad && (I_x + J_x)\cdot \dot{p} = K
\end{aligned} \tag{2.1}$$

The calculation of the forces and moments, included in the right part of these equations is complex and is described in section 2.2.

### 2.1.2 Coordinates conversion

As described before, two coordinates systems were used. All the calculations were done in the point-fixed system, because of the apparent simplicity of the equations. Thus, a conversion should be made to acquire the trajectory of the ship on the plane (i.e. sea).

In order to convert the coordinates, the $\dot{x}_{oG}$ and $\dot{y}_{oG}$ components of the velocity vector are calculated from the following equations, which include fixed-point system velocities $u$,$v$ and angle $\psi$ (given in eq. 7.3):

$$\begin{aligned}
x_{oG} &= \int_0^{\tau} (u \cdot \cos(\psi) - v \cdot \sin(\psi))dt \\
y_{oG} &= \int_0^{\tau} (v \cdot \cos(\psi) + u \cdot \sin(\psi))dt
\end{aligned} \tag{2.2}$$

$$\psi = \int_0^{\tau} r dt \tag{2.3}$$

By integrating the results from eq. (2.2), the result is the position of the ship. The plot of $x_{oG}$ versus $y_{oG}$ in time is the trajectory of the ship's movement (fig. 2.4).

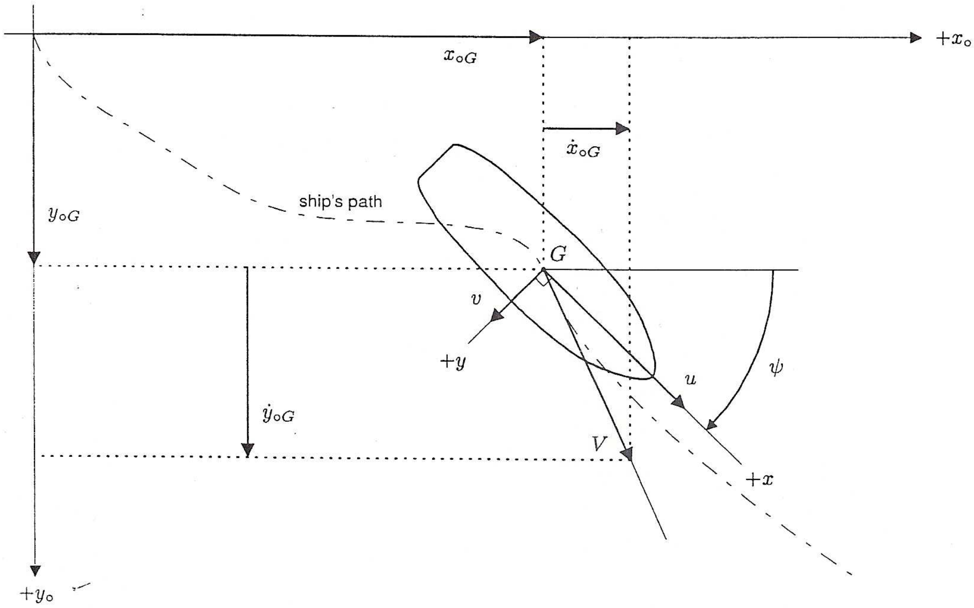


Figure 2.4 – Earthbound coordinate system

## 2.1.3 Notation

| | |
|---|---|
| $m$ | ship mass (displacement), [*kg*] |
| $m_x$ | ship's added mass (x-axis), [*kg*] |
| $m_y$ | ship's added mass (y-axis), [*kg*] |
| $I_z$ | moment of Inertia (z-axis) |
| $I_x$ | moment of Inertia (x-axis) |
| $J_z$ | added moment of Inertia (z-axis) |
| $J_x$ | added moment of Inertia (x-axis) |
| $u$ | surge velocity, [*m/s*] |
| $\dot{u}$ | surge acceleration [$m/s^2$] |
| $v$ | sway velocity, [*m/s*] |
| $\dot{v}$ | sway acceleration [$m/s^2$] |
| $r$ | yaw turn rate, [*rad/s*] |
| $\dot{r}$ | yaw acceleration [$rad/s^2$] |
| $p$ | roll turn rate, [*rad/s*] |
| $\dot{p}$ | roll acceleration [$rad/s^2$] |
| $X$ | total surge force [*N*] |
| $Y$ | total sway force [*N*] |
| $N$ | total yaw moment [*Nm*] |
| $K$ | total roll moment [*Nm*] |
| $\dot{x}_{oG}$ | ship velocity in x-axis of earthbound system, [*m/s*] |
| $\dot{y}_{oG}$ | Ship velocity in y-axis of earthbound system, [*m/s*] |
| $\psi$ | ship's heading in point-fixed system , [*rad*] |

## 2.2 Ship Forces

### 2.2.1 Introduction

The hydrodynamic forces acting on a ship (right hand side of eq. (2.1)), are the result of the interaction of the surrounding fluid (water) with the ship's hull, propeller and rudder. The calculation of them is demanding and thus two assumptions were made:

- the maneuver and generally all the motions are slow (usually true for the majority of commercial ships)
- the value of a force or moment at a time step *t* is independent from the value in the previous time step (*t-dt*)

The forces and moments consist of several parts, some of which are functions of the velocities and accelerations of the ship. The parts of the forces that were chosen for this model are the following:

- Resistance – ship resistance in calm water, plus wind resistance and the effect of shallow water

- Hull forces – hydrodynamic forces acting on ship, calculated with the use of hydrodynamic derivatives, plus the effect of shallow water

- Propeller forces – hydrodynamic forces acting on ship, due to operation of the propeller (thrust) and interaction with the hull

- Rudder forces – hydrodynamic forces acting on ship, due to application of the rudder, plus the effect of shallow water

- Other forces – external forces and environmental disturbances (i.e. shallow water)

As obvious the resistance is a part of hull forces but because of the its great significance and complexity it is described in a subnsequent paragraph.
Thus, the dynamics equations (2.1) become:

$$\begin{aligned}
&\text{Surge:} && (m+m_x)\cdot\dot{u} - m\cdot v\cdot r = R + X_H + X_P + X_R + X_{ext} \\
&\text{Sway:} && (m+m_y)\cdot\dot{v} - m\cdot u\cdot r = Y_H + Y_P + Y_R + Y_{ext} \\
&\text{Yaw:} && (I_z+J_z)\cdot\dot{r} = N_H + N_P + N_R + N_{ext} \\
&\text{Roll:} && (I_x+J_x)\cdot\dot{p} = K_H + K_P + K_R + K_{ext}
\end{aligned} \tag{2.4}$$

### 2.2.2 Resistance

The resistance of the ship was is defined as the force, which opposes the movement of the ship. Thus, it's direction is always opposite to the ship's direction.

For ahead movement of the ship the resistance is negative in vice versa. Usually it is calculated in calm water.

In this model, apart from the resistance in calm water, the effect on resistance of maneuvering through shallow water is considered.

### 2.2.2.1 Calm water resistance

The calculations were done according to the Holtrop method (Holtrop, Mennen, 1982) (Holtrop, 1984), which is the most frequently used method for such applications. This method was developed through a regression analysis of random model experiments and full-scale data.

The total resistance of the ship has been subdivided into:

$$R_{total} = R_F \cdot (1 + k_1) + R_{APP} + R_W + R_B + R_{TR} + R_A \tag{2.5}$$

where:

| | |
|---|---|
| $R_F$ | frictional resistance according to the ITTC-1957 friction formula |
| $(1+k_1)$ | form factor describing the viscous resistance of the hull form in relation to $R_F$ and speed dependent |
| $R_{APP}$ | resistance of appendices |
| $R_W$ | wave-making and wave-breaking resistance |
| $R_B$ | additional pressure resistance of bulbous bow near the water surface |
| $R_{TR}$ | additional pressure resistance of immersed transom stern |
| $R_A$ | model-ship correlation resistance |

The frictional resistance is calculated according to ITTC method as follows:

$$R_F = \frac{1}{2} \cdot \rho \cdot S \cdot u^2 \cdot C_F \tag{2.6}$$

where $\rho$ is the water density, u the ship's speed, S the wetted area of the ship and $C_F$ the frictional resistance factor. The latter is calculated as (ITTC – 1957 formula):

$$C_F = \frac{0.075}{(\log_{10} \mathrm{Re} - 2)^2} \tag{2.7}$$

where:

$$\mathrm{Re} = \frac{u \cdot L}{\nu} \tag{2.8}$$

The wetted area of the hull can be approximated well by:

$$S = L \cdot (2T + B)\sqrt{C_M}\left(0.453 + 0.4425 \cdot C_B - 0.2862 \cdot C_M - 0.003467\, B/T + 0.3696 \cdot C_{WP}\right) + \\ + 2.38\, A_{BT}/C_B \qquad (2.9)$$

where $C_B$ is the block coefficient based on the waterline length

The form factor $1+k$ is calculated as following:

$$1 + k = 0.93 + 0.487118 \cdot c_{14} \cdot (B/L)^{1.06806} \cdot (T/L)^{0.46106} \cdot (L/L_R)^{0.121563} \cdot \\ \cdot \left(L^3/\nabla\right)^{0.36486} \cdot (1 - C_P)^{-0.604247} \qquad (2.10)$$

In this formula $C_p$ is the prismatic coefficient based on the waterline length. The length of run $L_R$ is the length from the section of maximum area of the after end of parallel middle body to waterline termination. It can be approximated by:

$$L_R = L \cdot \left(1 - C_P + 0.06 \cdot C_P \cdot lcb/(4C_P - 1)\right) \qquad (2.11)$$

where *lcb* is the longitudinal position of the centre of buoyancy forward of *0.5L* as a percentage of *L*.

The coefficient $c_{14}$ accounts for the stern shape. It depends on the stern coefficient $C_{stern}$ for which the figures of table 2.1 can be given:

$$c_{14} = 1 + 0.011 \cdot C_{stern} \qquad (2.12)$$

*Table 2.1*

| Afterbody form | $C_{stern}$ |
|---|---|
| Pram with gondola | -25 |
| V-shaped sections | -10 |
| Normal section shape | 0 |
| U-shaped sections with Hogner stern | 10 |

The effect of ship speed on the form factor is taken into account by using the Froude Number. According to the "Report on the power performance committee", (ITTC,1990), the effect Y on the form factor is taken from extrapolation of the table 2.2. This value is multiplied with *k* in order to get the new form factor.

$$1 + k_F = 1 + k \cdot Y \qquad (2.13)$$

$$Fn = \frac{u}{\sqrt{gL}} \qquad (2.13)$$

*Table 2.2*

| Fn | Y | Fn | Y |
|---|---|---|---|
| 0.100 | 0.9300 | 0.35 | 0.5625 |
| 0.125 | 0.9395 | 0.40 | 0.3800 |
| 0.150 | 0.9513 | 0.45 | 0.2844 |
| 0.200 | 0.9500 | 0.50 | 0.2200 |
| 0.250 | 0.8744 | 0.60 | 0.1000 |
| 0.300 | 0.7500 | >0.8 | 0.0000 |

The appendage resistance can be determined from:

$$R_{APP} = 0.5 \cdot \rho \cdot u^2 \cdot S_{APP} \cdot (1 + k_2)_{eq} \cdot C_F \qquad (2.14)$$

where $1+k_2$ is the appendage resistance factor. The values of $1+k_2$ are given in table 2.3 for streamlined flow-oriented appendages.

*Table 2.3*

| Approximate $1+k_2$ values | |
|---|---|
| rudder behind skeg | 1.5-2.0 |
| rudder behind stern | 1.3-1.5 |
| twin-screw balance rudders | 2.8 |
| shaft barackets | 3.0 |
| skeg | 1.5-2.0 |
| strut bossings | 3.0 |
| hull bossings | 2.0 |
| shafts | 2.0-4.0 |
| stabilizer fins | 2.8 |
| dome | 2.7 |
| bilge keels | 1.4 |

The equivalent $1+k_2$ value for combination of appendages is determined from:

$$(1 + k_2)_{eq} = \frac{\sum (1 + k_2) S_{APP}}{\sum S_{APP}} \qquad (2.15)$$

In addition, the appendage resistance can be increased by the resistance of the bow thruster tunnel openings according to:

$$R_{THR} = 0.5 \cdot \rho \cdot u^2 \cdot \pi \cdot d^2 \cdot C_{BTO} \qquad (2.16)$$

where d is the tunnel diameter. The coefficient $C_{BTO}$ ranges from 0.003 to 0.012. For openings in the cylindrical part of the bulbous bow the lower figures should be used.

The calculation of wave resistance differs respectively to the Froude number. There are three different regions of *Fn* (*Fn>0.55, Fn<0.40, 0.40<Fn<0.55),* for each of which a different method is used. For fast ships (*Fn>0.55*) it is calculated as following:

$$R_{W-B} = c_{17} \cdot c_2 \cdot c_5 \cdot \nabla \cdot \rho \cdot g \cdot \exp\left\{m_3 \cdot F_n^d + m_4 \cos\left(\lambda \cdot F_n^{-2}\right)\right\} \tag{2.17}$$

where:

$$c_{17} = 6919.3 \cdot C_M^{-1.3346} \left(\nabla / L^3\right)^{2.00977} \left(L/B - 2\right)^{1.40692} \tag{2.18}$$

$$m_3 = -7.2035 \cdot \left(B/L\right)^{0.326869} \left(T/B\right)^{0.605375} \tag{2.19}$$

$$c_2 = \exp\left(-1.89\sqrt{c_3}\right) \tag{2.20}$$

$$c_5 = 1 - 0.8A_T / \left(B \cdot T \cdot C_M\right) \tag{2.21}$$

$$\lambda = 1.446C_P - 0.03\,L/B \tag{2.22}$$

when $L/B < 12$

$$\lambda = 1.446C_P - 0.36$$

when $L/B > 12$

$$d = -0.9 \tag{2.23}$$

$$c_3 = 0.56A_{BT}^{1.5} / \left\{BT\left(0.31\sqrt{A_{BT}} + T_F - h_B\right)\right\} \tag{2.24}$$

$$m_4 = c_{15}\, 0.4 \exp\left(-0.034F_n^{-3.29}\right) \tag{2.25}$$

$$c_{15} = -1.69385 \tag{2.26}$$

when $L^3/\nabla < 512$

$$c_{15} = -1.69385 + \left(L/\nabla^{1/3} - 8\right)/2.36$$

when $512 < L^3/\nabla < 1726.91$

$$c_{15} = 0$$

when $L^3/\nabla > 1726.91$

In these expressions $c_2$ is a parameter which accounts for the reduction of the wave resistance due to the action of bulbous bow, $c_5$ expresses the influence of a transom stern on the wave resistance. The coefficient $c_3$ determines the influence of the bulbous bow. The expression $A_T$ is the immersed part of the transverse area of the transom at zero speed and $h_B$ is the vertical position of the centre of $A_{BT}$ above the keel plane. The value of $h_B$ should not exceed $0.6T_F$.

For slower ships with low Froude number ($F_n<0.4$), wave resistance is calculated by the modified formula:

$$R_{W-A} = c_1 \cdot c_2 \cdot c_5 \cdot \nabla \cdot \rho \cdot g \cdot \exp\left\{m_1 \cdot F_n^d + m_4 \cos\left(\lambda \cdot F_n^{-2}\right)\right\} \tag{2.27}$$

where:

$$c_1 = 2223105 \cdot c_7^{3.78613} \left(T/B\right)^{1.07961} \left(90 - i_E\right)^{-1.37565} \tag{2.28}$$

$$c_7 = 0.229577 \cdot \left(B/L\right)^{0.33333} \tag{2.29}$$

when $B/L < 0.11$

$$c_7 = B/L$$

when $0.11 < B/L < 0.25$

$$c_7 = 0.5 - 0.0625\,L/B$$
when $B/L > 0.25$

$$m_1 = 0.0140407 \cdot L/T - 1.75254 \cdot \nabla^{1/3}/L - 4.79323\,B/L - c_{16} \qquad (2.30)$$

$$c_{16} = 8.07981 \cdot C_P - 13.8673 \cdot C_P^2 + 6.984388 \cdot C_P^3 \qquad (2.31)$$
when $C_P < 0.8$

$$c_{16} = 1.73014 - 8.07981 \cdot C_P$$
when $C_P > 0.8$

The half angle of entrance $i_E$ can be approximated by the following formula:

$$i_E = 1 + 89\exp\left\{-\left(L/B\right)^{0.80856}\left(1 - C_{WP}\right)^{0.30484}\left(1 - C_P - 0.0225lcb\right)^{0.6367}\left(L_R/B\right)^{0.34574} \left(100\nabla/L^3\right)^{0.16302}\right\} \qquad (2.32)$$

The calculation of the wave resistance for the speed range *0.40<Fn<0.55* is done with the interpolation formula:

$$R_W = R_{W-A,0.4} + \left(10F_n - 4\right)\left(R_{W-B,0.55} - R_{W-A,0.4}\right)/1.5 \qquad (2.33)$$

,where $R_{W-A,0.4}$ is the wave resistance for $F_n = 0.4$ and $R_{W-B,0.55}$ for $F_n = 0.55$ according to eq. (2.27) and (2.17) respectively.

The additional pressure resistance of a bulbous bow near the surface is determined from:

$$R_B = 0.11\exp\left(-3P_B^{-2}\right)F_{ni}^3 A_{BT}^{1.5}\,\rho g/\left(1 + F_{ni}^2\right) \qquad (2.34)$$

where the coefficient $P_B$ is a measure for the emergence of the bow and $F_{ni}$ is the Froude number based on the immersion:

$$P_B = 0.56\sqrt{A_{BT}}/\left(T_F - 1.5h_B\right) \qquad (2.35)$$

and

$$F_{ni} = V/\sqrt{g\left(T_F - h_B - 0.25\sqrt{A_{BT}}\right) + 0.15 \cdot u^2} \qquad (2.36)$$

The additional pressure resistance due to the immersed transom can be calculated as following:

$$R_{TR} = 0.5 \cdot \rho \cdot u^2 \cdot A_T \cdot c_6 \qquad (2.37)$$

The coefficient $c_6$ has been related to the Froude number based on the transom immersion:

$$c_6 = 0.2\left(1 - 0.2F_{nT}\right) \quad \text{when } F_{nT} < 5 \qquad (2.38)$$
$$c_6 = 0 \quad \text{when } F_{nT} \geq 5$$

where:

$$F_{nT} = V \Big/ \sqrt{2gA_T / (B + B \cdot C_{WP})} \tag{2.39}$$

Finally, the model-ship correlation resistance $R_A$ describes primarily the effect of the hull roughness and the still-air resistance. It can be calculated by the following equation:

$$R_A = 0.5 \cdot \rho \cdot u^2 \cdot S \cdot C_A \tag{2.40}$$

where $C_A$ is the correlation allowance coefficient approximated by:

$$C_A = 0.006(L + 100)^{-0.16} - 0.00205 + 0.003\sqrt{L/7.5} \cdot C_B^4 \cdot c_2 (0.04 - c_4) \tag{2.41}$$

with

$$\begin{aligned} c_4 &= T_F / L && \text{when } T_F / L \le 0.04 \\ c_4 &= 0.04 && \text{when } T_F / L > 0.04 \end{aligned} \tag{2.42}$$

In addition, $C_A$ might be increased due to the effect of a larger hull rougness than the standard. The ITTC-1978 formula can be used for roughness values greater than the standard $k_s = 150 \mu m$:

$$C_A = \left(0.105 \cdot k_s^{1/3} - 0.005579\right) \Big/ L^{1/3} \tag{2.43}$$

where $k_s$ and $L$ are given in metres.

### 2.2.2.3 Shallow water effect

As described in previous paragraphs, the model developed in this thesis includes the effect of maneuvering through shallow water on resistance. When a ship moves through shallow water there is an increase of the resistance due to confinement of the maximum depth of the sea. This increase of the resistance is expressed through the increase of the form factor (see eq. 2.44) based on the maximum depth. An estimation of the new form factor the following formula by Millward(1989,see also Kobayashi,1995) can be used:

$$k_{shallow} = k_{deep} + 0.644 \left( \frac{T}{h} \right)^{1.72} \tag{2.44}$$

where $k_{shallow}$ and $k_{deep}$ are form factors for resistance in shallow and deep water respectively.

## 2.2.3 Hydrodynamic Forces

### 2.2.3.1 Introduction

This paragraph describes the forces acting on the hull of the ship, due to the interaction of the ship with the surrounding fluid (i.e. sea, channel etc). As stated before, the hydrodynamic forces are functions of velocities and/or accelerations. The general form of the hydrodynamic forces is the following:

$$X = f\left(u, v, r, \dot{u}, \dot{v}, \dot{r}\right) \quad (2.45)$$

$$Y = f\left(u, v, r, \dot{u}, \dot{v}, \dot{r}\right) \quad (2.46)$$

$$N = f\left(u, v, r, \dot{u}, \dot{v}, \dot{r}\right) \quad (2.47)$$

$$K = f\left(u, v, r, \dot{u}, \dot{v}, \dot{r}\right) \quad (2.48)$$

The hydrodynamic forces of the hull can be calculated using the hydrodynamic derivatives. This method is described in the following sub-paragraph.

### 2.2.3.2 Hydrodynamic derivatives

The equations (2.45)-(2.48) are complex, hard to define and of low practical usage for calculations. In order to be used, they must be reduced to a useful mathematical form. They can be approximated by the Taylor expansion for functions with several variables. The form of the Taylor expansion for functions with a single variable is the following:

$$f(x) = f(x_1) + \Delta x \frac{df_x}{dx} + \frac{\Delta x^2}{2!}\frac{d^2 f_x}{dx^2} + \frac{\Delta x^3}{3!}\frac{d^3 f_x}{dx^3} + ... + + \frac{\Delta x^n}{n!}\frac{d^n f_x}{dx^n} \quad (2.49)$$

where :

| | |
|---|---|
| $f(x)$ | value of function at *x* close to $x_1$ |
| $f(x_1)$ | value of function at $x = x_1$ |
| $\Delta x$ | $= x - x_1$ |
| $\frac{d^n f_x}{dx^n}$ | n-th derivative of function evaluated at $x = x_1$ |

If $\Delta x$ is sufficiently small the higher order terms become too small and can be neglected. In general the smaller the $\Delta x$ gets the higher order terms can be neglected. The Taylor expansion for functions with two variables (i.e. x,y) has the following form:

$$f(x, y) = f(x_1, y_1) \quad (2.50)$$

$$
\begin{aligned}
&+\Delta x\frac{\partial f_{x,y}}{\partial x}+\Delta y\frac{\partial f_{x,y}}{\partial y}\\
&+\Delta x\Delta y\frac{\partial f_{x,y}}{\partial xy}+\frac{\Delta x^2}{2!}\frac{\partial^2 f_{x,y}}{\partial x^2}+\frac{\Delta y^2}{2!}\frac{\partial^2 f_{x,y}}{\partial y^2}\\
&+\frac{\Delta x^3}{3!}\frac{\partial^3 f_{x,y}}{\partial x^3}+\frac{\Delta y^3}{3!}\frac{\partial^3 f_{x,y}}{\partial y^3}+\ldots
\end{aligned}
$$

The application of this equation on the ship force $X$ gives:

$$
\begin{aligned}
X &= X_C + X_{\dot{u}}\Delta\dot{u} + X_{\dot{v}}\Delta\dot{v} + X_{\dot{r}}\Delta\dot{r} + X_{\dot{p}}\Delta\dot{p}\\
&+ X_u\Delta u + X_v\Delta v + X_r\Delta r + X_p\Delta p\\
&+ X_\phi\Delta\phi + X_\delta\Delta\delta\\
&+\frac{X_{\dot{u}^2}}{2}(\Delta\dot{u})^2+\frac{X_{\dot{v}^2}}{2}(\Delta\dot{v})^2+\frac{X_{\dot{r}^2}}{2}(\Delta\dot{r})^2+\frac{X_{\dot{p}^2}}{2}(\Delta\dot{p})^2\\
&+\frac{X_{u^2}}{2}(\Delta u)^2+\frac{X_{v^2}}{2}(\Delta v)^2+\frac{X_{r^2}}{2}(\Delta r)^2+\frac{X_{p^2}}{2}(\Delta p)^2\\
&+ X_{\dot{u}\dot{v}}\Delta\dot{u}\Delta\dot{v} + X_{\dot{u}\dot{r}}\Delta\dot{u}\Delta\dot{r} + X_{\dot{u}\dot{p}}\Delta\dot{u}\Delta\dot{p}\\
&+ X_{\dot{u}u}\Delta\dot{u}\Delta u + X_{\dot{u}v}\Delta\dot{u}\Delta v + X_{\dot{u}r}\Delta\dot{u}\Delta r + X_{\dot{u}p}\Delta\dot{u}\Delta p\\
&+ X_{\dot{u}\phi}\Delta\dot{u}\Delta\phi + X_{\dot{u}\delta}\Delta\dot{u}\Delta\delta + X_{\dot{u}\dot{r}}\Delta\dot{u}\Delta\dot{r}\\
&+\ldots
\end{aligned}
\tag{2.51}
$$

where the simplified derivative notation of SNAME is used, i.e $\partial X/\partial u = X_u$, $\partial Y/\partial uv = Y_{uv}$ and $\Delta u = u - u_m$. The expression $u_m$ is the value of the $u$ the measurements were made and typically the measurements are taken at a specific ship speed $u_m$, while all the other variables are equal to zero, as stated in the following:

$$u_m \neq 0 \tag{2.52}$$

$$\dot{u}_m = \dot{v}_m = \dot{r}_m = \dot{p}_m = v_m = r_m = p_m = \phi_m = \delta_m = 0 \tag{2.53}$$

By using eq. (2.53) in (2.51):

$$
\begin{aligned}
X &= X_C + X_{\dot{u}}\dot{u} + X_{\dot{v}}\dot{v} + X_{\dot{r}}\dot{r} + X_{\dot{p}}\dot{p}\\
&+ X_u\Delta u + X_v v + X_r r + X_p p\\
&+ X_\phi\phi + X_\delta\delta\\
&+\frac{X_{\dot{u}^2}}{2}\dot{u}^2+\frac{X_{\dot{v}^2}}{2}\dot{v}^2+\frac{X_{\dot{r}^2}}{2}\dot{r}^2+\frac{X_{\dot{p}^2}}{2}\dot{p}^2\\
&+\frac{X_{u^2}}{2}(\Delta u)^2+\frac{X_{v^2}}{2}v^2+\frac{X_{r^2}}{2}r^2+\frac{X_{p^2}}{2}p^2\\
&+ X_{\dot{u}\dot{v}}\dot{u}\dot{v} + X_{\dot{u}\dot{r}}\dot{u}\dot{r} + X_{\dot{u}\dot{p}}\dot{u}\dot{p}
\end{aligned}
\tag{2.54}
$$

$$
\begin{aligned}
&+ X_{\dot{u}u}\dot{u}\Delta u + X_{\dot{u}v}\dot{u}v + X_{\dot{u}r}\dot{u}r + X_{\dot{u}p}\dot{u}p \\
&+ X_{\dot{u}\phi}\dot{u}\phi + X_{\dot{u}\delta}\dot{u}\delta + X_{\dot{u}\dot{r}}\dot{u}\dot{r} \\
&+ \ldots
\end{aligned}
$$

The same expansion can be made for the remaining forces and moments acting on the ship, $Y$, $N$ and $K$. The derivative terms on the right hand side of this equation (i.e. $X_u$, $Y_{uv}$) are called hydrodynamic derivatives. As can be seen by equation (2.54), by knowing the value of them and of the velocities and accelerations, the hydrodynamic force can be calculated. In general, terms over the fourth order are neglected, as their contribution to the total force is usually insignificant.

The decision of which of them should be chosen is not standard and may vary. Depending on the desired precision or simplicity some terms can be neglected while others are essential.

The values of the hydrodynamic derivatives can be obtained experimentally for a specific ship.

### 2.2.3.3 Estimation Methods

Obtaining the values of the hydrodynamic derivatives experimentally would make this method inefficient for generic simulation purposes. Even for a specific ship, the calculation of the hydrodynamic forces would require complex and time-consuming experiments. Thus, many estimation methods have been developed over the past years. These methods provide the required formulas for the estimations and also make a proposition for which derivatives should be chosen.

The model of this thesis includes five methods, appearing in:

- Tae-II Lee, Kyoung-Soo Ahn, Hyoung-Suk Lee, On Deuk-Joon Yum, (2003) An Empirical Prediction Of Hydrodynamic Coefficients For Modern Ship Hulls, MARSIM 2003, Kanazawa, Japan.
- Inoue, S., Hirano, M., Kijima, K., and Takashina, J. (1981). A Practical Calculation Method of Ship Manoeuvring Motion. International Shipbuilding Progress, 28(325), 207–222.
- Clarke, D., Gedling, P. and Hine, G. (1983). The application of manoeuvring criteria in hull design using linear theory. The Naval Architect, pp. 45–68.
- Michael D Woodward, David Clarke, Mehmet Atlar, (2003) On The Manoeuvring Prediction Of Pod Driven Ships, MARSIM 2003, Kanazawa, Japan.
- Adapted method of Inoue et al by ITTC

The formulas of each method are tabulated in tables 2.7-2.11 respectively. The first column of each table includes the formulas and the second the variables included in each formula. All the results are non-dimensional values of derivatives and in order to be used in equation 2.58, they must be multiplied with the following expression:

$$\text{forces } (X,Y): \qquad \frac{1}{2}\cdot\rho\cdot L\cdot T\cdot u^2 \qquad (2.55)$$

$$\text{moments } (N,K): \quad \frac{1}{2}\cdot\rho\cdot L^2\cdot T\cdot u^2 \qquad (2.56)$$

*Table 2.7 – method by (T.I. Lee et al, 2003)*

| Hydrodynamic derivative formula | Variables |
|---|---|
| $X'_{vv}=0.223-0.011\cdot\frac{L}{T}$ | L, T |
| $X'_{rr}=0.038-0.001\cdot\frac{L}{T}$ | L, T |
| $X'_{vr}=0.12\cdot c_B\cdot\frac{B}{T}-0.018\cdot\frac{L}{T}+0.443$ | L, B, T, $c_B$ |
| $Y'_v=-0.145-2.25\cdot\frac{T}{L}-0.2\cdot\Delta_{SR}$<br>$\Delta_{SR}=\frac{P_{SR}-S_R}{P_{SR}}$<br>$P_{SR}=28.7\cdot\frac{c_B\cdot B\cdot T}{L^2}+0.54$<br>$S_R=\frac{B_{P07}}{B_{PS}}$ | L, B, T, $c_B$,<br>$B_{PS}$: half breadth at the height of propeller shaft in 2.0 station,<br>$B_{P07}$: half breadth at the height of 0.7R(propeller radius) in 2.0 station |
| $Y'_r=\frac{2\cdot c_B\cdot B}{L}-0.282+0.1\cdot\Delta_{SR}+\left(0.0086\cdot\Delta_{B/L}+0.004\right)\cdot\frac{L}{T}$<br>$\Delta_{SR}=\frac{P_{SR}-S_R}{P_{SR}}$<br>$P_{SR}=28.7\cdot\frac{c_B\cdot B\cdot T}{L^2}+0.54 \qquad \Delta_{B/L}=\frac{0.18-\frac{B}{L}}{0.18}$<br>$S_R=\frac{B_{P07}}{B_{PS}}$ | L, B, T, $c_B$,<br>$B_{PS}$: half breadth at the height of propeller shaft in 2.0 station,<br>$B_{P07}$: half breadth at the height of 0.7R(propeller radius) in 2.0 station. |
| $Y'_{vvv}=1.281+0.031\cdot\frac{L}{T}$ | L, T |
| $Y'_{rrr}=0.029\cdot c_B\cdot\frac{B}{T}-0.004\cdot\frac{L}{T}$ | L, B, T, $c_B$ |
| $Y'_{rvv}=0.628\cdot c_B\cdot\frac{B}{T}-0.066\cdot\frac{L}{T}$ | L, B, T, $c_B$ |
| $Y'_{vrr}=0.4+0.007\cdot\frac{L}{T}$ | L, T |
| $N'_v=-\left(0.222+0.1\cdot\Delta_{SR}\right)+0.00484\cdot\frac{L}{T}$<br>$\Delta_{SR}=\frac{P_{SR}-S_R}{P_{SR}}$<br>$P_{SR}=28.7\cdot\frac{c_B\cdot B\cdot T}{L^2}+0.54$<br>$S_R=\frac{B_{P07}}{B_{PS}}$ | L, B, T, $c_B$,<br>$B_{PS}$: half breadth at the height of propeller shaft in 2.0 station,<br>$B_{P07}$: half breadth at the height of 0.7R(propeller radius) in 2.0 station. |

| | |
|---|---|
| $N'_r = -\left(0.0424 - 0.03 \cdot \Delta_{SR}\right) + \left(0.004 \cdot \Delta_{c_B} + 0.00027\right) \cdot \frac{L}{T}$ <br> $\Delta_{SR} = \frac{P_{SR} - S_R}{P_{SR}}$ <br> $P_{SR} = 28.7 \cdot \frac{c_B \cdot B \cdot T}{L^2} + 0.54$ <br> $S_R = \frac{B_{P07}}{B_{PS}}$ <br> $\Delta_{c_B} = \frac{P_{c_B} - c_B}{P_{c_B}}$ <br> $P_{c_B} = 1.12 \cdot \frac{T}{L} + 0.735$ | L, B, T, $c_B$, <br> $B_{PS}$: half breadth at the height of propeller shaft in 2.0 station, <br> $B_{P07}$: half breadth at the height of 0.7R(propeller radius) in 2.0 station. |
| $N'_{vvv} = 0.188 - 0.01 \cdot \frac{L}{T}$ | L, T |
| $N'_{rrr} = 0.014 \cdot c_B \cdot \frac{B}{T} - 0.002 \cdot \frac{L}{T}$ | L, B, T, $c_B$ |
| $N'_{rvv} = 0.178 \cdot c_B \cdot \frac{B}{T} - 0.037 \cdot \frac{L}{T}$ | L, B, T, $c_B$ |
| $N'_{vrr} = 0.158 - 0.005 \cdot \frac{L}{T}$ | L, T |

*Table 2.8 – method by (Inoue S. et al, 1981)*

| Hydrodynamic derivative formula | Variables |
|---|---|
| $Y'_v = -\left(\pi \cdot \frac{T}{L} + 1.4 \cdot c_B \cdot \frac{B}{L}\right) \cdot \left(1 + \frac{2 \cdot t}{3 \cdot T}\right)$ | L, B, T, $c_B$, <br> t: trim by stern |
| $Y'_r = \frac{\pi}{2} \cdot \frac{T}{L} \cdot \left(1 + 0.8 \cdot \frac{t}{T}\right)$ | L, T, <br> t: trim by stern |
| $Y'_{v\|v\|} = 0.09 - 6.5 \cdot \left(1 - c_B\right) \cdot \frac{T}{B}$ | B, T, $c_B$ |
| $Y'_{v\|r\|} = -0.44 + 1.78 \cdot \left(1 - c_B\right) \cdot \frac{T}{B}$ | B, T, $c_B$ |
| $N_v{}' = -\frac{2 \cdot T}{L} \cdot \left(1 - \frac{0.27}{\ell_\beta} \cdot \frac{t}{T}\right)$ <br> $\ell_\beta = \frac{2 \cdot T}{\pi \cdot T + 1.4 \cdot c_B \cdot B}$ | L, B, T, $c_B$, <br> t: trim by stern |
| $N'_r = -\left(0.54 \cdot \frac{2 \cdot T}{L} - \left(\frac{2 \cdot T}{L}\right)^2\right) \cdot \left(1 + 0.3 \cdot \frac{t}{T}\right)$ | L, T, <br> t: trim by stern |
| $N'_{r\|r\|} = \begin{cases} -0.060 & \frac{c_B \cdot B}{L} < 0.06 \\ -0.146 + 1.8 \cdot \frac{c_B \cdot B}{L} - 6 \cdot \left(\frac{c_B \cdot B}{L}\right)^2 & 0.06 \le \frac{c_B \cdot B}{L} \le 0.2 \\ -0.026 & \frac{c_B \cdot B}{L} > 0.2 \end{cases}$ | L, B, $c_B$ |
| $N'_{\dot{r}vv} = -0.2$ | N/A |

*Table 2.9 – method by (Clarke D. et al,1983)*

| Hydrodynamic derivative formula | Variables |
|---|---|
| $Y'_v = -\pi\left(\frac{T}{L}\right)^2 \cdot \left(1+0.40\cdot c_B \cdot \frac{B}{T}\right)\cdot\left(1+\frac{2\cdot t}{3\cdot T}\right)$ | L, B, T, $c_B$,<br>t: trim by stern |
| $Y'_v = -\pi\left(\frac{T}{L}\right)^2 \cdot \left(-0.5+2.2\cdot\frac{B}{L}-0.08\cdot\frac{B}{T}\right)\cdot\left(1+0.80\cdot\frac{t}{T}\right)$ | L, B, T<br>t: trim by stern |
| $Y'_{\dot{v}} = -\pi\left(\frac{T}{L}\right)^2 \cdot \left(1+0.16\cdot c_B \cdot \frac{B}{T}-5.1\cdot\left(\frac{B}{L}\right)^2\right)$ | L, B, T, $c_B$ |
| $N'_{\dot{v}} = -\pi\left(\frac{T}{L}\right)^2 \cdot \left(1.1\cdot\frac{B}{L}-0.041\cdot\frac{B}{T}\right)$ | L, B, T |
| $N'_{\dot{r}} = -\pi\left(\frac{T}{L}\right)^2 \cdot \left(\frac{1}{12}+0.017\cdot c_B \cdot \frac{B}{T}-0.33\cdot\frac{B}{L}\right)$ | L, B, T, $c_B$ |
| $Y'_{\dot{r}} = -\pi\left(\frac{T}{L}\right)^2 \cdot \left(0.67\cdot\frac{B}{L}-0.0033\cdot\left(\frac{B}{T}\right)^2\right)$ | L, B, T |
| $N'_v = -\pi\left(\frac{T}{L}\right)^2 \cdot \left(-0.5+2.4\cdot\frac{T}{L}\right)\cdot\left(1-\frac{0.27}{\ell_\beta}\cdot\frac{t}{T}\right)$<br>$\ell_\beta = \frac{2\cdot T}{\pi\cdot T+1.4\cdot c_B\cdot B}$ | L, B, T, $c_B$,<br>t: trim by stern |
| $N'_r = -\pi\left(\frac{T}{L}\right)^2 \cdot \left(0.25+0.039\cdot\frac{B}{T}-0.56\cdot\frac{B}{L}\right)\cdot\left(1+0.3\cdot\frac{t}{T}\right)$ | L, B, T,<br>t: trim by stern |

*Table 2.10 – method by (Woodward M.D. et al, 2003)*

| Hydrodynamic derivative formula | Variables |
|---|---|
| $Y'_v = -\pi\left(\frac{T}{L}\right)^2 \cdot \left(1+6.4\cdot\frac{B}{L}+4.9\cdot\frac{\sigma_A\cdot B}{L}-108.3\cdot\frac{T\cdot B}{L^2}\right)$<br>$\sigma_A = \frac{1-c_{WPa}}{1-c_{Pa}}$ | L, B, T, $c_{WPa}$, $c_{Pa}$ |
| $Y'_r = m' - \pi\left(\frac{T}{L}\right)^2 \cdot \left(0.7+2.1\cdot\frac{B}{L}+0.6\cdot\frac{c_B\cdot B}{T}-0.4\cdot\frac{D_{pr}}{T}\right)$ | L, B, T, $c_B$, $D_{pr}$,<br>m' |
| $Y'_{\dot{v}} = m' - \pi\left(\frac{T}{L}\right)^2 \cdot \left(0.66-0.43\cdot\frac{B}{T}-3.62\cdot\frac{\sigma_A\cdot B}{L}+1.60\cdot\frac{c_B\cdot B}{T}\right)$<br>$\sigma_A = \frac{1-c_{WPa}}{1-c_{Pa}}$ | L, B, T, $c_{WPa}$, $c_{Pa}$, $c_B$, m’ |
| $N'_v = -\pi\left(\frac{T}{L}\right)^2 \cdot \left(0.8-1.8\cdot\frac{\sigma_A\cdot B}{L}-0.3\cdot\frac{D_{pr}}{T}\right)$<br>$\sigma_A = \frac{1-c_{WPa}}{1-c_{Pa}}$ | L, B, T, $c_{WPa}$, $c_{Pa}$, $D_{pr}$ |

| | |
|---|---|
| $N'_r = m' \cdot x'_g$ <br> $-\pi\left(\frac{T}{L}\right)^2 \cdot \left(0.4 - 1.7 \cdot \frac{c_B \cdot B}{L} + 1.3 \cdot \frac{\sigma_A \cdot B}{L} - 1.7 \cdot \frac{c_B \cdot x'_g \cdot B}{T}\right)$ <br> $\sigma_A = \frac{1 - c_{WPa}}{1 - c_{Pa}}$ | L, B, T, $c_{WPa}$, $c_{Pa}$, $c_B$, LCG |
| $N_{\dot{r}} = I'_{zz} - \pi\left(\frac{T}{L}\right)^2 \cdot \left(0.04 - 0.02 \cdot \frac{B}{T} - 0.38 \cdot \frac{\sigma_A \cdot B}{L} + 0.08 \cdot \frac{c_B \cdot B}{T}\right)$ <br> $\sigma_A = \frac{1 - c_{WPa}}{1 - c_{Pa}}$ | L, B, T, $c_{WPa}$, $c_{Pa}$, $c_B$, $I_{zz}$’ |

*Table 2.11 – method by ITTC*

| Hydrodynamic derivative formula | Variables |
|---|---|
| $Y'_v = -\frac{\pi}{2}\mathrm{k} + 1.4\mathrm{C_B}\frac{B}{L}$ | k,L,B,$C_B$ |
| $Y'_r = -\frac{\pi}{4}\mathrm{k}$ | k |
| $Y'_{vv} = 2.5\frac{T}{B}(1 - C_B) + \frac{1}{2}$ | B,T,$C_B$ |
| $Y'_{rr} = 0.343\frac{T}{B}C_B - 0.07$ | B,T,$C_B$ |
| $Y'_{vvr} = -114\left(\frac{T}{B}C_B\right)^2 + 62.12\frac{T}{B}(1 - C_B) - 8.20$ | B,T,$C_B$ |
| $Y'_{vrr} = -5.95\frac{T}{B}(1 - C_B)$ | B,T,$C_B$ |
| $N'_v = -\mathrm{k}$ | k |
| $N'_r = -(0.54k - k^2)$ | k |
| $N'_{vv} = 78\left(\frac{T}{B}(1 - C_B)\right)^2 - 19.0\frac{T}{B}(1 - C_B) - 8.20$ | B,T,$C_B$ |
| $N'_{rr} = 0.473\frac{B}{L}C_B - 0.089$ | L,B,$C_B$ |
| $N'_{vvr} = -120\left(\frac{B}{L}C_B\right)^2 + 35.22\frac{B}{L}C_B - 2.72$ | L,B,$C_B$ |
| $N'_{vrr} = 0.50\frac{T}{B}C_B - 0.05$ | B,T,$C_B$ |

### 2.2.3.4 Shallow water effect

Similarly to the calm water resistance, maneuvering through shallow or confined waters affects the residue hydrodynamic forces of the hull. The effect is not as simple as the increase of the form factor in resistance. In shallow waters the value of some of the hydrodynamics derivatives must be corrected (Ankudinov, 1990). The following correction should be made for $1.085 < h/T < 5$ and $C_B \leq 0.85$, where $h$ is the maximum sea depth:

- Linear sway-yaw terms:

$$\frac{Y'_{\dot{v}}}{\left(Y'_{\dot{v}}\right)_\infty} = gv\,; \quad \frac{Y'_{\dot{r}}}{\left(Y'_{\dot{r}}\right)_\infty} = gv\,; \quad \frac{N'_{\dot{v}}}{\left(N'_{\dot{v}}\right)_\infty} = gv$$
$$\frac{N'_{\dot{r}}}{\left(N'_{\dot{r}}\right)_\infty} = gnr\,; \quad \frac{Y'_{v}}{\left(Y'_{v}\right)_\infty} = fyv\,; \quad \frac{Y'_{r}}{\left(Y'_{r}\right)_\infty} = fyr \tag{2.57}$$
$$\frac{N'_{v}}{\left(N'_{v}\right)_\infty} = fnv\,; \quad \frac{N'_{r}}{\left(N'_{r}\right)_\infty} = fnr$$

- Non-linear sway-yaw terms:

$$\frac{Y'_{v|v|}}{\left(Y'_{v|v|}\right)_\infty} = \frac{9}{4} fnv - \frac{5}{4}\,; \qquad \frac{Y'_{r|r|}}{\left(Y'_{r|r|}\right)_\infty} = fnr \tag{2.58}$$
$$\frac{Y'_{vrr}}{\left(Y'_{vrr}\right)_\infty} = \frac{Y'_{r|v|}}{\left(Y'_{r|v|}\right)_\infty} = fyv$$
$$\frac{N'_{v|v|}}{\left(N'_{v|v|}\right)_\infty} = \frac{9}{4} fnv - \frac{5}{4}\,; \qquad \frac{N'_{r|r|}}{\left(N'_{r|r|}\right)_\infty} = gv$$
$$\frac{N'_{vrr}}{\left(N'_{vrr}\right)_\infty} = \frac{N'_{r|v|}}{\left(N'_{r|v|}\right)_\infty} = gnr$$

- Surge terms:

$$\frac{X'_{\dot{u}}}{\left(X'_{\dot{u}}\right)_\infty} = gv\,; \qquad \frac{X'_{rr}}{\left(X'_{rr}\right)_\infty} = gnr \tag{2.59}$$
$$\frac{X'_{vr}}{\left(X'_{vr}\right)_\infty} = frv\,; \qquad \frac{X'_{vv}}{\left(X'_{vv}\right)_\infty} = frv$$

with

$$gv = K_0 + \frac{2}{3} K_1 \frac{B_1}{T} + \frac{8}{15} K_2 \left(\frac{B_1}{T}\right)^2 \tag{2.60}$$

$$gnr = K_0 + \frac{8}{15} K_1 \frac{B_1}{T} + \frac{40}{105} K_2 \left(\frac{B_1}{T}\right)^2 \tag{2.61}$$

$$fyv = 1.5\, fnv - 0.5 \tag{2.62}$$

$$fyr = K_0 + \frac{2}{5} K_1 \frac{B_1}{T} + \frac{24}{105} K_2 \left(\frac{B_1}{T}\right)^2 \tag{2.63}$$

$$fnv = K_0 + K_1 \frac{B_1}{T} + K_2 \left( \frac{B_1}{T} \right)^2 \qquad (2.64)$$

$$fnr = K_0 + K_1 \frac{B_1}{T} + \frac{1}{3} K_2 \left( \frac{B_1}{T} \right)^2 \qquad (2.65)$$

where:

$$K_0 = 1 + \frac{0.0775}{F^2} - \frac{0.0110}{F^3} + \frac{0.000068}{F^5} \qquad (2.63)$$

$$K_1 = -\frac{0.0643}{F} + \frac{0.0724}{F^2} - \frac{0.0113}{F^3} + \frac{0.0000765}{F^5} \qquad (2.64)$$

$$K_2 = \frac{0.0342}{F}; \qquad \text{for } \frac{B}{T} > 4 : K_2 = \frac{0.137}{F} \frac{T}{B} \qquad (2.65)$$

with $F = \frac{h}{T} - 1; \qquad B_1 = C_B B \left( 1 + \frac{B}{L} \right)^2$ (2.66)

## 2.2.4 Propeller Forces

### 2.2.4.1 Introduction

This paragraph describes the forces acting on the ship due to the operation of the propeller. The only force that the propeller produces is the thrust, which is always on the x-axis, part of X total force and positive for "pushing" the ship ahead. Apart from the thrust another interesting expression of the propeller’s operation is the torque acting on the main shaft.

