SINTEF

Report

User Manual
ShipX Station Keeping Plug-In

Author(s)
Edvard Ringen

SINTEF Ocean AS

SINTEF Ocean AS

Address:
Postboks 4762 Torgarden
NO-7465 Trondheim
NORWAY

Switchboard: +47 46415000

Enterprise /VAT No:
NO 937 357 370 MVA

Report

User Manual
ShipX Station Keeping Plug-In

REPORT NO.	PROJECT NO.	VERSION	DATE
		7.0.5	2022-01-19

KEYWORDS:

AUTHOR(S)
Edvard Ringen

CLIENT(S)
Multi-client

CLIENT'S REF.		NUMBER OF PAGES/APPENDICES:
		119

CLASSIFICATION	CLASSIFICATION THIS PAGE	ISBN

ABSTRACT
This report describes the theory behind- and the usage of the ShipX Station Keeping Plug-In. Investigating the station keeping capabilities of a ship in the early design process is important with respect to dimensioning, positioning and usage of the propulsors and power system on the ship. This software eases investigations into scaling and moving propulsors around the ship including investigating propulsor fault scenarios. The ShipX Station Keeping Plug-In is designed to be a tool where a station keeping capability calculation is performed with a minimum amount of input data.

It is highly recommended to read chapter 3 of this user manual to get a quick guide into how to use the software. The examples in chapter 18 are also a good way into start using the program.

PREPARED BY	CHECKED BY	APPROVED BY
Edvard Ringen		

Document History

VERSION	DATE	VERSION DESCRIPTION
7.0.0	2019-12-13	Initial version of new plug-in design.
7.0.1	2020-07-13	Added DNVGL DPCAP 2018 Level 2 and Level 2 Site.
7.0.2	2021-09-30	Added information about running batch calculations with variable input. Changed SKP legend to April 2020 (no calculation changes). Added some file formats. Fixed power module recursive switch board bug.
7.0.3	2021-10-04	Improved allocation logic to attempt different strategies and select the best result.
7.0.5	2022-01-19	Added DNV DPCAP 2021 Level 1, 2 and 2 Site. Added information on different ways of running multiple calculations and/or parameter studies, see chapter 10.

SINTEF

Table of Contents

SINTEF

SINTEF

1 INTRODUCTION

1.1 General

This manual describes the usage of the ShipX Station Keeping Plug-In. Investigating the station keeping capabilities of a ship in the early design process is important with respect to dimensioning, positioning and usage of the propulsors on the ship. This software eases investigations into scaling and moving propulsors around the ship including investigating propulsor fault scenarios. The calculations are performed as static calculations in the horizontal plane.

Several propulsors are implemented in the ShipX Station Keeping Plug-In. The ShipX Station Keeping Plug-In has built in functionality enabling the users to define their own propulsor models in addition to the already implemented propulsors. User defined propulsors will be seamlessly integrated into the plug-in user interface and in the calculations. This user guide also gives the complete step-by-step procedure required for creating and implementing your own propulsors.

The ShipX Station Keeping Plug-In is designed to be a tool where a station keeping capability calculation is performed with a minimum amount of input data.

It is highly recommended to read chapter 3 of this user manual and then follow the example(s) in chapter 18 to get a quick guide into how to use the software.

Also make sure to read chapter 15 in this manual thoroughly before you start using the station keeping plug-in.

*Please note that this user manual explains all features of the plug-in. Depending on your license, some of these features may be unavailable to you. Some features may be locked to a single user or a group of users. Features of limited availability are marked with * in the feature description. If you would like to add any features to your version of the plug-in, please contact SINTEF Ocean to investigate the feasibility.*

2 GENERAL INFORMATION

2.1 General

Before starting the ShipX Station Keeping Plug-In, one must make sure that all input data are available.

Required input for station keeping calculations:

- Hull lines including superstructure data
- Propulsor data
- Power system layout

For more information about working with ShipX, read the ShipX general documentation found in the Help menu in ShipX.

In general, hover the mouse pointer on any object in the user interface to get further description on that object.

2.2 Coordinate System

The ShipX Station Keeping Plug-In uses two coordinate systems, one for input (standardized in ShipX) and one for calculation and reporting.

2.3 Coordinate System For Hull Fixed Input And Output

This coordinate system is used throughout the plug-in graphical user interface. It follows the standard ShipX coordinate system with positive x-axis forward, positive y-axis to starboard and positive z-axis up. This coordinate system is located at baseline at the aft perpendicular. It is used for defining positional input for hull fixed input such as propulsors and point of attack for constant tension external forces. The reports will reflect these inputs in the same coordinate system as they were defined in.

2.4 Coordinate System For Global Input And Output

This coordinate system is used for definition of global environmental direction, during calculations and in reporting of forces and moments. The coordinate system is described in Figure 1.