Practically, the force $X_P$ is not the whole thrust that the propeller produces but a part of due to hydrodynamic losses that depend on the geometry of the hull and the appendages. This is expressed through the thrust-deduction factor $t$ and the propeller forces is given by:

$$X_P = T(1 - t_R) \qquad (2.67)$$

### 2.2.4.2 Mathematical Model

The method that has been followed (M.W.C. Oosterveld, 1975) for calculation of the thrust and torque, is based on the Wageningen B-screw series. The open-water characteristics are obtained from open-water test results, for about 120 propellers. They are given in a conventional way in the form of thrust and torque coefficients $K_T$ and $K_Q$ respectively.

$$K_T = \frac{T}{\rho n^2 D_P^4} \qquad (2.68)$$

$$K_Q = \frac{Q}{\rho n^2 D_P^5} \qquad (2.69)$$

where $n$ is the revolutions of the propeller per second and $D_P$ is the propeller diameter. In order to calculate the trhust and the torque, firstly the values of the coefficient must be obtained. They are given by the polynomials in the following equations.

$$K_T = \sum C_{s,t,u,v} \cdot (J)^s \cdot (P/D)^t \cdot (A_E/A_0)^u \cdot (z)^v \qquad (2.70)$$

$$K_Q = \sum C_{s,t,u,v} \cdot (J)^s \cdot (P/D)^t \cdot (A_E/A_0)^u \cdot (z)^v \qquad (2.71)$$

where $s,t,u,v$ are the exponents of the above figures, $J$ is the advance ratio (given by eq. (2.72)), $P/D$ is the pitch diameter ratio of the propeller, $A_E/A_O$ is the expanded area ratio of the propeller and $z$ is the number of fins. The values of the coefficient $C_{s,t,u,v}$ and of $s,t,u,v$ are given in tables 2.12 and 2.13 for the thrust and torque respectively. The advance ratio is calculated by:

$$J = \frac{V_A}{nD_P} \qquad (2.72)$$

where $V_A$ is the velocity of advance.

*Table 2.12 – coefficient and power values for $K_T$*

| Thrust coefficient $K_T$ | | | | |
|---|---|---|---|---|
| Cs,t,u,v | s | t | u | v |
| | (J) | (P/D) | ($A_E/A_O$) | (Z) |
| +0.00880496 | 0 | 0 | 0 | 0 |
| -0.204554 | 1 | 0 | 0 | 0 |
| +0.166351 | 0 | 1 | 0 | 0 |
| +0.158114 | 0 | 2 | 0 | 0 |
| -0.147581 | 2 | 0 | 1 | 0 |
| -0.481497 | 1 | 1 | 1 | 0 |
| +0.415437 | 0 | 2 | 1 | 0 |
| +0.0144043 | 0 | 0 | 0 | 1 |
| -0.0530054 | 2 | 0 | 0 | 1 |
| +0.0143481 | 0 | 1 | 0 | 1 |
| +0.0606826 | 1 | 1 | 0 | 1 |
| -0.0125894 | 0 | 0 | 1 | 1 |
| +0.0109689 | 1 | 0 | 1 | 1 |
| -0.133698 | 0 | 3 | 0 | 0 |
| +0.00638407 | 0 | 6 | 0 | 0 |
| -0.00132718 | 2 | 6 | 0 | 0 |
| +0.168496 | 3 | 0 | 1 | 0 |
| -0.0507214 | 0 | 0 | 2 | 0 |
| +0.0854559 | 2 | 0 | 2 | 0 |
| -0.0504475 | 3 | 0 | 2 | 0 |
| +0.010465 | 1 | 6 | 2 | 0 |
| -0.00648272 | 2 | 6 | 2 | 0 |
| -0.00841728 | 0 | 3 | 0 | 1 |
| +0.0168424 | 1 | 3 | 0 | 1 |
| -0.00102296 | 3 | 3 | 0 | 1 |
| -0.0317791 | 0 | 3 | 1 | 1 |
| +0.018604 | 1 | 0 | 2 | 1 |
| -0.00410798 | 0 | 2 | 2 | 1 |
| -0.000606848 | 0 | 0 | 0 | 2 |
| -0.0049819 | 1 | 0 | 0 | 2 |
| +0.0025983 | 2 | 0 | 0 | 2 |
| -0.000560528 | 3 | 0 | 0 | 2 |
| -0.00163652 | 1 | 2 | 0 | 2 |
| -0.000328787 | 1 | 6 | 0 | 2 |
| +0.000116502 | 2 | 6 | 0 | 2 |
| +0.000690904 | 0 | 0 | 1 | 2 |
| +0.00421749 | 0 | 3 | 1 | 2 |
| +0.0000565229 | 3 | 6 | 1 | 2 |
| -0.00146564 | 0 | 3 | 2 | 2 |

*Table 2.13 – coefficient and power values for $K_Q$*

| Torque coefficient $K_Q$ | | | | |
|---|---|---|---|---|
| Cstuv | s | t | u | v |
| | (J) | (P/D) | ($A_E/A_O$) | (Z) |
| +0.00379368 | 0 | 0 | 0 | 0 |
| +0.00886523 | 2 | 0 | 0 | 0 |
| -0.032241 | 1 | 1 | 0 | 0 |
| +0.00344778 | 0 | 2 | 0 | 0 |
| -0.040881 | 0 | 1 | 1 | 0 |
| -0.108009 | 1 | 1 | 1 | 0 |
| -0.0885381 | 2 | 1 | 1 | 0 |
| +0.188561 | 0 | 2 | 1 | 0 |
| -0.00370871 | 1 | 0 | 0 | 1 |
| +0.00513696 | 0 | 1 | 0 | 1 |
| +0.0209449 | 1 | 1 | 0 | 1 |
| +0.00474319 | 2 | 1 | 0 | 1 |
| -0.00723408 | 2 | 0 | 1 | 1 |
| +0.00438388 | 1 | 1 | 1 | 1 |
| -0.0269403 | 0 | 2 | 1 | 1 |
| +0.0558082 | 3 | 0 | 1 | 0 |
| +0.0161886 | 0 | 3 | 1 | 0 |
| +0.00318086 | 1 | 3 | 1 | 0 |
| +0.015896 | 0 | 0 | 2 | 0 |
| +0.0471729 | 1 | 0 | 2 | 0 |
| +0.0196283 | 3 | 0 | 2 | 0 |
| -0.0502782 | 0 | 1 | 2 | 0 |
| -0.030055 | 3 | 1 | 2 | 0 |
| +0.0417122 | 2 | 2 | 2 | 0 |
| -0.0397722 | 0 | 3 | 2 | 0 |
| -0.00350024 | 0 | 6 | 2 | 0 |
| -0.0106854 | 3 | 0 | 0 | 1 |
| +0.00110903 | 3 | 3 | 0 | 1 |
| -0.000313912 | 0 | 6 | 0 | 1 |
| +0.0035985 | 3 | 0 | 1 | 1 |
| -0.00142121 | 0 | 6 | 1 | 1 |
| -0.00383637 | 1 | 0 | 2 | 1 |
| +0.0126803 | 0 | 2 | 2 | 1 |
| -0.00318278 | 2 | 3 | 2 | 1 |
| +0.00334268 | 0 | 6 | 2 | 1 |
| -0.00183491 | 1 | 1 | 0 | 2 |
| +0.000112451 | 3 | 2 | 0 | 2 |
| -0.0000297228 | 3 | 6 | 0 | 2 |
| +0.000269551 | 1 | 0 | 1 | 2 |
| +0.00083265 | 2 | 0 | 1 | 2 |
| +0.00155334 | 0 | 2 | 1 | 2 |
| +0.000302683 | 0 | 6 | 1 | 2 |
| -0.0001843 | 0 | 0 | 2 | 2 |
| -0.000425399 | 0 | 3 | 2 | 2 |
| +0.0000869243 | 3 | 3 | 2 | 2 |
| -0.0004659 | 0 | 6 | 2 | 2 |
| +0.0000554194 | 1 | 6 | 2 | 2 |

The determination of the polynomials for $K_T$ and $K_Q$ was done for Reynolds number $2x10^6$. Thus for *Rn* greater than this value a correction should be made in the coefficients according to the following:

$$\begin{aligned}\Delta K_{\mathrm{T}} = {} & 0.000353485 \\ & - 0.00333758(A_E/A_O)J^2 \\ & - 0.00478125(A_E/A_O)(P/D)J \\ & + 0.000257792(\log R_n - 0.301)^2(A_E/A_O)J^2 \\ & + 0.0000643192(\log R_n - 0.301)(P/D)^6 J^2 \\ & - 0.0000110636(\log R_n - 0.301)^2(P/D)^6 J^2 \\ & - 0.0000276305(\log R_n - 0.301)^2 z(A_E/A_O)J^2 \\ & + 0.0000954(\log R_n - 0.301)z(A_E/A_O)(P/D)J \\ & + 0.0000032049(\log R_n - 0.301)z^2(A_E/A_O)(P/D)^3 J\end{aligned} \tag{2.73}$$

$$\begin{aligned}\Delta K_Q = {} & -0.000591412 \\ & + 0.00696898(P/D) \\ & - 0.0000666654(P/D)^6 z \\ & + 0.0160818(A_E/A_O)^2 \\ & - 0.000938091(\log R_n - 0.301)(P/D) \\ & - 0.00059593(\log R_n - 0.301)(P/D)^2 \\ & + 0.0000782099(\log R_n - 0.301)^2(P/D)^2 \\ & + 0.0000052199(\log R_n - 0.301)z(A_E/A_O)J^2 \\ & - 0.00000088528(\log R_n - 0.301)^2 z(A_E/A_O)(P/D)J \\ & + 0.0000230171(\log R_n - 0.301)z(P/D)^6 \\ & - 0.00000184341(\log R_n - 0.301)^2 z(P/D)^6 \\ & - 0.00400252(\log R_n - 0.301)(A_E/A_O)^2 \\ & + 0.000220915(\log R_n - 0.301)^2(A_E/A_O)^2\end{aligned} \tag{2.74}$$

The Reynolds number included in these equations is the *Re* at 0.75R of the blade and given by:

$$\mathrm{Re}_{0.75R} = \frac{C_{0.75R}\sqrt{V_A^2 + (0.75\pi \cdot \mathrm{n} \cdot \mathrm{D_P})^2}}{\nu} \tag{2.75}$$

where $c_{0.75R}$ is the chord length of the blade at 0.75R and $\nu$ the kinematical viscosity.

## 2.2.5 Rudder Forces

The rudder plays a very important role in the maneuverability of the ship. The rudder doesn't turn the ship by itself but positions the hull in such a way, so the flow of the surrounding water will turn it. The rudder forces and moments including the hydrodynamic forces and moments induced on the ship by the rudder action are the following (Inoue, 1981):

$$X_R = -F_N \sin\delta \tag{2.76}$$

$$Y_R = -(1 + a_H) F_N \cos\delta \tag{2.77}$$

$$N_R = -(1 + a_H) x_R F_N \cos\delta \tag{2.78}$$

$$K_R = (1 + a_H) z_R F_N \cos\delta \tag{2.79}$$

where $F_N$ is the normal rudder force, $\delta$ is the rudder angle, $x_R$ and $z_R$ are the longitudinal and vertical distance of the rudder's center of gravity(c.o.g.) from the c.o.g of the ship respectively, $\lambda$ is the rudder aspect ratio and $a_H$ is a rudder-hull interaction coefficient based on experimental results that can be approximated by the following formula (T.I. Lee, 2003):

$$a_H = 40\nabla / L^3 \tag{2.80}$$

The normal rudder force is the component force on the c.o.g. of the rudder and is given by:

$$F_N = \frac{1}{2}\rho \frac{6.13}{\lambda + 2.25} \mathrm{A_R} \mathrm{U_R^2} \sin a_R \tag{2.81}$$

where $\rho$ is the water density, $A_R$ is the mid-section area of the rudder, $U_R$ and $a_R$ are the effective rudder inflow velocity and angle respectively. The velocity $U_R$ is the component velocity of $u_R$ and $v_R$, which are the axial and lateral effective rudder inflow velocities respectively. Thus:

$$\mathrm{U_R} = \sqrt{u_R^2 + v_R^2} \tag{2.82}$$

The axial velocity is given by (T.I. Lee, 2003):

$$\mathrm{u_R} = \varepsilon u_P \sqrt{n_H \left[1 + \kappa\left(\sqrt{1 + \frac{8\mathrm{K_T}}{\pi J^2}} - 1\right)\right]^2 + 1 - n_H} \tag{2.83}$$

$$\kappa = 0.6/\varepsilon \tag{2.84}$$

$$n_H = \frac{D_P}{H} \tag{2.85}$$

$$\mathrm{u_p} = u(1 - w_P) \tag{2.86}$$

where $D_P$ is the propeller diameter, $H$ is the rudder height and $\varepsilon$ represents the wake ratio between the propeller and the rudder and can be approximated by the empirical formula:

$$\varepsilon = 2.71 C_B \frac{B}{L} + 0.92 \tag{2.87}$$

The lateral component of the effective rudder inflow velocity $v_R$ is given by (Eloot K, 2004):

$$\mathrm{v_R} = \delta_0 u_R + \gamma(v + x'_R \cdot r) \tag{2.88}$$

Taking the flow rectifying effect into consideration $a_R$ can be expressed in the following form (Inoue, 1981):

$$\mathrm{a_R} = \delta + \delta_0 - \gamma \beta'_R \tag{2.89}$$

where $\beta'_R$ is defined as $\beta'_R = \beta - 2\mathrm{x'_R} r'$. In this expression *r'* is the non-dimensioned yaw turn rate and $\beta$ is the drift angle given by:

$$\beta = a\sin\left(\frac{v}{\sqrt{u^2 + v^2}}\right) \tag{2.90}$$

The expression $\delta_0$ is the neutral rudder angle and can be approximated by (T.I. Lee, 2003) by:

$$\delta_0 = -\frac{\pi \cdot \mathrm{s}}{90} \tag{2.91}$$

$$s = 1 - u(1 - w_p)/(n \cdot P) \tag{2.92}$$

where *n* are the propeller revolutions, *P* is the propeller's pitch and $w_p$ is the corrected wake fraction at the propeller estimated by (Inoue, 1981):

$$w_p = w_{p0} \exp(K_1 \beta_P^2), \quad K_1 = -4.0 \tag{2.93}$$

Where the effect of the maneuvering motion on $w_p$ is considered with the geometrical inflow angle at propeller position $\beta_P$, which is defined as $\beta_P = \beta - 2\mathrm{x'_P} r'$. The expression $\mathrm{x'_P}$ is the non-dimensioned distance of the propeller from the ship's c.o.g.

The flow rectifying effect can be substituted by two kinds of factors (Inoue, 1981). One is the flow-rectifying effect due to ship hull and the other is due to the propeller. Thus, the flow –rectification factor $\gamma$ can be written:

$$\gamma = C_P \cdot C_S \tag{2.94}$$

The propeller flow-rectification coefficient $C_P$ is given by:

$$C_P = 1 \Big/ \left[1 + 0.6 n_H (2 - 1.4s) s / (1 - s)^2\right]^{1/2} \tag{2.95}$$

with

$$s = 1 - u(1 - w_P)/(nP) \tag{2.96}$$

The ship hull flow-rectification coefficient $C_S$ is given in the following form based on some model experiment results:

$$C_S = K_3 \beta'_R \qquad \text{for } \beta'_R \le C_{SO}/K_3 \tag{2.97}$$
$$C_S = C_{SO} \qquad \text{for } \beta'_R > C_{SO}/K_3$$

with $K_3 = 0.45$ and $C_{SO} = 0.5$.

## 2.2.6 Notation

| | |
|---|---|
| $R_F$ | frictional resistance according to the ITTC-1957 friction formula |
| $(1+k_1)$ | form factor describing the viscous resistance of the hull form in relation to $R_F$ and speed dependent |
| $R_{APP}$ | resistance of appendices |
| $R_W$ | wave-making and wave-breaking resistance |
| $R_B$ | additional pressure resistance of bulbous bow near the water surface |
| $R_{TR}$ | additional pressure resistance of immersed transom stern |
| $R_A$ | model-ship correlation resistance |
| $u$ | surge velocity, [*m/s*] |
| $\rho$ | water density, [$kg/m^3$] |
| $S$ | wetted area of hull, [$m^2$] |
| $C_F$ | frictional resistance coefficient, [-] |
| $L$ | ship length, [*m*] |
| $B$ | ship breadth, [*m*] |
| $T$ | ship depth, [*m*] |
| $C_B$ | block coefficient, [-] |
| $C_M$ | midship section coefficient, [-] |
| $C_{WP}$ | waterplane coefficient, [-] |
| $C_p$ | prismatic coefficient, [-] |
| $\nabla$ | ship displacement, [$m^3$] |
| $1+k$ | form factor, [-] |
| $L_R$ | length of run |
| $lcb$ | center of buoyancy, [-] |
| $1+k_2$ | appendages' resistance factor, [-] |
| $S_{APP}$ | wetted area of appendages, [$m^2$] |
| $C_{BTO}$ | bow thruster tunnel opening coefficient, [-] |
| $A_{BT}$ | transverse sectional area of the bulb at the position where the still-water surface intersects the stern, [$m^2$] |
| $A_T$ | immersed part of the transverse area of the transom at zero speed, [$m^2$] |
| $T_F$ | ship depth at forepeak , [*m*] |
| $h_B$ | vertical position of the centre of $A_{BT}$ above the keel plane, [*m*] |
| $i_E$ | length of run, [-] |

| | |
|---|---|
| $F_n$ | Froude number, [-] |
| $P_B$ | coefficient for the emergence of the bow |
| $F_{ni}$ | Froude number based on the immersion |
| $F_{nT}$ | Froude number based on the transom immersion |
| $k_s$ | hull roughness factor, [*m*] ( usually [$\mu m$]) |
| $\rho$ | density of water, [$kg/m^3$] |
| $h$ | maximum see water depth, [*m*] |
| $t_R$ | thrust deduction factor, [-] |
| $T$ | thrust, [*N*] |
| $Q$ | Torque, [*Nm*] |
| $K_T$ | thrust coefficient, [-] |
| $K_Q$ | torque coefficient, [-] |
| $n$ | propeller revolutions, [*rps*] |
| $D_P$ | Propeller diameter, [*m*] |
| $J$ | advance ratio, [-] |
| $P/D$ | propeller pitch-diameter ratio, [-] |
| $A_E/A_O$ | propeller expanded area ratio, [-] |
| $z$ | propeller's number of fins, [-] |
| $V_A$ | advance velocity, [*m/s*] |
| $c_{0.75R}$ | chord length of the blade at 0.75R, [*m*] |
| $\nu$ | kinematical viscosity, [ |
| $F_N$ | rudder normal force, [*N*] |
| $\delta$ | rudder angle, [*rad*] |
| $x_R$ | longitudinal distance of rudder from ship’s c.o.g., [*m*] |
| $z_R$ | transverse distance of rudder from ship’s c.o.g., [*m*] |
| $a_H$ | rudder-hull interaction coefficient, [-] |
| $\lambda$ | rudder aspect ratio, [-] |
| $\mathrm{A}_R$ | mid-section area of the rudder, [$m^2$] |
| $U_R$ | effective rudder inflow velocity, [*m/s*] |
| $a_R$ | effective rudder inflow angle, [*rad*] |
| $u_R$ | axial component of $U_R$, [*m/s*] |
| $v_R$ | lateral component of $U_R$, [*m/s*] |
| $H$ | rudder height, [*m*] |
| $\delta_0$ | neutral rudder angle, [*rad*] |
| $\beta$ | drift angle, [*rad*] |
| $P$ | propeller pitch, [*m*] |
| $\gamma$ | flow-rectifying factor, [-] |
| $C_S$ | ship hull rectification coefficient, [-] |
| $C_P$ | propeller rectification coefficient, [-] |

## 2.3 External Forces

### 2.3.1 Introduction

Apart from the forces acting on ship due to its operation and maneuvering, there are also forces induced on ship that are independent from its operation. Such forces are caused by environmental disturbances, i.e. maneuvering through shallow water, wind. In addition, there are external forces that are caused by non environmental disturbances such as the force acting on a ship by a tug boat.

During the navigation of a ship, some environmental phenomena can affect its maneuvering and operation. For example by navigation through a place with frequent and strong winds the resistance will be increased significantly, altering the ship's behavior very much. Another environmental issue is the maneuvering through shallow waters that has been described in subparagraph 2.2.3.4. In such situations, all the forces acting on ship are changed.

When a tug boat assists a ship an external force is acting on the ship, which usually is constant. There is also a case that a ship will have to navigate through ice or even break through ice. In such situations some new forces have to be added to the total forces and moments.

The overall model described in the present thesis includes a submodel for additional resistance due to wind.

### 2.3.2 Additional wind resistance

The wind resistance is the force acting on the ship due to the effect of wind. It does not include the air resistance, which exists, even with no wind, and through the movement of the ship through air.

The method used for calculation (Isherwood, 1972) estimates the components of wind force ($F_{wind,x}$ & $F_{wind,y}$) and wind force induced yawing moment ($N_{wind}$) on any merchant ship for a wind from any direction. All three of them are calculated with use coefficients, as following:

$$F_{wind,x} = 0.5 \cdot \rho_{air} \cdot u_{wind}^2 \cdot A_{TP} \cdot C_X \qquad (2.98)$$

where $F_{wind,x}$ is the fore and aft component of the wind force, considered positive when directed from bow to stern, $\rho_{air}$ is the air density, $u_{air}$ is the wind speed, $A_{TP}$ is the transverse projected area and $C_x$ the wind resistance coefficient for x-axis.

$$F_{wind,y} = 0.5 \cdot \rho_{air} \cdot u_{wind}^2 \cdot A_L \cdot C_Y \qquad (2.99)$$

where $F_{wind,y}$ is the lateral component of the wind force, considered positive when directed away from the ship, $A_L$ is the lateral projected area and $C_y$ the wind resistance coefficient for y-axis.

$$N_{wind} = 0.5 \cdot \rho_{air} \cdot u_{wind}^2 \cdot A_L \cdot L_{OA} \cdot C_N \qquad (2.100)$$

where $N_{wind}$ is the wind-induced yawing moment about amidships overall, considered positive when tending to turn the bow away from the wind and $L_{OA}$ is the overall length of the ship.

The coefficients included in equations (2.47-2.49) are calculated as following:

$$C_X = A_0 + A_1 \frac{2A_L}{L_{OA}^2} + A_2 \frac{2A_{TP}}{B^2} + A_3 \frac{L_{OA}}{B} + A_4 \frac{S_P}{L_{OA}} + A_5 \frac{C}{L_{OA}} + A_6 M \qquad (2.101)$$

$$C_Y = B_0 + B_1 \frac{2A_L}{L_{OA}^2} + B_2 \frac{2A_{TP}}{B^2} + B_3 \frac{L_{OA}}{B} + B_4 \frac{S_P}{L_{OA}} + B_5 \frac{C}{L_{OA}} + B_6 \frac{A_{SS}}{A_L} \qquad (2.102)$$

$$C_N = C_0 + A_1 \frac{2A_L}{L_{OA}^2} + C_2 \frac{2A_{TP}}{B^2} + C_3 \frac{L_{OA}}{B} + C_4 \frac{S_P}{L_{OA}} + C_5 \frac{C}{L_{OA}} \qquad (2.103)$$

The constants $A_0$ to $A_6$, $B_0$ to $B_6$ and $C_0$ to $C_6$ are tabulated in tables 2.4-2.6 at $\gamma_R = 0°,10°......180°$, where $\gamma_R$ is the relative wind off bow. The expression $A_{ss}$ is the lateral projected area of superstructure, $S_P$ is the length of perimeter of lateral projection of model excluding waterline and slender bodies such as masts and ventilators, $C$ is the distance form bow to centroid of lateral projected area, $M$ is the number of distinct groups of masts or kingposts seen in lateral projection, without the kingposts close against the bridge.

*Table 2.14 – Fore and aft component of wind force*

| $\gamma_R$ | $A_0$ | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ |
|---|---|---|---|---|---|---|---|
| 0 | 2.152 | -5.00 | 0.243 | -0.164 | - | - | - |
| 10 | 1.714 | -3.33 | 0.145 | -0.121 | - | - | - |
| 20 | 1.818 | -3.97 | 0.211 | -0.143 | - | - | 0.033 |
| 30 | 1.965 | -4.81 | 0.243 | -0.154 | - | - | 0.041 |
| 40 | 2.333 | -5.99 | 0.247 | -0.190 | - | - | 0.042 |
| 50 | 1.726 | -6.54 | 0.189 | -0.173 | 0.348 | - | 0.048 |
| 60 | 0.913 | -4.68 | - | -0.104 | 0.482 | - | 0.052 |
| 70 | 0.457 | -2.88 | - | -0.068 | 0.346 | - | 0.043 |
| 80 | 0.341 | -0.91 | - | -0.031 | - | - | 0.032 |
| 90 | 0.355 | - | - | - | -0.247 | - | 0.018 |
| 100 | 0.601 | - | - | - | -0.372 | - | -0.020 |
| 110 | 0.651 | 1.29 | - | - | -0.582 | - | -0.031 |
| 120 | 0.564 | 2.54 | - | - | -0.748 | - | -0.024 |
| 130 | -0.142 | 3.58 | - | 0.047 | -0.700 | - | -0.028 |
| 140 | -0.677 | 3.64 | - | 0.069 | -0.529 | - | -0.032 |
| 150 | 0.723 | 3.14 | - | 0.064 | -0.475 | - | -0.032 |
| 160 | -2.148 | 2.56 | - | 0.081 | - | 1.27 | -0.027 |
| 170 | -2.707 | 3.97 | -0.175 | 0.126 | - | 1.81 | - |
| 180 | -2.529 | 3.76 | -0.174 | 0.128 | - | 1.55 | - |

*Table 2.15 – Lateral component of wind force*

| $\gamma_R$ | $B_0$ | $B_1$ | $B_2$ | $B_3$ | $B_4$ | $B_5$ | $B_6$ |
|---|---|---|---|---|---|---|---|
| 10 | 0.096 | 0.22 | - | - | - | - | - |
| 20 | 0.176 | 0.71 | - | - | - | - | - |
| 30 | 0.225 | 1.38 | - | 0.023 | - | -0.29 | - |
| 40 | 0.329 | 1.82 | - | 0.043 | - | -0.59 | - |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 50 | 1.164 | 1.26 | 0.121 | - | -0.242 | -0.95 | - |
| 60 | 1.163 | 0.96 | 0.101 | - | -0.177 | -0.88 | - |
| 70 | 0.916 | 0.53 | 0.069 | - | - | -0.65 | - |
| 80 | 0.844 | 0.55 | 0.082 | - | - | -0.54 | - |
| 90 | 0.889 | - | 0.138 | - | - | -0.66 | - |
| 100 | 0.799 | - | 0.155 | - | - | -0.55 | - |
| 110 | 0.797 | - | 0.151 | - | - | -0.55 | - |
| 120 | 0.996 | - | 0.184 | - | -0.212 | -0.66 | 0.34 |
| 130 | 1.014 | - | 0.191 | - | -0.280 | -0.69 | 0.44 |
| 140 | 0.784 | - | 0.166 | - | -0.209 | -0.53 | 0.38 |
| 150 | 0.536 | - | 0.176 | -0.029 | -0.1633 | - | 0.27 |
| 160 | 0.251 | - | 0.106 | -0.022 | - | - | - |
| 170 | 0.125 | - | 0.046 | -0.012 | - | - | - |

*Table 2.16 – wind induced yawing moment*

| $\gamma_R$ | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
|---|---|---|---|---|---|---|
| 10 | 0.0596 | 0.061 | - | - | - | -0.074 |
| 20 | 0.1106 | 0.204 | - | - | - | -0.170 |
| 30 | 0.2258 | 0.245 | - | - | - | -0.380 |
| 40 | 0.2017 | 0.457 | - | 0.0067 | - | -0.472 |
| 50 | 0.1759 | 0.573 | - | 0.0118 | - | -0.523 |
| 60 | 0.1925 | 0.480 | - | 0.0115 | - | -0.546 |
| 70 | 0.2133 | 0.315 | - | 0.0081 | - | -0.526 |
| 80 | 0.1827 | 0.254 | - | 0.0053 | - | -0.443 |
| 90 | 0.2627 | - | - | - | - | -0.508 |
| 100 | 0.2102 | - | -0.0195 | - | 0.0335 | -0.492 |
| 110 | 0.1567 | - | -0.0258 | - | 0.0497 | -0.457 |
| 120 | 0.0801 | - | -0.0311 | - | 0.0740 | -0.396 |
| 130 | -0.0189 | - | -0.0488 | 0.0101 | 0.1128 | -0.420 |
| 140 | 0.0256 | - | -0.0422 | 0.0100 | 0.0889 | -0.463 |
| 150 | 0.0552 | - | -0.0381 | 0.0109 | 0.0689 | -0.476 |
| 160 | 0.0881 | - | -0.0306 | 0.0091 | 0.0366 | -0.415 |
| 170 | 0.0851 | - | -0.0122 | 0.0025 | - | -0.220 |

## 2.3.3 Notation

$\rho$ density of water, [$kg/m^3$]
$F_{wind,x}$ wind resistance x-axis, [$N$]
$\rho_{air}$ density of air, [$kg/m^3$]
$u_{wind}$ wind speed [$m/s$]

| | |
|---|---|
| $A_{TP}$ | transverse projected area, [$m^2$] |
| $C_X$ | x-axis resistance coefficient, [-] |
| $F_{wind,y}$ | wind resistance y-axis, [$N$] |
| $A_L$ | lateral projected area, [$m^2$] |
| $C_Y$ | y-axis resistance coefficient, [-] |
| $N_{wind}$ | wind induced yawing moment, [$Nm$] |
| $C_N$ | wind induced yawing moment coefficient, [-] |
| $L_{OA}$ | overall length, [$m$] |
| $S_p$ | length of perimeter of lateral projection of ship excluding waterline and slender bodies such as masts and ventilators |
| $C$ | distance form bow to centroid of lateral projected area |
| $M$ | number of distinct groups of masts or kingposts seen in lateral projection, without the kingposts close against the bridge |
| $A_{SS}$ | lateral projected area of superstructure |

## 2.4 Miscellaneous Calculations

### 2.4.1 Introduction

There are some several specific calculations that are needed throughout the model. Such data are the added masses and moments of Inertia, the propulsion coefficients etc. The model described in this thesis includes a sub-model, in which all these values are calculated. This sub-model is called *Utility block*.

The following paragraphs describe the method, with which such data are calculated. The values are the following:

- Propulsion coefficient, including wake fraction, thrust deduction factor, relative-rotative coefficient
- Added mass in x and y-axis
- Added moment of inertia in z-axis
- Moments of inertia in x and z-axis
- Ship mass

### 2.4.2 Propulsion coefficients

As the propulsion coefficients are very relevant to the resistance, the method used for calculation is the same as for the calculations of the resistance (Holtrop, Mennen, 1982) (Holtrop, 1984).

For single screw ships with a conventional stern the formula is the following (Holtrop, 1984):

$$w = c_9 c_{20} C_V \frac{L}{T_A}\left(0.050776 + 0.93405 c_{11} \frac{C_V}{\left(1 - C_{P1}\right)}\right) + 0.27915 c_{20} \sqrt{\frac{B}{L\left(1 - C_{P1}\right)}} + c_{19} c_{20} \qquad (2.104)$$

The coefficient $c_9$ depends on the coefficient $c_8$ defined as:

$$c_8 = B \cdot S / \left(L \cdot D \cdot T_A\right), \text{ when } B/T_A < 5 \qquad (2.105)$$

or

$$c_8 = S\left(7B/T_A - 25\right) / \left(L \cdot D \cdot \left(B/T_A - 3\right)\right), \text{ when } B/T_A > 5 \qquad (2.106)$$

$$c_9 = c_8, \text{ when } c_8 < 28 \qquad (2.107)$$

or

$$c_9 = 32 - 16/\left(c_8 - 24\right), \text{ when } c_8 > 28 \qquad (2.108)$$

$$c_{11} = T_A/D \text{, when } T_A/D < 2 \quad (2.109)$$

or

$$c_{11} = 0.0833333(T_A/D)^3 + 1.33333 \text{, when } T_A/D > 2 \quad (2.110)$$

$$c_{19} = 0.12997/(0.95 - C_B) - 0.11056/(0.95 - C_P) \text{, when } C_P < 0.7 \quad (2.111)$$

or

$$c_{19} = 0.18567/(1.3571 - C_M) - 0.71276 + 0.38648 C_P \text{, when } C_P > 0.7 \quad (2.112)$$

$$c_{20} = 1 + 0.015 C_{stern} \quad (2.113)$$

$$C_{P1} = 1.45 C_P - 0.315 - 0.0225 lcb \quad (2.114)$$

The coefficient $C_V$ is the viscous resistance coefficient with:

$$C_V = (1+k) C_F + C_A \quad (2.115)$$

The thrust deduction factor and relative-rotative coefficient of single screw ships is given by the following equations:

$$t = 0.25014 (B/L)^{0.28956} \left(\sqrt{BT}/D\right)^{0.2624} \Big/ (1 - C_P + 0.0225 lcb)^{0.01762} + 0.0015 \cdot C_{stern} \quad (2.116)$$

$$n_R = 0.992 - 0.05908 A_E/A_O + 0.07424 (C_P - 0.0225 lcb) \quad (2.117)$$

For single screw ships with open stern the following equations can be used:

$$w = 0.3 C_B + 10 C_V C_B - 0.1 \quad (2.118)$$

$$t = 0.10 \quad (2.119)$$

$$n_R = 0.98 \quad (2.120)$$

Finally, for twin screw ships the values are calculated by the following:

$$w = 0.3095 C_B + 10 C_V C_B - 0.23 D/\sqrt{BT} \quad (2.121)$$

$$t = 0.325 C_B - 0.1885 D/\sqrt{BT} \quad (2.122)$$

$$n_R = 0.9737 + 0.111 (C_P - 0.0225 lcb) + 0.06325 P/D \quad (2.123)$$

### 2.4.3 Mass & Moment of inertia

The ship's mass is calculated using the following formula:

$$m_{ship} = \nabla \cdot \rho \tag{2.124}$$

where: $\nabla$ : ship's displacement, [$m^3$]
$\rho$ : water density, [$kg/m^3$]

The non-dimensional added mass $m_{xx}$ is calculated according to T.I. Lee (2003) by the following equation:

$$m_{xx} = \frac{2.7 \cdot \rho}{L^2} \cdot \left(\mathrm{C_B} \cdot L \cdot B \cdot T\right)^{5/3} \tag{2.125}$$

Similarly, the same reference suggests an equation for the calculation of the added mass on y-axis $m_{yy}$:

$$m_{yy} = \frac{\pi \cdot \rho \cdot \mathrm{L} \cdot \mathrm{T}^2}{2} \cdot \left(1 + 0.16 \cdot \mathrm{C_B} \cdot \frac{B}{T} - \frac{5.1}{\left(L/B\right)^2}\right) \tag{2.126}$$

but in order to have Prime I non-dimensioning the equation is transformed into the following (Clarke, 1983):

$$m_{yy} = \frac{\pi \cdot \rho \cdot \mathrm{T}^3}{2} \cdot \left(1 + 0.16 \cdot \mathrm{C_B} \cdot \frac{B}{T} - \frac{5.1}{\left(L/B\right)^2}\right) \tag{2.127}$$

The added moment of inertia in z-axis $J_{zz}$ is calculated by the equation also suggested by T.I. Lee (2003):

$$J_{zz} = \frac{\pi \cdot \rho \cdot \mathrm{L}^3 \cdot \mathrm{T}^2}{24} \cdot \left(1 + 0.2 \cdot \mathrm{C_B} \cdot \frac{B}{T} - \frac{4}{\left(L/B\right)}\right) \tag{2.128}$$

Finally, the non-dimensional forms used in this diploma thesis are shown in table 2.7.

*Table 2.17 – Non dimensional Forms*

| Term | Expression |
|---|---|
| Length, $x$ | $x' = \frac{x}{L}$ |
| Masses, $m$ | $m' = \frac{m}{\frac{1}{2} \cdot \rho \cdot \mathrm{L}^2 \cdot T}$ |
| Moments of inertia, $I$ | $I' = \frac{I}{\frac{1}{2} \cdot \rho \cdot \mathrm{L}^4 \cdot T}$ |
| Forces, $X,Y$ | $X' = \frac{X}{\frac{1}{2} \cdot \rho \cdot \mathrm{L} \cdot T \cdot V^2}$ |
| Moments, $N,K$ | $N' = \frac{N}{\frac{1}{2} \cdot \rho \cdot \mathrm{L}^2 \cdot T \cdot V^2}$ |

## 2.4.4 Notation

| | |
|---|---|
| $\rho$ | density of water, [$kg/m^3$] |
| $L$ | ship length, [m] |
| $B$ | ship bredth, [m] |
| $T$ | ship depth, [m] |
| $T_A$ | depth aft, [m] |
| $S$ | wetted surface, [m$^2$] |
| $D_P$ | propeller's diameter, [m] |
| $C_B$ | block coefficient, [-] |
| $C_P$ | prismatic coefficient, [-] |
| $C_M$ | mid-section coefficient, [-] |
| $C_{stern}$ | stern coefficient, [-] |
| $lcb$ | Longitudinal centre of buoyancy , [-] |
| $C_F$ | friction resistance coefficient, [-] |
| $C_A$ | model-ship correlation resistance coefficient, [-] |
| $C_V$ | viscous resistance coefficient, [-] |
| $w$ | wake fraction, [-] |
| $t$ | thrust deduction factor, [-] |
| $n_r$ | Relative-rotative factor, [-] |
| $A_E/A_O$ | expanded blade area ratio, [$N$] |
| $\nabla$ | ship displacement, [$m^3$] |
| $m_{ship}$ | ship mass, [$kg$] |
| $m_{xx}$ | ship's added mass, x-axis, [$kg$] |
| $m_{yy}$ | ship's added mass, y-axis, [$kg$] |

| | |
|---|---|
| $J_{zz}$ | ship's added moment of inertia, z-axis, [$kg\ m^2$] |
| $C$ | distance form bow to centroid of lateral projected area |
| $M$ | number of distinct groups of masts or kingposts seen in lateral projection, without the kingposts close against the bridge |
| $A_{SS}$ | lateral projected area of superstructure |
| $\rho$ | density of water, [kg/m3] |
| $C_V$ | viscous resistance coefficient, [-] |
| $C_P$ | prismatic coefficient, [-] |
| $\rho_{air}$ | density of air, [kg/m3] |
| $u_{wind}$ | wind speed [m/s] |
| $A_{TP}$ | transverse projected area, [m2] |
| $C_X$ | x-axis resistance coefficient, [-] |
| $F_{wind,y}$ | wind resistance y-axis, [N] |
| $A_L$ | lateral projected area, [m2] |
| $C_Y$ | y-axis resistance coefficient, [-] |
| $N_{wind}$ | wind induced yawing moment, [Nm] |
| $C_N$ | wind induced yawing moment coefficient, [-] |
| $L_{OA}$ | overall length, [m] |
| $S_p$ | length of perimeter of lateral projection of ship excluding waterline and slender bodies such as masts and ventilators |
| $C$ | distance form bow to centroid of lateral projected area |
| $M$ | number of distinct groups of masts or kingposts seen in lateral projection, without the kingposts close against the bridge |
| $A_{SS}$ | lateral projected area of superstructure |

## 2.5 Modeling of Maneuvers

### 2.5.1 Introduction

Traditionally the maneuvering performance of a ship received little attention by the designers. This was mainly a result of the lack of some maneuvering standards for the designer to follow and of regulatory authorities to enforce. Thus, some ships were built with poor maneuvering quality, which led to accidents and loss of life. The designers often relied on the ship handling abilities of the ship's crew. In order to avoid such incidents, some standards were implemented mostly by IMO.

The standards were selected so that they are simple, practical and do not require significant increase in trial time or complexity. They are based on the premise that the maneuverability of a ship can be adequately judged by the results of the typical sea trial maneuvers. If a ship fails the trials, depending on the extent of the inadequacy, the ship owner can decrease the payment or even not accept the ship. For great inadequacy, the IMO does not let the ship operate and the ship must be re-inserted in the dock in order to correct the faults.

There are three main categories of tests required by the standards:

- Turning tests
- Zig-zag tests
- Stopping tests

The present model can simulate the first two maneuvers,as described in the next paragraphs

### 2.5.2 Turning circle maneuver

A turning circle maneuver is to be perfomed to both starboard and port with 35˚ rudder angle or the maximum rudder angle permissible at the test speed. The rudder angle is executed following a steady approach with zero yaw rate. The essential information to be obtained from this manoeuvre is the tactical diameter, the advance and the transfer, as shown in fig. 2.5.

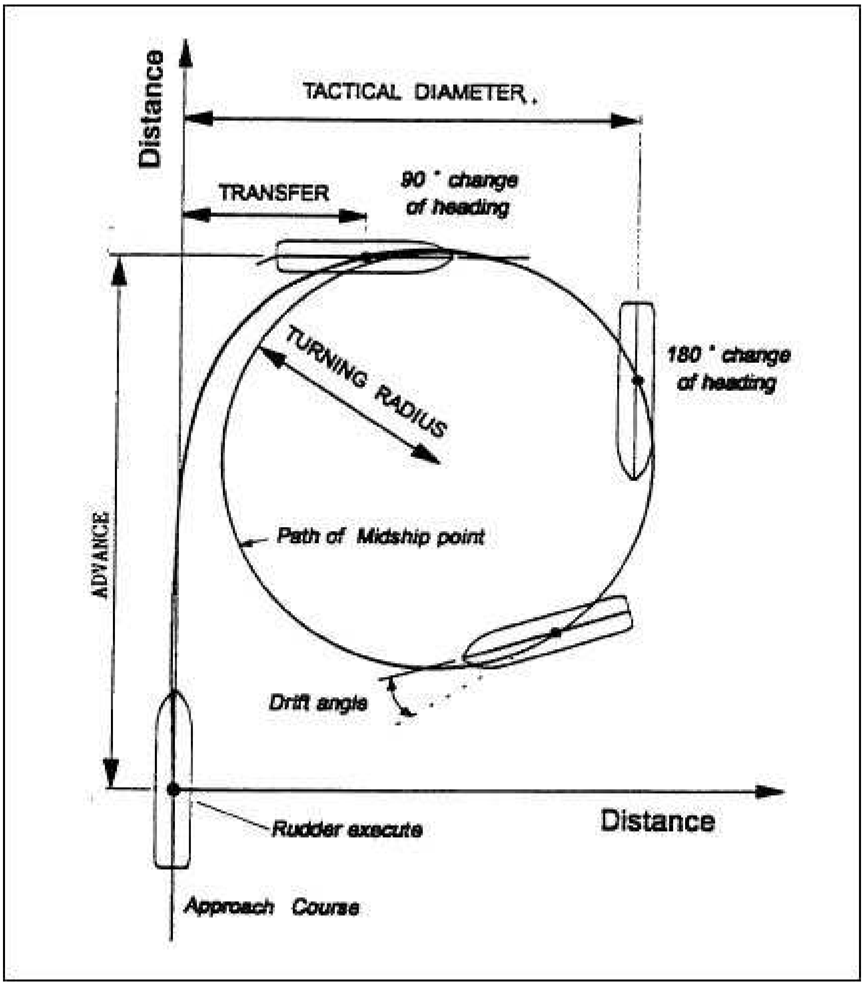

*Fig. 2.5 – Turning circle maneuver definitions*

Trials shall be made to port and starboard using maximum rudder angle without changing engine control setting from the initial speed. The following general procedure is recommended (IMO, 2002):

1. The ship is brought to a steady course and speed according to the specific approach condition
2. The recording of data starts
3. The maneuver is started by ordering the rudder to the maximum rudder angle. Rudder and engine controls are kept constant during the turn.
4. The turn continues until 360° change of heading has been completed. It is, however, recommended that in order to fully assess environmental effects a 720° turn be completed.
5. Recording of data is stopped and the maneuver is terminated.

### 2.5.3 Zig-zag maneuver

A zig-zag test should be initiated to both starboard and port and begins by applying a specified amount of rudder angle to an initially straight approach ("first execute"). The rudder angle is then alternately shifted to either side after a specified deviation from the ship's original heading is reached ("second execute" and following).

There are two kinds of zig-zag tests that are included in the standards, the 10°/10° and 20°/20° zig-zag tests. The 10°/10° test uses rudder angles of 10° to either side following a heading deviation of 10° and similarly the 20°/20°. The essential information to be obtained by these tests is the overshoot angles, initial turning time to second execute and the time to check yaw.

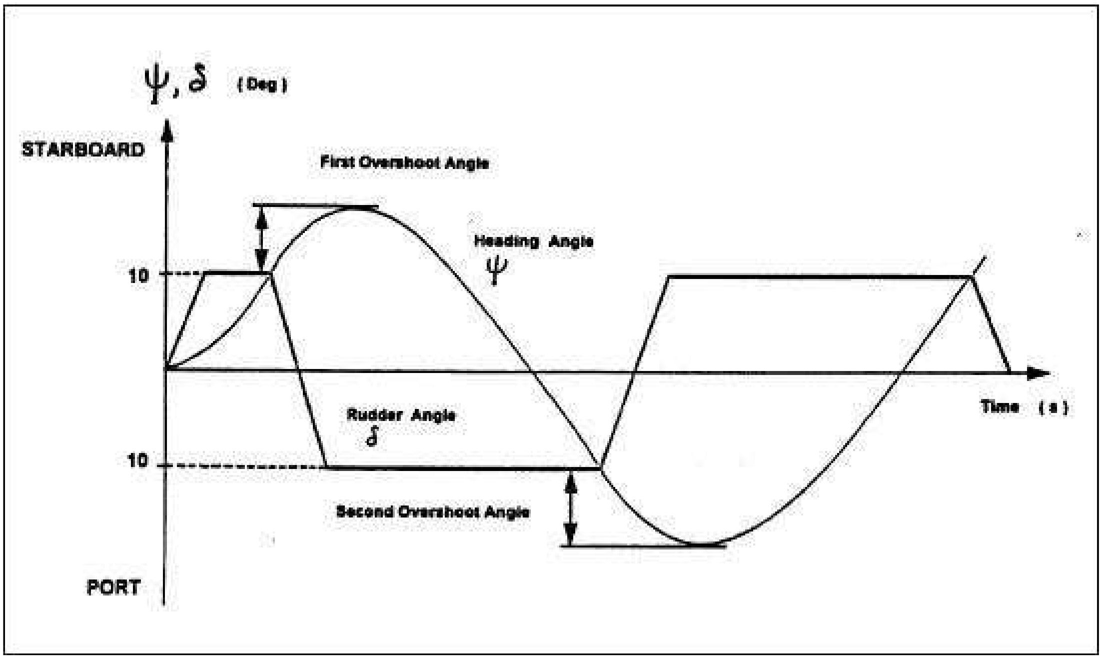


*Fig. 2.6 – Zig-zag maneuver definition*

The given rudder and change of heading angle for the following procedure is 10°. This value can be replaced for alternative or combined zig-zag maneuvers by other angles such as 20° for other zig-zag tests. The following general procedure is recommended (IMO, 2002)::

1. The ship is brought to a steady course and speed according to the specific approach condition.
2. The recording of the data starts.
3. The rudder is ordered to 10° to starboard/port.
4. When the heading has changed by 10° off the base course, the rudder is shifted to 10° to port/starboard. The ship's yaw will be checked and a turn in the opposite direction (port/starboard) will begin. The ship will continue in the turn and the original heading will be crossed.
5. When the heading is 10° port/starboard off the base course, the rudder is reversed as before.
6. The procedure is repeated until ship heading has passes the base course no less than two times.
7. Recording of data is stopped and the maneuver is terminated.

### 2.5.4 Standard maneuvers' terminology

The various terms that are needed for the modelling of the trials as well as their simulation are the following:

1. The test speed $V$ used in the standards is a speed of at least 90% of the ship's speed corresponding to 85% of the maximum engine output.
2. Turning circle maneuver is the maneuver to be performed to both starboard anf port with 35° rudder angle or the maximum rudder angle permissible at the test speed following a steady approach with zero yaw rate.
3. Advance is the distance travelled in the direction of the original course by the midship point of a ship from the position at which the rudder order is given to the position at which the heading has changed 90° from the original course.
4. Tactical diameter is the distance travelled by the midship point of a ship from the position at which the rudder order is given to the position at whivh the heading has changed 180° from the original course. It is measured in a direction perpendicular to original heading of the ship.

### 2.5.5 Criteria

The maneuverability of a ship is considered satisfactory if the following criteria are complied with:

1. Turning ability

   The advance should not exceed 4.5 ship lengths and the tactical diameter should not exceed 5 ship lengths in the turning circle maneuver.

2. Initial turning ability

   With the application of 10° rudder angle to port/starboard, the ship should not have travelled more than 2.5 ship lengths by the time the heading has changed by 10° from the original heading.

3. Yaw-checking and course-keeping abilities

   1. The value of the first overshoot angle in the 10°/10° zig-zag test should not exceed:

      1. 10° if $L/V$ is less than 10 seconds;
      2. 20° if $L/V$ is 30 seconds or more;
      3. $(5+1/2(L/V))°$ if $L/V$ is 10 seconds or more than 30seconds.

   2. The value of the second overshoot angle in the 10°/10° zig-zag should not exceed:

1. 25° if *L/V* is less than 10 seconds;
2. 40° if *L/V* is 30 seconds or more;
3. $(17.5 + 0.75(L/V))°$ if *L/V* is 10 seconds or more than 30seconds.

3. The value of the first overshoot angle in the 20°/20° zig-zag test should not exceed 25°

# Chapter 3

# Simulation Model

## 3.1 Introduction

The model described in this thesis was developed in Matlab and Simulink. In Matlab, the programs are developed, using some programming principles and rules, in m-files. These m-files can be executed without compiling processes. Thus, every sub-model described in chapter 2 was programmed in a separate m-file.

In order to have a graphical interface of the model, Simulink was used. Simulink works with blocks, which connect to each other. Every sub-model was organized in a block. Simulink blocks can be hierarchically synthesized to groups of models in various levels. Every block can be replaced without the loss of functionality of the whole model.

Five main blocks were used, one for the dynamics, one for the ship forces, one for external ship forces, one for coordinates conversion and a last one called *Utility*, where some auxiliary values are calculated. The overall structure of the model is shown in fig. 3.1.

The Dynamics block includes the dynamics equations. Thus it calculates the velocities by having as input the forces and moments acting on the ship.

The Ship Forces block includes all the sub-models described in section 2.2. There are four sub-blocks for Resistance, Hull forces, Propeller forces and Rudder forces, each of which calculates the ship forces and moments. In every sub-block, the m-file of each model was inserted with the use of s-functions. The output of each model is added to produce the total force or moment for every direction ($X_{tot}$, $Y_{tot}$, $N_{tot}$, $K_{tot}$). Apart from the geometrical and functional inputs needed for the calculation this block also needs the ship velocities. Thus the input of this block is the output of the dynamics block and vice versa.

The External Ship Forces block includes the model of the wind resistance, which calculates the forces and moments acting on the ship due to wind and the ouputs are added in the total forces and moments. If the wind speed is set to zero the output of this sub-block is neglected.

The coordinates conversion block includes the process described in paragraph 2.1.2, and produces the matrix, which includes the $x$ and $y$ position of the ship every simulation step.

Finally, the Miscellaneous calculations block includes the calculation of some values needed for the simulation such as the additional masses, the wake fraction etc.

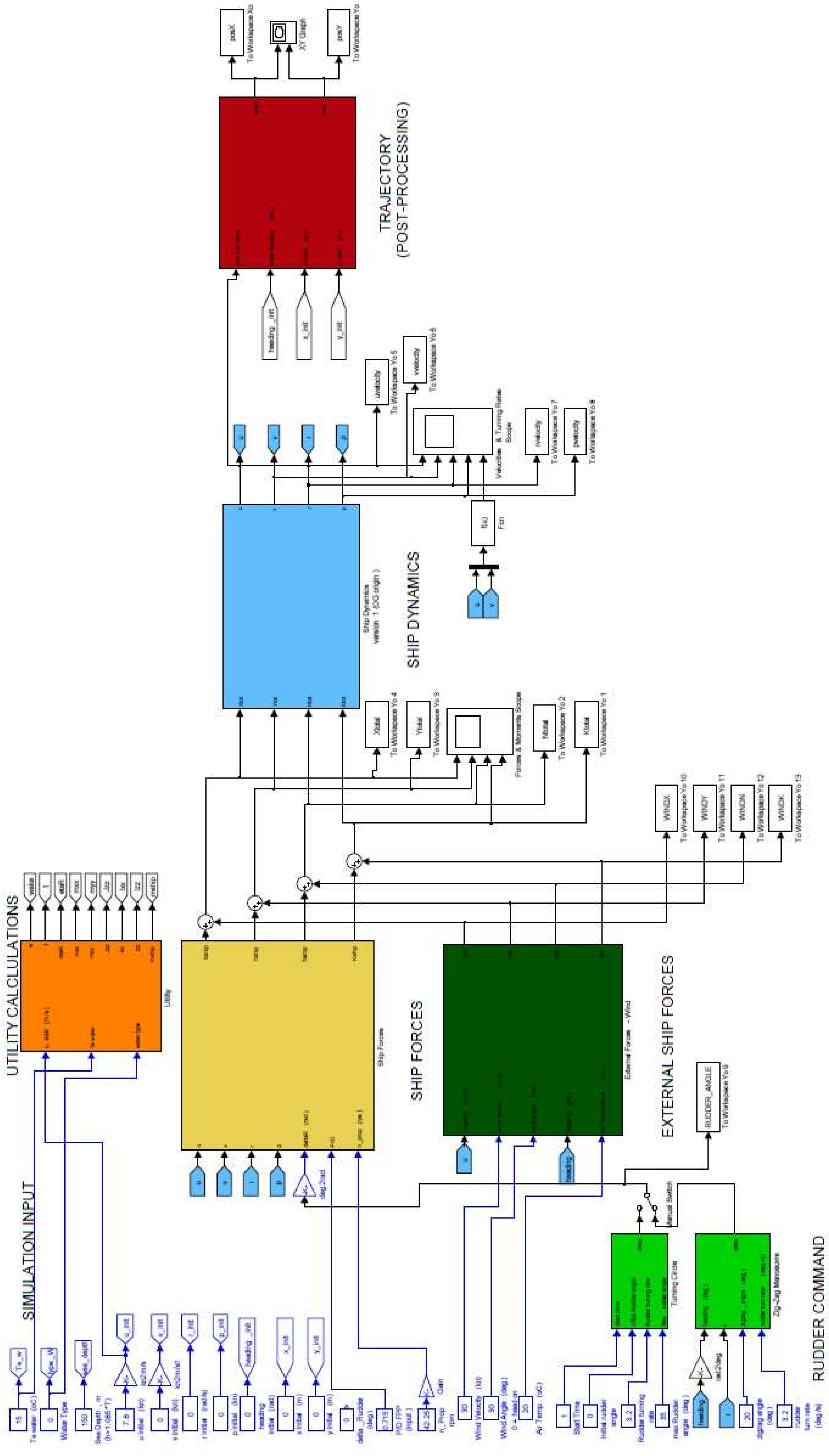


*Fig. 3.1 – Overall structure of the model*

## 3.2 Input file

To perform a calculation for a particular case, all the previously mentioned blocks need input values. Some of them are geometrical particulars of the ship, such as breadth, propeller diameter etc, and the rest of them are operational parameters, which must be able to be altered during the navigation of the ship such as the rudder angle, propeller revolutions etc. The geometrical parameters are given with the use of an input file called "input.txt" which includes all the needed input values in a specific order. The operational parameters are given in the simulink model by the user or by a user-defined sequence.

The input file is organized according to the sub-models, meaning that the first sector is the input for the resistance sub-model, the second for the hull forces and so on. The user has to define all the ship's particulars and create the input file. The form of the input file is shown in table 3.1.

*Table 3.1 – Input file index*

| index | Input value | Description |
|---|---|---|
| 1 | LBP | Lenght bewtween perpendiculars, [*m*] |
| 2 | LWL | Lenght in waterline, [*m*] |
| 3 | B | Breadth, [*m*] |
| 4 | TF | Depth at for perpendicular, [*m*] |
| 5 | TA | Depth at aft perpendicular, [*m*] |
| 6 | VOL | Displacement, [$m^3$] |
| 7 | LCB | Longitudal Centre of Buoyancy, [*m*] |
| 8 | CWP | Waterplane Coefficient, [-] |
| 9 | CM | Midship Section Coefficient, [-] |
| 10 | S | Wetted Hull Area, [$m^2$] |
| 11 | CSTERN | cstern , [-] |
| 12 | NRUD | Number of Rudders, [-] |
| 13 | SRUD | Wetted area of rudder, [$m^2$] |
| 14 | CRUD | coefficient of rudder, [-] |
| 15 | SAPP i | Wetted area of rudder behind skeg, [$m^2$] |
| 16 | | Wetted area of rudder behind stern, [$m^2$] |
| 17 | | Wetted area of twin-screw balance rudders, [$m^2$] |
| 18 | | Wetted area of shaft brackets, [$m^2$] |
| 19 | | Wetted area of skeg, [$m^2$] |
| 20 | | Wetted area of strut bossings, [$m^2$] |
| 21 | | Wetted area of hull bossings, [$m^2$] |
| 22 | | Wetted area of shafts, [$m^2$] |
| 23 | | Wetted area of stabilizer fins, [$m^2$] |
| 24 | | Wetted area of dome, [$m^2$] |
| 25 | | Wetted area of bilge keels, [$m^2$] |
| 26 | ABT | cross sectional area of bulbous bow, [$m^2$] |
| 27 | HB | centroid of bulbous bow cross section to keel, [*m*] |
| 28 | AT | area of immersed transom, [$m^2$] |
| 29 | LR | Lenght of run, [*m*] |
| 30 | IE | half angle of entrance, [-] |
| 31 | NBT | Number of Bow thruster, [-] |
| 31+1 | DBT (1$^{st}$) | Diameter of 1$^{st}$ bow thruster, [*m*] |
| 31+2 | DBT (2$^{nd}$) | Diameter of 2$^{nd}$ bow thruster, [*m*] |
| … | | |

| | | |
|---|---|---|
| 31+i | DBT (i) | Diameter of i bow thruster, [*m*] |
| 31+i+1 | NPROP | Number of propellers, [-] |
| 31+i+2 | DP | Propeller Diameter, [*m*] |
| 31+i+3 | AAE | Propeller Expanded blade area ratio, [-] |
| 31+i+4 | PPD | Propeller pitch-diameter ratio, [-] |
| 31+i+5 | LOA | Length overall, [*m*] |
| 31+i+6 | AL | Lateral projeced wind area, [$m^2$] |
| 31+i+7 | AS | Lateral projectef area of superstructure, [$m^2$] |
| 31+i+8 | AT | Transverse projected wind area, [$m^2$] |
| 31+i+9 | S | Length of perimeter of lateral projection, [*m*] |
| 31+i+10 | C | Distance from bow to centroid of lateral projected area, [*m*] |
| 31+i+11 | M | Number of distinct groups of masts or king posts, [-] |
| 31+i+12 | NDER | Total Number of derivatives on all directions, [-] |
| 31+i+9+j*4 | dir | direction of the derivative (X,Y,N,K) , [-] |
| 31+i+10+j*4 | Hd | value of the derivative, [-] |
| 31+i+11+j*4 | index | index of the derivativ (i.e. uv2) , [-] |
| 31+i+12+j*4 | method | method of calc. (0:user,1,2,3,5) , [-] |
| … | dir | |
| … | Hd | |
| … | index | |
| … | method | |
| 31+i+j*4+13 | CWPA | aft waterplane coefficient , [-] |
| 31+i+j*4+14 | CPA | aft prismatic coefficient, [-] |
| 31+i+j*4+15 | LCG | longitudinal center of gravity as percentage from half length, positive for, negative aft, [*%*] |
| 31+i+j*4+16 | BP07 | half breadth at the height of 0.7R(propeller radius) in 2.0 station,[*m*] |
| 31+i+j*4+17 | BPS | half breadth at the height of propeller shaft in 2.0 station,[*m*] |
| 31+i+j*4+18 | Ixx | moment of inertia at x-axis, [kg $m^2$] |
| 31+i+j*4+19 | Izz | moment of inertia at z-axis, [kg $m^2$] |
| 31+i+j*4+20 | xP | distance from propeller to ship, [*m*] |
| 31+i+j*4+21 | xR | distance from rudder to ship, [*m*] |
| 31+i+j*4+22 | zR | vertical distance of rudder, [*m*] |
| 31+i+j*4+23 | HR | rudder height, [*m*] |
| 31+i+j*4+24 | lamda | rudder aspect ratio, [*m*] |
| 31+i+j*4+25 | AR | rudder surface, [*m*] |
| 31+i+j*4+26 | w0 | wake fraction,(user defined,0=calculation), [-] |
| 31+i+j*4+27 | t0 | thrust deduction factor,(user defined,0=calculation), [-] |
| 31+i+j*4+28 | nR0 | relative rotative factor,(user defined,0=calculation), [-] |
| 31+i+j*4+29 | mxx0 | added mass in x-axis(user defined,0=calculation), [*kg*] |
| 31+i+j*4+30 | myy0 | added mass in y-axis(user defined,0=calculation), [*kg*] |
| 31+i+j*4+31 | Jzz0 | added moment of inertia in y-axis(user defined,0=calculation), [*kg*] |

As shown in table 3.1 two values, the number of bow thrusters (NBT) and the number of hydrodynamic derivatives (NDER), are not defined a priori and affect the index of every value. Thus, the assignment of the input values to matrices had to take into consideration the fact that the index of every value is not fixed. For example the index of every value after the NBT should be increased by NBT, because the input file includes the diameter of each bow thruster. So, if the ship has three bow thrusters, three diameters must be given and thus all the subsequent indices will be increased by three.

The assisting parameters *i* and *j* in table 3.1 were inserted as NBT and NDER in order to make the index generic.

The part of the input file, where the user gives the hydrodynamic derivatives, was designed so as the user can define which derivatives should be included and also which is the method of calculation. There are four values that should be given for every hydrodynamic derivative. The first one is the direction, where the values should be given according to table 3.2.

*Table 3.2*

| Corresponding values of "*dir*" | | |
|---|---|---|
| Value | direction | description |
| 1 | X | Hydr. Derivative of force *X* |
| 2 | Y | Hydr. Derivative of force *Y* |
| 3 | N | Hydr. Derivative of force *N* |
| 4 | K | Hydr. Derivative of force *K* |

The second value is the actual value of the derivative. If the method is selected to be a calculation method and not user-defined, this value will be neglected.

The third value is the derivative index, meaning the definition of the hydrodynamic derivative (i.e. $X_{uv}$). The user must give eight positive real numbers, separated with a gap. Every number indicates the power that each velocity or acceleration should be raised, the sequence of whom is $u, v, r, p, \dot{u}, \dot{v}, \dot{r}, \dot{p}$. For example the derivative $Y_{uv^2}$ is decribed by the sequence *[1 2 0 0 0 0 0 0]*.

Finally, the user must define which method should be used for calculations. The corresponding values are shown in the following table:

*Table 3.3*

| Corresponding values of "*method*" | |
|---|---|
| Value | method |
| 0 | user-defined value |
| 1 | Inoue, S., Hirano, M., Kijima, K., and Takashina, J. (1981). A Practical Calculation Method of Ship Manoeuvring Motion. International Shipbuilding Progress, 28(325), 207–222. |
| 2 | Clarke, D., Gedling, P. and Hine, G. (1983). The application of manoeuvring criteria in hull design using linear theory. The Naval Architect, pp. 45–68. |
| 3 | Tae-II Lee, Kyoung-Soo Ahn, Hyoung-Suk Lee, On Deuk-Joon Yum, (2003) An Empirical Prediction Of Hydrodynamic Coefficients For Modern Ship Hulls, MARSIM 2003, Kanazawa, Japan. |
| 4 | Adapted method of Inoue et al by ITTC |
| 5 | Michael D Woodward, David Clarke, Mehmet Atlar, (2003) On The Manoeuvring Prediction Of Pod Driven Ships, MARSIM 2003, Kanazawa, Japan. |

The last six values, which are wake fraction, thrust deduction factor, relative-rotative factor, added masses on x and y axes and added moment of inertia in z-axis can be given by the user, but also can be calculated by the model (*utility* block). If the user gives the value 0 then these values are calculated by the model.

The loading of these values on the workspace is done by running the in.m m-file (Appendix 1.1). In this program the values are assigned to the matrix *A* and some logical values are converted to real values accordingly (i.e. stern coefficient at Holtrop method). Every sub-model was programmed to assign all the values it needs from matrix *A* to a matrix (i.e. $L_{BP}$=A(1)), taking into consideration the index increase due to altering *NBT* and *NDER* values.

The other input values, called simulation inputs are given directly in the model in the far left side (*fig. 3.1*). Especially for the rudder angle, two blocks were built. These two blocks control the rudder angle in such a way so that the ship will execute two maneuvers, the turning circle and zig-zag as defined by IMO.

## 3.3 Dynamics block

### 3.3.1 Introduction

The dynamics block, shown in figure 3.2, includes the dynamics equations of the model (eq. 2.1). The inputs of this subsystem are the forces and moments acting on ship, that are calculated in the Ship Forces block and the External Forces block every simulation step. Using the dynamics equations it calculates the velocities of the ship at every simulation step, which are then taken as input for the Ship dynamics block. The yaw turn rate r is also taken as input in the coordinates conversion block.

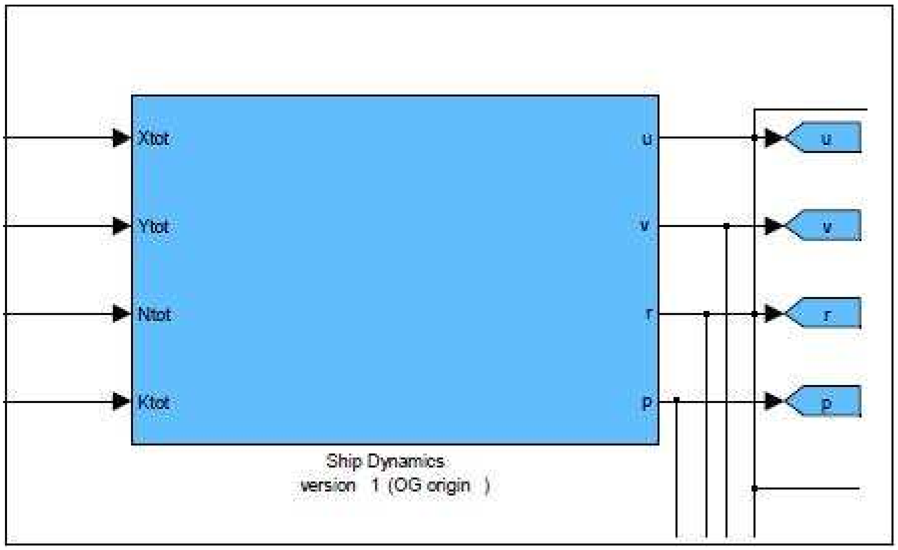


*Figure 3.2- Ship dynamics blok*

## 3.3.2 Mathematical Model

Into this block there are four more sub-blocks, one for every equation (fig.3.3). Each sub-block takes as input the respective force or moment and two or none velocities and gives as output the respective velocity or turn rate at that time step.

As shown in the following figure, the first sub-block which includes the surge equation has the total surge force *X*, velocity *v* and turn rate *r* as inputs. Similarly the second sub-block of sway has total sway force *Y*, velocity *u* and turn rate *r* as inputs. The latter two sub-blocks include yaw and roll equation respectively. The yaw sub-block takes the yaw moment *N* as input and the roll sub-block the roll moment *K*. The output of the yaw block is feeding the first two blocks with the yaw turn rate *r*.

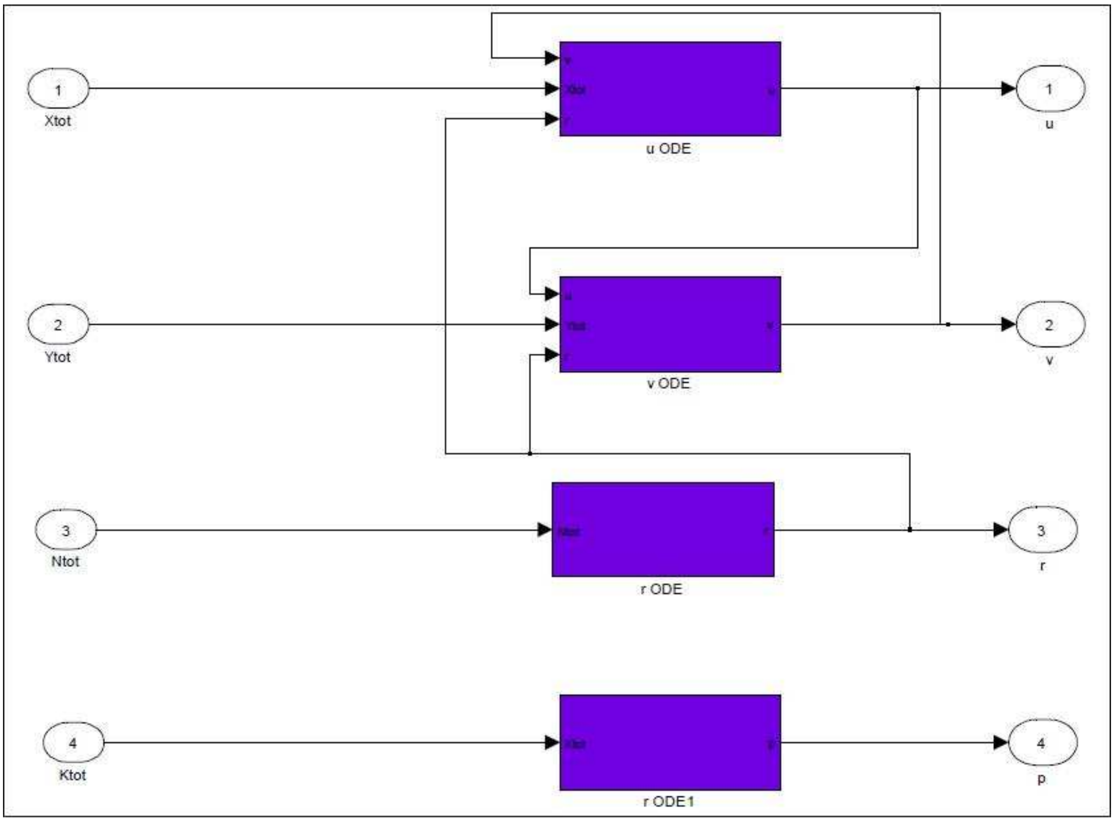


*Figure 3.3 – Dynamics Block interior*

*<u>Surge equation block</u>*

This block includes the model of the surge equation. In order to create the equation in Simulink it had to be rewritten in the following form:

$$\dot{u} = \frac{X + m \cdot v \cdot r}{(m + m_x)} \tag{3.1}$$

where it is solved after the surge acceleration. As shown in the following figure, first, the right hand side part of the equation is created. After that the value is integrated in order to provide the surge velocity. In the integrator, the initial value of the surge velocity, as given by the user in the simulation inputs, is used as initial condition. The value $m_x$ is the added mass in x-axis, which is calculated in the Utility block or given by the user in the input file.

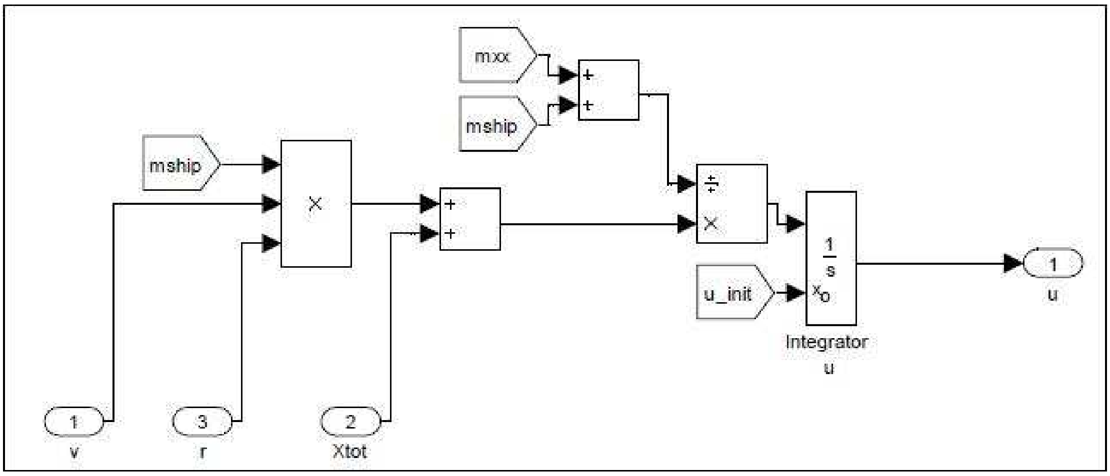


*Figure 3.4 – Surge equation block*

<u>*Sway equation block*</u>

This block includes the model of the sway equation. In a similar way as in the surge block the equation had to be rewritten in the following form:

$$\dot{v} = \frac{Y + m \cdot u \cdot r}{\left(m + m_y\right)} \tag{3.2}$$

where it is solved after the sway acceleration. As shown in the following figure, the right hand side part of the equation is created and after that the value is integrated in order to provide the sway velocity. In the integrator the initial value of the sway velocity, as given by the user in the simulation inputs, is used as initial condition. The value $m_{xy}$ is the added mass in y-axis, which is calculated in the Utility block or given by the user in the input file.

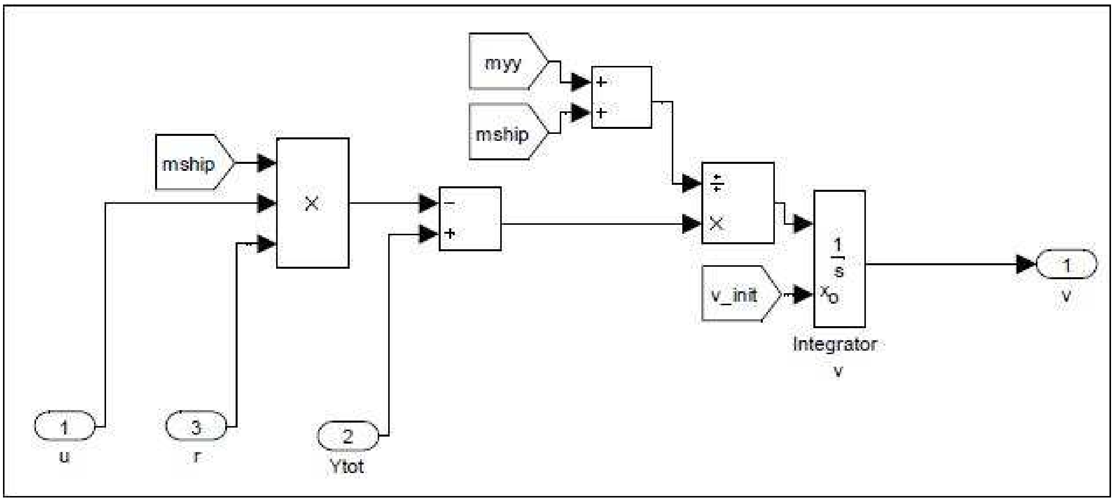


*Figure 3.5 – Sway equation block*

*Yaw equation block*

This block includes the model of the yaw equation. In a similar way as in the latter two blocks the equation had to be rewritten in the following form:

$$\dot{r} = \frac{N}{\left(I_z + J_z\right)} \tag{3.3}$$

where it is solved after the yaw acceleration. The yaw equation is much simpler than the other two, as it includes only the moment of inertia, the added moment of inertia, the total yaw moment and the yaw acceleration.

As shown in the following figure, the right hand side part of the equation is created and after that the value is integrated in order to provide yaw turn rate. In the integrator the initial value of the yaw turn rate, as given by the user in the simulation inputs, is used as initial condition. The value $J_z$ is the added moment of inertia in z-axis, which is calculated in the Utility block, or given by the user in the input file.

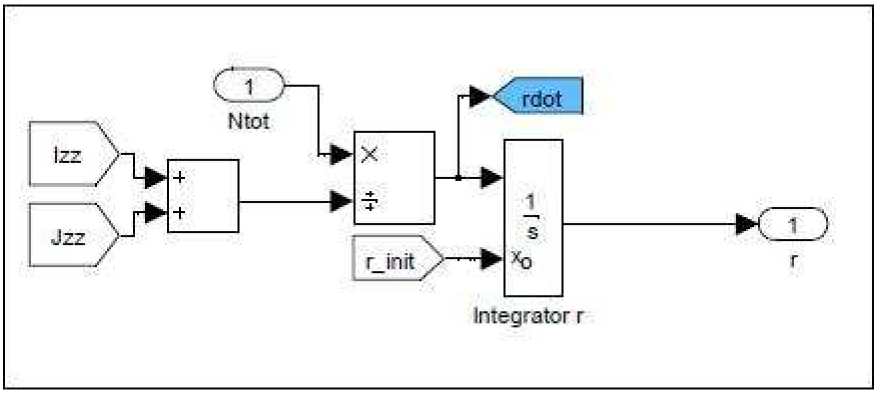

*Figure 3.6 – Yaw equation block*

*Roll equation block*

This block includes the model of the yaw equation. In a similar way as in the previous two blocks the equation had to be rewritten in the following form:

$$\dot{p} = \frac{K}{I_x} \tag{3.4}$$

where it is solved after the roll acceleration. The roll equation is much simpler than the other three, as it includes only the moment of inertia, the total roll moment and the roll acceleration.

As shown in the following figure, the right hand side part of the equation is created and after that the value is integrated in order to provide roll turn rate. In the integrator the initial value of the roll turn rate, as given by the user in the simulation inputs, is used as initial condition.

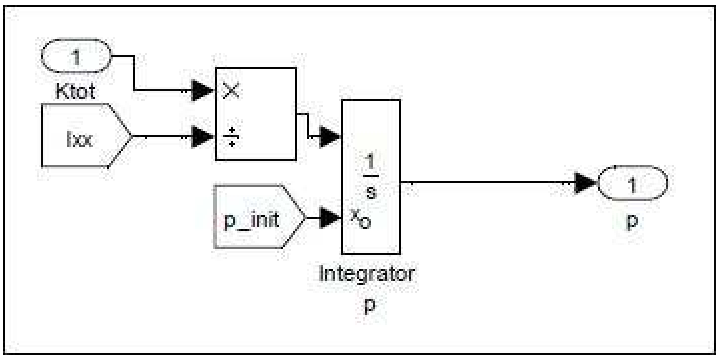


*Figure 3.7– Roll equation block*

## 3.3.3 Simulation input

As every block, the dynamics block uses a lot of values as inputs in order to calculate the desired outputs. There are three categories of input values. The main characteristics, which are given via the input file (i.e. ship length $L_{BP}$) , the simulation inputs (operational parameters), which are given by the user in the Simulink model (i.e. initial surge velocity), and finally the internal inputs, which are produced by another block and feed the dynamics block as input (i.e. total surge force $X$).

The values needed for calculation by the dynamics block are shown in the following table. The first column indicates the type and source of each value.

*Table 3.4*

| Type/Source | | Value | Description |
|---|---|---|---|
| Input file | | $I_z$ | moment of inertia (z-axis), [$kg\ m^2$] |
| Input file | | $I_x$ | moment of inertia (x-axis), [$kg\ m^2$] |
| Simulation input | | $u_{init}$ | initial surge velocity, [$m/s$] |
| Simulation input | | $v_{init}$ | initial sway velocity, [$m/s$] |
| Simulation input | | $r_{init}$ | initial yaw turn rate, [$rad/s$] |
| Simulation input | | $p_{init}$ | initial roll turn rate, [$rad/s$] |
| Internal input | Ship forces block | $X$ | total surge force, [$N$] |
| Internal input | Ship forces block | $Y$ | total sway force, [$N$] |
| Internal input | Ship forces block | $N$ | total yaw moment, [$Nm$] |
| Internal input | Ship forces block | $K$ | total roll moment, [$Nm$] |
| Internal input | Utility block | $m$ | ship mass, [$kg$] |
| Internal input or input file | Utility block | $m_x$ | ship added mass (x-axis), [$kg$] |
| Internal input or input file | Utility block | $m_y$ | ship added mass (y-axis), [$kg$] |
| Internal input or input file | Utility block | $J_z$ | added moment of inertia (z-axis), [$kg\ m^2$] |

### 3.3.4 Example

This paragraph includes a simple presentation of how the dynamics block works. The simulation was made for the vessel ESSO OSAKA case study, which is a reference set of experimental data for such studies (see 4.2.1). The characteristics of the ship are presented in chapter 4.

The forces and moments acting on ship, that are the input of the dynamics block, are shown in figures 3.8-3.11 as calculated by the model for this case study ship. The manoeuvre that was simulated was the left turning circle with no initial rudder turn and after one minute the rudder started to turn with turn rate of 2.34 deg/s and maximum angle of 34 degrees. The simulation time was 3600 seconds. The rest of the simulation inputs are shown in following table.

*Table 3.5*

| Type/Source | Value | |
|---|---|---|
| $u_{init}$ | 7.8 | kn |
| $v_{init}$ | 0 | kn |
| $r_{init}$ | 0 | kn |
| $p_{init}$ | 0 | kn |
| $Te_{water}$ | 15 | °C |
| *Water type* | salt | - |
| *Sea Depth* | 55 | m |
| *Initial heading* | 0 | rad |
| *P/D* | 0.715 | - |
| $N_{propeller}$ | 42.25 | rps |
| Wind velocity | 0 | kn |
| Wind angle | 0 | deg |
| Air temperature | 20 | °C |
| Start time | 60 | s |
| Initial rudder angle | 0 | deg |
| Rudder turning rate | 2.34 | deg/s |
| Max. rudder angle | 34 | deg |

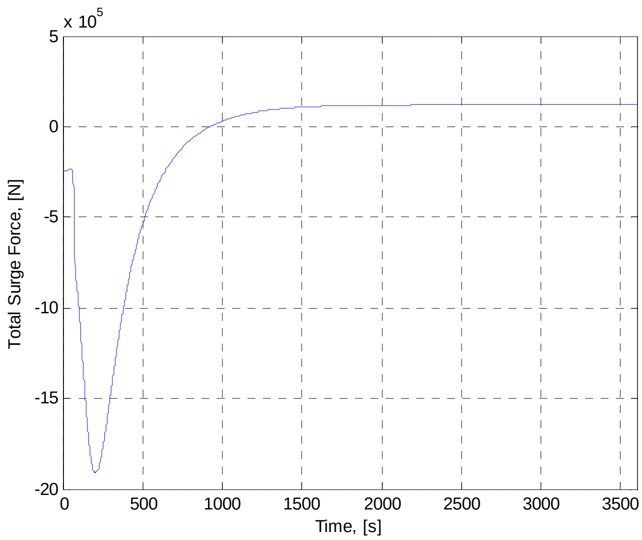


*Figure 3.8 – Total Surge force vs Time*

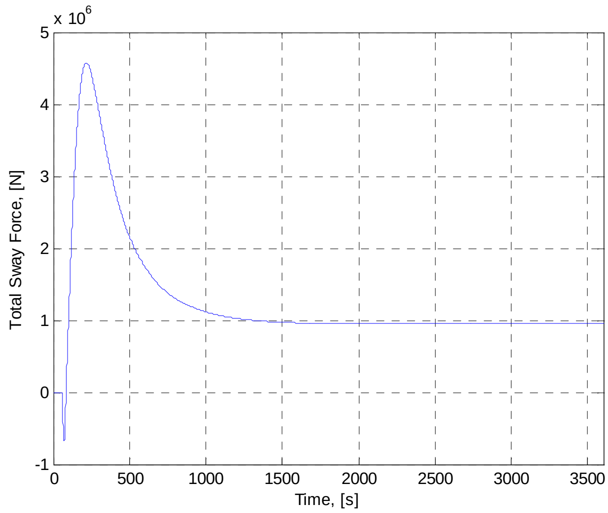


*Figure 3.9 – Total Sway force vs Time*

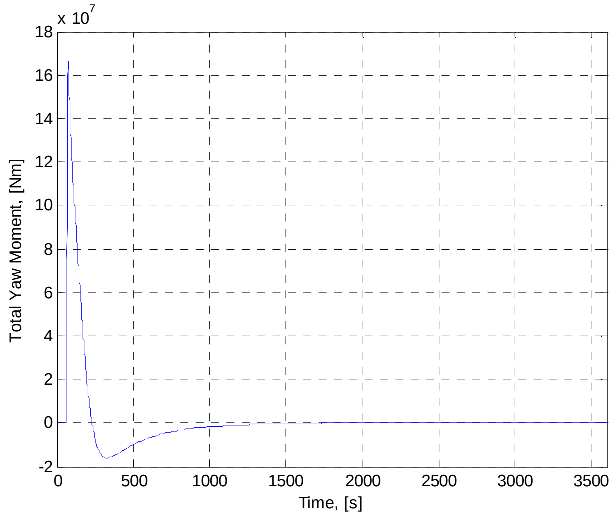


*Figure 3.10 – Total Yaw moment vs Time*

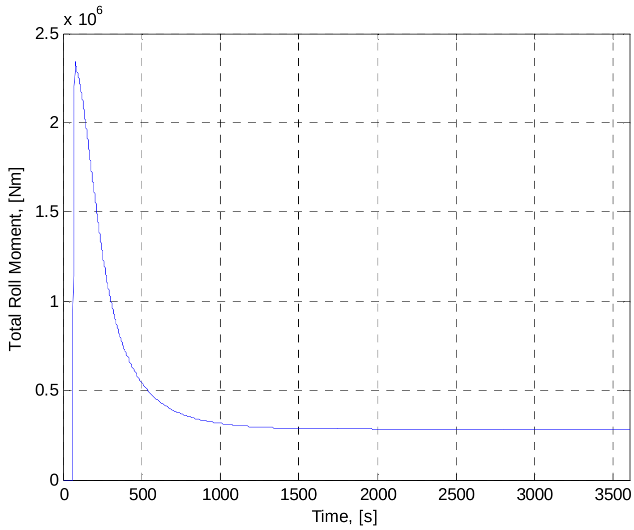


*Figure 3.11 – Total Surge force vs Time*

The output of the dynamics block, using as input the latter forces and moments is shown in the following figures. As stated before, these velocities and turn rates are inserted in the ship forces block and especially the yaw turn rate r is inserted in the post-processing block.