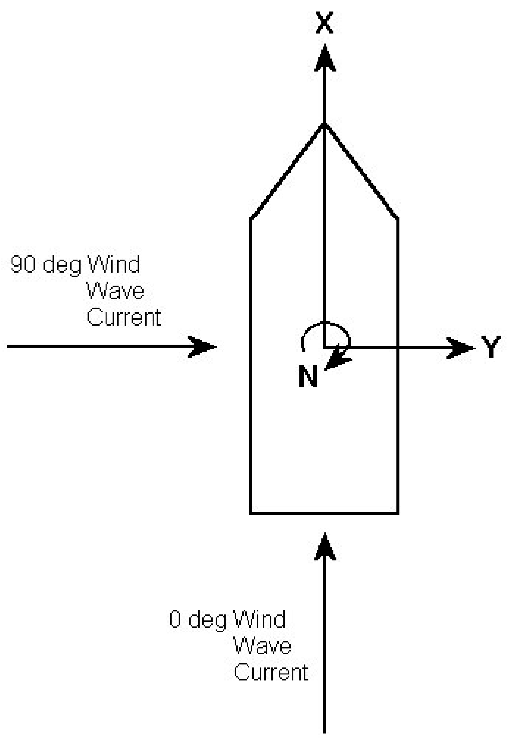


Figure 1, Force and moment coordinate system

The coordinate system is centered at midships with positive X-direction forward, positive Y- direction to starboard and positive Z-direction down. All environmental force directions are defined identically (see Figure 1) with a clockwise rotation.
The ship heading is defined at zero degrees as shown in Figure 1, with clockwise rotation.

2.5 Station Keeping Calculation Method Overview

There are two main ways of running the plug-in; iterate to find maximum withstandable weather (capability study) or run with a fixed weather (propulsor load). A graphical illustration of the two main calculation methods are shown below.

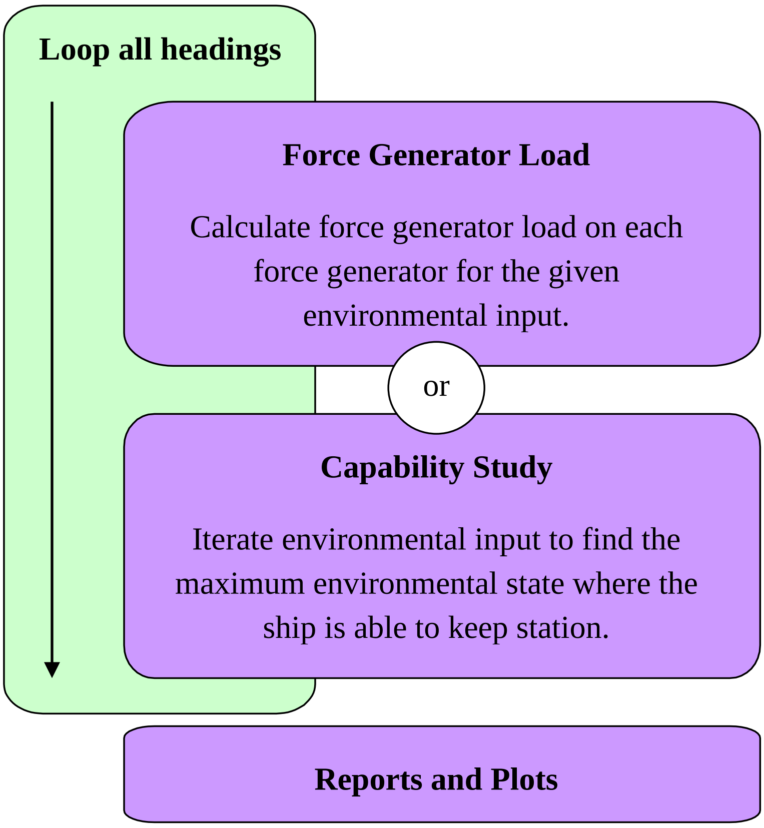


Figure 2, Calculation structure

There are many ways of running a capability study, from the user defined generic calculations to the strictly limited methods defined by class societies. Explore the options in the plug-in and in this user manual.

3 GETTING STARTED

In order to get started using the program you should familiar yourself with the user interface. Detailed step-by-step examples on how to create input for different types of calculations are outlined in chapter 18. In this chapter we will cover the process in a more superficial manner to get you started quickly.

The basic workflow is:

- Install ShipX
- Create a workspace
- Create a ship (either by importing a ship geometry file or by using main dimensions only)

You are now ready to create a Station Keeping run:

- Create the Station Keeping run
- Select calculation method and set required input
- Create Propulsors
- Create a Propulsor Configuration
- Create a Power Configuration
- Run the calculations

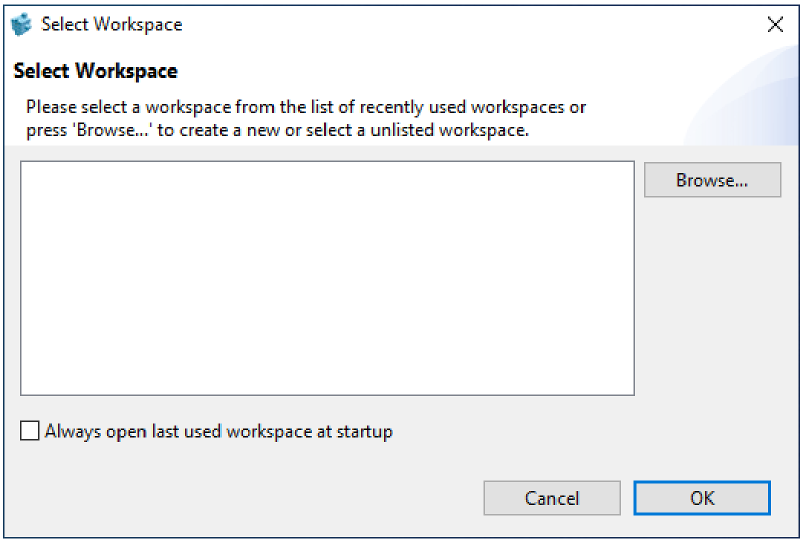
3.1 Install ShipX

In order to install ShipX, do the following:

- Save the license file you received from SINTEF Ocean to a location of your choice
- Download the latest ShipX release
- Unzip the downloaded zip-file to a folder of your choice
- Open the folder, right-click the **ShipX.exe** file, select **Send To** - **Desktop (create shortcut)**, name the shortcut **ShipX**
- Double-click the **ShipX** shortcut
- Create a workspace as outlined in chapter 3.2
- Specify the license file location (or server name) in the **Missing or Invalid FLEXlm License** dialog

3.2 Create A Workspace

A workspace is where the data you are working on is saved (similar to the database in the old ShipX). The first time you start ShipX you will be asked to create a workspace.