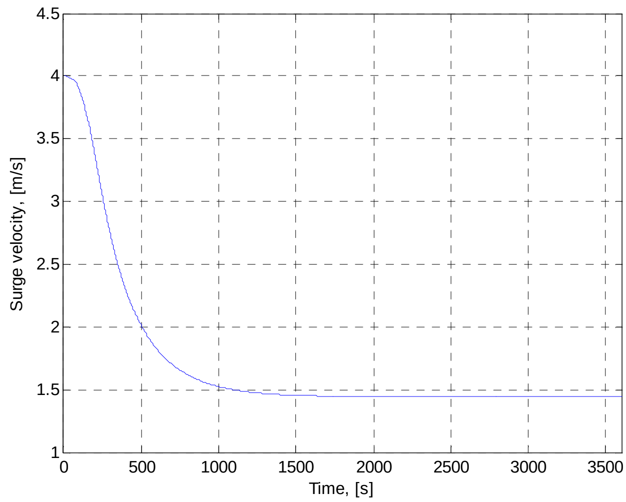


*Figure 3.12 – Surge velocity vs Time*

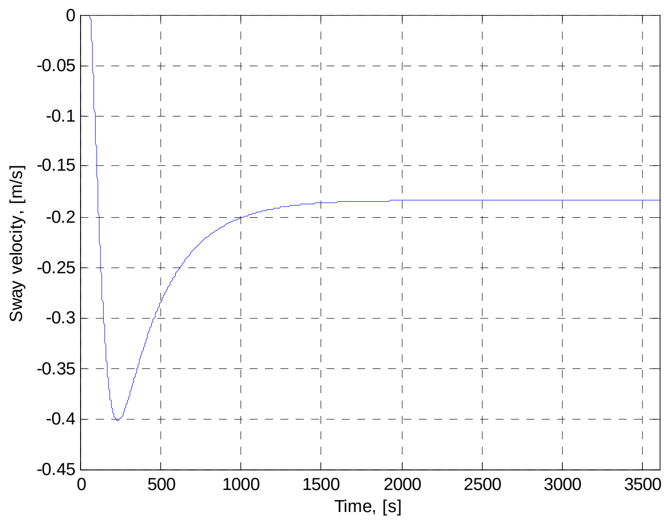


*Figure 3.13 – Swayvelocity vs Time*

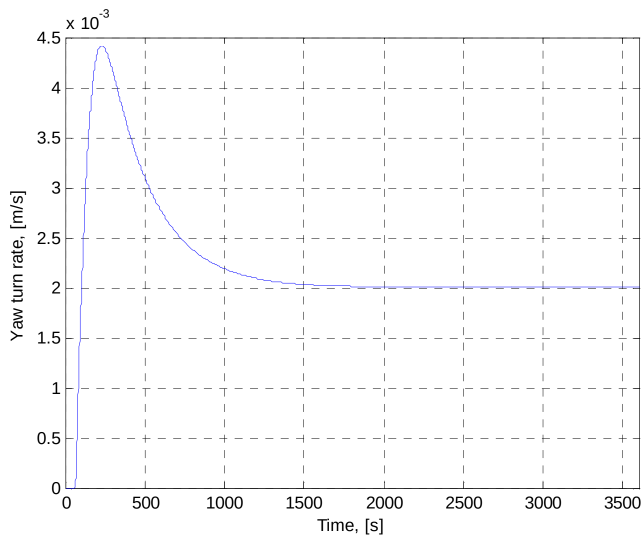


*Figure 3.14 – Yaw turn rate vs Time*

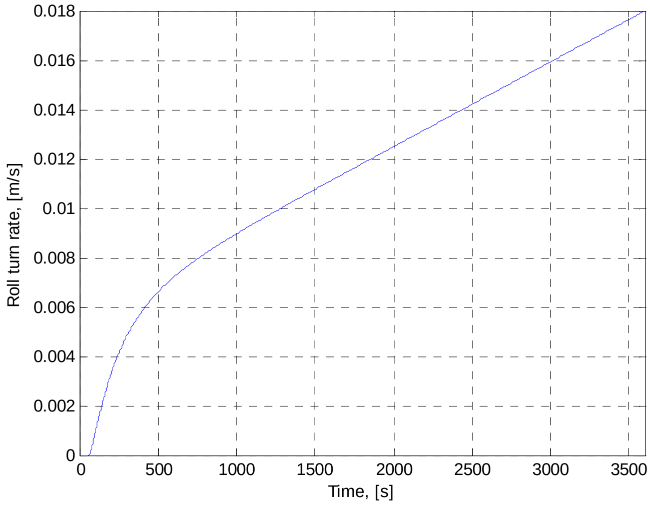


*Figure 3.15 – Roll turn rate vs Time*

## 3.3 Ship forces Block

### 3.3.1 Introduction

The ship forces block includes four main sub-blocks. As shown in the following figure, the four sub-blocks are the Resistance, Hydrodynamic forces, Propeller forces and finally Rudder forces sub-blocks including the corresponding model.

For consistency the output of the resistance block and the X force of the hydrodynamic forces block are added together, as the resistance is in fact a hydrodynamic force acting on the ship. Also, the propeller's sway force $Y_P$, yaw moment $N_P$, roll moment $K_P$ are being neglected in the adding blocks because their value is zero.

As shown in the following figure, all four blocks take as input the velocities and other geometrical or operational parameters, and produce the forces and moments acting by each ship component. The calculations take place in the s-function block. Each one of them includes the matlab code (Appendix A) that describes the model, as presented in Chapter 2.

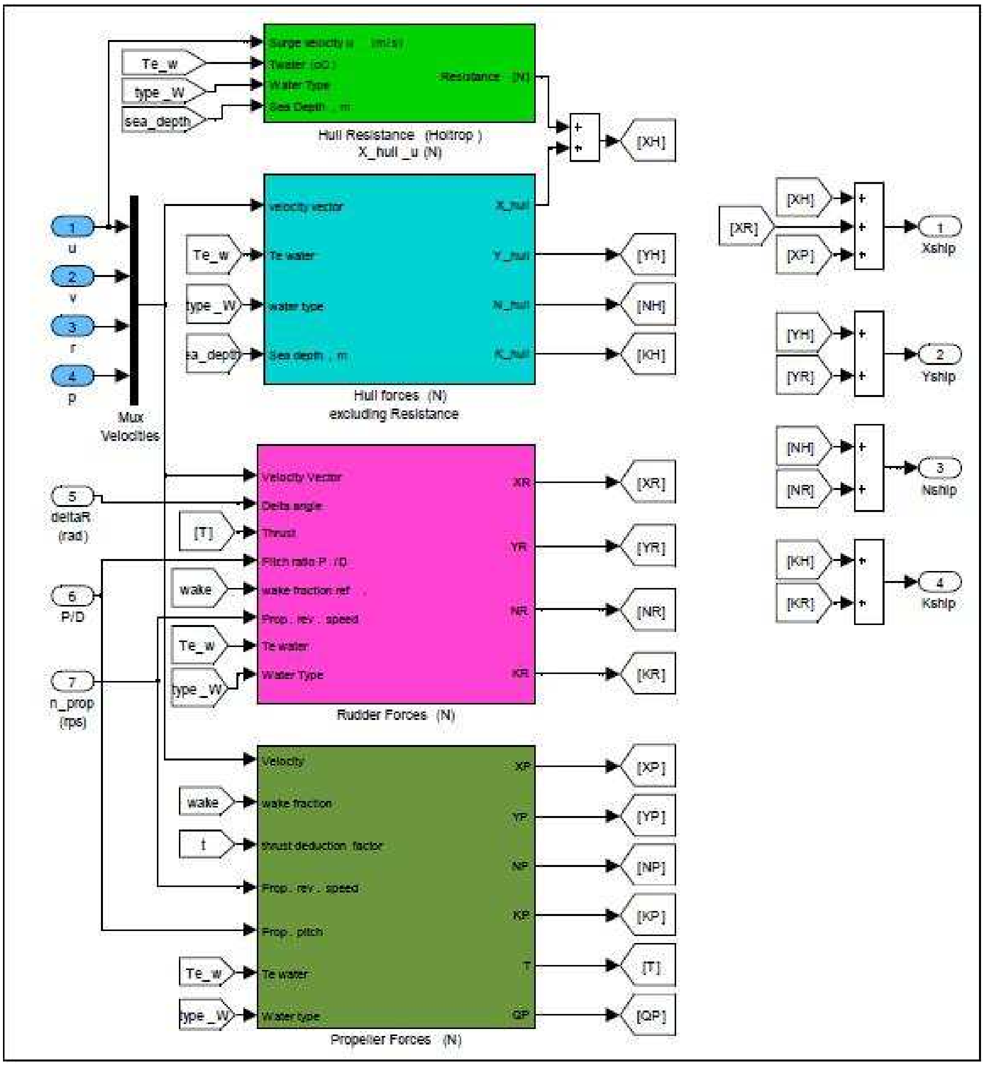


*Figure 3.16 – Ship forces block interior*

## 3.3.2 Resistance Block

### 3.3.2.1 Mathematical Model

This block calculates the ship Resistance, using the Holtrop method, as described in paragraph 2.2.2. As shown in the following figure apart from the S-function block, where the calculations are made,another system was also used, which is actually a *logical if* system. S-funtions are blocks that use Matlab code in Simulink.

Fistly, the ship's surge velocity $u$ is going into the "if" block, where the sign is checked. If the velocity is positive, the resistance is negative, so the output of the s-function block is multiplied by *-1*. Otherwise it is multiplied by *1.2*, which is based on the estimation of a 20% increase of the resistance due to the fact that the ship hull is not so hydrodynamically efficient when moving astern. Due to lack of data in models regarding astern resistance, a commonly used approach is to consider the astern resistance to be 120% forward resistance.

The input of the s-function is the absolute value of surge velocity and thus the output is also positive. After the multiplication block, accordingly to the ship direction the right sign is given to the force. The Matlab code included in the s-function block is shown in appendix A1.

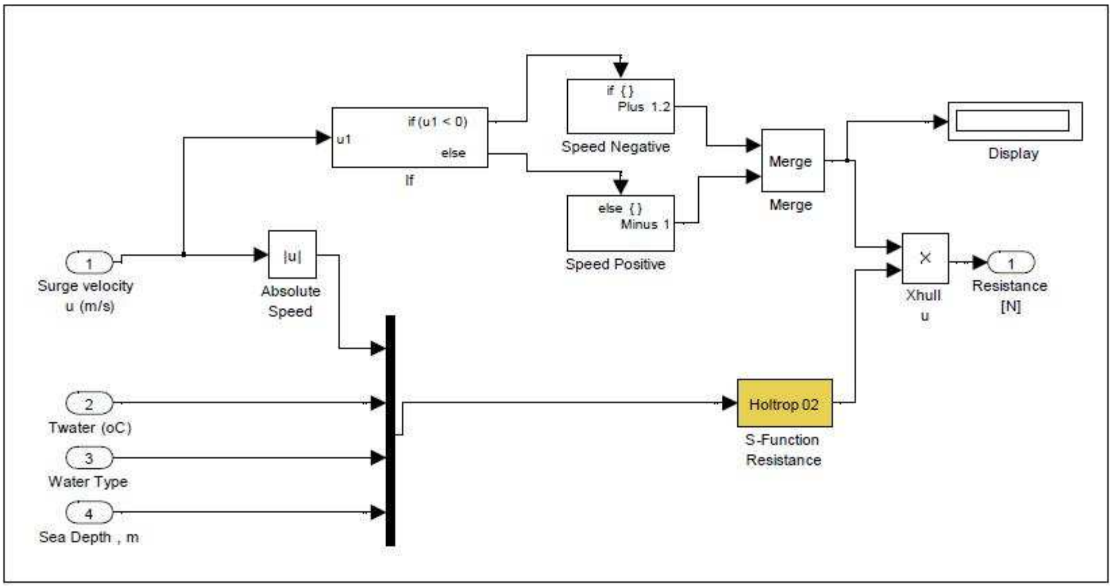


*Figure 3.17 – Resistance block*

### 3.3.2.2 Simulation input

As described previously there are three categories of input values. The main characteristics, which are given via the input file (i.e. ship length $L_{BP}$) , the simulation inputs (operational parameters), which are given by the user in the simulink model (i.e. initial surge velocity), and finally the internal inputs, which are produced by another block and feed the dynamics block as input (i.e. total surge force $X$).

The values needed for calculation by the Resistance block are shown in the following table. The first column indicates the type and source of each value.

As presented in the previous figure there are only four simulation inputs. The water temperature and type (fresh or sea) are needed for the calculation of the water's density, whereas the sea depth for the calculation of the shallow water effect. Finally, in the first time step the initial surge velocity, as given by the user is used, and afterwards the block uses as input the value, as calculated by the dynamics block.

*Table 3.6*

| Type/Source | | Value | Description |
|---|---|---|---|
| Input file | | *LBP* | Lenght bewtween perpendiculars, [m] |
| Input file | | *LWL* | Lenght in waterline, [m] |
| Input file | | *B* | Breadth, [m] |
| Input file | | *TF* | Depth at for perpendicular, [m] |
| Input file | | *TA* | Depth at aft perpendicular, [m] |
| Input file | | *VOL* | Displacement, [m3] |
| Input file | | *LCB* | Longitudal Centre of Buoyancy, [m] |
| Input file | | *CWP* | Waterplane Coefficient, [-] |
| Input file | | *CM* | Midship Section Coefficient, [-] |
| Input file | | *S* | Wetted Hull Area, [m2] |
| Input file | | *CSTERN* | cstern , [-] |
| Input file | | *NRUD* | Number of Rudders, [-] |
| Input file | | *SRUD* | Wetted area of rudder, [m2] |
| Input file | | *CRUD* | coefficient of rudder, [-] |
| Input file | | *SAPP i* | Wetted area of rudder behind skeg, [m2] |
| Input file | | | Wetted area of rudder behind stern, [m2] |
| Input file | | | Wetted area of twin-screw balance rudders, [m2] |
| Input file | | | Wetted area of shaft brackets, [m2] |
| Input file | | | Wetted area of skeg, [m2] |
| Input file | | | Wetted area of strut bossings, [m2] |
| Input file | | | Wetted area of hull bossings, [m2] |
| Input file | | | Wetted area of shafts, [m2] |
| Input file | | | Wetted area of stabilizer fins, [m2] |
| Input file | | | Wetted area of dome, [m2] |
| Input file | | | Wetted area of bilge keels, [m2] |
| Input file | | *ABT* | cross sectional area of bulbous bow, [m2] |
| Input file | | *HB* | centroid of bulbous bow cross section to keel, [m] |
| Input file | | *AT* | area of immersed transom, [m2] |
| Input file | | *LR* | Lenght of run, [m] |
| Input file | | *IE* | half angle of entrance, [-] |
| Input file | | *NBT* | Number of Bow thrusters, [-] |
| Input file | | *DBT (1st)* | Diameter of 1st bow thrusters, [m] |
| Input file | | *DBT (2nd)* | Diameter of 2nd bow thrusters, [m] |
| Input file | | | |
| Input file | | *DBT (i)* | Diameter of i bow thrusters, [m] |
| Simulation input | | $u_{init}$ | initial surge velocity, [*m/s*] |
| Simulation input | | $T_{water}$ | water temperature, [˚*C*] |
| Simulation input | | $Type_{water}$ | fresh or sea water, [-] |
| Simulation input | | $Sea_d$ | sea depth, [*m*] |
| Internal input | Dynamics block | $u$ | surge velocity, [*m/s*] |

### 3.3.2.3 Example

In this subparagraph a simple presentation of how this block works is shown. The ship used for this simulation is the *Hollandia* which has the following particulars, given via the input file(Table 3.7). This ship was selected, because it was used in the paper on which the model is based (Holtrop et al. 1982).

*Table 3.7*

| Type/Source | Value | Description |
|---|---|---|
| 193,1 | *LBP* | Lenght bewtween perpendiculars, [m] |
| 196,7 | *LWL* | Lenght in waterline, [m] |
| 30,8 | *B* | Breadth, [m] |
| 9 | *TF* | Depth at for perpendicular, [m] |
| 9 | *TA* | Depth at aft perpendicular, [m] |
| 31282 | *VOL* | Displacement, [m3] |
| -0,03 | *LCB* | Longitudal Centre of Buoyancy, [m] |
| 0,708 | *CWP* | Waterplane Coefficient, [-] |
| 0,965 | *CM* | Midship Section Coefficient, [-] |
| 0 | *S* | Wetted Hull Area, [m2] |
| 3 | *CSTERN* | cstern , [-] |
| 1 | *NRUD* | Number of Rudders, [-] |
| 60 | *SRUD* | Wetted area of rudder, [m2] |
| 1,4 | *CRUD* | coefficient of rudder, [-] |
| 0 | *ABT* | cross sectional area of bulbous bow, [m2] |
| 4,84 | *HB* | centroid of bulbous bow cross section to keel, [m] |
| 0 | *AT* | area of immersed transom, [m2] |
| 0 | *LR* | Lenght of run, [m] |
| 0 | *IE* | half angle of entrance, [-] |
| 1 | *NBT* | Number of Bow thrusters, [-] |
| 2,6 | *DBT (1st)* | Diameter of 1st bow thrusters, [m] |

The simulation was done for a speed range from 2 to 20 kn per 2. The water temperature was 15°C and the water type was salt. Finally, the sea depth was chosen 150m in order to eliminate the shallow water effect. The wind was absent, and the manoeuvre was just the straight line, with no deviation of the rudder. These simulation inputs are shown in table 3.8.

*Table 3.8*

| Type/Source | Value | |
|---|---|---|
| $u_{init}$ | 7.8 | kn |
| $v_{init}$ | 0 | kn |
| $r_{init}$ | 0 | kn |
| $p_{init}$ | 0 | kn |
| $Te_{water}$ | 15 | °C |
| *Water type* | salt | - |
| *Sea Depth* | 150 | m |
| *Initial heading* | 0 | rad |
| *P/D* | 0.715 | - |
| $N_{propeller}$ | 42.25 | rps |
| Wind velocity | 0 | kn |
| Wind angle | 0 | deg |
| Air temperature | 20 | °C |
| Start time | 60 | s |
| Initial rudder angle | 0 | deg |
| Rudder turning rate | 0 | deg/s |
| Max. rudder angle | 0 | deg |

The results of the simulation are given in table 3.9. It includes the results of the present model as well as the results of the paper by Holtrop. As shown in fig. 3.18 the result are acceptable as the deviation is not great. As observed the deviation gets greater at larger speed.

*Table 3.9*

| Velocities [kn] | Resistance [kN] | |
|---|---|---|
| | Model | Holtrop |
| 1 | 2,5 | 3,3 |
| 2 | 9,2 | 12,2 |
| 3 | 19,7 | 26,3 |
| 4 | 33,9 | 45,2 |
| 5 | 51,5 | 68,9 |
| 6 | 72,7 | 97,4 |
| 7 | 97,3 | 130,4 |
| 8 | 125,3 | 168,0 |
| 9 | 156,8 | 210,1 |
| 10 | 191,7 | 256,8 |
| 11 | 230,3 | 308,3 |
| 12 | 272,9 | 365,2 |
| 13 | 320,2 | 428,3 |
| 14 | 373,1 | 498,7 |
| 15 | 432,8 | 578,3 |
| 16 | 500,7 | 668,9 |
| 17 | 578,6 | 773,1 |
| 18 | 668,6 | 893,6 |
| 19 | 771,8 | 1032,2 |
| 20 | 890,3 | 1191,5 |
| 21 | 1030,4 | 1380,0 |
| 22 | 1199,6 | 1607,2 |
| 23 | 1395,5 | 1870,1 |
| 24 | 1603,1 | 2148,7 |
| 25 | 1806,4 | 2423,0 |

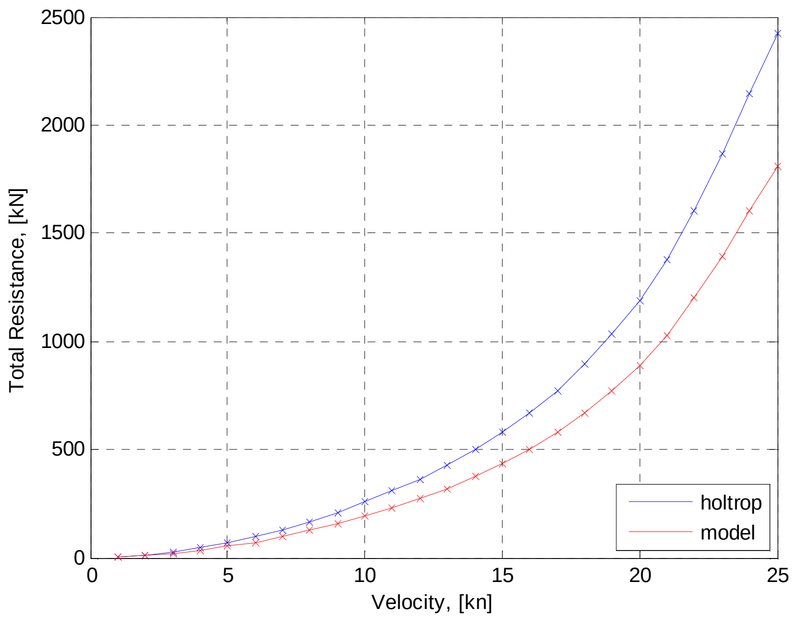


*Fig. 3.18 – Comparison of the results of the model and of Holtrop*

### 3.3.3 Hydrodynamic Forces Block

#### 3.3.3.1 Mathematical Model

By the rotational motion of the propeller, a force is produced known as thrust. This force is on the x-axis and it's direction is based on the direction of the propeller's rotation. Thus the forces and moments in all other directions are zero. The calculation of the total surge force is done in this block. As shown in the following figure this block only includes the inputs, the outputs and the s-function.

The s-function block includes the matlab code that implements the hull forces model based on paragraph 2.2.4 as shown in appendix A2.

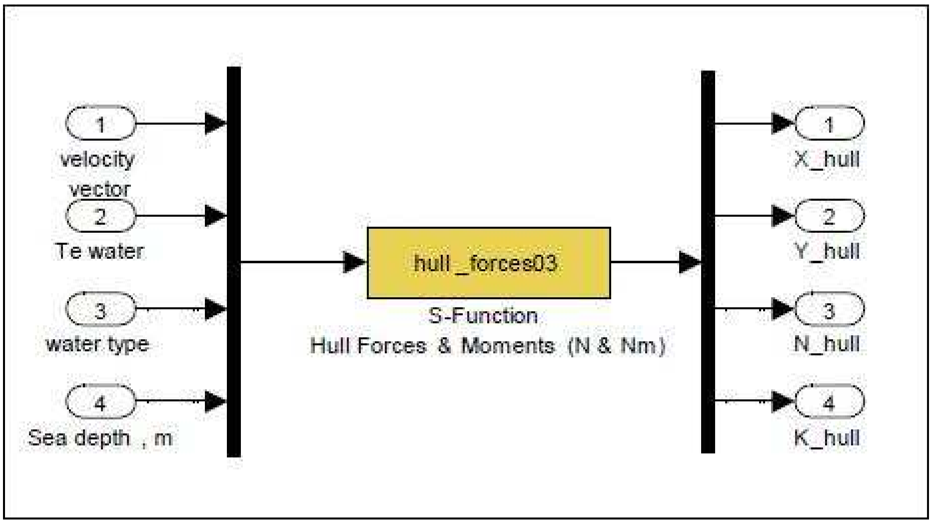


*Figure 3.19 – Hydrodynamic Forces Block interior*

#### 3.3.3.2 Simulation Input

The first and most important input of this block is the velocity vector, which includes all the velocity of the ship at each time step, as calculated by the dynamics block. Also there are the water temperature and type, needed for the calculation of the exact value of water density. Finally, there is the Sea depth needed for the calculation of the shallow water effect.

Apart from these inputs there are also the geometrical characteristics given by the input file. These include also the hydrodynamic derivatives. As described in section 3.2 the user defines the number of derivatives and after that there are four values for each derivative, the direction, the actual value (only if the calculation method is *user defined*), the index of the derivative and finally the method of calculation. Also, there is the *Number of bow thrusters* because it is needed for the variable indexing of the input file.

Finally, there are also the simulation inputs, given by the user in the model. All of them are used as input in the block. The initial velocities are used in the first time step and afterwards the velocities are calculated in the dynamics block. All the inputs of the block are shown in table 3.10.

*Table 3.10*

| Type/Source | | Value | Description |
|---|---|---|---|
| Input file | | *LBP* | Lenght bewtween perpendiculars, [m] |
| Input file | | *LWL* | Lenght in waterline, [m] |
| Input file | | *B* | Breadth, [m] |
| Input file | | *TF* | Depth at for perpendicular, [m] |
| Input file | | *TA* | Depth at aft perpendicular, [m] |
| Input file | | *VOL* | Displacement, [m3] |
| Input file | | *NBT* | Number of Bow thruster, [-] |
| Input file | | *NDER* | Number of hydrodynamic derivatives, [-] |
| Input file | | dir | direction of the derivative (X,Y,N,K) , [-] |
| Input file | | Hd | value of the derivative, [-] |
| Input file | | index | index of the derivative (i.e. uv2) , [-] |
| Input file | | method | method of calc. (0:user,1,2,3,5) , [-] |
| Input file | | dir | … |
| Input file | | Hd | … |
| Input file | | index | … |
| Input file | | method | … |
| Input file | | *DP* | Propeller diameter [*m*] |
| Input file | | $C_{WPA}$ | Aft. Waterplane coefficient, aft, [-] |
| Input file | | $C_{PA}$ | Aft. Prismatic coefficient, [-] |
| Input file | | *LCG* | longitudinal center of gravity as percentage from half length, positive for, negative aft, [*%*] |
| Input file | | $BP_{07}$ | half breadth at the height of 0.7R(propeller radius) in 2.0 station,[*m*] |
| Input file | | $BP_S$ | half breadth at the height of propeller shaft in 2.0 station,[*m*] |
| Input file | | $I_{zz}$ | moment of inertia at x-axis, [kg $m^2$] |
| Simulation input | | $u_{init}$ | initial surge velocity, [*m/s*] |
| Simulation input | | $v_{init}$ | initial sway velocity, [*m/s*] |
| Simulation input | | $r_{init}$ | initial yaw turn rate, [*rad/s*] |
| Simulation input | | $p_{init}$ | initial roll turn rate, [*rad/s*] |
| Simulation input | | $T_{water}$ | water temperature, [˚*C*] |
| Simulation input | | $Type_{water}$ | fresh or sea water, [-] |
| Simulation input | | $Sea_d$ | sea depth, [*m*] |
| Internal input | Dynamics block | $u$ | surge velocity, [*m/s*] |
| Internal input | Dynamics block | $v$ | sway velocity, [m/s] |
| Internal input | Dynamics block | $r$ | yaw turn rate, [rad/s] |
| Internal input | Dynamics block | $p$ | roll turn rate, [rad/s] |

### 3.3.3.3 Example

This sub-paragraph includes a simple presentation of how the hydrodynamic forces block works. The simulation was made for ESSO OSAKA case study. The characteristics of the ship are shown in chapter 4.

The simulation inputs are shown in table 3.11. The hydrodynamic derivatives where chosen to be estimated from various methods, as shown in table 3.12. They were all calculated by the model. It executed the turning circle manoeuvre and the simulation time was 3600s.

*Table 3.11*

| Type/Source | Value | |
|---|---|---|
| $u_{init}$ | 7.8 | kn |
| $v_{init}$ | 0 | kn |
| $r_{init}$ | 0 | kn |
| $p_{init}$ | 0 | kn |
| $Te_{water}$ | 15 | °C |
| *Water type* | salt | - |
| *Sea Depth* | 55 | m |
| *Initial heading* | 0 | rad |
| *P/D* | 0.715 | - |
| $N_{propeller}$ | 42.25 | rps |
| Wind velocity | 0 | kn |
| Wind angle | 0 | deg |
| Air temperature | 20 | °C |
| Start time | 60 | s |
| Initial rudder angle | 0 | deg |
| Rudder turning rate | 2.34 | deg/s |
| Max. rudder angle | 34 | deg |

*Table 3.12*

| derivative | method |
|---|---|
| $m_x$ | 3 |
| $m_y$ | 3 |
| $J_{zz}$ | 2 |
| $X_{vv}$ | 3 |
| $X_{rr}$ | 3 |
| $X_{vr}$ | 3 |
| $Y_v$ | 4 |
| $Y_r$ | 4 |
| $Y_{vvv}$ | 3 |
| $Y_{rrr}$ | 3 |
| $Y_{vvr}$ | 3 |
| $Y_{vrr}$ | 4 |
| $N_v$ | 4 |
| $N_r$ | 4 |
| $N_{vvv}$ | 3 |
| $N_{rrr}$ | 3 |
| $N_{vvr}$ | 3 |
| $N_{vrr}$ | 3 |

The output of the hydrodynamic forces block is shown in the following figures. As there are no roll hydrodynamic derivatives, the roll moment is zero, so the respective figure is neglected.

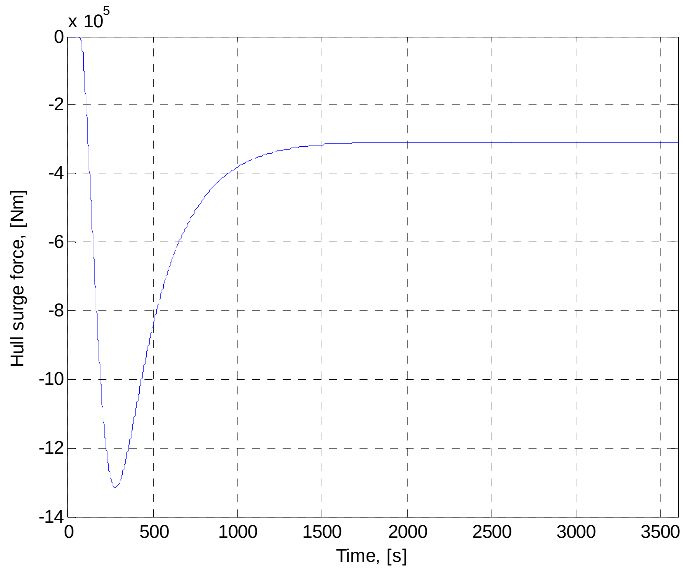


*Figure 3.20 – Total Surge force vs Time*

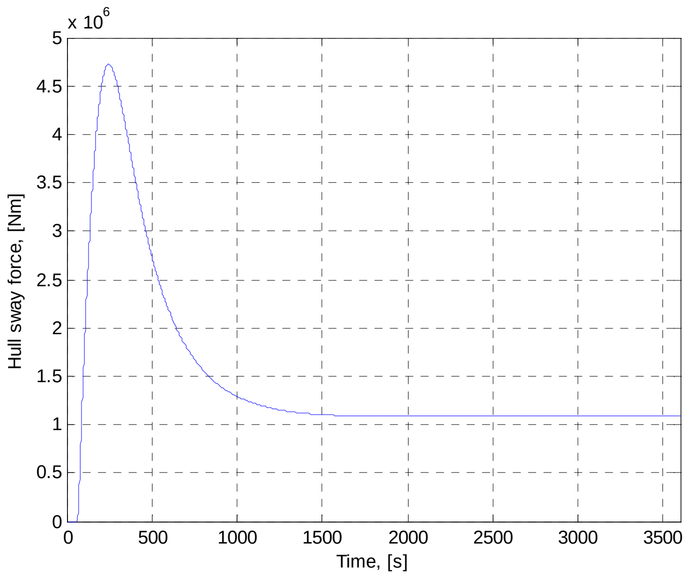


*Figure 3.21 – Total Sway force vs Time*

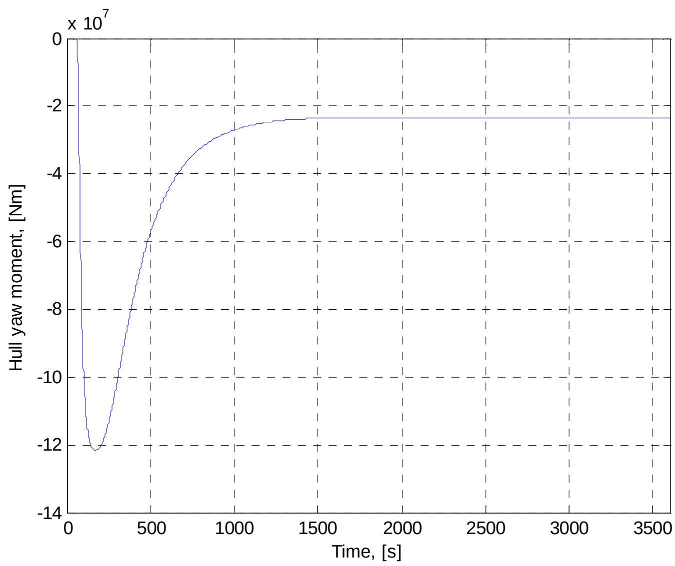


*Figure 3.22 – Total Yaw moment vs Time*

### 3.3.4 Propeller Forces Block

#### 3.3.4.1 Mathematical Model

Apart from the resistance, which is a hydrodynamic force but only on x-axis, there also other hydrodynamic forces acting on ship due to interaction of the ship with the surrounding fluid. These forces are calculated by this block. As shown in the following figure this block is much simpler than the resistance block, as it only includes the inputs, the outputs and the s-function.

The s-function block includes the matlab code based on paragraph 2.2.4 as shown in appendix A3.

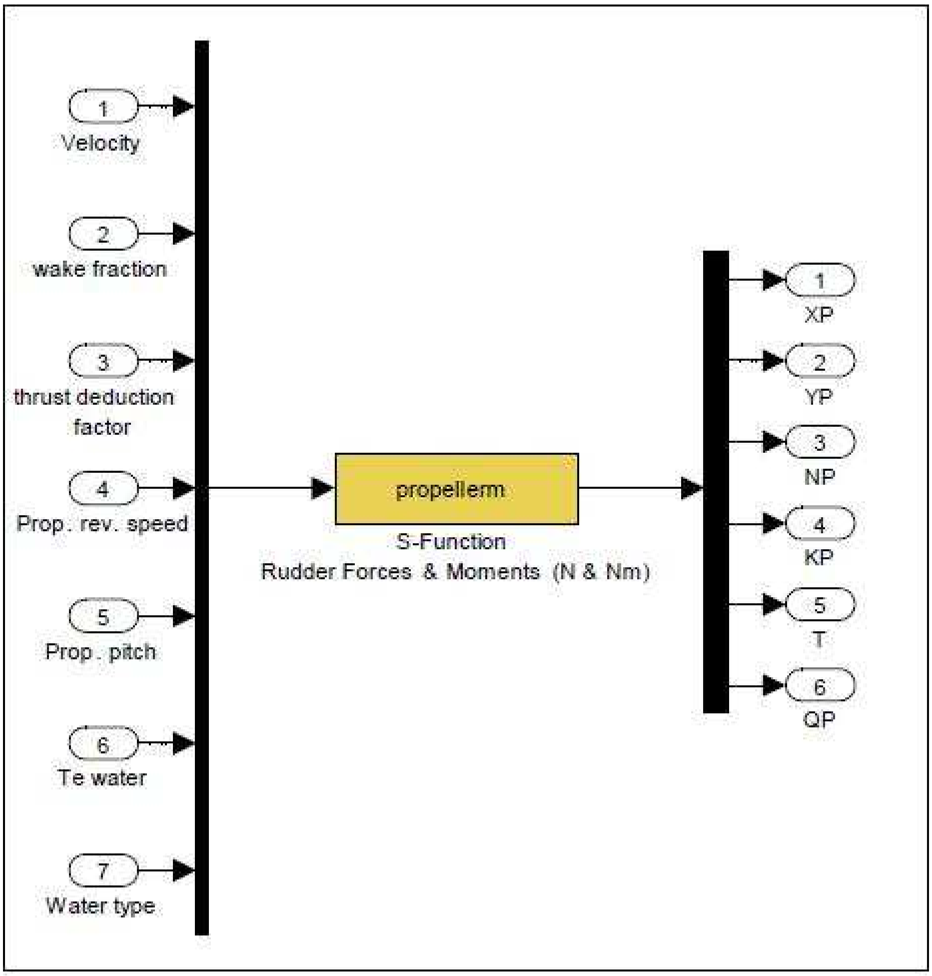


*Figure 3.23 – Propeller Forces Block interior*

#### 3.3.4.2 Simulation Input

As shown in the block figure there are some inputs. The first one is the surge velocity of the ship. After that, there are the wake fraction and the thrust deduction factor, which can be calculated in the *Utility block* or can be given by the user via the input file (for calculation by the block the user has to give the value 0). The rest of the block’s inputs are simulation inputs given by the user in the model and they are the propeller’s rotational speed, the propeller’s pitch amd the water type and temperature. The pitch is given as a simulation input in order to simulate a ship with controllable pitch propeller (CPP).

There are also some geometrical inputs given via the input file and some characteristics of the propeller. The block also needs the *Number of bow thrusters* and the *Number of Derivatives* from the input file in order to assign the proper values to the proper variables due to the variable indexing of the input file.

Finally, there are also the simulation inputs, given by the user in the model. All of them are used as input in the block. The initial velocity are used in the first time step and afterwards the velocities are calculated in the dynamics block. All the inputs of the block are shown in table 3.8.

*Table 3.13*

| Type/Source | | Value | Description |
|---|---|---|---|
| Input file | | *LBP* | lenght bewtween perpendiculars, [m] |
| Input file | | *NBT* | number of Bow thruster, [-] |
| Input file | | *NDER* | number of hydrodynamic derivatives, [-] |
| Input file | | *NPROP* | number of propellers , [-] |
| Input file | | *DP* | propeller diameter , [*m*] |
| Input file | | *AAE* | propeller expanded blade area ratio, [-] |
| Input file | | *NZ* | number of propeller’s fins , [-] |
| Input file | | *xP* | longitudinal distance of propeller from ship’s center , [-] |
| Simulation input | | $u_{init}$ | initial surge velocity, [*m/s*] |
| Simulation input | | $N_{prop}$ | propeller’s revolution speed, [*rps*] |
| Simulation input | | *Pitch* | propeller's pitch, [*m*] |
| Simulation input | | $T_{water}$ | water temperature, [˚*C*] |
| Simulation input | | $Type_{water}$ | fresh or sea water, [-] |
| Internal input | Dynamics block | $u$ | surge velocity, [*m/s*] |
| Internal input/ input file | Utility block/- | $w$ | wake fraction, [-] |
| Internal input/ input file | Utility block/- | $t$ | thrust deduction factor, [-] |

### 3.3.4.3 Example

This sub-paragraph includes an example of how the propeller forces block works. The simulation was made for ESSO OSAKA case study. The characteristics of the ship are given in chapter 4.

The simulation inputs are given in table 3.11. The pitch-diameter ratio and the propeller’s rotational speed were chosen based on actual values of the ESSO OSAKA and especially the rotational speed was also based on actual values depending on the velocity the ship in order to correspond to true values. There was no wind and the ship executed the zigzag manoeuvre and the simulation time was 3600s.

*Table 3.14*

| Type/Source | Value | |
|---|---|---|
| $u_{init}$ | 7.8 | kn |
| $v_{init}$ | 0 | kn |
| $r_{init}$ | 0 | kn |
| $p_{init}$ | 0 | kn |
| Tewater | 15 | °C |
| Water type | salt | - |
| Sea Depth | 55 | m |
| Initial heading | 0 | rad |
| P/D | 0.715 | - |
| Npropeller | 42.25 | rps |
| Wind velocity | 0 | kn |
| Wind angle | 0 | deg |
| Air temperature | 20 | °C |
| Rudder turning rate | 2.34 | deg/s |
| Zigzag angle | -20 | deg |

As described in previous paragraphs this block has only one force as output, which is the surge force. Traditionally, apart from the thrust $T$, the other item that is interesting for the engineer is the propeller's moment $Q$. Thus the following output figure includes also the moment.

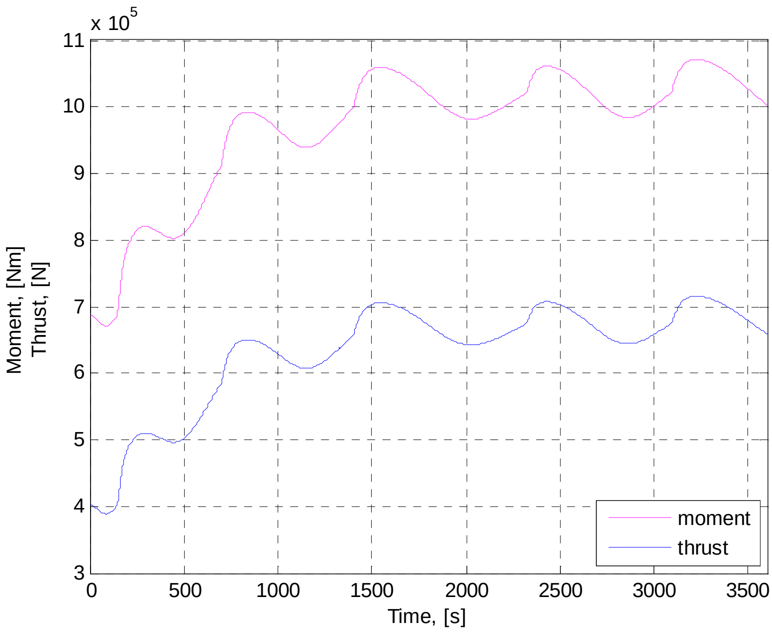


*Figure 3.24 – Propeller's Moment and Thrust vs Time*

## 3.3.5 Rudder Forces Block

### 3.3.5.1 Mathematical Model

By the operation of the rudder, its surface is turned around an axis. The flow encounters the surface and a force is induced, called the *normal force*. This force can be analyzed into two component forces on x and y axes and transferred to the ship's centre of gravity producing two moments. The value of the forces and moments depends on the rudder angle.

The block includes the inputs, the outputs and the s-function block. The latter includes the matlab code which calculates the forces using the model described in paragraph 2.2.5(Appendix A4).

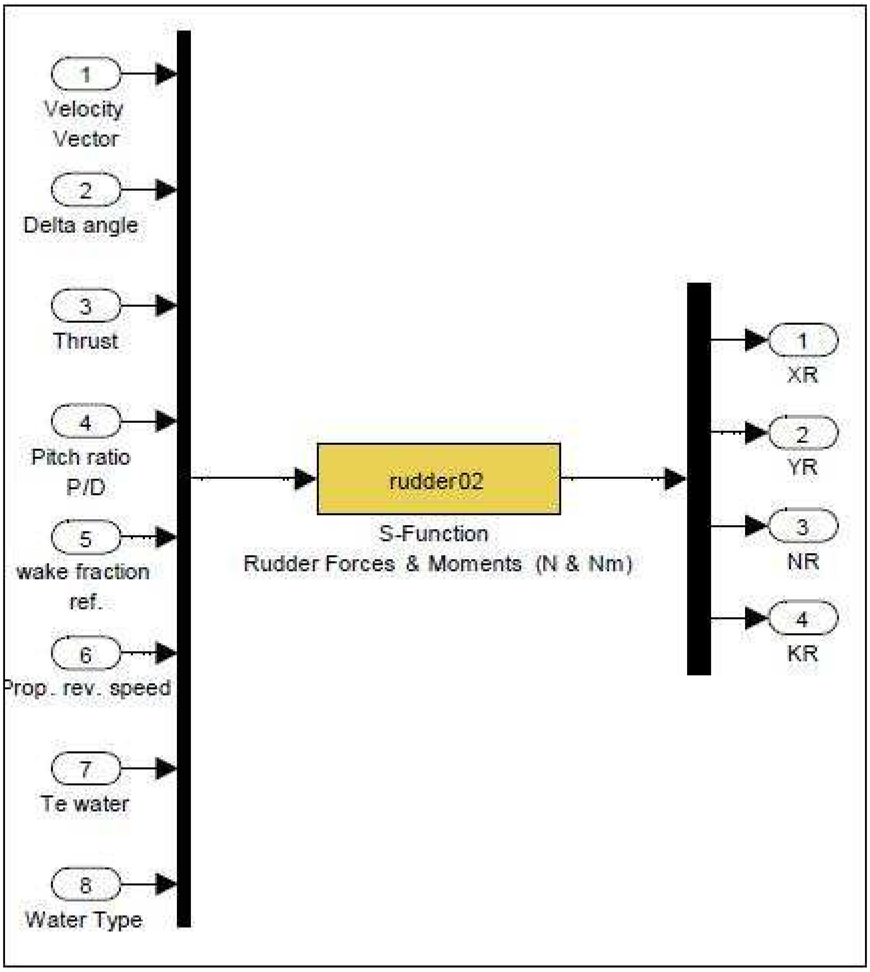


*Figure 3.25 – Rudder Forces Block interior*

### 3.3.5.2 Simulation Input

As shown in the block figure there are some inputs. The first one is the velocity vector, which includes both surge and sway velocities and both yaw and roll turn rates. In the first time step the block uses the initial values, as given by the user in the model, and after that it uses the values calculated in the *Dynamics* Block.The angle of rudder *delta*, $\delta$ is also an input as well as the *Thrust*, $T$. The Thrust is calculated by the propeller block. The pitch ratio is given as a simulation input, in order to simulate also ships with CPP. The wake fraction can be given by the user via the input file or can be calculated by the *Utility Block* (user gives value 0 in the input file). The propeller’s rotational speed is a simulation input, given by the user in the model and finally the type and temperature of the water are needed in order to calculate the exact water density.