To create a workspace, do the following:

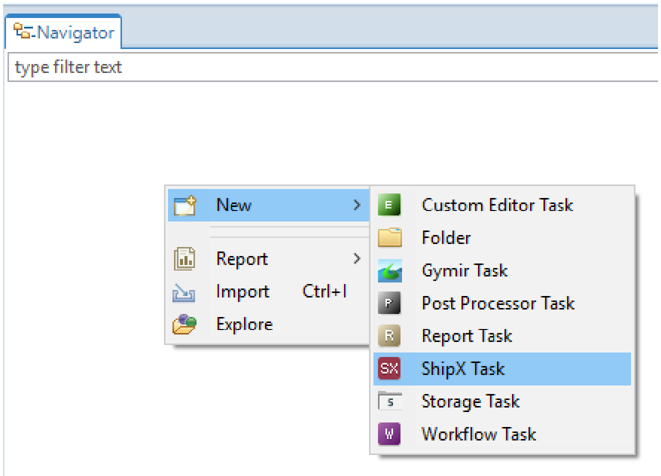
- Click the **Browse...** button
- Select the desired folder (or create a new one) in the **Select Workspace Directory** dialog

To manually create a new workspace later on, use the **File** - **Open / Create Workspace** menu in ShipX.

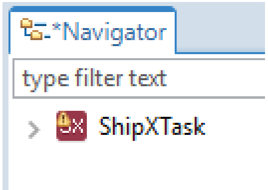
3.3 Create A Ship

Before we can create a ship, we must make a task in ShipX into which we will create the ship. A task is a logical divider in a workspace into which you can sort your projects/work. To create a task:

- Right-click the Navigator window
- Select **New** – **ShipX Task**

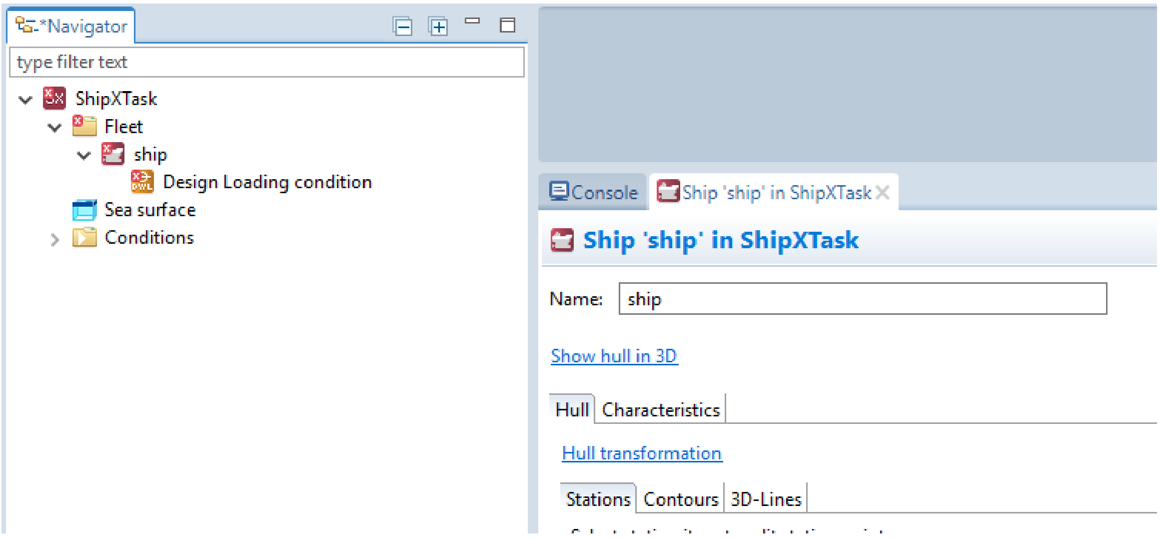


- In the wizard that appears, give the task a name and click **Finish**. You will now have a ShipX task in your **Navigator** window.



The Station Keeping plug-in in ShipX requires a ship as basis, either defined by a geometry file or by main dimensions. Using a geometry file (right-click the ShipX task, select **Import** – **ShipX Geometry**) will aid you in defining the input as some input will be automatically derived from the geometry file. For simplicity sake, in this example we will create a ship without a geometry file. To create a ship without a geometry file, do the following:

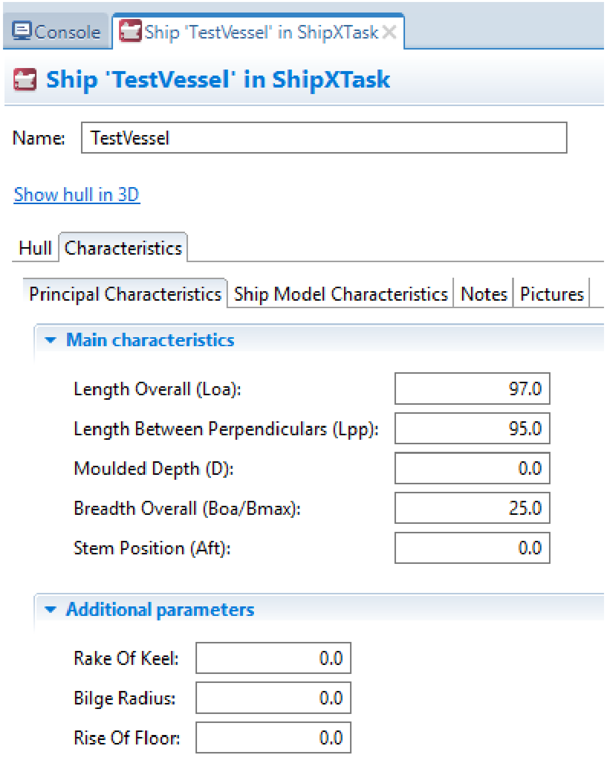
- Right-click the ShipX task
- Select **New** – **Ship**



Since there is no geometry file present, the ship main dimensions are unknown to ShipX. You must therefore enter the following input required to perform a Station Keeping calculation:

In the **Characteristics** tab of the ship:

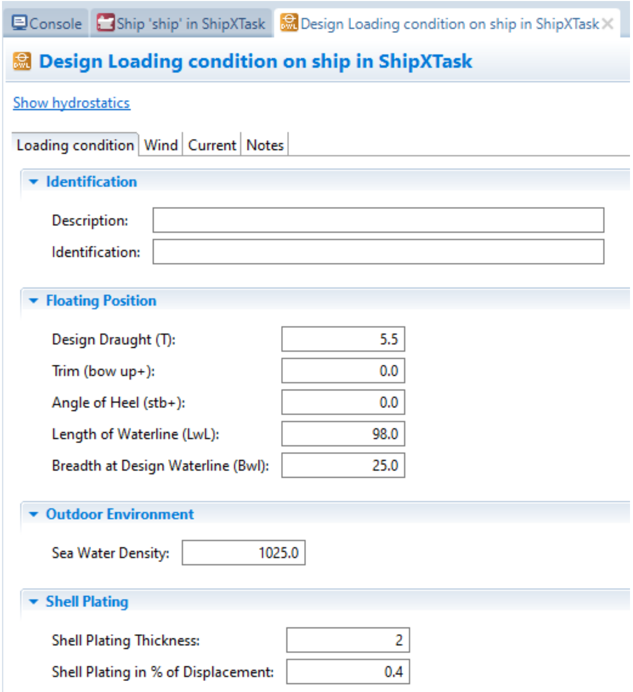
- Length Overall (Loa)
- Length Between Perpendiculars (Lpp)
- Breadth Overall (Boa)

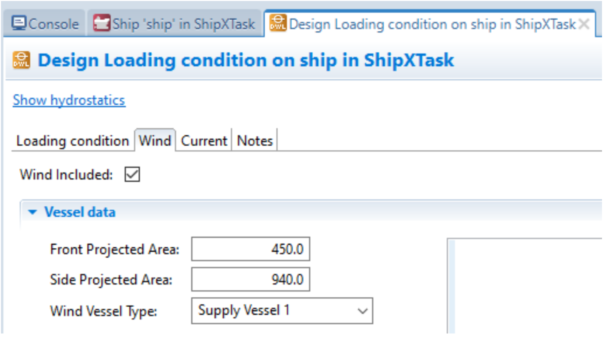


Regardless if you use a geometry file as input or simply use main dimensions as input, you will have to define the loading condition manually. Double click the **Design Loading Condition** in the **Navigator** window and enter the following input:

- Design Draught (T)
- Length of Waterline (Lwl) (right-click and select **Set to editable** first)
- Breadth of Waterline (Bwl) (right-click and select **Set to editable** first)
- Wind Projected Areas
- Wind Ship Type

In the loading condition dialog, you also define the ships current coefficients, but for now we won't be needing those.





Since there is no hull geometry file, the **Hydrostatics** data cannot be calculated.

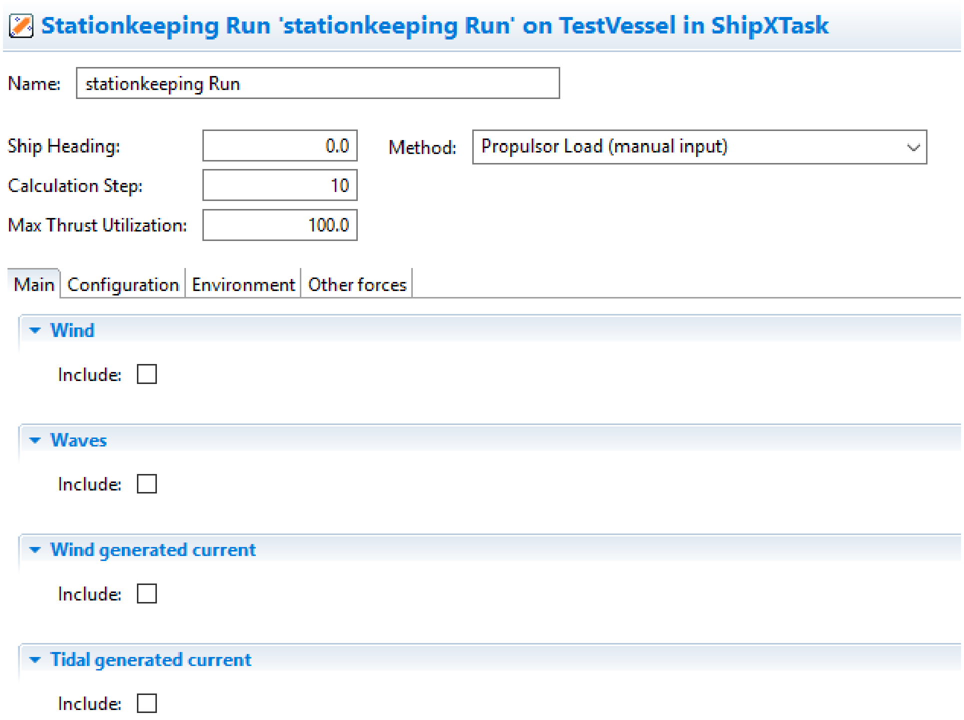
Note that a ship merely defined by main dimensions is not generally acceptable for use in other plug-ins. In Station Keeping plug-in it is sufficient, but in general, a ship defined by a hull geometry file is always preferred.