There are also some geometrical inputs given via the input file and some characteristics of the propeller and especially the rudder, such as the rudder height, surface etc. The block also needs the *Number of bow thrusters* and the *Number of Derivatives* from the input file in order to assign the proper values to the proper variables due to the variable indexing of the input file.

Finally, there are also the simulation inputs, given by the user in the model. All of them are used as input in the block. All the inputs of the block are shown in table 3.9.

*Table 3.15*

| Type/Source | | Value | Description |
|---|---|---|---|
| Input file | | *LBP* | Lenght bewtween perpendiculars, [m] |
| Input file | | *B* | Breadth, [m] |
| Input file | | *TF* | Depth at for perpendicular, [m] |
| Input file | | *TA* | Depth at aft perpendicular, [m] |
| Input file | | *VOL* | Displacement, [m3] |
| Input file | | *NBT* | number of Bow thruster, [-] |
| Input file | | *NDER* | number of hydrodynamic derivatives, [-] |
| Input file | | *DP* | propeller diameter , [*m*] |
| Input file | | $x_P$ | longitudinal distance of propeller from ship’s center , [*m*] |
| Input file | | $x_R$ | longitudinal distance of rudder from ship’s center,[*m*] |
| Input file | | $z_R$ | vertical distance of propeller from ship’s center,[*m*] |
| Input file | | $H_R$ | rudder height , [*m*] |
| Input file | | $\lambda$ | rudder aspect ratio, [-] |
| Input file | | $A_R$ | rudder surface , [$m^2$] |
| Simulation input | | $u_{init}$ | initial surge velocity, [*m/s*] |
| Simulation input | | $v_{init}$ | initial sway velocity, [*m/s*] |
| Simulation input | | $r_{init}$ | initial yaw turn rate, [*rad/s*] |
| Simulation input | | $p_{init}$ | initial roll turn rate, [*rad/s*] |
| Simulation input | | $u_{init}$ | initial surge velocity, [*m/s*] |
| Simulation input | | $v_{init}$ | initial sway velocity, [*m/s*] |
| Simulation input | | $\delta$ | rudder angle, [*rad*] |
| Simulation input | | *Pitch* | propeller's pitch, [*m*] |
| Simulation input | | $N_{prop}$ | propeller’s revolution speed, [*rps*] |
| Simulation input | | $T_{water}$ | water temperature, [*˚C*] |
| Simulation input | | $Type_{water}$ | fresh or sea water, [-] |
| Internal input | Ship Forces block (propeller sub-block) | *T* | Thrust, [*N*] |
| Internal input | Dynamics block | *u* | surge velocity, [*m/s*] |
| Internal input | Dynamics block | *v* | sway velocity, [m/s] |
| Internal input | Dynamics block | *r* | yaw turn rate, [rad/s] |
| Internal input | Dynamics block | *p* | roll turn rate, [rad/s] |
| Internal input/ input file | Utility block/- | *w* | wake fraction, [-] |

### 3.3.5.3 Example

This sub-paragraph includes a simple example of how the rudder forces block works. The simulation was made for ESSO OSAKA case study. The characteristics of the ship are shown in chapter 4.

The simulation inputs are shown in table 3.16. There was no wind and the ship executed the turning zigzag manoeuvre.

*Table 3.16*

| Type/Source | Value | |
|---|---|---|
| $u_{init}$ | 7.8 | kn |
| $v_{init}$ | 0 | kn |
| $r_{init}$ | 0 | kn |
| $p_{init}$ | 0 | kn |
| $Te_{water}$ | 15 | °C |
| *Water type* | salt | - |
| *Sea Depth* | 55 | m |
| *Initial heading* | 0 | rad |
| *P/D* | 0.715 | - |
| $N_{propeller}$ | 42.25 | rps |
| Wind velocity | 0 | kn |
| Wind angle | 0 | deg |
| Air temperature | 20 | °C |
| Start time | 60 | s |
| Initial rudder angle | 0 | deg |
| Rudder turning rate | 2.34 | deg/s |
| Zigzag angle | 10 | deg |

The output of the block is shown in the following figures. Due to the direct dependence of the rudder forces and moments on the rudder angle, it is also shown the deviation of the rudder angle through time.

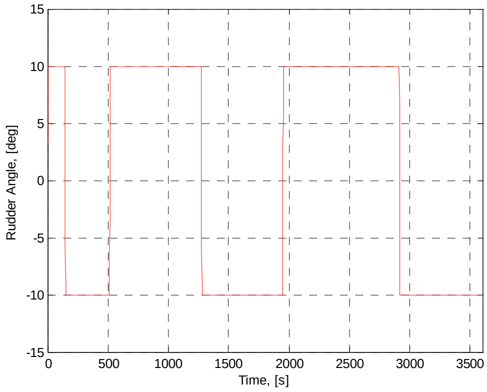


*Figure 3.26 – Rudder angle vs Time*

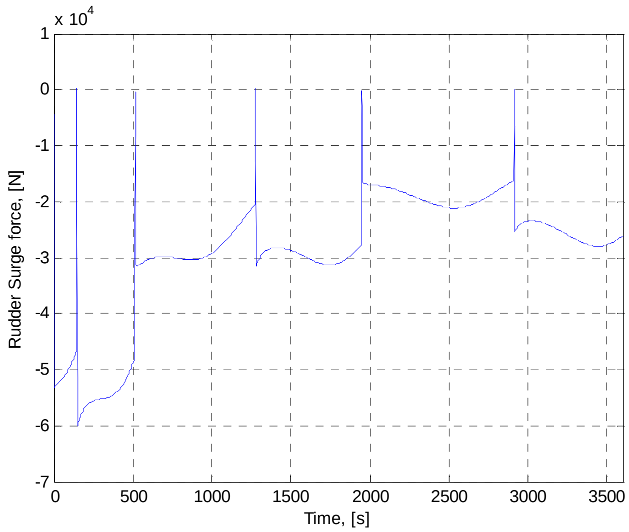


*Figure 3.27 – Total Surge force vs Time*

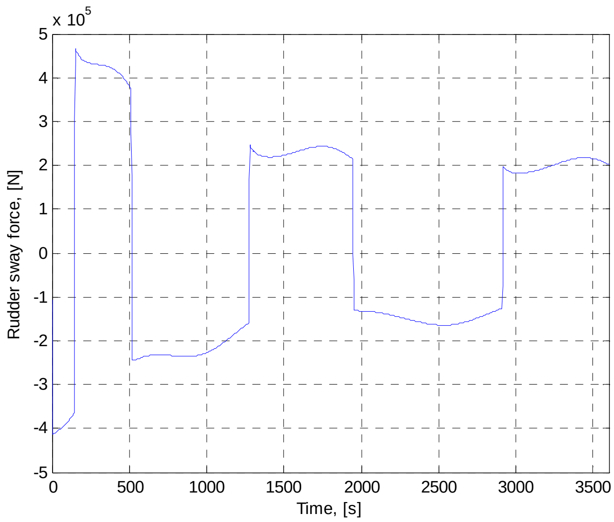


*Figure 3.28 – Total Sway force vs Time*

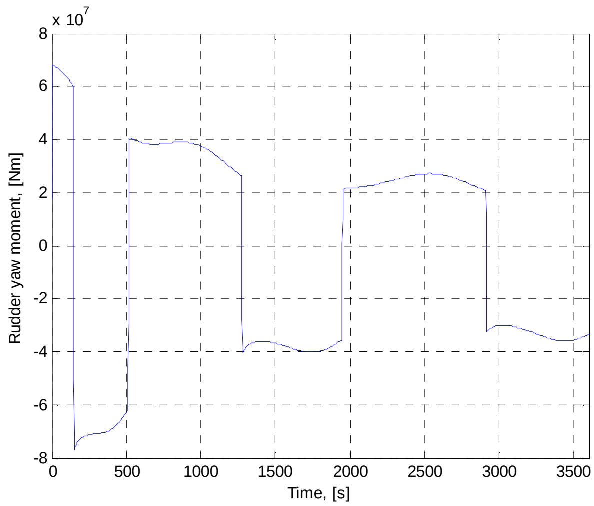


*Figure 3.29 – Total Yaw moment vs Time*

## 3.4 External Ship Forces

### 3.4.1 Mathematical model

Apart from the forces acting on ship due to its operation, there are also forces acting on ship due to external disturbances. Such forces can be wind, shallow water, bad weather etc. This block calculates the forces and moments acting on ship due to wind. As described in paragraph 2.3.2 the forces and moments depend on the speed and angle with which the wind hits the ship, but also on the ship's speed

The block and how it is connected with the rest of the models is depicted in the following figures. It produces two forces and two moments which are added to the forces and moments calculated by the *Ship Forces Block* respectively.If a future developer wants to extend the spectrum of external forces that the model simulates, it can be done by simply adding more blocks which produce forces and moments which can also be added in the total values.

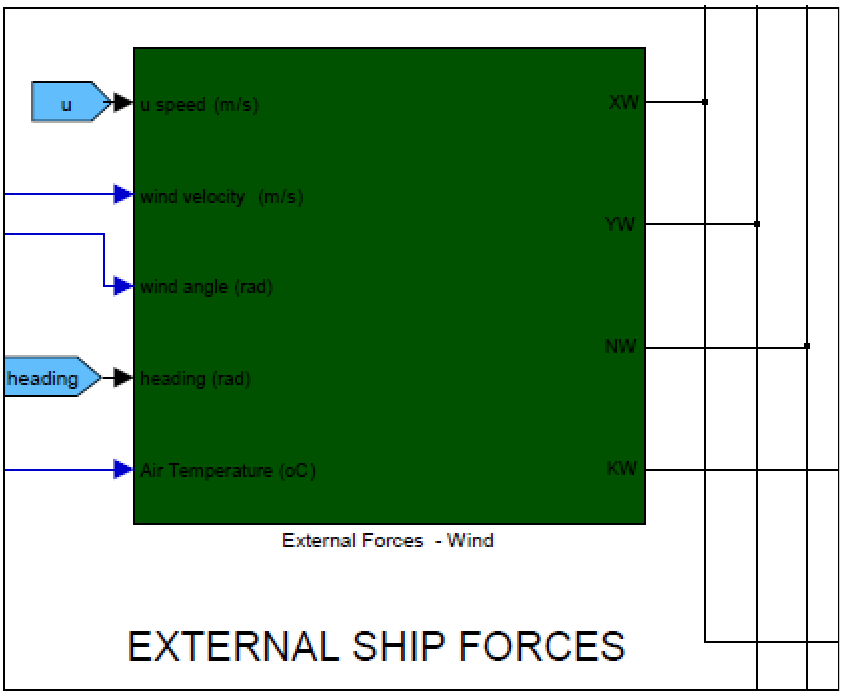


*Fig. 3.30 – external ship forces block*

The interior of this block is simple, as can be seen in the following figure. It includes the inputs, the outputs and the s-function block. The inputs are described in the next paragraph. The outputs are the forces and moments due to wind and are later added to the total values. The s-function block includes the matlab code which calculates the desired forces, using the model that is decribed in paragraph 2.3.2 (Appendix A5).

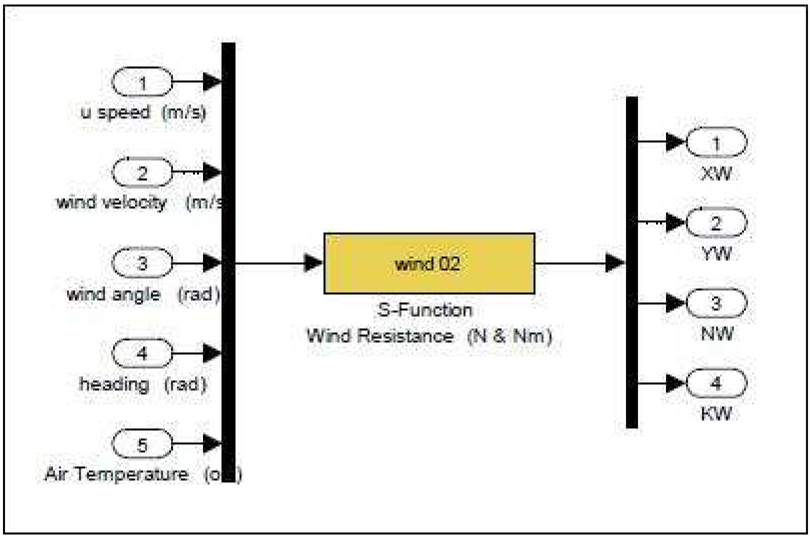


*Figure 3.31 – Wind Resistance Block interior*

## 3.4.2 Input file

As shown in the figure, this block has five inputs. The ship's surge velocity is first given by the user as a simulation input but after the first time step it is given by the dynamics block. The rest of the input are given by the user in the model. The wind velocity and angle are used for interpolation in the matrices (see par. 2.3.2), the heading is needed in order to find the relative angle that the wind hits the ship and the air temperature is used for the calculation of the air density.

Apart from the simulation inputs this block uses also some inputs from the input file. They are generally some geometrical characteristics of the hull, mostly areas and distances. Also the *Number of Bow thrusters* is needed due to the altering index of the input file.

*Table 3.17*

| Type/Source | | Value | Description |
|---|---|---|---|
| Input file | | *NBT* | Number of Bow thruster, [-] |
| Input file | | *LOA* | Length overall, [*m*] |
| Input file | | *AL* | Lateral projeced wind area, [$m^2$] |
| Input file | | *AS* | Lateral projectef area of superstructure, [$m^2$] |
| Input file | | *AT* | Transverse projected wind area, [$m^2$] |
| Input file | | *S* | Length of perimeter of lateral projection, [*m*] |
| Input file | | *C* | Distance from bow to centroid of lateral projected area, [*m*] |
| Input file | | *M* | Number of distinct groups of masts or king posts, [-] |
| Simulation input | | $u_{init}$ | initial surge velocity, [*m/s*] |
| Simulation input | | $w$ | wind velocity, [*m/s*] |
| Simulation input | | $w_{angle}$ | initial surge velocity, [*rad*] |
| Simulation input | | $heading$ | ship's heading, [*rad*] |
| Simulation input | | $T_{air}$ | air temperature, [˚*C*] |
| Internal input | Dynamics block | $u$ | surge velocity, [*m/s*] |

### 3.4.3 Example

This paragraph includes an exapmle of how this block works. The simulation was made for ESSO OSAKA case study. The characteristics of the ship are shown in chapter 4.

The simulation inputs are shown in table 3.18. There was high wind, meaning number 7 in the Beaufort scale and 30kn. The ship executed the turning circle manoeuvre.

*Table 3.18*

| Type/Source | Value | |
|---|---|---|
| $u_{init}$ | 7.8 | kn |
| $v_{init}$ | 0 | kn |
| $r_{init}$ | 0 | kn |
| $p_{init}$ | 0 | kn |
| $Te_{water}$ | 15 | °C |
| *Water type* | salt | - |
| *Sea Depth* | 55 | m |
| *Initial heading* | 0 | rad |
| *P/D* | 0.715 | - |
| $N_{propeller}$ | 42.25 | rps |
| Wind velocity | 30 | kn |
| Wind angle | 30 | deg |
| Air temperature | 20 | °C |
| Start time | 60 | s |
| Initial rudder angle | 0 | deg |
| Rudder turning rate | 2.34 | deg/s |
| Max. rudder angle | -34 | deg |

The output of the block is shown in the following figures.

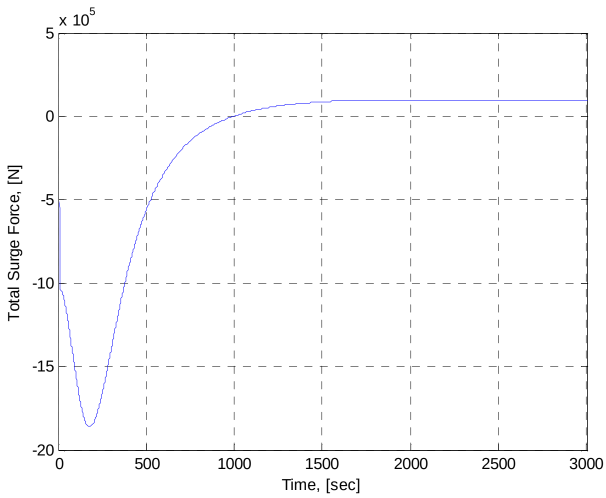


*Figure 3.32 – Total Surge force vs Time*

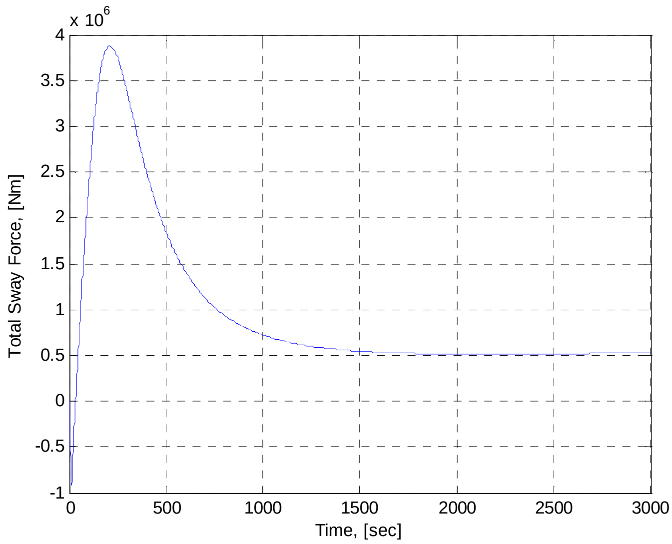


*Figure 3.33 – Total Sway force vs Time*

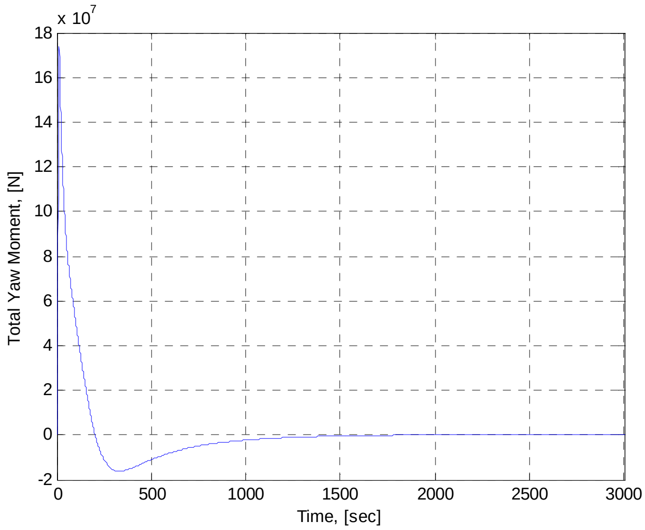


*Figure 3.34 – Total Yaw moment vs Time*

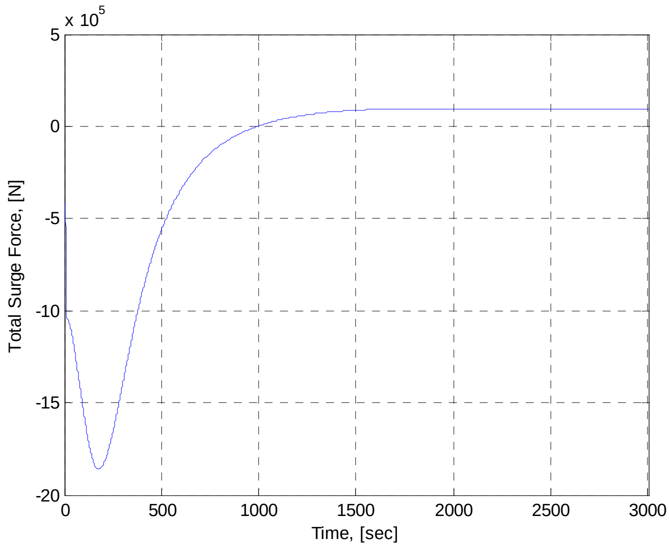


*Figure 3.35 – Total Surge force vs Time*

## 3.5 Post-processing Block

### 3.5.1 Mathematical Model

As described in section 2.1 the model is using a body-fixed coordinates system for the calculations. In order for the user to understand the results better and to have a practical use and meaning this block was built. Its job is to convert the results form body-fixed to earth-fixed system.

As presented in the following picture the block was set in the far right region of the model, as it is the last block in the sequence of the calculations. It connects with the rest of the model with the yaw turn rate as deprived by the dynamics block and the simulation inputs. The outputs are *posX* and *posY*, which describe the position of the ship in x and y-axis respextively. These two values are calculated at each time step and are stored in a variable in matlab workspace for further use (such as plots, figures, etc). Also both of them are inserted in a *plot block* which designs the ship's trajectory.

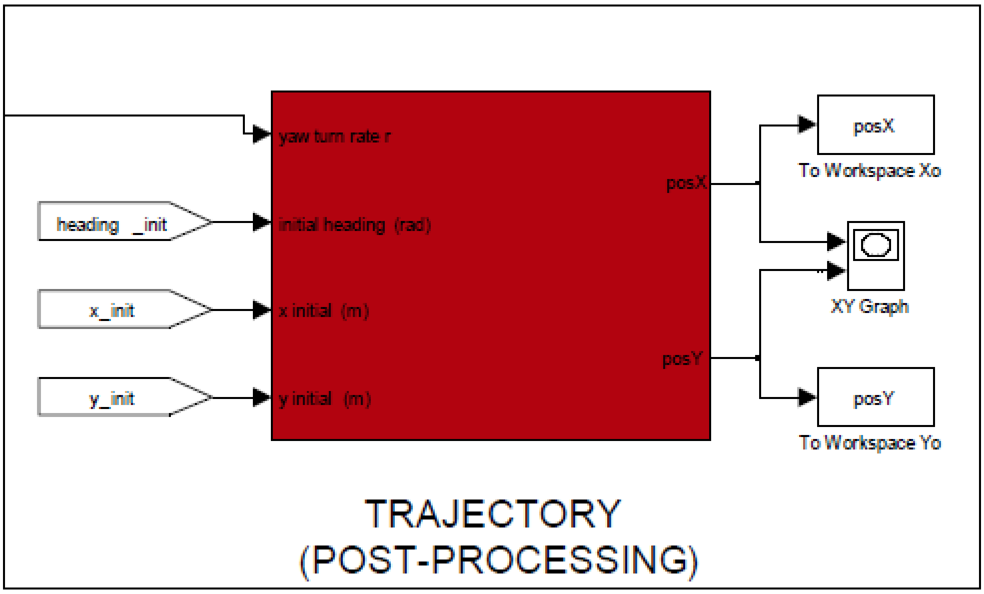

*Fig 3.36. – Post Processing block*

The lower level of the block is presented in the following figure 3.37. It was built based on equations (2.2) and (2.3). First, the yaw turn rate is going into an integrator, which uses the initial heading as initial condition, in order to provide the heading $\psi$. After that it is inserted in a *sin* block as well as a *cos* block. Each value is multiplied with surge and sway velocities and are added in such a manner as the equation (2.2) defines. The two values are then integrated with initial *x-position* and initial *y-position* as initial conditions respectively, in order to provide *posX* and *posY*.

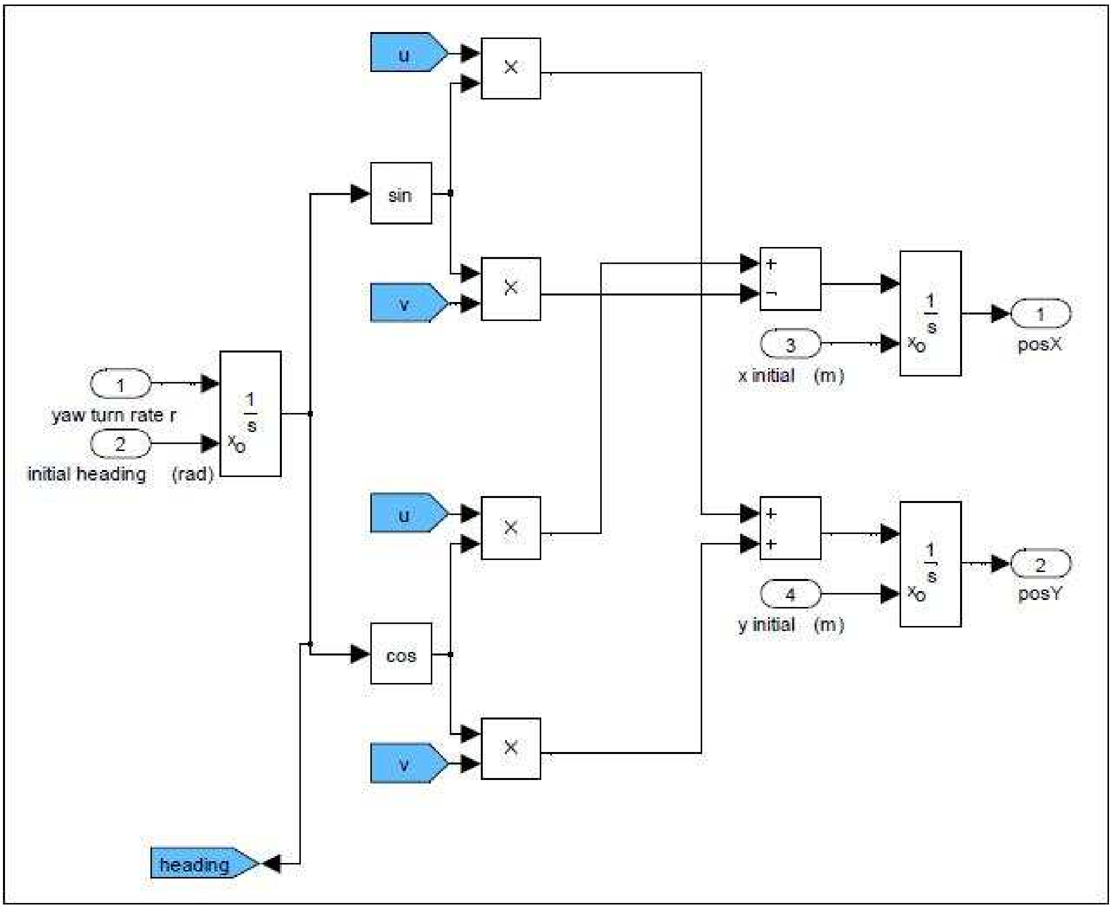

*Figure 3.37 - Coordinates conversion Block*

## 3.5.2 Input File

This block does not use any inputs given by the input file, but only simulation and internal inputs. As this block converts the coordinate, the initial heading and position are needed in order to produce the correct values. The other inputs are the surge and sway velocities used to produce equation (2.2).

*Table 3.19*

| Type/Source | | Value | Description |
|---|---|---|---|
| Simulation input | | $heading$ | ship's heading, [$rad$] |
| Simulation input | | $x - initial$ | ship's initial position in x-axis, [$m$] |
| Simulation input | | $y - initial$ | ship's initial position in y-axis, [$m$] |
| Internal input | Dynamics block | $u$ | surge velocity, [m/s] |
| Internal input | Dynamics block | $v$ | sway velocity, [m/s] |
| Internal input | Dynamics block | $r$ | yaw turn rate, [rad/s] |
| Internal input | Dynamics block | $p$ | roll turn rate, [rad/s] |

## 3.6 Miscellaneuos Calculations Block

### 3.6.1 Mathematical model

As described previously there is a block called *Utility block* which carries out some general purpose calculations of figures that are needed by the other blocks. As shown in the following Figure 3.38 the block has only three simulation inputs and a large number of outputs, including some hydrodynamic coefficients, added masses and added moments of Inertia. As stated in paragraph 3.6.2 the largest amount of the inputs needed for the calculation are given via the input file.

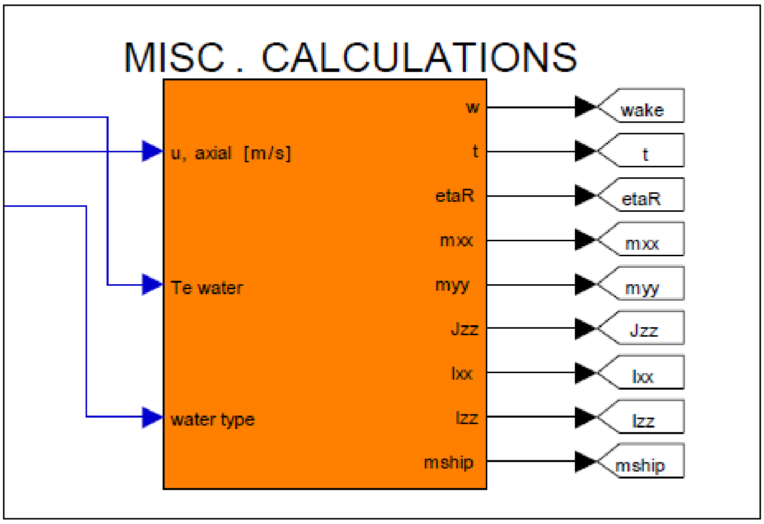


*Fig 3.38*

The following Figure 3.39 shows the interior of the block, which is very simple. It includes only input, output and the s-function block. The latter includes the matlab code (appendix A6) which calculates the values, based on section 2.4.

The outputs include the hydrodynamic coefficients which are the wake fraction, the thrust deduction factor and the relative-rotative coefficient. There also the added masses on x and y- axis and the added moment of inertia on z-axis. Finally, there are the moments of Inertia on x and z-axis and the ship mass. For all the values the user can choose if the values will be calculated or given in the input file. If the value given as input data is zero then this value will be calculated.

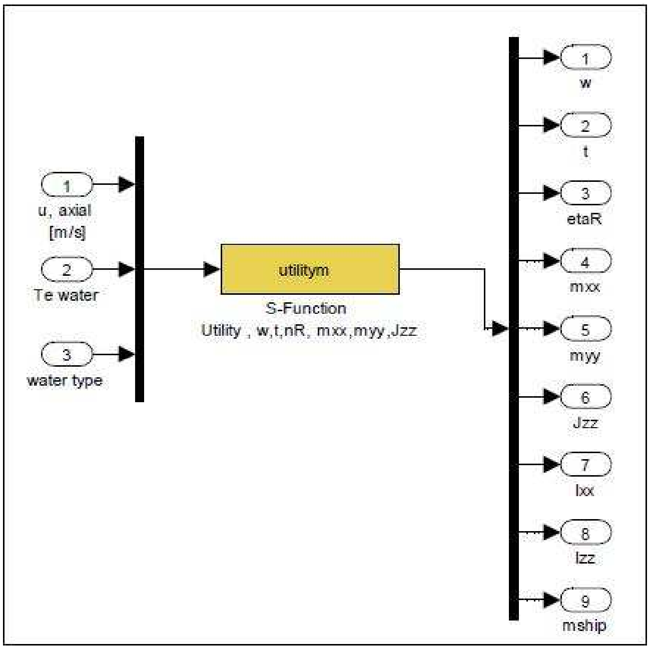


*Figure 3.39*

### 3.6.2 Input file

The block has only three simulation inputs, which are the surge velocity and the type and temperature of the water. There are some general geometrical characteristics of the hull and the propeller and also the values which are calculated. If the user gives the value zero then these values will be calculated, otherwise the value given by the user will be used as output.

The block also needs the *Number of bow thrusters* and the *Number of Derivatives* from the input file in order to assign the proper values to the proper variables due to the variable indexing of the input file.

*Table 3.20*

| index | Input value | Description |
|---|---|---|
| Input file | LBP | Lenght bewtween perpendiculars, [m] |
| Input file | LWL | Lenght in waterline, [m] |
| Input file | B | Breadth, [m] |
| Input file | TF | Depth at for perpendicular, [m] |
| Input file | TA | Depth at aft perpendicular, [m] |
| Input file | VOL | Displacement, [m3] |
| Input file | LCB | Longitudal Centre of Buoyancy, [m] |
| Input file | CWP | Waterplane Coefficient, [-] |
| Input file | CM | Midship Section Coefficient, [-] |
| Input file | S | Wetted Hull Area, [m2] |
| Input file | ABT | cross sectional area of bulbous bow, [m2] |
| Input file | HB | centroid of bulbous bow cross section to keel, [m] |
| Input file | AT | area of immersed transom, [m2] |
| Input file | LR | Lenght of run, [m] |
| Input file | IE | half angle of entrance, [-] |
| Input file | NBT | Number of Bow thruster, [-] |
| Input file | NPROP | Number of propellers, [-] |
| Input file | DP | Propeller Diameter, [m] |
| Input file | AAE | Propeller Expanded blade area ratio, [-] |
| Input file | PPD | Propeller pitch-diameter ratio, [-] |

| | | |
|---|---|---|
| Input file | NDER | Total Number of derivatives on all directions, [-] |
| Input file | Ixx | moment of inertia at x-axis, [kg m2] |
| Input file | Izz | moment of inertia at z-axis, [kg m2] |
| Input file | lamda | rudder aspect ratio, [m] |
| Input file | w0 | wake fraction,(user defined,0=calculation), [-] |
| Input file | t0 | thrust deduction factor,(user defined,0=calculation), [-] |
| Input file | nR0 | relative rotative factor,(user defined,0=calculation), [-] |
| Input file | mxx0 | added mass in x-axis(user defined,0=calculation), [kg] |
| Input file | myy0 | added mass in y-axis(user defined,0=calculation), [kg] |
| Input file | Jzz0 | added moment of inertia in y-axis(user defined,0=calculation), [kg] |
| Simulation input | $u_{init}$ | initial surge velocity, [m/s] |
| Simulation input | $T_{water}$ | water temperature, [˚C] |
| Simulation input | $Type_{water}$ | fresh or sea water, [-] |

## 3.7 Maneuvers' Blocks

The blocks described in previous paragraphs calculate forces, moments and velocities. They are also used for the simulation of the ship motions. In order for the ship to execute the desired manoeuvres there must be a deviation from the original course. This is done by steering the rudder and altering the rudder angle. The model includes two standard manoeuvres, which are customizable, the turning circle and the zig-zag manoeuvre, as described in section 2.5. They are included in the manoeuvring block, as shown in the following figure.

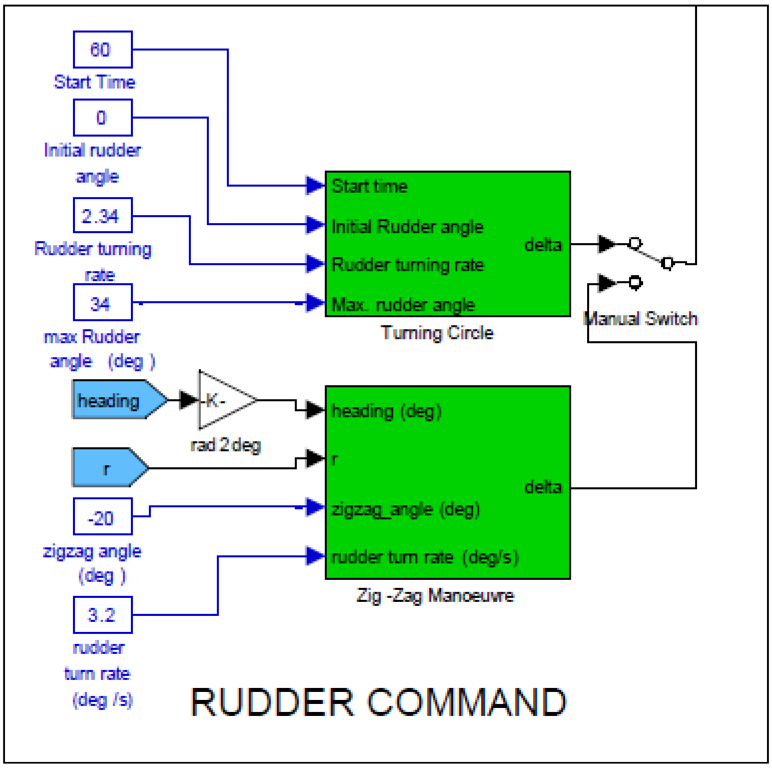


*Fig.3.40 – Maoever's blocks*

### 3.7.1 Turning circle Block

#### 3.7.1.1 Mathematical model

This block includes the turning circle maneuver. As shown in the following figure the block is rather simple, as it only includes inputs, output and the s-function block. The later includes the code that generates the rudder angle. It is written as the same template for s-function that has been used in the rest blocks, as described in Appendix 1. The methodology for calculating the angle at each time step is the following (matlab code can be found in appendix A7.2):

1. Check if the turn has been initiated
2. Calculate the angle by using the formula: $\delta = rate \cdot t + \delta_{init}$
3. Check if the calculated angle exceeds the angle threshold(maximum rudder angle) using if block

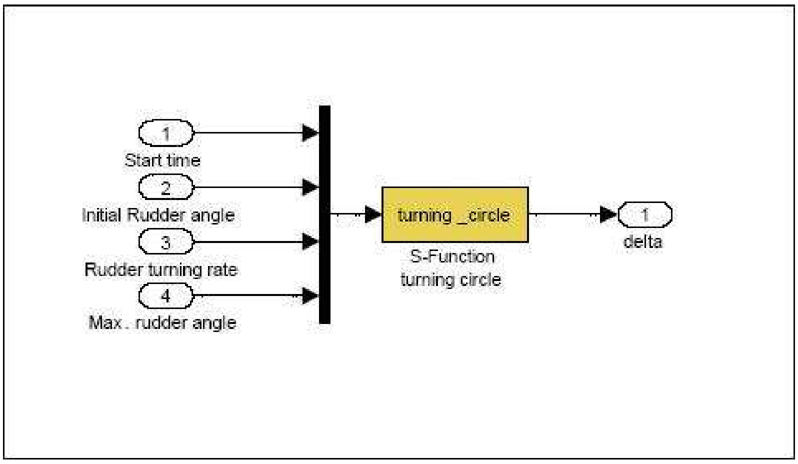

*Fig. 3.41 – Turning Circle Interior*

### 3.7.1.2 Input file

This block has four simulation inputs. First, it's the time that the maneuver will initiate. Then there is the initial rudder angle, and two rudder characteritics, which is the rudder turning rate and the maximum rudder angle, as shown in the following table.

*Table 3.21*

| index | Input value | Description |
|---|---|---|
| Simulation input | $T_{start}$ | Start time of the maneuver, [s] |
| Simulation input | $\delta_{init}$ | initial rudder angle, [deg] |
| Simulation input | $\dot{\delta}$ | rudder angle turn rate, [deg/s] |
| Simulation input | $\delta_{max}$ | Maximum rudder angle, [deg] |

## 3.7.2 Zig-zag maneuver Block

### 3.7.2.1 Mathematical Model

This block includes the zig-zag maneuver. As shown in the following figure this block is more complex than the turning circle. This maneuver is much more complex, since the rudder angle cannot be predefined, but changes dynamically based on the heading. It was built with various *if blocks*. It also includes inputs, output and the s-function block.

In this block the S-function block calculates the sign of rudder's turning rate. It is then inserted as input in some "if" blocks used to calculate the angle at each time step. The methodology for the calculations requires the actual ship's heading and turn rate in order to compare it with the desired values (the matlab code is presented in appendix A7.2).

1. Check if the zig-zag angle $\delta_{zigzag}$ is positive.
2. Then follows a double check on ship's heading $\psi$ and turn rate $r_{rud}$, which may have four combinations of logical results.
3. Depending of the combination the rate can be produced as presented in table 3.14.

*Table 3.22 – Rate sign calculation methodology*

| $\psi_{ref}$ | heading | Actual rate | Rudder turning rate sign |
|---|---|---|---|
| Positive | $\psi \le \delta_{zigzag}$ | $\dot{\delta} \ge 0$ | + |
| | $\psi > \delta_{zigzag}$ | $\dot{\delta} > 0$ | - |
| | $\psi \ge -\delta_{zigzag}$ | $\dot{\delta} < 0$ | - |
| | $\psi < -\delta_{zigzag}$ | $\dot{\delta} < 0$ | + |
| Negative | $\psi \ge \delta_{zigzag}$ | $\dot{\delta} \le 0$ | + |
| | $\psi < \delta_{zigzag}$ | $\dot{\delta} < 0$ | - |
| | $\psi \le -\delta_{zigzag}$ | $\dot{\delta} > 0$ | - |
| | $\psi > -\delta_{zigzag}$ | $\dot{\delta} > 0$ | + |

The output of this s-function is added with the rudder angle of the previous time step, and insterted in the *if block*. The second value that is inserted in this block is the absolute value of the zigzag angle. If the first value is greater than the second the first *if-action block* is enabled and the ouput value is the positive zig-zag angle. If it is less than the negative of the second value, then the second *if-action block* is enabled and the output is the negative zig-zag angle. Else there is no saturation and the output is the value just after the addition of the s-function block and the previous time step value.

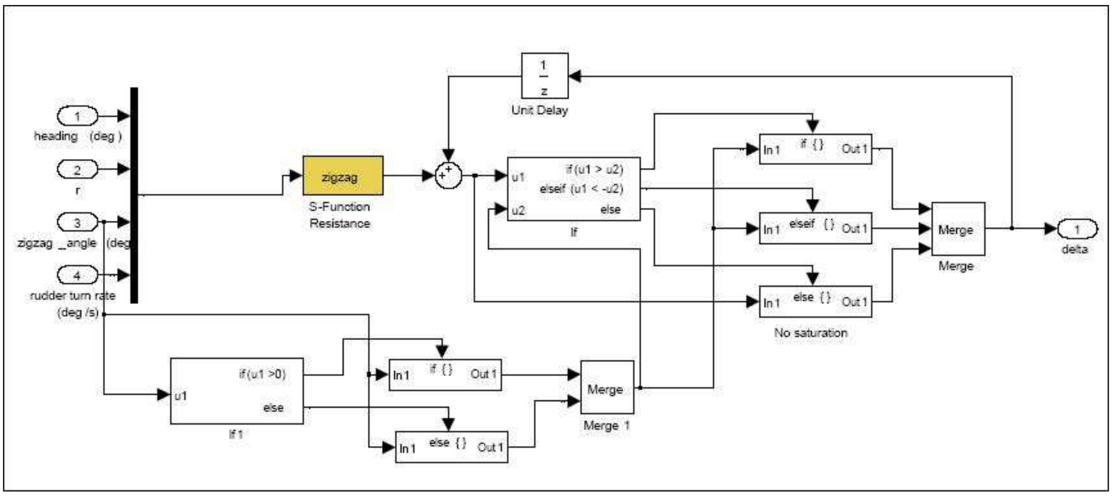


*Figure 3.42 – Zigzag maneuver block interior*

### 3.7.2.2 Input file

This block has four simulation inputs. First, it's the heading and the rudder angle at a time step. Then it's the zig-zag angle (i.e 10° or 20°) and finally the rudder turn rate, as shown in 1the following table.

*Table 3.23*

| index | Input value | Description |
|---|---|---|
| Simulation input | *heading*, $\psi$ | ship's heading, [deg] |
| Simulation input | $\delta$ | rudder angle, [deg] |
| Simulation input | $\delta_{zigzag}$ | Zigzag angle, [deg] |
| Simulation input | $\dot{\delta}$ | rudder angle turn rate, [deg/s] |

# Chapter 4

# Case Studies & Simulations

## 4.1 Introduction

The previous chapters described on what theory the model was based and how it was built. This chapter includes the case studies that were chosen and presents the results in form of comparative diagrams with the sea trials experimental results.