For more information about working with ShipX, read the ShipX general documentation. Some general tips are outlined in chapter 16.

3.4 Create A Station Keeping Run

We now have a vessel defined and a loading condition defined for the vessel. We can now create a Station Keeping Plug-In run.

- Right-click the **Design Loading Condition**
- Select **New** - **Stationkeeping Run**



3.5 Select A Calculation Method And Set Required Input

There are many different calculation methods. In this example we'll use the default method **Propulsor Load (manual input)**. Add the following input:

- Include **Wind** and let it rotate 360 degrees
- Select the **Environment** tab and use the **Beaufort Wind Scale** to define a wind speed, use the Beaufort scale 8

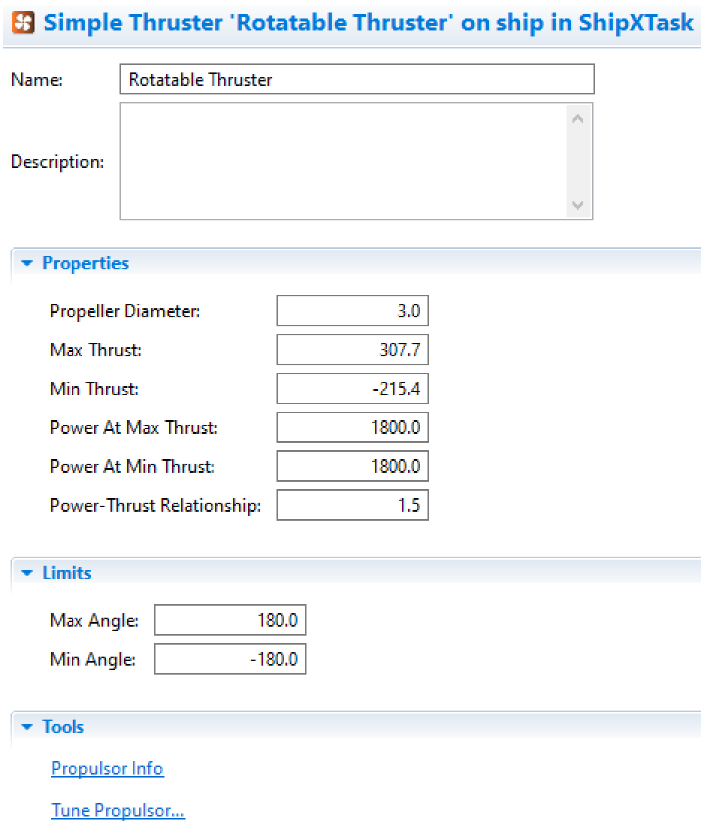
The input remaining now are the **Propulsor Configuration** and the **Power Configuration**. We will deal with these in the next chapters.

3.6 Create Propulsors

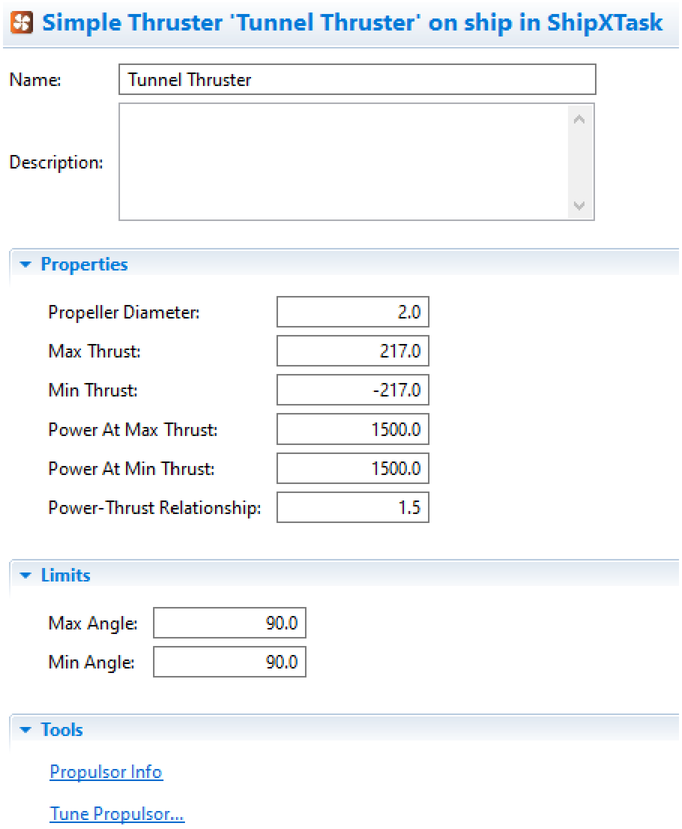
In order to create a Propulsor Configuration we need to define some propulsors. The propulsors are defined on the ship and can thus be used by all plug-in calculations for the ship. In this example we will create two propulsor types and use multiple instances of them when defining the Propulsor Configuration.

In order to create a propulsor:

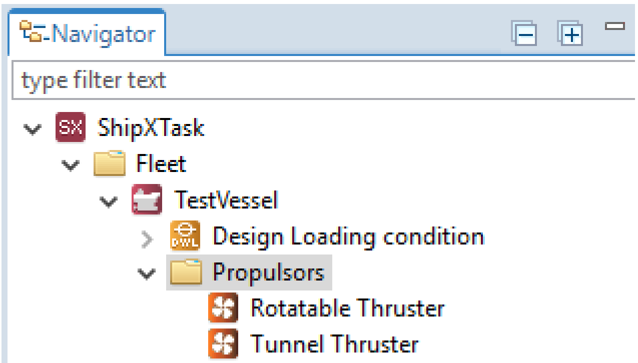
- Right-click the ship in the **Navigator** window
- Select **New** – **Static Propulsor** – **Simple Thruster**
- Give relevant input as in image below (you can try the **Tune Propulsor** button if you like)



We will also create a tunnel thruster based on the same Simple Thruster model by repeating above procedure:



We have now defined two types of propulsors. We can now make multiple instances of them in the Propulsor Configuration in the next chapter.



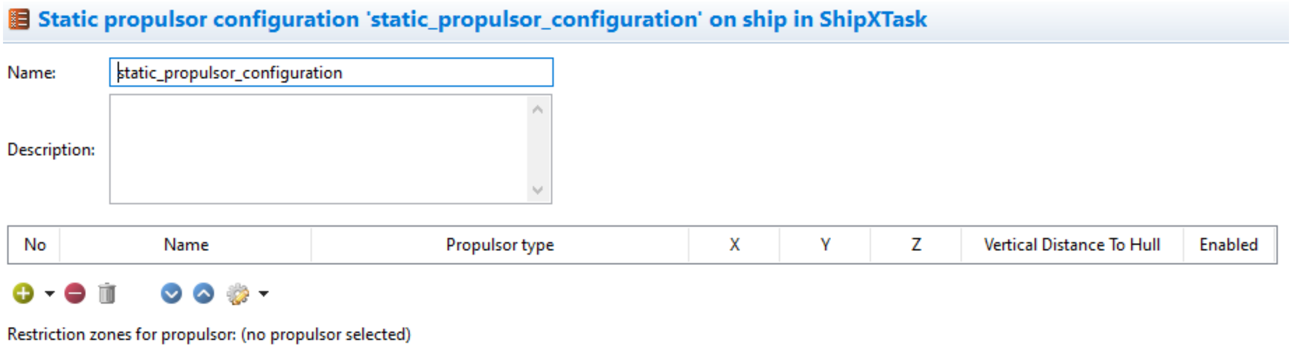
Note that in this example we have only selected static propulsors (they are sufficient since the station keeping calculations are static calculations). The static propulsors cannot be used in dynamic calculations, but dynamic propulsors may support static calculations. We could therefore have defined dynamic propulsors, but in this simple example we will use only static propulsors.

3.7 Create A Propulsor Configuration

A propulsor configuration is a combination of multiple propulsors defining the propulsor configuration of the ship in question. The propulsor configuration is also defined on the ship level, so it can be used by all plug-ins running calculations on this ship.

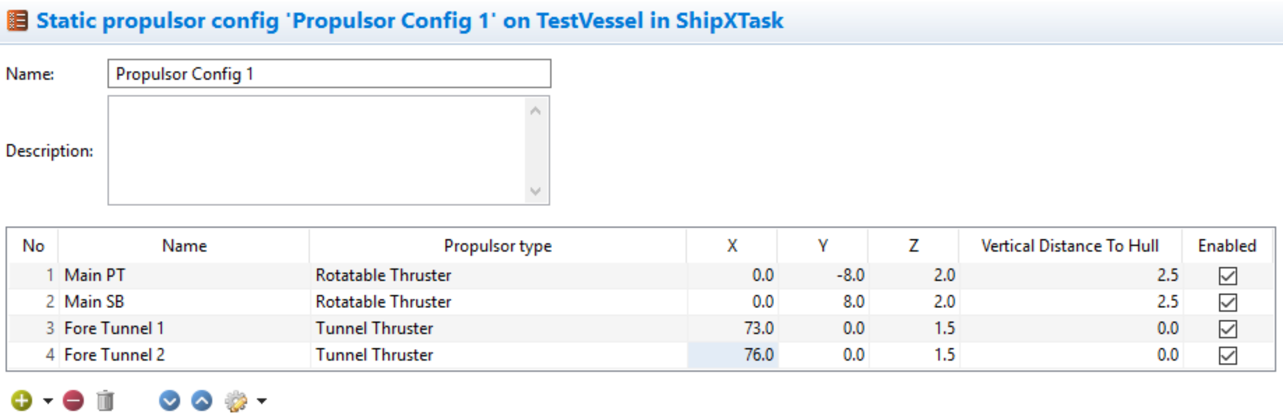
To create a propulsor configuration, do the following:

- Right-click the ship in the **Navigator** window
- Select **New** – **Static propulsor configuration**