The case studies were selected in order to include two of the most popular type of ships, tankers and containerships. In addition there had to be an extensive set of data and real conditions results in order to compare a wide spectrum of parameters and values. Thus the three case studies that were chosen are ESSO OSAKA, MOERI tanker KVLCC1 and MOERI container-ship KCS.

In the following sections there is the description of each case study, including all the particulars and the input file. In addition, there are also the comparison diagrams for various values, i.e. the ship's trajectory, ship's speed.

## 4.2 Methodology

The input file includes an extensive amount of the ship’s characteristics that are required for the simulation. Some of those are main particulars , such as ship’s length, breadth and draft. There are propeller’s characteristics, rudder’s characteristics and hydrodynamic derivatives. Those were found in each case study’s reference tables. However, the hydrostatic figures of the ship could not be found in the same way as they are dependent on the ship’s waterline.

In order to acquire these hydrostatic figures there was extensive use of RhinoMarine. This procedure requires the design of each case study in digital format. The design was inserted in RhinoMarine along with the basic flotation’s state figures, including desing draft, aft. and for. Perpendiculars. Thus, RhinoMarine calculated figures such as form factors(block coef. $C_B$, waterplane coef. $C_{WP}$, etc).

Apart from the hydrostatic figures, designs in Rhino were used to accurately measure all the distances and areas required. Such figures are the distances of rudder and propeller from ship’s center and the area of the section of bulbous bow

## 4.2 Case Study: ESSO OSAKA

### 4.2.1 Description & Input file

This ship was chosen by the Maneuvering Committee of the 21st ITTC in Trondheim, Norway, as a benchmark ship to operate as a reference for every maneuvering simulation model. The reason behind this choice were mainly that there was an unusually extensive set of sea trials at full load draft and all of which were conducted with unusual attention to measurement accuracy, including correction of the trials' results for the effects of the ocean current. Also, trials were conducted in deep water and water depths equal to 1.5 and 1.2 times trials' draft.

The majority of the values, such as the main particulars, various coefficients, propeller and rudder characteristics and other were acquired from “The Specialist Committee of the ESSO OSAKA”,(ITTC,2002). For the remaining values more complicated work was needed. The .igs file (Initial graphics specification), was inserted in Rhinoceros program (fig 4.2.1). With the use of Rhinomarine the hydrostatics needed were calculated (i.e. LCG,$C_{WPA}$), and the rest of values such as the distance of rudder from center etc were calculated using dimension, section and other tools.

All the particulars of ESSO OSAKA, as inserted in the model via the input file are shown in the following table.

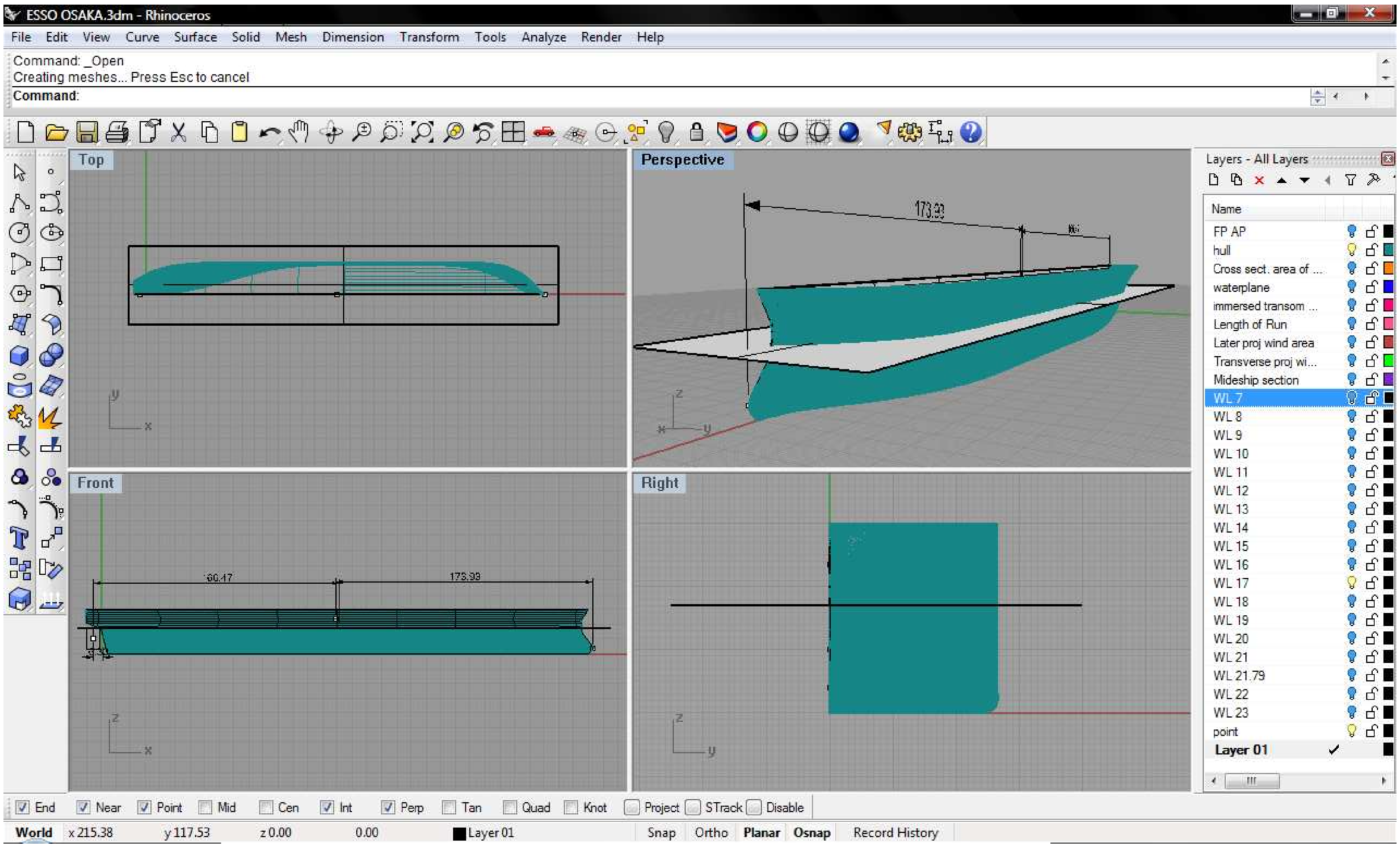


*Figure 4.2.1 – Imported .igs file of ESSO OSAKA in Rhinoceros*

*Table 4.1*

| index | Input value | Description |
|---|---|---|
| 1 | LBP | 325 |
| 2 | LWL | 334.75 |
| 3 | B | 53.06 |
| 4 | TF | 21.79 |
| 5 | TA | 21.79 |
| 6 | VOL | 319089.2 |
| 7 | LCB | 0.045 |
| 8 | CWP | 0.871 |
| 9 | CM | 0.997 |
| 10 | S | 28862 |
| 11 | CSTERN | 3 |
| 12 | NRUD | 1 |
| 13 | SRUD | 239.64 |
| 14 | CRUD | 1.4 |
| 15 | SAPP i | 0 |
| 16 | | 0 |
| 17 | | 0 |
| 18 | | 0 |
| 19 | | 0 |
| 20 | | 0 |
| 21 | | 0 |
| 22 | | 0 |
| 23 | | 0 |
| 24 | | 0 |
| 25 | | 0 |
| 26 | ABT | 42.07 |
| 27 | HB | 4 |
| 28 | AT | 0.33 |
| 29 | LR | 172.25 |
| 30 | IE | 0 |
| 31 | NBT | 0 |
| 31+1 | DBT ($1^{st}$) | - |
| 31+2 | DBT ($2^{nd}$) | - |
| … | | - |
| 31+i | DBT (i) | - |
| 31+i+1 | NPROP | 1 |
| 31+i+2 | DP | 9.1 |
| 31+i+3 | AAE | 0 |
| 31+i+4 | PPD | 0.715 |
| 31+i+5 | LOA | 342.7 |
| 31+i+6 | AL | 2766.6 |
| 31+i+7 | AS | 0 |
| 31+i+8 | AT | 435.2 |
| 31+i+9 | S | 691.23 |
| 31+i+10 | C | 174.29 |
| 31+i+11 | M | 2 |
| 31+i+12 | NDER | 5 |
| 31+i+9+j*4 | dir | 1 |
| 31+i+10+j*4 | Hd | 2.3 |
| 31+i+11+j*4 | index | 1 0 0 0 1 0 0 0 |

| | | |
|---|---|---|
| 31+i+12+j*4 | method | 0 |
| … | dir | 1 |
| … | Hd | 2.3 |
| … | index | 1 0 0 0 1 0 0 0 |
| … | method | 0 |
| 31+i+j*4+13 | CWPA | 0.7 |
| 31+i+j*4+14 | CPA | 0.7 |
| 31+i+j*4+15 | LCG | 0.04 |
| 31+i+j*4+16 | BP07 | 0.07 |
| 31+i+j*4+17 | BPS | 0.08 |
| 31+i+j*4+18 | Ixx | 82755846147 |
| 31+i+j*4+19 | Izz | 2160982123245 |
| 31+i+j*4+20 | xP | 160 |
| 31+i+j*4+21 | xR | 166 |
| 31+i+j*4+22 | zR | 2 |
| 31+i+j*4+23 | HR | 13.85 |
| 31+i+j*4+24 | lamda | 1.539 |
| 31+i+j*4+25 | AR | 119.817 |
| 31+i+j*4+26 | w0 | 0 |
| 31+i+j*4+27 | t0 | 0 |
| 31+i+j*4+28 | nR0 | 0 |
| 31+i+j*4+29 | mxx0 | 0 |
| 31+i+j*4+30 | myy0 | 0 |
| 31+i+j*4+31 | Jzz0 | 0 |

### 4.2.2 Simulations

Several simulations were carried out. All the results were compared with diagrams from “The Specialist Committee of the ESSO OSAKA”,(ITTC,2002), which include sea trial tests and results from other simulation models. In general the results of the present thesis’ model were acceptable. There is a match in patterns in the various figures, whereas there are deviations as far as the absolute values. These deviations are justified as the model was validated only in order to work properly and produce logical results close to other models’ results.

In addition, throughout the validation process it was observed that the results are highly dependent on the selection of the hydrodynamic derivatives, both on which will be chosen and the values of these. As depicted in figure 4.2.2 the deviation between two different derivatives’ sets are great even though the only differences of the input files were on the values of three derivatives. Thus, a better validation can be achieved for each vessel by selecting a better set of hysrodynamic derivatives

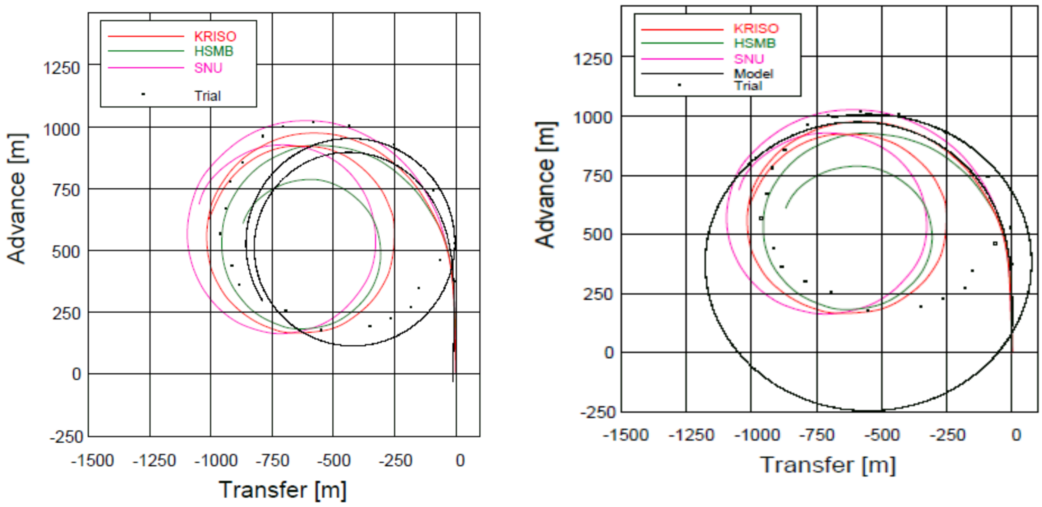


*Fig. 4.2.2 – Comparison of two different hydrodynamic derivatives sets*

The simulation were carried out for four maneuvers variations which are the following:

- Port Turning Circle on 7.7 kn with 35° maximum rudder angle
- Starboard Turning Circle on 10 kn with 35° maximum rudder angle
- Zig-zag maneuver with 10° angle on 7.5 kn
- Zig-zag maneuver with 20° angle on 7.8 kn

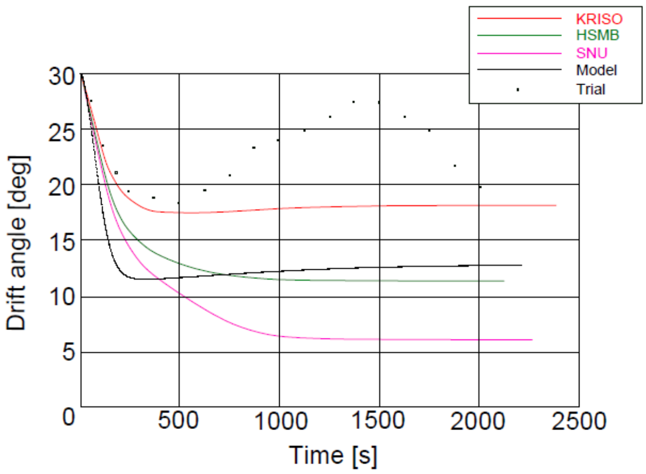


*Figure 4.2.3 – Drift angle vs Time (Turning circle, 7.7kn)*

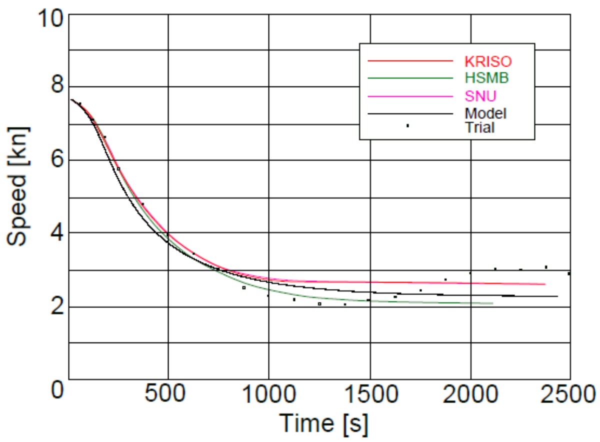


*Figure 4.2.4 – Speed vs Time (Turning circle, 7.7kn)*

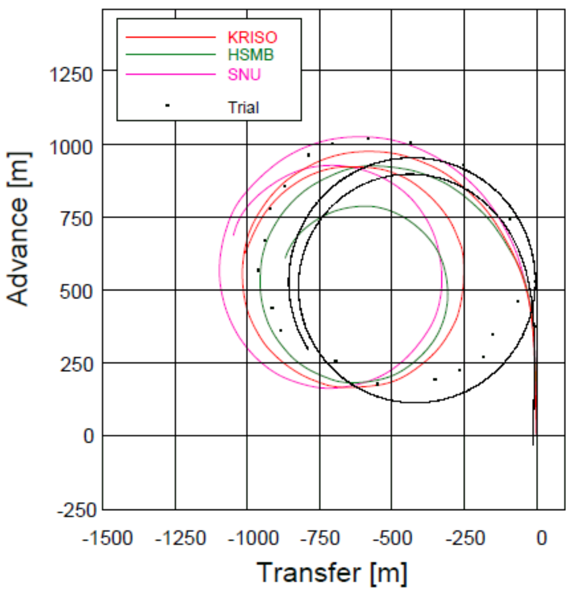


*Figure 4.2.5 -Trajectory (Turning circle, 7.7kn)*

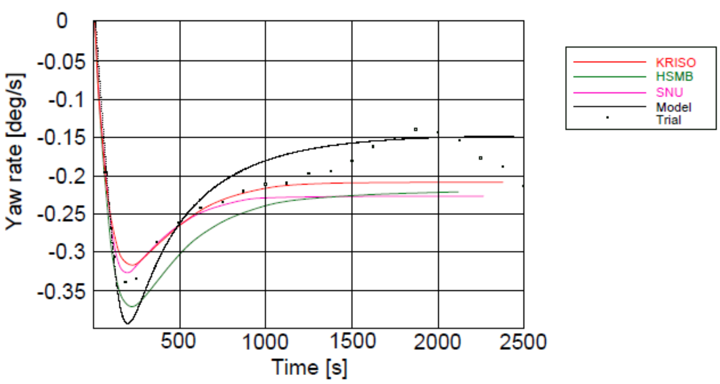


*Figure 4.2.6 – Yaw rate vs Time (Turning circle, 7.7kn)*

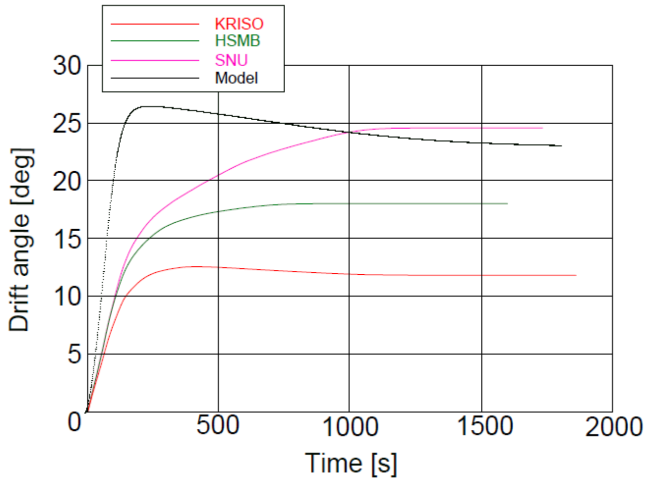


*Figure 4.2.7 – Drift angle vs Time (Turning circle, 10kn)*

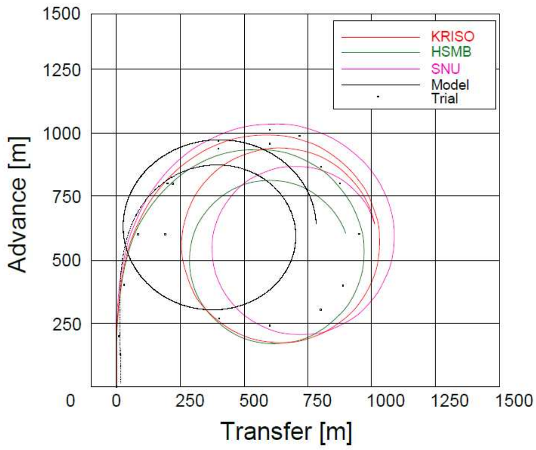


*Figure 4.2.8 – Trajectory (Turning circle, 10 kn)*

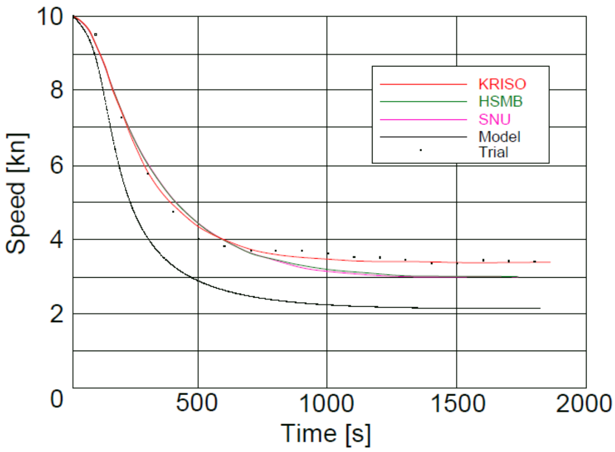


*Figure 4.2.9 – Speed vs Time (Turning circle, 10 kn)*

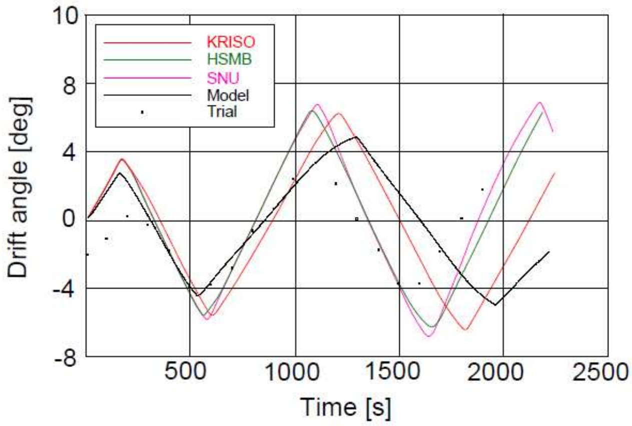


*Figure 4.2.10 – Drift vs Time (Zig-zag 10 deg)*

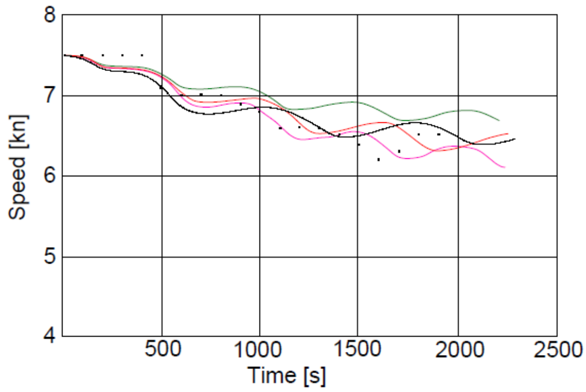


*Figure 4.2.11 – Speed vs Time (Zig-zag 10 deg)*

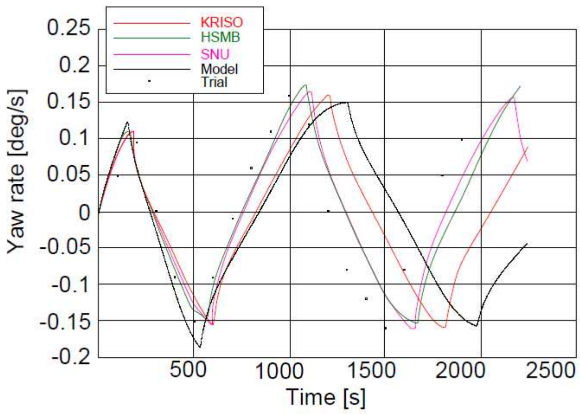


*Figure 4.2.12 – Yaw rate vs Time (Zig-zag 10 deg)*

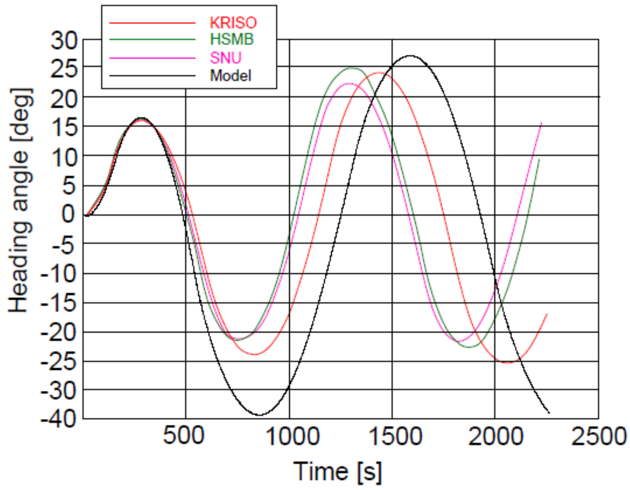


*Figure 4.2.13 – Heading angle vs Time (Zig-zag 10 deg)*

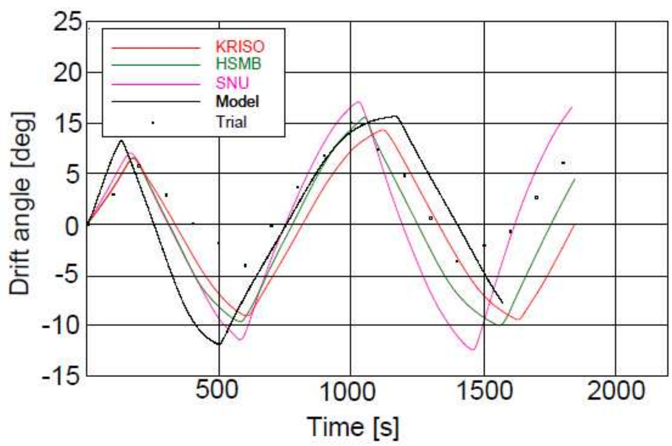


*Figure 4.2.14 – Drift angle vs Time (Zig-zag 20 deg)*

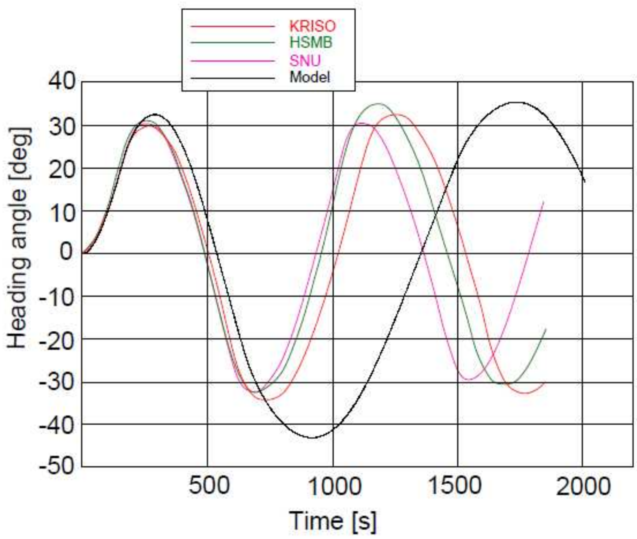


*Figure 4.2.15 – Heading angle vs Time (Zig-zag 20 deg)*

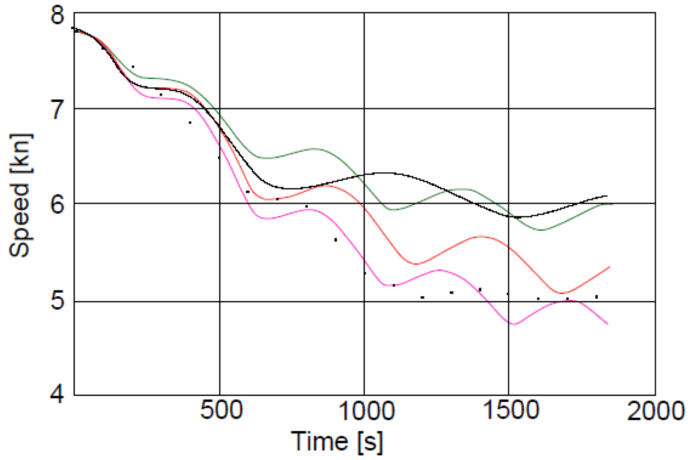


*Figure 4.2.16 – Speed vs Time (Zig-zag 20 deg)*

# 4.3 Case Study: MOERI Tanker KVLCC1

## 4.3.1 Description & Input file

The MOERI VLCC was selected to provide data for both explanation of flow physics and CFD validation for a modern (ca. 1997) 300 K tanker ship with bulbous bow and stern. Two stern variants were designed: KVLCC1 has barge type stern frame-lines with a fine stern end bulb i.e. relatively V-shaped frame-lines, while KVLCC2 has more U-shaped stern frame-lines. No full-scale ship exists.

Apart from the general particulars [15], some values mainly concerning the propeller and the rudder were taken from the designs of the ship as in Esso Osaka, fig (4.3.1). The input file is shown in the following table:

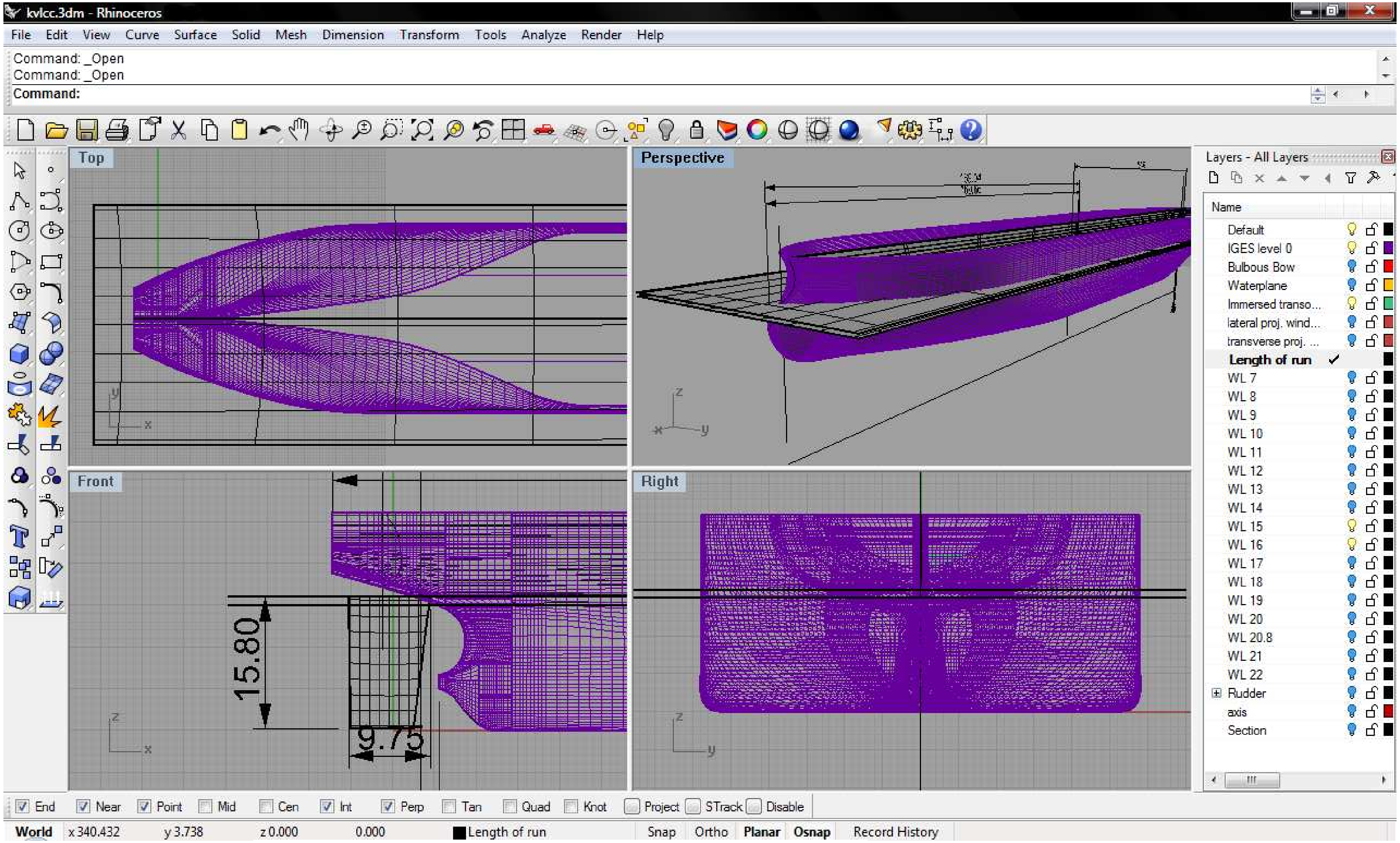


*Figure 4.3.1 – Imported .igs file of KVLCC  in Rhinoceros*

*Table 4.2*

| index | Input value | value |
|---|---|---|
| 1 | LBP | 320 |
| 2 | LWL | 325.554 |
| 3 | B | 58.002 |
| 4 | TF | 20.8 |
| 5 | TA | 20.8 |
| 6 | VOL | 311439.0 |
| 7 | LCB | 0.043 |
| 8 | CWP | 0.891 |
| 9 | CM | 0.999 |
| 10 | S | 27695.94 |
| 11 | CSTERN | 3 |
| 12 | NRUD | 1.000 |
| 13 | SRUD | 273.3 |
| 14 | CRUD | 1.4 |
| 15 | SAPP i | 0 |
| 16 | | 0 |
| 17 | | 0 |
| 18 | | 0 |
| 19 | | 0 |
| 20 | | 0 |
| 21 | | 0 |
| 22 | | 0 |
| 23 | | 0 |
| 24 | | 0 |
| 25 | | 0 |
| 26 | ABT | 86.847 |
| 27 | HB | 10.67 |
| 28 | AT | 7.877 |
| 29 | LR | 136.7 |
| 30 | IE | 0 |
| 31 | NBT | 0 |
| 31+1 | DBT ($1^{st}$) | - |
| 31+2 | DBT ($2^{nd}$) | - |
| … | | - |
| 31+i | DBT (i) | - |
| 31+i+1 | NPROP | 1 |
| 31+i+2 | DP | 9.86 |
| 31+i+3 | AAE | 0.431 |
| 31+i+4 | PPD | 0.721 |
| 31+i+5 | LOA | 333.47 |
| 31+i+6 | AL | 1695.8 |
| 31+i+7 | AS | 0 |
| 31+i+8 | AT | 301.6 |
| 31+i+9 | S | 663.7 |
| 31+i+10 | C | 170.48 |
| 31+i+11 | M | 2 |
| 31+i+12 | NDER | 5 |
| 31+i+9+j*4 | dir | 1 |
| 31+i+10+j*4 | Hd | 2.3 |
| 31+i+11+j*4 | index | 1 0 0 0 1 0 0 0 |

| | | |
|---|---|---|
| 31+i+12+j*4 | method | 0 |
| … | dir | 1 |
| … | Hd | 2.3 |
| … | index | 1 0 0 0 1 0 0 0 |
| … | method | 0 |
| 31+i+j*4+13 | CWPA | 0.891 |
| 31+i+j*4+14 | CPA | 0.820 |
| 31+i+j*4+15 | LCG | 0.035 |
| 31+i+j*4+16 | BP07 | 0.7 |
| 31+i+j*4+17 | BPS | 0.9 |
| 31+i+j*4+18 | Ixx | 80771765287 |
| 31+i+j*4+19 | Izz | 2109172330124 |
| 31+i+j*4+20 | xP | 161.00 |
| 31+i+j*4+21 | xR | 164.00 |
| 31+i+j*4+22 | zR | 2.00 |
| 31+i+j*4+23 | HR | 15.80 |
| 31+i+j*4+24 | lamda | 1.72 |
| 31+i+j*4+25 | AR | 126.90 |
| 31+i+j*4+26 | w0 | 0 |
| 31+i+j*4+27 | t0 | 0 |
| 31+i+j*4+28 | nR0 | 0 |
| 31+i+j*4+29 | mxx0 | 0 |
| 31+i+j*4+30 | myy0 | 0 |
| 31+i+j*4+31 | Jzz0 | 0 |

### 4.3.2 Simulations

Due to lack of simulation data on KVLCC there could be any comparison of the model's results. Thus, some simulations were made as reference for future studies.

The simulations include port turning circle on 10kn with maximum rudder angle of 35° and a zig-zag maneuver with 10° angle on 7.8kn.

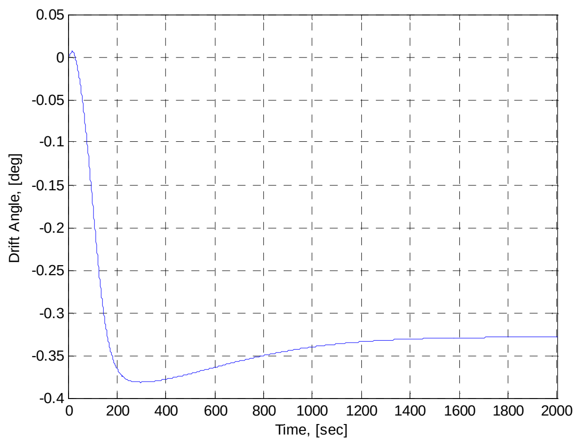


*Figure 4.3.2 – Drift angle – turning circle on 7.8kn*

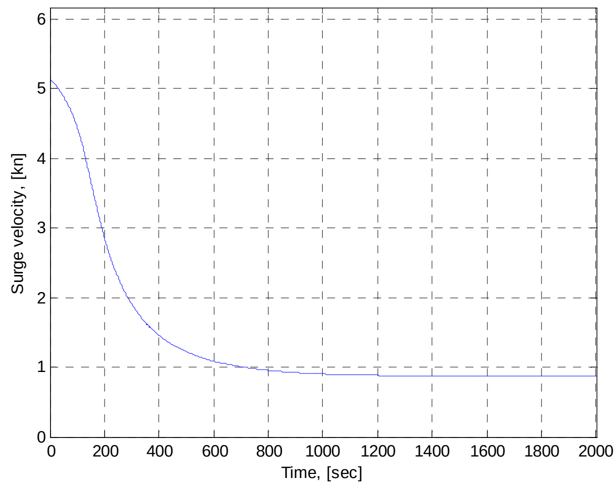


*Figure 4.3.3 – Speed – turning circle on 7.8kn*

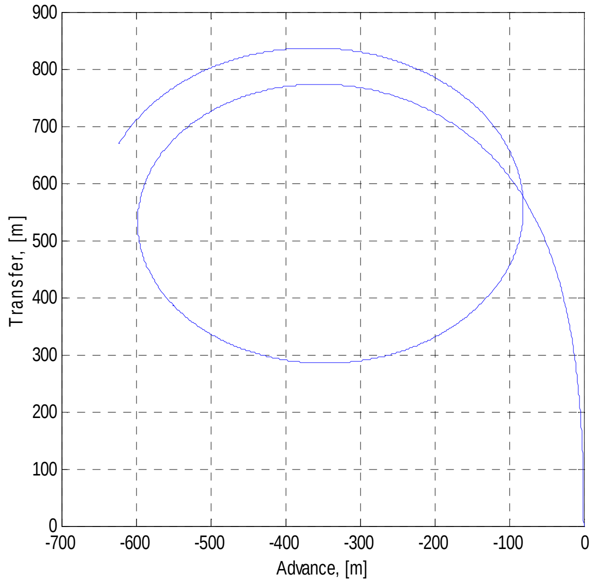


*Figure 4.3.4 – Trajectory– turning circle on 7.8kn*

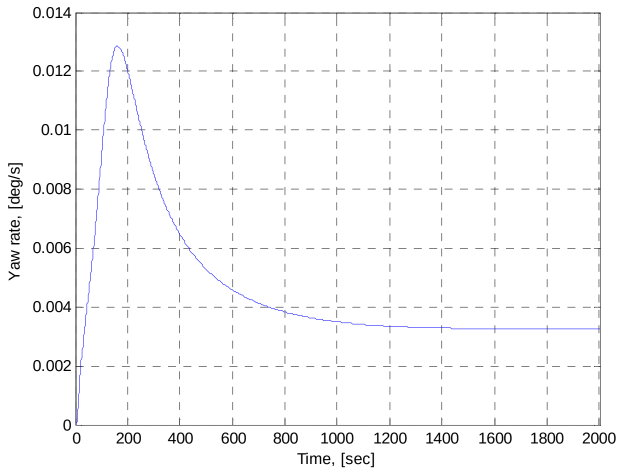


*Figure 4.3.5 – Yaw rate – turning circle on 7.8kn*

## 4.4 Case Study: MOERI Containership KCS

### 4.4.1 Description & Input file

Similarly with the previous case study, the KCS was conceived to provide data for both explanation of flow physics and CFD validation for a modern container ship with bulb bow and stern (i.e., ca. 1997). The conditions include bare hull and fixed model. No full-scale ship exists.

The particulars were taken from [15] and the other values, as in ESSO OSAKA, concerning the rudder, the propeller, hydrostatics etc were acquired from the designs provided. The igs file was imported in Rhinoceros (fig. 4.4.1) and the hydrostatics were calculated with Rhinomarine. They are shown in the following table.

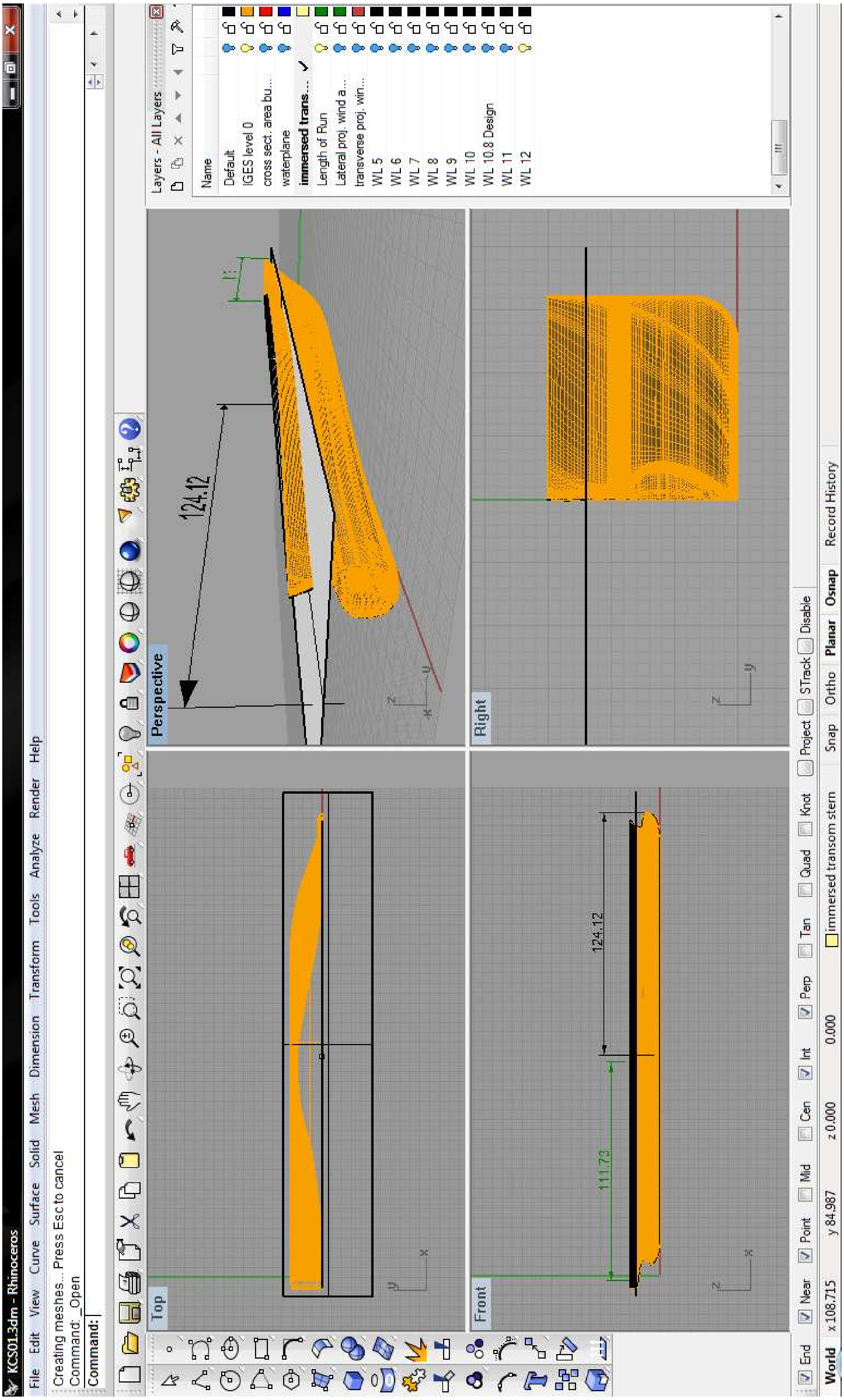


*Figure 4.4.1 – Imported .igs file of KCS in Rhinoceros*

*Table 4.3*

| index | Input value | value |
|---|---|---|
| 1 | LBP | 230.00 |
| 2 | LWL | 232.443 |
| 3 | B | 32.203 |
| 4 | TF | 10.80 |
| 5 | TA | 10.80 |
| 6 | VOL | 51723.10 |
| 7 | LCB | -0.09 |
| 8 | CWP | 0.821 |
| 9 | CM | 0.985 |
| 10 | S | 9490.50 |
| 11 | CSTERN | 3.00 |
| 12 | NRUD | 1.00 |
| 13 | SRUD | 115 |
| 14 | CRUD | 1.40 |
| 15 | SAPP i | 0 |
| 16 | | 0 |
| 17 | | 0 |
| 18 | | 0 |
| 19 | | 0 |
| 20 | | 0 |
| 21 | | 0 |
| 22 | | 0 |
| 23 | | 0 |
| 24 | | 0 |
| 25 | | 0 |
| 26 | ABT | 14.82 |
| 27 | HB | 6.09 |
| 28 | AT | 0.00 |
| 29 | LR | 111.73 |
| 30 | IE | 0.00 |
| 31 | NBT | 0.00 |
| 31+1 | DBT (1st) | - |
| 31+2 | DBT (2nd) | - |
| … | | - |
| 31+i | DBT (i) | - |
| 31+i+1 | NPROP | 1 |
| 31+i+2 | DP | 7.9 |
| 31+i+3 | AAE | 0.8 |
| 31+i+4 | PPD | 0.997 |
| 31+i+5 | LOA | 243 |
| 31+i+6 | AL | 998 |
| 31+i+7 | AS | 0 |
| 31+i+8 | AT | 67.77 |
| 31+i+9 | S | 486.8 |
| 31+i+10 | C | 124.25 |
| 31+i+11 | M | 2 |
| 31+i+12 | NDER | 5 |
| 31+i+9+j*4 | dir | 1 |
| 31+i+10+j*4 | Hd | 2.3 |
| 31+i+11+j*4 | index | 1 0 0 0 1 0 0 0 |

| | | |
|---|---|---|
| 31+i+12+j*4 | method | 0 |
| … | dir | 1 |
| … | Hd | 2.3 |
| … | index | 1 0 0 0 1 0 0 0 |
| … | method | 0 |
| 31+i+j*4+13 | CWPA | 0.821 |
| 31+i+j*4+14 | CPA | 0.640 |
| 31+i+j*4+15 | LCG | -0.01 |
| 31+i+j*4+16 | BP07 | |
| 31+i+j*4+17 | BPS | |
| 31+i+j*4+18 | Ixx | 13414396055 |
| 31+i+j*4+19 | Izz | 350286673629 |
| 31+i+j*4+20 | xP | 106.80 |
| 31+i+j*4+21 | xR | 111.60 |
| 31+i+j*4+22 | zR | 2.00 |
| 31+i+j*4+23 | HR | 10.00 |
| 31+i+j*4+24 | lamda | 1.81 |
| 31+i+j*4+25 | AR | 54.45 |
| 31+i+j*4+26 | w0 | 0 |
| 31+i+j*4+27 | t0 | 0 |
| 31+i+j*4+28 | nR0 | 0 |
| 31+i+j*4+29 | mxx0 | 0 |
| 31+i+j*4+30 | myy0 | 0 |
| 31+i+j*4+31 | Jzz0 | 0 |

### 4.4.2 Simulations

Due to lack of simulation data on KCS there could not be any comparison of the model's results. Thus, some simulations were made as reference for future studies.