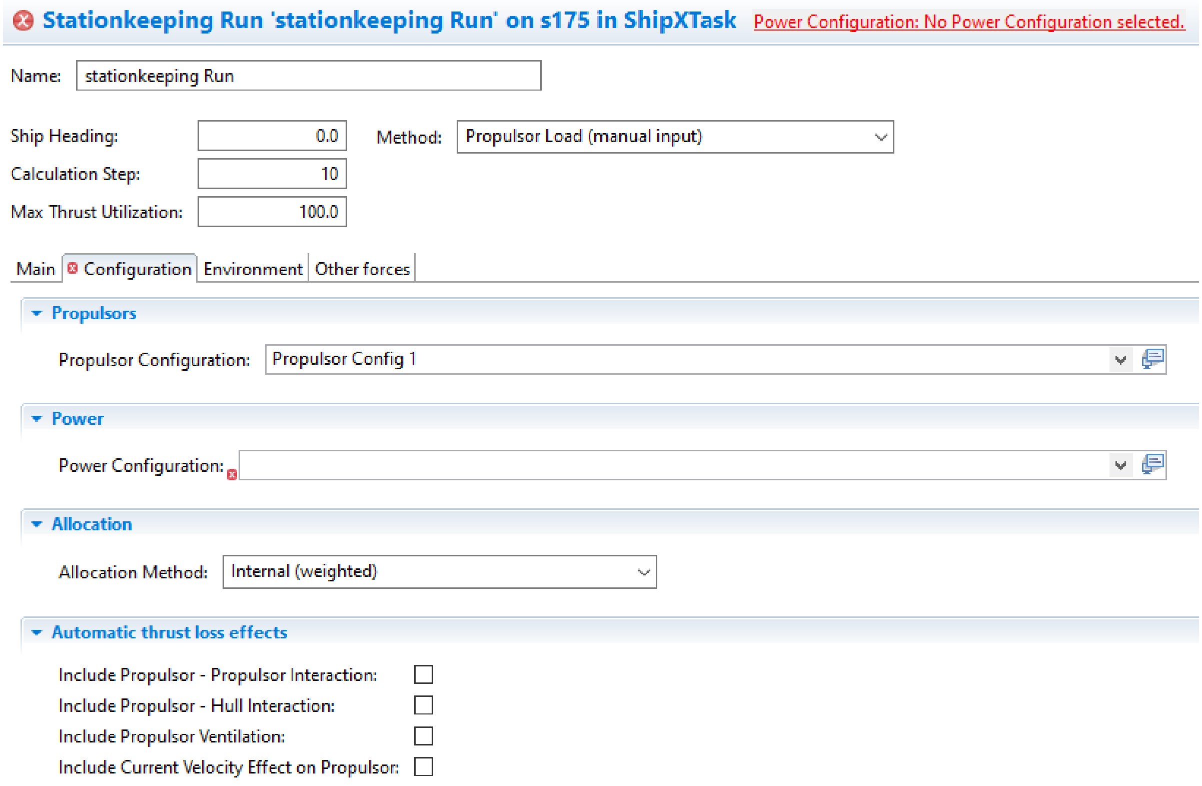
In order to add propulsors to the configuration:

- Click the + sign on the left
- Select a **Propulsor Type**
- Give it a **Name**
- Provide the position data for the propulsor
- Repeat the process for all propulsors you want to define



No	Name	Propulsor type	X	Y	Z	Vertical Distance To Hull	Enabled
1	Main PT	Rotatable Thruster	0.0	-8.0	2.0	2.5	☑
2	Main SB	Rotatable Thruster	0.0	8.0	2.0	2.5	☑
3	Fore Tunnel 1	Tunnel Thruster	73.0	0.0	1.5	0.0	☑
4	Fore Tunnel 2	Tunnel Thruster	76.0	0.0	1.5	0.0	☑

We have now defined the Propulsor Configuration and can now go into the Station Keeping input and set the Propulsor Configuration we want to use in the calculation.



Now we're only missing the Power Configuration (note the red errors). This will be defined in the next chapter.

3.8 Create A Power Configuration

The Power Configuration defines how the propulsors receive their power and may influence the calculation a lot if there are limited power resources. The Power Configuration is defined on the ship level, so other plug-ins might use it as well. The user can model quite complicated power systems, but in this simple example we're not interested in the power system. We will therefore create a dummy power system that will guarantee that the propulsors have the power they need available. For more advanced power system designs, see chapter 18.

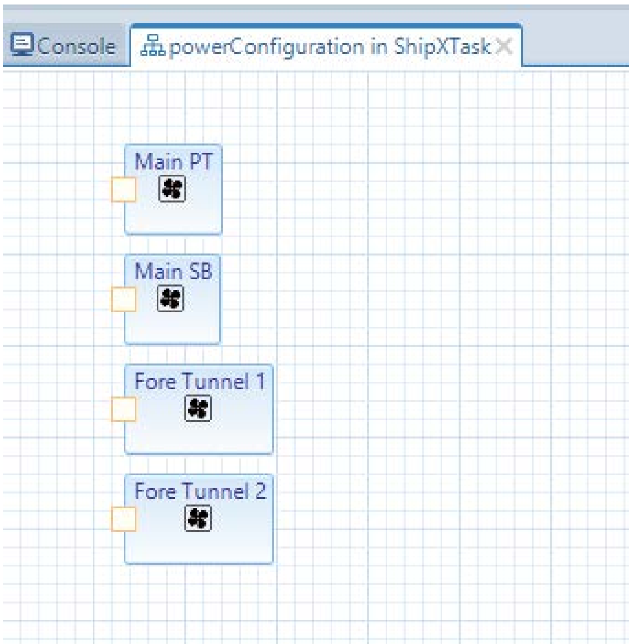
To create a Power Configuration, do the following:

- Right-click the ship in the **Navigator** window
- Select **New** – **Power Configuration**

Observe that the power configuration tab is started in the topmost window of the ShipX workbench. You can drag it down to the bottom window to have it in place with the other input tabs. You can rename the power configuration by right-clicking it in the **Navigator** window and selecting **Rename**.