The simulations include turning circle on 7.5kn with maximum rudder angle of 35°. In this figures it's observed,that the pattern seem in the right order and the turning circle radius is within normal values' range.

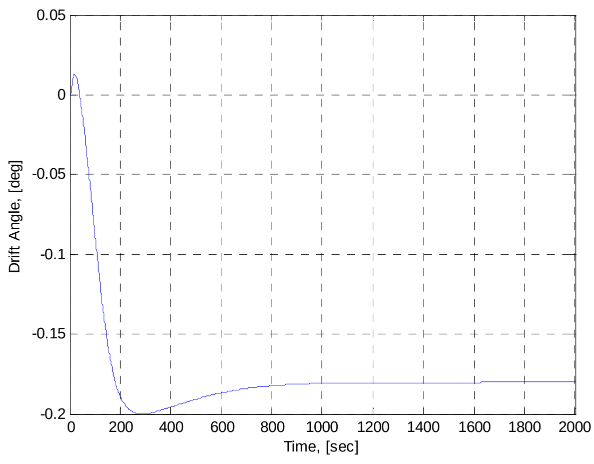


*Figure 4.4.2 – Drift angle – turning circle on 10kn*

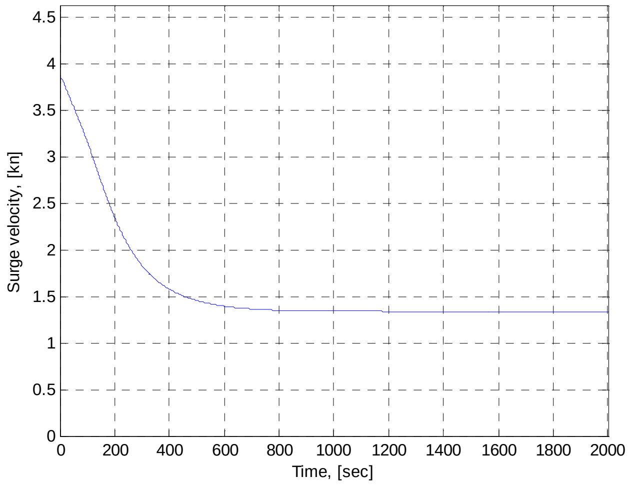


*Figure 4.4.3 – Speed – turning circle on 10kn*

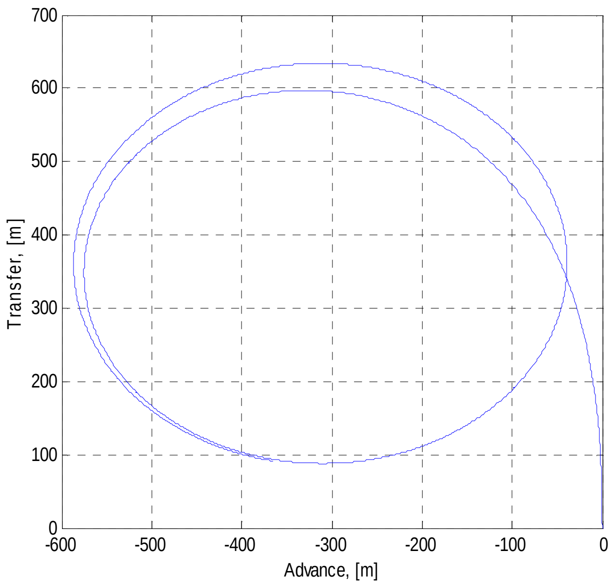


*Figure 4.4.4 – Trajectory – turning circle on 10kn*

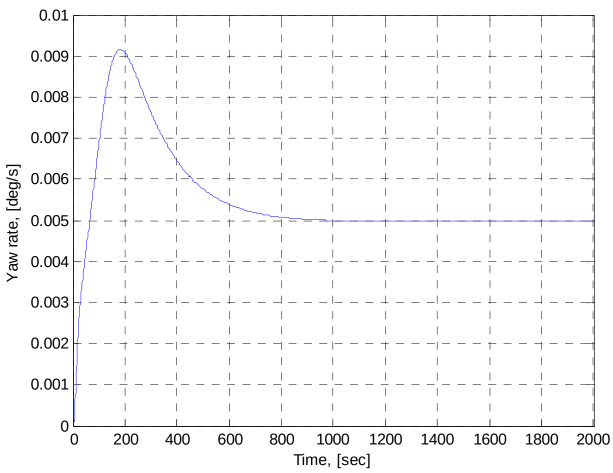


*Figure 4.4.5 – Yaw rate – turning circle on 10kn*

Chapter 5

# Conclusions

In this thesis a method for modeling ship's motions and maneuvering using a scalable and extenable block approach was presented.

After an extensive search in current bibliographic sources, the individual models were selected in order to be up to date, accurate and generally accepted. The mathematical models were used as basis to create programming code. For the implementation of the model the programming environment MATLAB/Simulink was used. Each sub-model was modeled in a separate block, all built in a hierarchical structure. Two common maneuvers were also modeled in order to calclulate the rudder angle, the turning circle and the zig-zag maneuver.

The model was shown to work properly and produce reasonable results when compared published available experimental data.

Various Extensions to the proposed method are possible. The method can be extended at the interface level, the block element and the mathematical model element.

A possible extension is the insertion of an engine block simulating the engine operation in detail. Such a block would affect the rudder forces block and the propeller forces block as it would produce more detailed propeller rotational speed and Torque estimation. An engine model has been developed in another thesis in LME.

Also, improved mathematical models for the elements presented in this thesis may be introduced. Some of the blocks could be replaced by more sophisticated models or simpler models could also be introduced in order to avoid some input figures that are difficult to find.

A possible extension is the insertion of new blocks. For example a ice-collision block could be used, simulating the ship's movement through ice and during breaking ice navigation. Another external forces block could be a wave block simulating the effect of waves on the ship's navigation.

Another possible extension is the insertion of Graphical User Interface (GUI), giving the user a much friendlier and relaxing environment to provide the large amount of input figures and the simulation inputs. Also, an improved output interface would make simpler to get the results.

A further extension of the GUI is the real time rudder control. In such a case the user could alter the rudder angle during the simulation and perform a navigation maneuver that is not predefined

# Appendix A – matlab codes

Table of Contets



As described in the main chapters, all the matlab codes that simulate the individual models were built in the form of s-functions. There's a pattern that was used for all of them, in terms of structure. It is shown in A1. The code that is included in the section "`function sys=mdlOutputs(t,x,u)`" is the translation of the mathematical models in matlab code. The other section that differs from one model to the other is the "`function [sys,x0,str,ts]=mdlInitializeSizes`" section where the number of inputs and outputs is defined. Also, at the end of some of the models, there are some internal functions, needed for the calculations, i.e. water density and viscosity calculation. These function are also shown in all the models.

In the S-funtion there was used a universal templates. Thus, the whole code of the s-function is shown only for the resistance block, as in the rest the repeated parts were neglected.

## A1 – Resistance block

```
function [sys,x0,str,ts] = holtrop02(t,x,u,flag)
% calculates sys,x0,str,ts
%
switch flag,

  case 0,
    [sys,x0,str,ts]=mdlInitializeSizes;

  case 1,
    sys=mdlDerivatives(t,x,u);

  case 2,
    sys=mdlUpdate(t,x,u);

  case 3,
    sys=mdlOutputs(t,x,u);

  case 4,
    sys=mdlGetTimeOfNextVarHit(t,x,u);
```

```
  case 9,
    sys=mdlTerminate(t,x,u);

  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end


function [sys,x0,str,ts]=mdlInitializeSizes

sizes = simsizes;

sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 4;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];

function sys=mdlDerivatives(t,x,u)

sys = [];

function sys=mdlUpdate(t,x,u)

sys = [];

function sys=mdlOutputs(t,x,u)

global a VP VC
% function [RT, RF, RAP, RW, RB, RTR, RA, R]= holtrop02
(V,a,VC,VP,Te,W)
% Holtrop Method, with rho,ni function, Fn-dependent form factor

%% Assign of geometrical variables
V = u(1);   % surge velocity of sheep
Te = u(2);  % Water Temperature
W = u(3);   % Water type (0 - salt, 1 - fresh)
h = u(4);   % Sea Depth (surface to bottom)

LBP=a(1);   % Lenght bewtween perpendiculars
LWL=a(2);   % Lenght in waterline
B=a(3);     % Breadth
TF=a(4);    % T for
TA=a(5);    % T aft
T=(TF+TA)/2;% T mid
VOL=a(6);   % Displacement
LCB=a(7);   % Longitudal Centre of Buoyancy
CWP=a(8);   % Waterplane Coefficient
CM=a(9);    % Midship Section Coefficient
S=a(10);    % Wetted Hull Area
```

```
ABT=a(26);  % cross sectional area of bulbous bow
HB=a(27);   % centroid of bulbous bow cross section to keel
AT=a(28);   % area of immersed transom
LR=a(29);   % Lenght of run
IE=a(30);   % half angle of entrance
NBT=a(31);  % Number of Bow thruster

if NBT>=1
    DBT=zeros(NBT,1);   % Diameter of bow thruster
    for i=1:NBT                 % for is used because the number of
bowthrusters is not predefined
        DBT(i)=a(31+i);
    end
else
    i=0;
    DBT=0;
end

NPROP=a(31+i+1);    % Number of propellers
DP=a(31+i+2);       % Propeller Diameter
AAE=a(31+i+3);      % Propeller Expanded blade area ratio
PPD=a(31+i+4);      % Propeller pitch-diameter ratio

%% Assign of coefficients
CSTERN=VC(1);
CAP=VC(2);
SAP=VC(3);

for i=1:NBT;
    CBTT(i)=VC(3+i);
end

% %% Density & Viscocity ni,rho
[rho,ni]=const(Te,W);
g=9.80665;


%% Main Programm

% Friction Resistance RF

Re=V*LWL/ni;
CF=VP(1)./((log10(Re)+VP(2)).^2);
CB=VOL/(LWL*B*T);

if S==0         % if S is given, the following equation is not used
S=LWL*(VP(3)*T+B)*sqrt(CM)*(VP(4)+VP(5)*CB+VP(6)*CM+VP(7)*(B/T)+VP(8)
*CWP)+VP(9)*(ABT/CB); % Wetted surface
end

RF=0.5*rho*S*(V.^2).*CF;
RF=RF';

% Form Factor (1+k1)==ff Froude dependent

c14=VP(18)+VP(19)*CSTERN;
CP=CB/CM;
LR=LWL*(VP(20)-CP+VP(21)*CP*LCB/(VP(22)*CP+VP(23)));
```

```
ff=VP(10)+VP(11)*c14* ((B/LWL)^VP(12))* ((T/LWL)^VP(13))
*((LWL/LR)^VP(14)) *((LWL^3/VOL)^VP(15))* ((VP(16)-CP)^VP(17));
ffo=ff-1;                 % form factor calculated with holtrop method
Fn=V/(sqrt(LWL*g));      % Froude number
Fnff=[0.100; 0.125; 0.150; 0.200; 0.250; 0.300; 0.35; 0.40; 0.45;
0.50; 0.60; 0.80]; % ref ITTC ???
Yff =
[0.9300;0.9395;0.9513;0.9500;0.8744;0.7500;0.5625;0.3800;0.2844;0.220
0;0.1000;0];
Yi=spline(Fnff,Yff,Fn);  % Correction for form factor, cubic spline
based on the froude number
ffo=Yi*ffo;              % New form factor
if h/T < 5
    ffo = ffo + 0.644*((T/h)^1.72); % Shallow water correction
else
    ffo = ffo;
end
ff = ffo+1;
ff=ff';

% Appendage Resistance RAP

CBTT=0.003;
Rthr=(rho*(V.^2)*pi)*sum(((DBT.^2)*CBTT));

RAP=0.5*rho*(V.^2)*(SAP*CAP).*CF+Rthr;
RAP=RAP';

% Wave resistance RW
% Creating "universal" data

c3=VP(35)*(ABT^VP(36)) /(B*T*(VP(37)*sqrt(ABT)+TF-HB));
c2=exp(VP(34)*sqrt(c3));
c5=(VP(38)+(VP(39)*AT/(B*T*CM)));
d=VP(51);

hp1=(LWL^3)/VOL;
    if hp1 < VP(49)
        c15=VP(46);
    elseif (hp1 > VP(49)) && (hp1 < VP(50))
        c15=VP(46)+(((LWL/VOL)^(1/3))+VP(47))/VP(48);
    else
        c15=0;
    end
m4=c15*VP(43)*exp(VP(44)*(Fn.^VP(45)));         % Fn is V
dependent==> m4=[vector]

hp2=LWL/B;
    if hp2 < VP(55)
        lamda=VP(52)*CP + VP(53)*hp2;
    else
        lamda=VP(52)*CP + VP(54);
    end

% Fn - dependent RW
sizev=size(V);           % size and for are used because the number of
speeds is not predefined

for i=1:sizev(2);
```

```
if Fn(i) > 0.55

c17=VP(29)*(CM^VP(30))*((VOL/(LWL^3))^VP(31))*(LWL/B+VP(32))^VP(33);
    m3=VP(40)*((B/LWL)^VP(41))*((T/B)^VP(42));

RW(i)=c17*c2*c5*VOL*rho*g*exp(m3*(Fn(i)^d)+m4(i)*cos(lamda*(Fn(i)^VP(
28))));

elseif Fn(i) < 0.4

    hp3=B/LWL;
      if hp3 < VP(66)
         c7=VP(62)*((hp3)^VP(63));
      elseif (hp3>VP(66)) && (hp3<VP(67))
         c7=hp3;
      else
         c7=VP(64)+VP(65)*(LWL/B);
      end

      if IE==0
         IE=VP(68)+VP(69)*exp((-(LWL/B)^VP(70))*((VP(71)-
CWP)^VP(72))*((VP(73)-
CP+VP(74)*LCB)^VP(75))*((LR/B)^VP(76))*((VP(77)*VOL/(LWL^3))^VP(78)))
;
      end

    c1=VP(57)*(c7^VP(58))*((T/B)^VP(59))*((VP(60)-IE)^VP(61));

    if CP < VP(89)
        c16=VP(82)*CP + VP(83)*(CP^VP(84)) + VP(85)*(CP^VP(86));
    else
        c16=VP(87)+VP(88)*CP;
    end

    m1=VP(79)*LWL/T + VP(80)*(VOL^(1/3))/LWL + VP(81)*B/LWL - c16;

RW(i)=c1*c2*c5*VOL*rho*g*exp(m1*Fn(i)^d+m4(i)*cos(lamda*Fn(i)^VP(56))
);

elseif     Fn(i)>0.4 && Fn(i)<0.55

    Fn(i)=0.55;
      m4=c15*VP(43)*exp(VP(44)*(Fn.^VP(45)));

c17=VP(29)*(CM^VP(30))*((VOL/(LWL^3))^VP(31))*(LWL/B+VP(32))^VP(33);
      m3=VP(40)*((B/LWL)^VP(41))*((T/B)^VP(42));

RW055(i)=c17*c2*c5*VOL*rho*g*exp(m3*(Fn(i)^d)+m4(i)*cos(lamda*(Fn(i)^
VP(28))));

    Fn(i)=0.4;
      m4(i)=c15*VP(43)*exp(VP(44)*(Fn(i)^VP(45)));
      hp3=B/LWL;
```

```
        if hp3 < VP(66)
           c7=VP(62)*((hp3)^VP(63));
        elseif (hp3>VP(66)) && (hp3<VP(67))
           c7=hp3;
        else
           c7=VP(64)+VP(65)*(LWL/B);
        end
        if IE==0
           IE=VP(68)+VP(69)*exp((-(LWL/B)^VP(70))*((VP(71)-
CWP)^VP(72))*((VP(73)-
CP+VP(74)*LCB)^VP(75))*((LR/B)^VP(76))*((VP(77)*VOL/(LWL^3))^VP(78)))
;
        end
        c1=VP(57)*(c7^VP(58))*((T/B)^VP(59))*((VP(60)-IE)^VP(61));
        if CP < VP(89)
            c16=VP(82)*CP + VP(83)*(CP^VP(84)) + VP(85)*(CP^VP(86));
        else
            c16=VP(87)+VP(88)*CP;
        end
        m1=VP(79)*LWL/T + VP(80)*(VOL^(1/3))/LWL + VP(81)*B/LWL - c16;

RW04(i)=c1*c2*c5*VOL*rho*g*exp(m1*Fn(i)^d+m4(i)*cos(lamda*Fn(i)^VP(56
)));

    Fn(i)=V(i)/(sqrt(LWL*g));
    RW(i)=RW04(i) + (VP(90)*Fn(i)+VP(91))*(RW055(i)-RW04(i))/VP(92);
end
end
RW=RW';

% Additional pressure resistance of bulbous bow near the water
surface, RB

Pb=(VP(100)*sqrt(ABT))/(TF + VP(101)*HB);                    %
Coefficient Pb, measure for the emergence of the bow
Fni=V./(sqrt(g*(TF-HB+VP(102)*sqrt(ABT))+VP(103)*(V.*V)));   % Froude
Number based on immersion

RB=VP(93)*exp(VP(94)*(Pb^VP(95)))*(Fni.^VP(96))*(ABT^VP(97))*rho*g./(
VP(97)+(Fni.^VP(98)));
RB=RB';


% Additional pressure resistence of immersed transom stern, RTR

if AT ~= 0          % if AT=0 there is no additional pressure
resistance

    FnT=V./(sqrt(VP(110)*g*AT/(B+B*CWP)));

    for i=1:sizev(2);       % sizev is the size of matrix V, defined
in line 112
    if FnT(i) < VP(109)
        c6(i)=VP(105)*(VP(106)+VP(107)*FnT(i));
    else
        c6(i)=VP(108);
    end
    end
    RTR=VP(104)*rho*V.^2*AT.*c6;
```

```
else
    RTR=zeros(1, sizev(2));
end
RTR=RTR';

% Model-Ship correlation resistance, RA

hp4=TF/LWL;
if hp4 <= VP(120)
    c4=hp4;
else
    c4=VP(119);
end

Ks=120;
if Ks > VP(125)
    CAi=(VP(121)*(Ks^(1/VP(122)))+VP(123))/(LWL^(1/3));
else
    CAi=VP(124);
end
CA=VP(111)*((LWL+VP(112))^VP(113)) + VP(114) +
VP(115)*(sqrt(LWL/VP(116)))*(CB^VP(117))*c2*(VP(118)-c4)+CAi;

RA=0.5*rho*V.^2*S*CA;
RA=RA';


% RT=RF.*ff+RAP+RW+RB+RTR+RA;
RT=RF.*ff+RAP+RW+RB+RTR+RA;
% COEF=0.5*rho*V.^2*S;
% COEF=COEF'
% R=[RT RF RAP RW RB RTR RA]/1000;  % N--> kN

sys = [RT];  % muxed outputs


function sys=mdlGetTimeOfNextVarHit(t,x,u)

sampleTime = 1;    %  Example, set the next hit to be one second
later.
sys = t + sampleTime;

function sys=mdlTerminate(t,x,u)

sys = [];


function [rho, ni]=const(Te, W)
% Calculatin of density & viscocity
g=9.80665;
switch W
    case 0   % salt water
        ni=((1.023379273*0.001787*1000) / (1 + 0.033408772*Te +
0.0001681570669*(Te^2))*10^-6);   %m2/s
        rho= (-0.00048033168167*(Te^2) - 0.0076223076145*Te +
104.83341642)*g;                  %m/s2
```

```
    case 1   % fresh water
        ni=((1.001100823*0.001787*1000) / (1 + 0.034861885*Te +
0.0002110645367*(Te^2))*10^-6);   %m2/s
        rho= (-0.00060464910507*(Te^2) + 0.0035225376681*Te +
101.95448314)*g;                 %m/s2
end
```

## A2 – Hydrodynamic Forces block

```
function [sys,x0,str,ts] = hull_forces03(t,x,u,flag)
……


……
function [sys,x0,str,ts]=mdlInitializeSizes

sizes = simsizes;

sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 4
sizes.NumInputs      = 7;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
……


……
function sys=mdlOutputs(t,x,u)

global a
% function [XH YH NH KH] = black3 (vel,a,Te,W)
% Force Calculation - THE DERIVATIVES CAN GIVEN BY THE USERS AND BE
% CALCULATED BY A METHOD (DEFINED BY THE USER) & is in function form
with
% vel in function inputs

%Assign of input values
for i = 1:4
    vel(i) = u(i);      % u, v ,r, p
end
vel(5:8) = 0;            % because accelerations don't get value from
feedback

Te = u(5);  % Water temperature, oC
W = u(6);   % Water type, 0-salt, 1-fresh
h = u(7);   % Max depth of sea (from surface to bottom)

LBP=a(1);   % Lenght bewtween perpendiculars
LWL=a(2);   % Lenght in waterline
B=a(3);     % Breadth
TF=a(4);    % T for
TA=a(5);    % T aft
T=(TF+TA)/2;% T mid
VOL=a(6);   % Displacement
```

```
NBT=a(31);  % Number of bowthruster, it is needed to read correctly
the input file(it changes the position of each value)

NDER=a(43+NBT); % Number of Hydrodynamic Derivatives given by the
user

Dp = a(31+NBT+2);          % Propeller diameter (5 Yr Nv)
CWPA = a(44+NBT+4*NDER);     % 5 Yv
CPA  = a(45+NBT+4*NDER);     % 5 Yv
xg = a(46+NBT+4*NDER);       % 5 Nr
BP07 = a(47+NBT+4*NDER);
BPS = a(48+NBT+4*NDER);
Izz = a(50+NBT+4*NDER);        % 5 Nr'
TRIM=TF-TA;
CB=VOL/(LWL*B*T);

[rho,ni]=const(Te,W);        % Calculation of water density

% Calculate some values for shallow water   - ref. 6 p. 203
[gv gnr fnv fyv fyr fnr frv]= shallow (h,LBP,B,T,CB);

uvel = sqrt(vel(1)^2 + vel(2)^2);
vel(1) = vel(1) / uvel;
vel(2) = vel(2) / uvel;
vel(3) = vel(3) * LBP / uvel;
vel(4) = vel(4) * LBP / uvel;

m = VOL*rho;                  % MASS

mdim = m/(0.5*rho*(LBP^4)*T); % non dimensional mass
Izzdim = Izz/(0.5*rho*(LBP^4)*T);
%% Assign of the derivatives' inputs to matrices

for i=1:NDER

    dir(i) = a(40+NBT+4*i);  % Assign of the direction of the hydr.
derivative
    index(i,1:8) = a(42+NBT+4*i,1:8);   % Assign of vel. & acc. index
    method(i) = a(43+NBT+4*i);  % Assign of method

end
dir=dir';
index;
method=method';

% Assign and/or calculation of derivatives' value
for i=1:NDER
    switch method(i)
        case 0      %user defined derivatives
            if isequal(index(i,:),[0 2 0 0 0 0 0 0])==1 && dir(i)==1
% Xvv
                Hd(i) = a(41+NBT+4*i);
                if h/T < 5              % Shallow water correction
                    Hd(i) = Hd(i) * frv ;
                else
                    Hd(i) = Hd(i);
                end
```

```
            elseif isequal(index(i,:),[0 0 2 0 0 0 0 0])==1 &&
dir(i)==1 % Xrr
                Hd(i) = a(41+NBT+4*i);
                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gnr ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 1 1 0 0 0 0 0])==1 &&
dir(i)==1 %Xvr
                Hd(i) = a(41+NBT+4*i);
                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * frv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 &&
dir(i)==2 %Yv
                Hd(i) = a(41+NBT+4*i);
                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fyv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 &&
dir(i)==2 %Yr
                Hd(i) = a(41+NBT+4*i);

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fyr ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 2 0 0 0 0 0 0])==1 &&
dir(i)==2  %Yv|v|
                iHd = a(41+NBT+4*i);
                Hd(i) = iHd*sign(vel(2));

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * (9/4*fnv - 5/4) ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 2 0 0 0 0 0])==1 &&
dir(i)==2 %Yr|r|
                iHd = a(41+NBT+4*i);
                Hd(i) = iHd*sign(vel(3));

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fnr ;
                else
                    Hd(i) = Hd(i);
                end
```

```
            elseif isequal(index(i,:),[0 1 1 0 0 0 0 0])==1 &&
dir(i)==2 %Yr|v|
                if sign(vel(2)) > 0
                    Hd(i) = a(41+NBT+4*i);
                else
                    iHd = a(41+NBT+4*i);
                    Hd(i) = iHd*(-1);
                end

            elseif isequal(index(i,:),[0 1 2 0 0 0 0 0])==1 &&
dir(i)==2 %Yvrr
                Hd(i) = a(41+NBT+4*i);

                if h/T < 5                          % Shallow water correction
                    Hd(i) = Hd(i) * fyv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 0 0 0 1 0 0])==1 &&
dir(i)==2 %Yv'
                Hd(i) = a(41+NBT+4*i);

                if h/T < 5                          % Shallow water correction
                    Hd(i) = Hd(i) * gv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 0 0 0 0 1 0])==1 &&
dir(i)==2 %Yr'
                Hd(i) = a(41+NBT+4*i);

                if h/T < 5                          % Shallow water correction
                    Hd(i) = Hd(i) * gv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 &&
dir(i)==3 %Nv
                Hd(i) = a(41+NBT+4*i);

                if h/T < 5                          % Shallow water correction
                    Hd(i) = Hd(i) * fnv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 &&
dir(i)==3 %Nr
                Hd(i) = a(41+NBT+4*i);

                if h/T < 5                          % Shallow water correction
                    Hd(i) = Hd(i) * fnr ;
                else
                    Hd(i) = Hd(i);
                end
```

```
            elseif isequal(index(i,:),[0 2 0 0 0 0 0 0])==1 &&
dir(i)==3 %Nv|v|
                iHd = a(41+NBT+4*i);
                Hd(i) = iHd*sign(vel(2));

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * (9/4*fnv - 5/4) ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 2 0 0 0 0 0])==1 &&
dir(i)==3 %Nr|r|
                iHd = a(41+NBT+4*i);
                Hd(i) = iHd*sign(vel(3));

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 1 1 0 0 0 0 0])==1 &&
dir(i)==3 %Nr|v|
                if sign(vel(2)) > 0
                    Hd(i) = a(41+NBT+4*i);
                else
                    iHd = a(41+NBT+4*i);
                    Hd(i) = iHd*(-1);
                end

            elseif isequal(index(i,:),[0 1 2 0 0 0 0 0])==1 &&
dir(i)==3
                Hd(i) = a(41+NBT+4*i);            % Nvrr

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gnr ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 0 0 0 1 0 0])==1 &&
dir(i)==3 %Nv'
                Hd(i) = a(41+NBT+4*i);

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 2 1 0 0 0 1 0])==1 &&
dir(i)==3 %Nr'
                Hd(i) = a(41+NBT+4*i);

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gnr ;
                else
                    Hd(i) = Hd(i);
```

```
                end

            else
                Hd(i) = 0;

            end

        case 3
            if isequal(index(i,:),[0 2 0 0 0 0 0 0])==1 && dir(i)==1
                Hd(i) = 0.223-0.011*LBP/T;         % Xvv

                if h/T < 5
                    Hd(i) = Hd(i) * frv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 2 0 0 0 0 0])==1 &&
dir(i)==1
                Hd(i) = -0.038-0.001*LBP/T;            % Xrr

                if h/T < 5
                    Hd(i) = Hd(i) * gnr ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 1 1 0 0 0 0 0])==1 &&
dir(i)==1 %Xvr
                Hd(i) = 0.12*CB*B/T-0.018*LBP/T+0.443;

                if h/T < 5                    % Shallow water correction
                    Hd(i) = Hd(i) * frv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 &&
dir(i)==2 %Yv

                SR=BP07/BPS;
                PSR=28.7*CB*B*T/LBP/LBP+0.54;
                DSR=(PSR-SR)/PSR;
                Hd(i) =-0.145-2.25*T/LBP-0.2*DSR;

                if h/T < 5
                    Hd(i) = Hd(i) * fyv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 &&
dir(i)==2 %Yr
                SR=BP07/BPS;
                PSR=28.7*CB*B*T/LBP/LBP+0.54;
                DBL=(0.18-B/LBP)/0.18;
                DSR=(PSR-SR)/PSR;
                Hd(i) =2*CB*B/LBP-
0.282+0.1*DSR+(0.0086*DBL+0.004)*LBP/T;
```

```
                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fyr ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 3 0 0 0 0 0 0])==1 &&
dir(i)==2 % Yvvv
                Hd(i)=-(1.281+0.031*LBP/T);%sign correction according
to experimental data

            elseif isequal(index(i,:),[0 0 3 0 0 0 0 0])==1 &&
dir(i)==2 % Yrrr
                Hd(i)=0.029*CB*B/T-0.004*LBP/T;

            elseif isequal(index(i,:),[0 2 1 0 0 0 0 0])==1 &&
dir(i)==2 % Yrvv
                Hd(i)=0.628*CB*B/T-0.066*LBP/T;

            elseif isequal(index(i,:),[0 1 2 0 0 0 0 0])==1 &&
dir(i)==2 % Yvrr
                Hd(i)=-(0.4+0.007*LBP/T);%sign correction according
to experimental data

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fyv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 &&
dir(i)==3 %Nv
                SR=BP07/BPS;
                PSR=28.7*CB*B*T/LBP/LBP+0.54;
                DSR=(PSR-SR)/PSR;
                Hd(i) =-(0.222+0.1*DSR)+0.00484*LBP/T;

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fnv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 &&
dir(i)==3 %Nr
                SR=BP07/BPS;
                PSR=28.7*CB*B*T/LBP/LBP+0.54;
                DSR=(PSR-SR)/PSR;
                PCB=1.12*T/LBP+0.735;
                DCB=(PCB-CB)/PCB;
                Hd(i) =-(0.0424-0.03*DSR)+(0.004*DCB+0.00027)*LBP/T;

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fnr ;
                else
                    Hd(i) = Hd(i);
                end
```

```
            elseif isequal(index(i,:),[0 3 0 0 0 0 0 0])==1 &&
dir(i)==3
                Hd(i) = 0.188-0.01*LBP/T;              % Nvvv

            elseif isequal(index(i,:),[0 0 3 0 0 0 0 0])==1 &&
dir(i)==3
                Hd(i) = 0.029*CB*B/T-0.004*LBP/T;      % Nrrr

            elseif isequal(index(i,:),[0 2 1 0 0 0 0 0])==1 &&
dir(i)==3
                Hd(i) = 0.178*CB*B/T-0.037*LBP/T;      % Nrvv

            elseif isequal(index(i,:),[0 1 2 0 0 0 0 0])==1 &&
dir(i)==3
                Hd(i) = 0.158-0.005*LBP/T;             % Nvrr

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gnr ;
                else
                    Hd(i) = Hd(i);
                end

            else
                Hd(i)=0;

            end

        case 1
            if isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 && dir(i)==2
                Hd(i) = -(pi*T/LBP+1.4*CB*B/LBP)*(1+(2*TRIM)/(3*T));
% Yv

                if h/T < 5
                    Hd(i) = Hd(i) * fyv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 &&
dir(i)==2
                Hd(i) = (pi/2)*(T/LBP)*(1+0.8*TRIM/T);     %Yr

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fyr ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 2 0 0 0 0 0 0])==1 &&
dir(i)==2
                iHd = 0.09-6.5*(1-CB)*T/B;             %Yv|v|
                Hd(i) = iHd*sign(vel(2));

                if h/T < 5                  % Shallow water
correction
                    Hd(i) = Hd(i) * (9/4*fnv - 5/4) ;
                else
                    Hd(i) = Hd(i);
                end
```

```
        elseif isequal(index(i,:),[0 1 1 0 0 0 0 0])==1 && 
dir(i)==2
            iHd = -0.44+1.78*(1-CB)*T/B;          %Yv|r|

            if sign(v) > 0
                Hd(i) = iHd;
            else
                Hd(i) = iHd*(-1);
            end

        elseif isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 && 
dir(i)==3
            lb = (2*T)/(pi*T+1.4*CB*B);

            Hd(i) = -2*T/LBP*(1-0.27/lb*TRIM/T);      %Nv

            if h/T < 5                % Shallow water correction
                Hd(i) = Hd(i) * fnv ;
            else
                Hd(i) = Hd(i);
            end

        elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 && 
dir(i)==3
            Hd(i) = -(0.54*2*T/LBP-((2*T/LBP)^2))*(1+0.3*TRIM/T);
%Nr

            if h/T < 5                % Shallow water correction
                Hd(i) = Hd(i) * fnr ;
            else
                Hd(i) = Hd(i);
            end

        elseif isequal(index(i,:),[0 0 2 0 0 0 0 0])==1 && 
dir(i)==3
            hp1 = CB*B/LBP;

            if hp1<0.06                                  %Nr|r|
                iHd = -0.060;
            elseif hp1>=0.06 && hp1<=0.2
                iHd = -0.146+1.8*hp1-6*(hp1^2);
            elseif hp1>0.2
                iHd = -0.026;
            end

            Hd(i) = iHd*sign(vel(3));

            if h/T < 5                % Shallow water correction
                Hd(i) = Hd(i) * gv ;
            else
                Hd(i) = Hd(i);
            end

        elseif isequal(index(i,:),[0 2 0 0 0 0 1 0])==1 && 
dir(i)==3
            Hd(i) = -0.2;                                 %Nr'vv

        else
```

```
                Hd(i)=0;

            end

        case 2
            if isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 && dir(i)==2
%Yv
                Hd(i) = -
pi*((T/LBP)^2)*(1+0.4*CB*B/T)*(1+(2*TRIM)/(3*T));

                if h/T < 5
                    Hd(i) = Hd(i) * fyv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 &&
dir(i)==2 %Yr
                Hd(i) = -pi*((T/LBP)^2)*(-0.5+2.2*B/LBP-
0.08*B/T)*(1+0.8*TRIM/T);

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fyr ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 0 0 0 1 0 0])==1 &&
dir(i)==2 %Yv'
                Hd(i) = -pi*((T/LBP)^2)*(1+0.16*CB*B/T-
5.1*((B/LBP)^2));

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 0 0 0 0 1 0])==1 &&
dir(i)==2 %Yr'
                Hd(i) = -pi*((T/LBP)^2)*(0.67*B/LBP-
0.0033*((B/T)^2));

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 &&
dir(i)==3 %Nv
                lb = (2*T)/(pi*T+1.4*CB*B);
                Hd(i) = -pi*((T/LBP)^2)*(-0.5+2.4*T/LBP)*(1-
0.27/lb*TRIM/T);

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fnv ;
                else
                    Hd(i) = Hd(i);
```

```
                end

            elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 &&
dir(i)==3 %Nr
                Hd(i) = -(0.54*2*T/LBP-((2*T/LBP)^2))*(1+0.3*TRIM/T);

                if h/T < 5                  % Shallow water
correction
                    Hd(i) = Hd(i) * fnr ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 0 0 0 1 0 0])==1 &&
dir(i)==3 %Nv'
                Hd(i) = -pi*((T/LBP)^2)*(1.1*B/LBP-0.041*B/T);

                if h/T < 5                 % Shallow water correction
                    Hd(i) = Hd(i) * gv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 2 1 0 0 0 1 0])==1 &&
dir(i)==3 %Nr'
                Hd(i) = -pi*((T/LBP)^2)*(1/12+0.017*CB*B/T-
0.33*B/LBP);

                if h/T < 5                 % Shallow water correction
                    Hd(i) = Hd(i) * gnr ;
                else
                    Hd(i) = Hd(i);
                end

            else
                Hd(i) = 0;

            end

        case 4 % ref. 6
            k = 2*T/LBP;

            if isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 && dir(i)==2
%Yv
                Hd(i) =  -(pi/2*k+1.4*CB*B/LBP);

                if h/T < 5
                    Hd(i) = Hd(i) * fyv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 &&
dir(i)==2 %Yr
                Hd(i) =  pi/4*k;

                if h/T < 5                 % Shallow water correction
                    Hd(i) = Hd(i) * fyr ;
                else
```

```
                Hd(i) = Hd(i);
            end

        elseif isequal(index(i,:),[0 2 0 0 0 0 0 0])==1 &&
dir(i)==2 %Yv|v|
            iHd =  2.5*T/B*(1-CB)+0.5;
            Hd(i) =-(iHd*sign(vel(2))); %sign correction
according to experimental data

            if h/T < 5                % Shallow water correction
                Hd(i) = Hd(i) * (9/4*fnv - 5/4) ;
            else
                Hd(i) = Hd(i);
            end

        elseif isequal(index(i,:),[0 0 2 0 0 0 0 0])==1 &&
dir(i)==2 %Yr|r|
            iHd =  0.343*T/B*CB-0.07;
            Hd(i) = iHd*sign(vel(3));

            if h/T < 5                % Shallow water correction
                Hd(i) = Hd(i) * fnr ;
            else
                Hd(i) = Hd(i);
            end

        elseif isequal(index(i,:),[0 2 1 0 0 0 0 0])==1 &&
dir(i)==2 %Yvvr
            Hd(i) = -(-114*((T/B*CB)^2)+62.12*T/B*CB-8.20); %sign
correction according to experimental data

        elseif isequal(index(i,:),[0 1 2 0 0 0 0 0])==1 &&
dir(i)==2 %Yvrr
            Hd(i) =  -5.95*T/B*(1-CB);

            if h/T < 5                % Shallow water correction
                Hd(i) = Hd(i) * fyv ;
            else
                Hd(i) = Hd(i);
            end

        elseif isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 &&
dir(i)==3 %Nv
            Hd(i) =  -k;

            if h/T < 5                % Shallow water correction
                Hd(i) = Hd(i) * fnv ;
            else
                Hd(i) = Hd(i);
            end

        elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 &&
dir(i)==3 %Nr
            Hd(i) =  -(0.54*k-k^2);

            if h/T < 5                % Shallow water correction
                Hd(i) = Hd(i) * fnr ;
            else
                Hd(i) = Hd(i);
```

```
                end

            elseif isequal(index(i,:),[0 2 0 0 0 0 0 0])==1 &&
dir(i)==3 %Nv|v|
                iHd =  78*((T/B*(1-CB))^2)-19*T/B*(1-CB)-8.20;
                Hd(i) = iHd*sign(vel(2));
                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * (9/4*fnv - 5/4) ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 2 0 0 0 0 0])==1 &&
dir(i)==3 %Nr|r|
                iHd =  0.473*B/LBP*CB-0.089;
                Hd(i) = iHd*sign(vel(3));

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 2 1 0 0 0 0 0])==1 &&
dir(i)==3 %Nvvr
                Hd(i) =  -120*((B/LBP*CB)^2)+35.22*B/LBP*CB-2.72;

            elseif isequal(index(i,:),[0 1 2 0 0 0 0 0])==1 &&
dir(i)==3 %Nvrr
                Hd(i) = 0.5*T/B*CB-0.05;

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gnr ;
                else
                    Hd(i) = Hd(i);
                end

            end


        case 5
            if isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 && dir(i)==2
%Yv
                sa=(1-CWPA)/(1-CPA);
                Hd(i) = -pi*((T/LBP)^2)*(1+6.4*B/LBP+4.9*sa*B/LBP-
108.3*T*B/(LBP^2));

                if h/T < 5
                    Hd(i) = Hd(i) * fyv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 &&
dir(i)==2 %Yr
                Hd(i) = mdim-
pi*((T/LBP)^2)*(0.7+2.1*B/LBP+0.6*CB*B/T-0.4*Dp/T);

                if h/T < 5                % Shallow water correction
```

```
                    Hd(i) = Hd(i) * fyr ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 0 0 0 1 0 0])==1 &&
dir(i)==2 %Yv'
                sa=(1-CWPA)/(1-CPA);
                Hd(i) = mdim-pi*((T/LBP)^2)*(0.66-0.43*B/T-
3.62*sa*B/LBP+1.6*CB*B/T);

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 1 0 0 0 0 0 0])==1 &&
dir(i)==3 %Nv
                sa=(1-CWPA)/(1-CPA);
                Hd(i) = mdim-pi*((T/LBP)^2)*(0.8-1.8*sa*B/LBP-
0.3*Dp/T);

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fnv ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 1 0 0 0 0 0])==1 &&
dir(i)==3 %Nr
                sa=(1-CWPA)/(1-CPA);
                Hd(i) = mdim*Xg-pi*((T/LBP)^2)*(0.4-
1.7*CB*B/LBP+1.3*sa*B/LBP-1.7*CB*Xg*B/T);

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * fnr ;
                else
                    Hd(i) = Hd(i);
                end

            elseif isequal(index(i,:),[0 0 0 0 0 0 1 0])==1 &&
dir(i)==3 %Nr'
                sa=(1-CWPA)/(1-CPA);
                Hd(i) = Izzdim-pi*((T/LBP)^2)*(0.04-0.02*B/T-
0.38*sa*B/LBP+0.08*CB*B/T);

                if h/T < 5                % Shallow water correction
                    Hd(i) = Hd(i) * gnr ;
                else
                    Hd(i) = Hd(i);
                end

            else
                Hd(i)=0;

            end
        end
    end
```

```
% Assign of derivatives values to matrices Xd,Yd,Nd,Kd

Xd=zeros(1,NDER);      % Create matrix for hydr. derivatives in X
direction
Yd=zeros(1,NDER);     % Create matrix for hydr. derivatives in Y
direction
Nd=zeros(1,NDER);      % Create matrix for hydr. derivatives in N
direction
Kd=zeros(1,NDER);     % Create matrix for hydr. derivatives in K
direction

for i=1:NDER
    switch dir(i)
        case 1
            Xd(i) = Hd(i);

        case 2
            Yd(i) = Hd(i);

        case 3
            Nd(i) = Hd(i);

        case 4
            Kd(i) = Hd(i);
    end
end

Xddim=Xd';       % non dimensional hydr. derivatives
Yddim=Yd';       % non dimensional hydr. derivatives
Nddim=Nd';       % non dimensional hydr. derivatives
Kddim=Kd';       % non dimensional hydr. derivatives

% Dimensioning of non dimensional hydr. derivatives
%uvel = vel(1);     % surge speed

Xd = Xddim*(0.5*rho*LBP*T*(uvel^2));
Yd = Yddim*(0.5*rho*LBP*T*(uvel^2));
Nd = Nddim*(0.5*rho*(LBP^2)*T*(uvel^2));
Kd = Kddim*(0.5*rho*(LBP^2)*T*(uvel^2));

% Calculation of forces

hp3(1,1:NDER)=1;
for i=1:NDER
    hp2(i,:)=vel.^index(i,:);   % values that must * with hydr. der.
value to give Force
    for j=1:8
        hp3(i)=hp3(i)*hp2(i,j); % values that must * with hydr. der.
value to give Force
    end

end
hp3=hp3';

XH=sum(Xd.*hp3);
YH=sum(Yd.*hp3);
NH=sum(Nd.*hp3);
KH=sum(Kd.*hp3);
```

```
sys = [XH YH NH KH];  % muxed outputs
……


……
function sys=mdlTerminate(t,x,u)

sys = [];

function [rho, ni]=const(Te, W)
% Calculatin of density & viscocity
g=9.80665;
switch W
    case 0   % salt water
        ni=((1.023379273*0.001787*1000) / (1 + 0.033408772*Te +
0.0001681570669*(Te^2))*10^-6);   %m2/s
        rho= (-0.00048033168167*(Te^2) - 0.0076223076145*Te +
104.83341642)*g;                  %m/s2

    case 1   % fresh water
        ni=((1.001100823*0.001787*1000) / (1 + 0.034861885*Te +
0.0002110645367*(Te^2))*10^-6);   %m2/s
        rho= (-0.00060464910507*(Te^2) + 0.0035225376681*Te +
101.95448314)*g;                  %m/s2
end

function [gv gnr fnv fyv fyr fnr frv]= shallow (h,LBP,B,T,CB)
F = h/T-1;
B1 = CB*B*(1+B/LBP)^2;
K0 = 1 + (0.0775/(F^2)) - (0.0110/(F^3)) + (0.000068/(F^5));
K1 = -(0.0643/F) + (0.0724/(F^2)) - (0.0113/(F^3)) +
(0.0000765/(F^5));

if B/T > 4
    K2 = 0.0342/F;
else
    K2 = 0.137*T/F/B;
end

gv = K0 + 2/3*K1*B1/T + 8/15*K2*(B1/T)^2;
gnr = K0 + 8/15*K1*B1/T + 40/105*K2*(B1/T)^2;
fnv = K0 + K1*B1/T + K2*(B1/T)^2;
fyv = 1.5*fnv - 0.5;
fyr = K0 + 2/5*K1*B1/T + 24/105*K2*(B1/T)^2;
fnr = K0 + 0.5*K1*B1/T + 1/3*K2*(B1/T)^2;
frv = fnv;

% References

%6. Appendix A. Manoeuvering in shallow and confined waters,
Proceedings of
%   the 23rd ITTC - Volume I
```

## A3 – Propeller Forces block

```
function [sys,x0,str,ts] = propellerm(t,x,u,flag)
……

……
function [sys,x0,str,ts]=mdlInitializeSizes

sizes = simsizes;

sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 6;
sizes.NumInputs      = 10;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
……



……
function sys=mdlOutputs(t,x,u)

global a

% function [T Q] = propeller01 (
uvel = u(1);    % surge speed
v = u(2);       % sway speed
r = u(3);       % yaw rate
wp0 = u(5);     % wake fraction ref.(calculated by Utility)
t = u(6);       % thrust deduction factor
nrev = u(7);    % propeller rev. speed
PPD = u(8);     % Propeller pitch-diameter ratio
Te = u(9);      % Water temperature
W = u(10);      % water type

LBP = a(1);         % lenght between perpendiculars
NBT = a(31);        % Number of Bow thruster (needed for indexing of
inputs)
NDER = a(43+NBT);   % Number of derivatives (needed for indexing of
inputs)

NPROP = a(32+NBT);    % Number of propellers
DP = a(33+NBT);       % Propeller Diameter
AAE = a(34+NBT);      % Propeller Expanded blade area ratio
NZ = a(60+NBT+4*NDER);% Number of prop. fins
xP = a(51+NBT+4*NDER);% distance of rudder from ship's center

U = sqrt(uvel*uvel+v*v);
rdim = r/(U/LBP);       % non dimensional r

[rho,ni]=const(Te,W);   % calculation of density & kinematic
viscosity
```

```
%% Creation of the KT matrices of constants

CKT = [0.00880496; -0.204554; 0.166351; 0.158114; -0.147581; -
0.481497;
    0.415437; 0.0144043; -0.0530054; 0.0143481; 0.0606826; -
0.0125894;
    0.0109689; -0.133698; 0.00638407; -0.00132718; 0.168496; -
0.0507214;
    0.0854559; -0.0504475; 0.010465; -0.00648272; -0.00841728;
0.0168424;
    -0.00102296; -0.0317791; 0.018604; -0.00410798; -0.000606848;
    -0.0049819; 0.0025983; -0.000560528; -0.00163652; -0.000328787;
    0.000116502; 0.000690904; 0.00421749; 0.0000565229; -
0.001446564];

% s (J)
KTp1 = [0; 1; 0; 0; 2; 1; 0; 0; 2; 0; 1; 0; 1; 0; 0; 2; 3; 0; 2; 3;
1; 2;
    0; 1; 3; 0; 1; 0; 0; 1; 2; 3; 1; 1; 2; 0; 0; 3; 0];

% t (P/D)
KTp2 = [0; 0; 1; 2; 0; 1; 2; 0; 0; 1; 1; 0; 0; 3; 6; 6; 0; 0; 0; 0;
6; 6;
    3; 3; 3; 3; 0; 2; 0; 0; 0; 0; 2; 6; 6; 0; 3; 6; 3];

% u (AE/AO)
KTp3 = [0; 0; 0; 0; 1; 1; 1; 0; 0; 0; 0; 1; 1; 0; 0; 0; 1; 2; 2; 2;
2; 2;
    0; 0; 0; 1; 2; 2; 0; 0; 0; 0; 0; 0; 0; 1; 1; 1; 2];

% v (Z)
KTp4 = [0; 0; 0; 0; 0; 0; 0; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0; 0; 0; 0;
0; 0;
    1; 1; 1; 1; 1; 1; 2; 2; 2; 2; 2; 2; 2; 2; 2; 2;];

% KTmatrix = [CKT, KTp1, KTp2 KTp3 KTp4] % the whole matrix

%% Creation of the KQ matrices of constants

CKQ = [0.00379368; 0.00886523; -0.032241; 0.00344778; -0.0408811;
    -0.108009; -0.0885381; 0.188561; -0.00370871; 0.00513696;
0.0209449;
    0.00474319; -0.00723408; 0.00438388; -0.0269403; 0.0558082;
0.0161886;
    0.00318086; 0.015896; 0.0471729; 0.0196283; -0.0502782; -
0.030055;
    0.0417122; -0.0397722; -0.00350024; -0.0106854; 0.00110903;
    -0.000313912; 0.0035985; -0.00142121; -0.00383637; 0.0126803;
    -0.00318278; 0.00334268; -0.00183491; 0.000112451; -0.0000297228;
    0.000269551; 0.00083265; 0.00155334; 0.000302683; -0.0001843; -
0.000425399;
    0.0000869243; -0.0004659; 0.0000554194];

% s (J)
KQp1 = [0; 2; 1; 0; 0; 1; 2; 0; 1; 0; 1; 2; 2; 1; 0; 3; 0; 1; 0; 1;
3; 0;
    3; 2; 0; 0; 3; 3; 0; 3; 0; 1; 0; 2; 0; 1; 3; 3; 1; 2; 0; 0; 0; 0;
3; 0;
    1];
```

```
% t (P/D)
KQp2 = [0; 0; 1; 2; 1; 1; 1; 2; 0; 1; 1; 1; 0; 1; 2; 0; 3; 3; 0; 0;
0; 1;
    1; 2; 3; 6; 0; 3; 6; 0; 6; 0; 2; 3; 6; 1; 2; 6; 0; 0; 2; 6; 0; 3;
3; 6;
    6];

% u (AE/AO)
KQp3 = [0; 0; 0; 0; 1; 1; 1; 1; 0; 0; 0; 0; 1; 1; 1; 1; 1; 1; 2; 2;
2; 2;
    2; 2; 2; 2; 0; 0; 0; 1; 1; 2; 2; 2; 2; 0; 0; 0; 1; 1; 1; 1; 2; 2;
2; 2;
    2];

% v (Z)
KQp4 = [0; 0; 0; 0; 0; 0; 0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0; 0;
0; 0;
    0; 0; 0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 2; 2; 2; 2; 2; 2; 2; 2; 2;
2; 2;
    2];

% KQmatrix = [CKQ, KQp1, KQp2 KQp3 KQp4] % the whole matrix


%% Main Calculations
% wake fraction correction
beta = asin(v/U);
K1 = -4.0;
xPdim = xP/LBP;
betaP = beta - 2*xPdim*rdim;
wp = wp0*exp(K1*(betaP.^2));

% advance coef.
uveladv = uvel*(1-wp);       % advance velocity

J = uveladv/(nrev*DP);   % advance coef.

% Calculation of THRUST COEF., KT
hp1 = size(CKT);       % helping param.1 size of matrix (needed for
hp2 = hp1(1);
for i = 1:hp2
    iKT(i) =
CKT(i)*(J^(KTp1(i)))*(PPD^(KTp2(i)))*(AAE^(KTp3(i)))*(NZ^(KTp4(i)));
end

KT = sum(iKT);

% Calculation of MOMENT COEF., KQ
hp3 = size(CKQ);       % helping param.1 size of matrix (needed for
hp4 = hp3(1);
for i = 1:hp4
    iKQ(i) =
CKQ(i)*(J^(KQp1(i)))*(PPD^(KQp2(i)))*(AAE^(KQp3(i)))*(NZ^(KQp4(i)));
end

KQ = sum(iKQ);
```

```
%% Reynolds correction
% Calculation of Reynolds number, Rn
if NZ == 3
    c75R = 2.1475;
else
    c75R = 2.0570;
end

Rn = c75R * (sqrt((uveladv^2)+(0.75*pi*nrev*DP)^2))/ni;
% Rn = 2e6
% Calculation of thrust correction, DKT
DKT = 0.000353485 - ...
     0.00333758*(AAE*(J^2)) - ...
     0.00478125*(AAE*PPD*J) + ...
     0.000257792*((log(Rn)-0.301)^2)*AAE*(J^2) + ...
     0.0000643192*(log(Rn)-0.301)*(PPD^6)*(J^2) - ...
     0.0000110636*((log(Rn)-0.301)^2)*(PPD^6)*(J^2) - ...
     0.0000276305*((log(Rn)-0.301)^2)*NZ*AAE*(J^2) + ...
     0.0000954*(log(Rn)-0.301)*NZ*AAE*PPD*J + ...
     0.0000032049*(log(Rn)-0.301)*(NZ^2)*AAE*(PPD^3)*J;

% Calculation of thrust correction, DKT
DKQ = -0.000591412 +...
    0.00696898*PPD -...
    0.0000666654*NZ*(PPD^6) +...
    0.0160818*(AAE^2) -...
    0.000938091*(log(Rn)-0.301)*PPD -...
    0.00059593*(log(Rn)-0.301)*(PPD^2) +...
    0.0000782099*((log(Rn)-0.301)^2)*(PPD^2) +...
    0.0000052199*(log(Rn)-0.301)*NZ*AAE*(J^2) -...
    0.00000088528*((log(Rn)-0.301)^2)*NZ*AAE*PPD*J +...
    0.0000230171*(log(Rn)-0.301)*NZ*(PPD^6) -...
    0.00000184341*((log(Rn)-0.301)^2)*NZ*(PPD^6) -...
    0.00400252*(log(Rn)-0.301)*(AAE^2) +...
    0.000220915*((log(Rn)-0.301)^2)*(AAE^2);


KT = KT+DKT;
KQ = KQ+DKQ;


%% THRUST & MOMENT calculation

T = KT*rho*(nrev^2)*(DP^4);
Q = KQ*rho*(nrev^2)*(DP^5);

XP = (1-t)*T;
YP = 0;
NP = 0;
KP = 0;
QP = Q;

sys = [XP YP NP KP T QP];  % muxed outputs
……

……
function sys=mdlTerminate(t,x,u)
```

```
sys = [];

function [rho, ni]=const(Te, W)
% Calculatin of density & viscocity
g=9.80665;
switch W
    case 0   % salt water
        ni=((1.023379273*0.001787*1000) / (1 + 0.033408772*Te +
0.0001681570669*(Te^2))*10^-6);   %m2/s
        rho= (-0.00048033168167*(Te^2) - 0.0076223076145*Te +
104.83341642)*g;                  %m/s2

    case 1   % fresh water
        ni=((1.001100823*0.001787*1000) / (1 + 0.034861885*Te +
0.0002110645367*(Te^2))*10^-6);   %m2/s
        rho= (-0.00060464910507*(Te^2) + 0.0035225376681*Te +
101.95448314)*g;                  %m/s2
end
```

## A4 – Rudder Forces block

```
function [sys,x0,str,ts] = rudder02(t,x,u,flag)
……

……
function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 4;  % 8/2/07, gp
sizes.NumInputs      = 11; % 8/2/07, gp
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;   % at least one sample time is needed

sys = simsizes(sizes);
……

……
function sys=mdlOutputs(t,x,u)

global a

%Assign of input values
for i = 1:4
    vel(i) = u(i);      % u, v ,r, p
end
vel(5:8) = 0;             % because accelerations don't get value from
feedback

delta = u(5);
T = u(6);
PPD = u(7);
wp0 = u(8);
nP = u(9);
Te = u(10);
```

```
W = u(11);

% function [XR YR NR KR] = rudder2 (delta,vel,T,ppitch,wp0,nP,a,Te,W)
% Calculation of Forces induced by the Rudder

LBP=a(1);    % Length bewtween perpendiculars
B=a(3);      % Breadth
TF=a(4);     % T for
TA=a(5);     % T aft
T=(TF+TA)/2;% T mid
VOL=a(6);    % Displacement

NBT = a(31);             % Number of bowthrusters(number needed )
NDER = a(43+NBT);        % Number of hydr. deriv. (use

Dp = a(33+NBT);          % Propeller diameter
xP = a(51+NBT+4*NDER);   % distance from propeller to ship c.o.g.
xR = a(52+NBT+4*NDER);   % distance from rudder to ship
zR = a(53+NBT+4*NDER);   % vertical distance from rudder to ship
HR = a(54+NBT+4*NDER);   % Rudder height
lamda = a(55+NBT+4*NDER);%rudder aspect ratio
AR = a(56+NBT+4*NDER);   % rudder surface

wr0 = wp0;               % wake fraction of rudder is taken equal to
prop.'s
uvel = vel(1);           % surge velocity u
v = vel(2);              % sway velocity v
r = vel(3);              % yaw rate r
CB=(LBP*B*T)/VOL;        % block coef.
ppitch = PPD*Dp;         % propeller pitch
U = sqrt(uvel*uvel+v*v);% ship speed
rdim = r/(U/LBP);        % non dimensional r

%kHPR = 1;               % flow rectification factor

%% Calculation of water density
[rho,ni]=const(Te,W);

%% Calculation of aR = delta+delta0-gama*betaRdot
% calculation of betaRdot
beta = asin(v/sqrt(uvel*uvel+v*v));   % Spyrou
xRdim = xR/LBP;     % xR' - nondimensioned rudder-ship center
distance
betaRdot = beta - 2*xRdim*rdim;   % bR' - used for aR calculation

% calculation of gama = CP*CS
% calculation of CP
nH = Dp/HR;

K1 = -4.0;
xPdim = xP/LBP;     % xP' - nondimensioned prop.-ship center distance
betaP = beta - 2*xPdim*rdim;
wp = wp0*exp(K1*(betaP.^2)); % correction of the prop. wake fraction

s = 1-uvel*(1-wp)/(nP*ppitch);

CP = 1/(1+0.6*nH*(2-1.4*s)*s*((1-s)^2));

% calvulation of CS
```

```
K3 = 0.45;
CSO = 0.5;
hp1 = CSO/K3;

if betaRdot <= hp1
    CS = K3*betaRdot;
elseif betaRdot > hp1
    CS = CSO;
end

gama = CP*CS;

% calculation of delta0
%s==SO = 1-u*(1-wp)/(nP*ppitch);  % m/s /(rps m)=round
if delta == 0
    delta0 =0;
else
    delta0 = -(pi*s/90);         % bib 2, page 3
end
%  delta0 = -(pi*s/90);         % bib 2, page 3

% Test for alternative methods for straight seacourse
aR = delta + delta0 - gama*betaRdot;
% if delta == 0
%     aR=0;
% else
%     aR = delta + delta0 - gama*betaRdot;
% end
%% Calculation of entrance speed of rudder
uP = uvel*(1-wp0);                  % velocity of the flow in the
prop.
J = uP/(nP*Dp);                     % advance coef.
KT = T/(rho*(nP^2)*(Dp^4));         % thrust coef.
epsilon = 2.71*CB*B/LBP+0.92;       % bib 2, page 11
k = 0.6/epsilon;

uR = epsilon*uP*sqrt((nH*(1+k*(sqrt(1+(8*KT/(pi*(J^2))))-1))^2)+1-
nH); % surge
% velocity of the flow in the rudder

vR = delta0*uR + gama*(v+xRdim*r);  % lateral velocity of the flow in
the rudder

UR = sqrt((uR^2)+(vR^2));           % inflow velocity of the rudder

%% Calculation of rudder normal force coefficient fa
fa = (6.13*lamda)/(lamda+2.25);


%% Calculation of normal rudder force
FN = 0.5*rho*fa*AR*(UR^2)*(sin(aR));


%% Calculation of rudder forces XR,YR,NR,KR
Vcoef = VOL/(LBP^3);
aH = 40*Vcoef;

XR = -FN*sin(delta);
YR = -(1+aH)*FN*cos(delta);
```

```
NR = -(1+aH)*xR*FN*cos(delta);
KR = (1+aH)*zR*FN*cos(delta);

sys = [XR YR NR KR];  % muxed outputs
……


……
function sys=mdlTerminate(t,x,u)

sys = [];


function [rho, ni]=const(Te, W)
% Calculatin of density & viscocity
g=9.80665;
switch W
    case 0   % salt water
        ni=((1.023379273*0.001787*1000) / (1 + 0.033408772*Te +
0.0001681570669*(Te^2))*10^-6);   %m2/s
        rho= (-0.00048033168167*(Te^2) - 0.0076223076145*Te +
104.83341642)*g;                   %m/s2

    case 1   % fresh water
        ni=((1.001100823*0.001787*1000) / (1 + 0.034861885*Te +
0.0002110645367*(Te^2))*10^-6);   %m2/s
        rho= (-0.00060464910507*(Te^2) + 0.0035225376681*Te +
101.95448314)*g;                   %m/s2
end
```

## A5 – Wind resistance block

```
function [sys,x0,str,ts] = wind02(t,x,u,flag)
……


……
function [sys,x0,str,ts]=mdlInitializeSizes

sizes = simsizes;

sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 4;
sizes.NumInputs      = 5;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
……

……
function sys=mdlOutputs(t,x,u)

global a
% added on template, GP ----------------------------------
```

```
% function [FX, FY, N]= wind01 (VR,gf,a,Tea)
% version 01 = input variables are taken from input file of holtrop
% version 02 = user gives wind angle and speed, and relative values
are
% calculated
uvel = u(1);     % u velocity of the ship
wvel = u(2);     % wind velocity
wangle = u(3);  % angle of wind (0=head on)
heading = u(4); % heading of the ship, deg
Tea = u(5); % Air temperature, used for calculation of the air
density

VR = sqrt((uvel+wvel*cos(wangle))^2+(wvel*sin(wangle))^2);% Wind
speed relative to ship, m/s

wangle = wangle -heading;
gf = acos((uvel+(wvel*cos(wangle)))/VR);% Angle of relative wind off
bow, deg
gf = gf*180/pi;

%% 0) Assign values to variables

i=a(31);                  % Because the number of bow thrusters NBT
affects the following values...
                        % ...in the input file (see input file index)
LOA=a(31+i+5);          % Length overall, m
B=a(3);                 % Breadth, m
AL=a(31+i+6);           % Lateral projeced wind area, m2
ASS=a(31+i+7);           % Lateral projectef area od superstructure,
m2
AT=a(31+i+8);           % Transverse projected wind area, m2
S=a(31+i+9);            % Length of perimeter of lateral projection,
m
C=a(31+i+10);           % Distance from bow to centroid of lateral
projected area, m
M=a(31+i+11);           % Number of distinct groups of masts or king
posts, -


rhoair=360.77819*((Tea)^(-1.00336));       % Air density

%% 1) Create matrices

hgf=[0; 10; 20; 30; 40; 50; 60; 70; 80; 90; 100; 110; 120; 130; 140;
150; 160; 170; 180];



iA0=[2.152; 1.714; 1.818; 1.965; 2.333; 1.726; 0.913; 0.457; 0.341;
0.355; 0.601; 0.651; 0.564; -0.142; -0.677; -0.723; -2.148; -2.707; -
2.529];
iA1=[-5; -3.33; -3.97; -4.81; -5.99; -6.54; -4.68; -2.88; -0.91; 0;
0; 1.29; 2.54; 3.58; 3.64; 3.14; 2.56; 3.97; 3.76];
iA2=[0.243; 0.145; 0.211; 0.243; 0.247; 0.189; 0; 0; 0; 0; 0; 0; 0;
0; 0; 0; 0; -0.175; -0.174];
iA3=[-0.164; -0.121; -0.143; -0.154; -0.19; -0.173; -0.104; -0.068; -
0.031; 0; 0; 0; 0; 0.047; 0.069; 0.064; 0.081; 0.126; 0.128];
iA4=[0; 0; 0; 0; 0; 0.348; 0.482; 0.346; 0;-0.247; -0.372; -0.582; -
0.748; -0.7; -0.529; -0.475; 0; 0; 0];
```

```
iA5=[0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1.27; 1.81;
1.55];
iA6=[0; 0; 0.033; 0.041; 0.042; 0.048; 0.052; 0.043; 0.032; 0.018; -
0.02; -0.031; -0.024; -0.028; -0.032; -0.032; -0.027; 0; 0];
iASE=[0.086; 0.104; 0.096; 0.117; 0.115; 0.109; 0.082; 0.077; 0.09;
0.094; 0.096; 0.09; 0.1; 0.105; 0.123; 0.128; 0.123; 0.115; 0.112];

iB0=[0; 0.096; 0.176; 0.225; 0.329; 1.164; 1.163; 0.916; 0.844;
0.889; 0.799; 0.797; 0.996; 1.014; 0.784; 0.536; 0.251; 0.125; 0];
iB1=[0; 0.22; 0.71; 1.38; 1.82; 1.26; 0.96; 0.53; 0.55; 0; 0; 0; 0;
0; 0; 0; 0; 0; 0];
iB2=[0; 0; 0; 0; 0; 0.121; 0.101; 0.069; 0.082; 0.138; 0.155; 0.151;
0.184; 0.191; 0.166; 0.176; 0.106; 0.046; 0];
iB3=[0; 0; 0; 0.023; 0.043; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; -0.029; -
0.022; -0.012; 0];
iB4=[0; 0; 0; 0; 0; -0.242; -0.177; 0; 0; 0; 0; 0; -0.212; -0.28; -
0.209; -0.163; 0; 0; 0];
iB5=[0; 0; 0; -0.29; -0.59; -0.59; -0.88; -0.65; -0.54; -0.66; -0.55;
-0.55; -0.66; -0.69; -0.53; 0; 0; 0; 0];
iB6=[0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0.34; 0.44; 0.38; 0.27; 0;
0; 0];
iBSE=[0; 0.015; 0.023; 0.03; 0.054; 0.055; 0.049; 0.047; 0.046;
0.051; 0.05; 0.049; 0.047; 0.051; 0.06; 0.055; 0.036; 0.022; 0];

iC0=[0; 0.0596; 0.1106; 0.2258; 0.2017; 0.1759; 0.1925; 0.2133;
0.1827; 0.2627; 0.2102; 0.1567; 0.0801; -0.0189; 0.0256; 0.0552;
0.0081; 0.0851; 0];
iC1=[0; 0.061; 0.204; 0.245; 0.457; 0.573; 0.480; 0.315; 0.254; 0; 0;
0; 0; 0; 0; 0; 0; 0; 0];
iC2=[0; 0; 0; 0; 0; 0; 0; 0; 0; 0; -0.0195; -0.0258; -0.0311; -
0.0488; -0.0422; -0.0381; -0.0306; -0.0122; 0];
iC3=[0; 0; 0; 0; 0.0067; 0.0118; 0.0115; 0.0081; 0.0053; 0; 0; 0; 0;
0.0101; 0.0100; 0.0109; 0.0091; 0.0025; 0];
iC4=[0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0.0335; 0.0497; 0.074; 0.1128;
0.0889; 0.0689; 0.0366; 0; 0];
iC5=[0; -0.074; -0.170; -0.380; -0.472; -0.523; -0.546; -0.526; -
0.443; -0.508 ;-0.492 ;-0.457; -0.396; -0.420; -0.463; -0.476; -
0.415; -0.220; 0];
iCSE=[0; 0.0048; 0.0074; 0.0105; 0.0137; 0.0149; 0.0133; 0.0125;
0.0123; 0.0141; 0.0146; 0.0163; 0.0179; 0.0166; 0.0162; 0.0141;
0.0105; 0.0057; 0];

A0=[hgf, iA0];
A1=[hgf, iA1];
A2=[hgf, iA2];
A3=[hgf, iA3];
A4=[hgf, iA4];
A5=[hgf, iA5];
A6=[hgf, iA6];
ASE=[hgf, iASE];

B0=[hgf, iB0];
B1=[hgf, iB1];
B2=[hgf, iB2];
B3=[hgf, iB3];
B4=[hgf, iB4];
B5=[hgf, iB5];
B6=[hgf, iB6];
BSE=[hgf, iBSE];
```

```
C0=[hgf, iC0];
C1=[hgf, iC1];
C2=[hgf, iC2];
C3=[hgf, iC3];
C4=[hgf, iC4];
C5=[hgf, iC5];
CSE=[hgf, iCSE];

%% 2) Interpolation from matrices to find coef. A0-A6, B0-B6, C0-C6, 
ASE,
%% BSE and CSE
A0=interp1(A0(:,1),A0(:,2),gf);
A1=interp1(A1(:,1),A1(:,2),gf);
A2=interp1(A2(:,1),A2(:,2),gf);
A3=interp1(A3(:,1),A3(:,2),gf);
A4=interp1(A4(:,1),A4(:,2),gf);
A5=interp1(A5(:,1),A5(:,2),gf);
A6=interp1(A6(:,1),A6(:,2),gf);
ASE=interp1(ASE(:,1),ASE(:,2),gf);

AS=[A0 A1 A2 A3 A4 A5 A6 ASE]';

B0=interp1(B0(:,1),B0(:,2),gf);
B1=interp1(B1(:,1),B1(:,2),gf);
B2=interp1(B2(:,1),B2(:,2),gf);
B3=interp1(B3(:,1),B3(:,2),gf);
B4=interp1(B4(:,1),B4(:,2),gf);
B5=interp1(B5(:,1),B5(:,2),gf);
B6=interp1(B6(:,1),B6(:,2),gf);
BSE=interp1(BSE(:,1),BSE(:,2),gf);

BS=[B0 B1 B2 B3 B4 B5 B6 BSE]';

C0=interp1(C0(:,1),C0(:,2),gf);
C1=interp1(C1(:,1),C1(:,2),gf);
C2=interp1(C2(:,1),C2(:,2),gf);
C3=interp1(C3(:,1),C3(:,2),gf);
C4=interp1(C4(:,1),C4(:,2),gf);
C5=interp1(C5(:,1),C5(:,2),gf);
CSE=interp1(CSE(:,1),CSE(:,2),gf);

CS=[C0 C1 C2 C3 C4 C5 CSE]';


%% 3) Calculation of CX, CY, CN

CX = A0 + A1*(2*AL/(LOA^2)) + A2*(2*AT/(B^2)) + A3*(LOA/B) + 
A4*(S/LOA) + A5*(C/LOA) + A6*M;
CY = B0 + B1*(2*AL/(LOA^2)) + B2*(2*AT/(B^2)) + B3*(LOA/B) + 
B4*(S/LOA) + B5*(C/LOA) + B6*(ASS/AL);
CN = C0 + C1*(2*AL/(LOA^2)) + C2*(2*AT/(B^2)) + C3*(LOA/B) + 
C4*(S/LOA) + C5*(C/LOA);


%% 4) Calculation of FX, FY, N

FX = 0.5*rhoair*(VR.^2)*AT.*CX/1e3;
FY = 0.5*rhoair*(VR.^2)*AT.*CY/1e3;
N = 0.5*rhoair*(VR.^2)*AT*LOA.*CN/1e3;
```

```
K = 0;

sys = [FX FY N K ];  % muxed outputs
……

……
function sys=mdlTerminate(t,x,u)

sys = [];

% end mdlTerminate

function [rho, ni]=const(Te, W)
% Calculatin of density & viscocity
g=9.80665;
switch W
    case 0   % salt water
        ni=((1.023379273*0.001787*1000) / (1 + 0.033408772*Te +
0.0001681570669*(Te^2))*10^-6);   %m2/s
        rho= (-0.00048033168167*(Te^2) - 0.0076223076145*Te +
104.83341642)*g;                  %m/s2

    case 1   % fresh water
        ni=((1.001100823*0.001787*1000) / (1 + 0.034861885*Te +
0.0002110645367*(Te^2))*10^-6);   %m2/s
        rho= (-0.00060464910507*(Te^2) + 0.0035225376681*Te +
101.95448314)*g;                  %m/s2
end
```

## A6 – Miscellaneous calculations Block

```
function [sys,x0,str,ts] = utilitym(t,x,u,flag)
……

……
function [sys,x0,str,ts]=mdlInitializeSizes

sizes = simsizes;

sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 9;
sizes.NumInputs      = 3;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
……

……
function sys=mdlOutputs(t,x,u)

global a VC VP
```

```
% added on template, GP ----------------------------------

%Assign of input values

uvel = u(1);      % u, v ,r, p
Te = u(2);
W = u(3);


% function [w t nR mxx myy Jzz] = utility(uvel,a,VC,VP,Te,W)
% Calculation of w,t,nR and mxx,myy,Jzz

%%Calculation of propulsion factors w,t,nR

%% Assign of geometrical variables

LBP=a(1);   % Lenght bewtween perpendiculars
LWL=a(2);   % Lenght in waterline
B=a(3);     % Breadth
TF=a(4);    % T for
TA=a(5);    % T aft
T=(TF+TA)/2;% T mid
VOL=a(6);   % Displacement
LCB=a(7);   % Longitudal Centre of Buoyancy
CWP=a(8);   % Waterplane Coefficient
CM=a(9);    % Midship Section Coefficient
S=a(10);    % Wetted Hull Area
ABT=a(26);  % cross sectional area of bulbous bow
HB=a(27);   % centroid of bulbous bow cross section to keel
AT=a(28);   % area of immersed transom
LR=a(29);   % Lenght of run
IE=a(30);   % half angle of entrance
NBT=a(31);  % Number of Bow thruster

NPROP=a(32+NBT);    % Number of propellers
DP=a(33+NBT);       % Propeller Diameter
AAE=a(34+NBT);      % Propeller Expanded blade area ratio
PPD=a(35+NBT);      % Propeller pitch-diameter ratio

NDER = a(43+NBT);
Ixx = a(49+NBT+4*NDER);
Izz = a(50+NBT+4*NDER);
w0 = a(57+NBT+4*NDER);
t0 = a(58+NBT+4*NDER);
nR0 = a(59+NBT+4*NDER);
mxx0 = a(60+NBT+4*NDER);
myy0 = a(61+NBT+4*NDER);
Jzz0 = a(62+NBT+4*NDER);

CSTERN = VC(1);      % Stern type coef.

CB=VOL/(LWL*B*T);
CP=CB/CM;

[rho,ni]=const(Te,W);
g=9.80665;

mship = VOL*rho;
```

```
if S==0          % if S is given, the following equation is not used
S=LWL*(VP(3)*T+B)*sqrt(CM)*(VP(4)+VP(5)*CB+VP(6)*CM+VP(7)*(B/T)+VP(8)
*CWP)+VP(9)*(ABT/CB); % Wetted surface
end

Re=uvel*LWL/ni;
CF=VP(1)./((log10(Re)+VP(2)).^2);

c14=VP(18)+VP(19)*CSTERN;
LR=LWL*(VP(20)-CP+VP(21)*CP*LCB/(VP(22)*CP+VP(23)));
ff=VP(10)+VP(11)*c14* ((B/LWL)^VP(12))* ((T/LWL)^VP(13))
*((LWL/LR)^VP(14)) *((LWL^3/VOL)^VP(15))* ((VP(16)-CP)^VP(17));
ffo=ff-1;                    % form factor calculated with holtrop method
Fn=uvel/(sqrt(LWL*g));         % Froude number
Fnff=[0.100; 0.125; 0.150; 0.200; 0.250; 0.300; 0.35; 0.40; 0.45;
0.50; 0.60; 0.80];
Yff =
[0.9300;0.9395;0.9513;0.9500;0.8744;0.7500;0.5625;0.3800;0.2844;0.220
0;0.1000;0];
Yi=spline(Fnff,Yff,Fn);  % Correction for form factor, cubic spline
based on the froude number
ff=1+Yi*ffo;             % New form factor
k = ff-1;
c3=VP(35)*(ABT^VP(36)) /(B*T*(VP(37)*sqrt(ABT)+TF-HB));
c2=exp(VP(34)*sqrt(c3));
hp4=TF/LWL;
if hp4 <= VP(120)
    c4=hp4;
else
    c4=VP(119);
end
Ks=120;
if Ks > VP(125)
    CAi=(VP(121)*(Ks^(1/VP(122)))+VP(123))/(LWL^(1/3));
else
    CAi=VP(124);
end
CA=VP(111)*((LWL+VP(112))^VP(113)) + VP(114) +
VP(115)*(sqrt(LWL/VP(116)))*(CB^VP(117))*c2*(VP(118)-c4)+CAi;

if NPROP == 1

    if CSTERN == 10

        if w0 <= 0
            CV = (1+k)*CF + CA;
            w = 0.3*CB + 10*CV*CB - 0.1;
        else
            w =w0;
        end

        if t0 <= 0
            t = 0.10;
        else
            t = t0;
        end

        if nR0 <= 0
            nR = 0.98;
```

```
        else
            nR =nR0;
        end

    else
        % Calculation of wake fraction, w

        if w0 <= 0

            hp1 = B/TA;
            if hp1<5
                c8 = B*S/(LBP*DP*TA);
            else
                c8 = S*(7*B/TA-25)/(LBP*DP*(B/TA-3));
            end

            if c8 < 28
                c9 = c8;
            else
                c9 = 32-16/(CB-24);
            end

            hp2 = TA/DP;
            if hp2 < 2
                c11 = TA/DP;
            else
                c11 = 0.08333333*((TA/DP)^3)+1.33333;
            end

            if CP < 0.7
                c19 = 0.12997/(0.95 - CB) - 0.11056/(0.95 - CP);
            else
                c19 = 0.18567/(1.3571 - CM) - 0.71276 + 0.38648*CP;
            end

            c20 = 1 + 0.015*CSTERN;
            CP1 = 1.45*CP - 0.315 - 0.0225*LCB;
            CV = (1+k)*CF + CA;

            w = c9*c20*CV*LBP/TA*(0.050776 + 0.93405*c11*CV/(1-CP1))
+ 0.27915*c20*sqrt(B/LBP/(1-CP1)+c19*c20);

        else
            w = w0;
        end

        % Calculation of thrust deduction, t
        if t0 <= 0
            t =
0.25014*((B/LBP)^0.28956)*(((sqrt(B*T))/DP)^0.2624)/((1-
CP+0.0225*LCB)^0.01762)+0.0015*CSTERN;
        else
            t = t0;
        end

        % Calculation of relative-rotative efficiency
        if nR0 <= 0
            nR = 0.9922 - 0.05908*AAE + 0.07424*(CP - 0.0225*LCB) ;
        else
```

```
            nR = nR0;
        end
    end

elseif NPROP == 2

    if w0 <= 0
        CV = (1+k)*CF + CA;
        w = 0.3095*CB + 10*CV*CB - 0.23*DP/(sqrt(B*T));
    else
        w = w0;
    end

    if t0 <= 0
        t = 0.325*CB - 0.1885*DP/(sqrt(B*T));
    else
        t = t0;
    end

    if nR0 <= 0
        nR = 0.9737 + 0.111*(CP - 0.0225*LCB) - 0.06325*PPD;
    else
        nR = nR0;
    end

else
    if w0 <= 0
        CV = (1+k)*CF + CA;
        w = 0.3095*CB + 10*CV*CB - 0.23*DP/(sqrt(B*T));
    else
        w = w0;
    end

    if t0 <= 0
        t = 0.325*CB - 0.1885*DP/(sqrt(B*T));
    else
        t = t0;
    end

    if nR0 <= 0
        nR = 0.9737 + 0.111*(CP - 0.0225*LCB) - 0.06325*PPD;
    else
        nR = nR0;
    end

end

%% Calculation of added masses mxx,myy and added moment of inertia
Jzz

% Density & Viscocity ni,rho
[rho,ni]=const(Te,W);
g=9.80665;

% Calculation of non-dimensional added mass mxx [kg]
if mxx0 <= 0
    mxx = 2.7*rho/(LBP^2)*((CB*LBP*B*T)^(5/3));   %ref 3, 4
    %mxx = 0.5*rho*(LBP^2)*T*138.5e-5
else
```

```
    mxx = 0.5*rho*(LBP^2)*T*mxx0;
end


% Calculation of non-dimensional added mass myy [kg]
if myy0 <= 0
    myy = 0.5*pi*rho*(T^3)*(1+0.16*CB*B/T -
5.1/((LBP/B)^2));%corrected to Prime I ref 2
    %myy = 0.5*rho*(LBP^2)*T*1423.5e-5
else
    myy = 0.5*rho*(LBP^2)*T*myy0;
end


% Calculation of non-dimensional added moment of inertia Jzz
if Jzz0 <= 0
    Jzz = 0.5*pi*rho*(LBP^3)*(T^2)/24*(1+0.2*CB*T/B - 4/(LBP/B));
%ref 3
    %Jzz = 0.5*rho*(LBP^4)*T*47.5e-5
    %Jzz = 0.5*pi*rho*(LBP^2)*(T^3)*(1/12+0.017*CB*B/T-0.33*B/LBP);
%other method ref 2
else
    Jzz = 0.5*rho*(LBP^4)*T*Jzz0;
end

sys = [w t nR mxx myy Jzz Ixx Izz mship];  % muxed outputs
……


……
function sys=mdlTerminate(t,x,u)

sys = [];

% end mdlTerminate

function [rho, ni]=const(Te, W)
% Calculatin of density & viscocity
g=9.80665;
switch W
    case 0   % salt water
        ni=((1.023379273*0.001787*1000) / (1 + 0.033408772*Te +
0.0001681570669*(Te^2))*10^-6);   %m2/s
        rho= (-0.00048033168167*(Te^2) - 0.0076223076145*Te +
104.83341642)*g;                   %m/s2

    case 1   % fresh water
        ni=((1.001100823*0.001787*1000) / (1 + 0.034861885*Te +
0.0002110645367*(Te^2))*10^-6);   %m2/s
        rho= (-0.00060464910507*(Te^2) + 0.0035225376681*Te +
101.95448314)*g;                   %m/s2
end
```

# A7 – Manouevering Block

## A7.1 – Turning Circle

```
start_time = u(1);      % Time, when rudder starts steering, s
init_rud = u(2);        % initial rudder angle, deg
rate = u(3);            % turning rate of rudder, deg/s
max_rud = u(4);         % max. rudder angle, deg

if t < start_time
    delta = init_rud;
else
    delta = rate*(t-start_time) + init_rud;
end

 if delta > max_rud
     delta = max_rud;
 elseif delta < - max_rud
     delta = - max_rud;
 else
     delta = delta;
 end
```

## A7.2 Zig-Zag Manoeuver

```
heading = u(1);      % heading of the ship, deg
rvel = u(2);         % r velocity
ref_heading = u(3);  % reference heading for zig zag manouevre,deg
r0 = u(4);           % turning rate of rudder, deg/s

if ref_heading > 0

    if heading <= ref_heading && rvel >= 0
        rate = r0;
     elseif heading > ref_heading && rvel >  0
        rate = -r0;
     elseif heading >= -ref_heading && rvel < 0
         rate = -r0;
     elseif heading < -ref_heading && rvel < 0
         rate = r0;
     end

 elseif ref_heading < 0

     if heading >= ref_heading && rvel <= 0
         rate = -r0;
     elseif heading < ref_heading && rvel <  0
         rate = r0;
     elseif heading <= -ref_heading && rvel > 0
         rate = r0;
     elseif heading > -ref_heading && rvel > 0
         rate = -r0;
     end
 end
```

# Appendix B – Abbreviations

| | |
|---|---|
| MOERI | Maritime and Ocean Engineering research institute |
| SNAME | Society of Naval Architectures and Marine Engineers |
| IMO | International Maritime Organization |
| KRISO | Korea Research Institute of Ships and Ocean Engineering |
| MARIN | Maritime Research Institute Netherlands |
| KCS | KRISO Containership |
| KVLCC | Kriso Very Large Crude-oil Carrier |
| ITTC | International Towing Tank Comitee |
| FPP | Fixed Pitch Propeller |
| CPP | Controllable Pitch Propeller |
| DOF | Degrees Of Freedom |
| COG | Center Of Gravity |
| NBT | Number of Bow Thrusters |
| NDER | Number of hydrodynamice DERivatives |
| GUI | Graphical User Interface |

References

[1] Ankudinov, V.K., Miller, E.R., Jakobsen, B.K. and Daggett, L.L., 1990, "Manoeuvring performance of tug/barge assemblies in restricted waterways", Proceedings MARSIM & ICMS 90, Tokyo, Japan, pp. 515-525.

[2] Clarke, D., Gedling, P. and Hine, G. (1983). The application of manoeuvring criteria in hull design using linear theory. The Naval Architect, pp. 45–68.

[3] Holtrop, J., Mennen, G.G.J, "An Approximate Power prediction Method", International Shipbuilding Progress, Vol.29, No. 335, July 1982

[4] Holtrop, J., "A Statistical Re-Analysis of Resistance and Propulsion Data", International Shipbuilding Progress, Vol.31, No. 363, Nov 1984

[5] Inoue, S., Hirano, M., Kijima, K., and Takashina, J. "A Practical Calculation Method of Ship Manoeuvring Motion" International Shipbuilding Progress, 28(325), 207–222., 1981

[6] Isherwood, R. M., 1972: Wind resistance of merchant ships. Trans. of the Royal Institution of Naval Architects. 115, 327- 338, 1972

[7] ITTC, "Report of the Power Performance Comitee", 1990

[8] ITTC, "The manoeuvring comitee", 23rd international towing tank conference, p. 206, 2002

[9] ITTC, “The Specialist Comitee of ESSO OSAKA”, 23rd international towing tank conference, p. 581-743, 2002

[10] M.D. Woodward, D. Clarke, M. Atlar, "On The Manoeuvring Prediction Of Pod Driven Ships", MARSIM 2003, Kanazawa, Japan, 2003

[11] T.I. Lee, K.S. Ahn, H.S Lee, O.D.J Yum, "An empirical prediction of hydrodynamic coefficients for modern ship hulls", MARSIM 2003, Kanazawa, Japan, 2003

[12] Appendix A. Manoeuvering in shallow and confined waters, Proceedings of the 23rd ITTC - Volume I

[13] M.W.C. Oosterveld, P.V. Oossanen, "Further Computer-Analyzed Data of the Wageningen B-Screw Series" ,(I.S.P., Vol. 22, No. 251, July 1975)

[14] Eloot K, Vantorre M (2004) Prediction of low-speed manoeuvring based on captive model tests: opportunities and limitations. 31st annual general meeting of IMSF, Antwerp Maritime Academy & Flanders Hydraulics Research, Antwerp, 13–17 September 2004

[15] http://www.simman2008.dk/

[16] IMO, “Explanatory notes to the standards for ship manieuvrability” Ref. T4/3.01 , 2002
[17] V.P. Lambropoulos, “Modular Simulation of marine Propulsion Systems”. Ph.D Thesis, LME, NTUA, March 2000,
[18] T. Perez, M. Blanke, “Mathematical Ship Modeling for Control Applications”, Technical Report, 2002