The power configuration is closely related to the propulsor configuration. When using both in a calculation, they must match, that is, all propulsor in the propulsor configuration must also be defined in the power configuration. When you create a new power configuration it is empty. The easiest way of populating the power configuration with the propulsors you need is this:

- In the **Navigator** window, locate the **Propulsor Configuration** you want to use (the one we defined in chapter 3.7)
- Select it, then drag it on top of the **Power Configuration** window and drop it there



The power configuration consists of four main components:

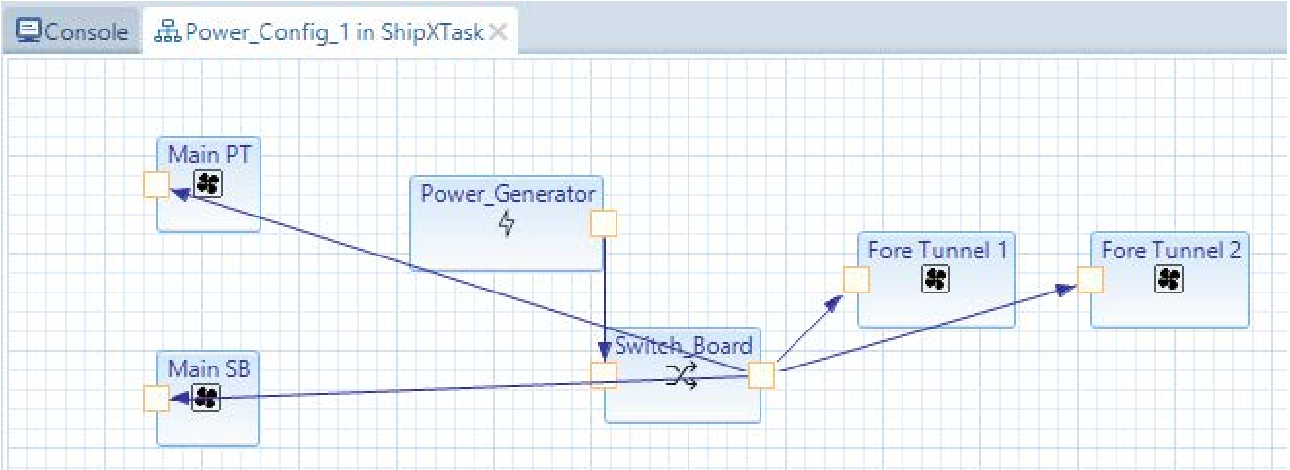
- **Propulsor**: propulsor defined in the propulsor configuration, consumes power from the system
- **Heavy Consumer**: constant power consumer (hotel, winch etc)
- **Power Generator**: engine/unit producing power into the system
- **Switch Board**: connects the power producing units with the power consuming units

All components/boxes can be dragged around to sort them more logically. You can also double-click the boxes to edit relevant input for that component.

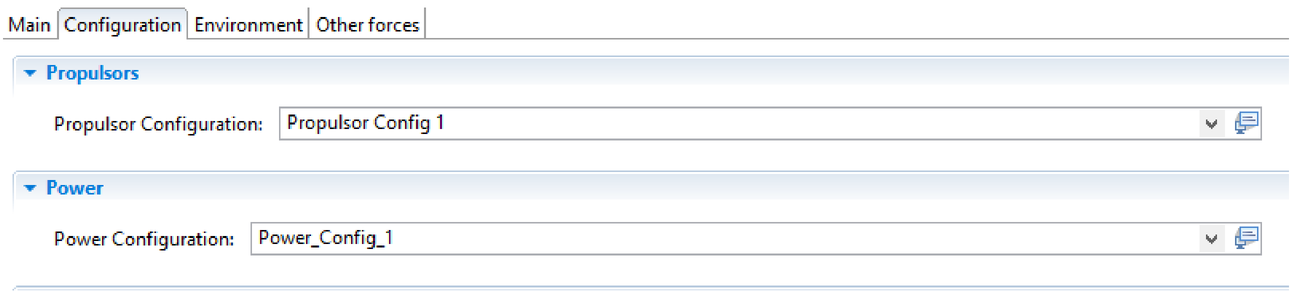
Components have input- (left orange box) and/or output (right orange box) connections. To add a component, simply locate it in the Components folder on the right side of the screen and drag it into the power configuration window. To connect components, simply click and hold the output box of one component and drag to the input box of another component and release the mouse. An arrow will be created indicating the connection and the power direction flow.

We will now make (as mentioned earlier) a simple power configuration to assure all propulsors have sufficient power at all times:

- Drag one **Power Generator** into the power configuration
- Drag one **Switch Board** into the power configuration
- Double-click the **Power Generator** and set the **Power** to 10000 kW (exceeds the sum of all the propulsors connected to it).
- Do the same for the **Power Limit** on the **Switch Board**
- Connect the **Power Generator** to the **Switch Board**
- Connect the **Switch Board** to all the **Propulsors**



We have now defined the Power Configuration and can now go into the Station Keeping input and set the Power Configuration we want to use in the calculation.



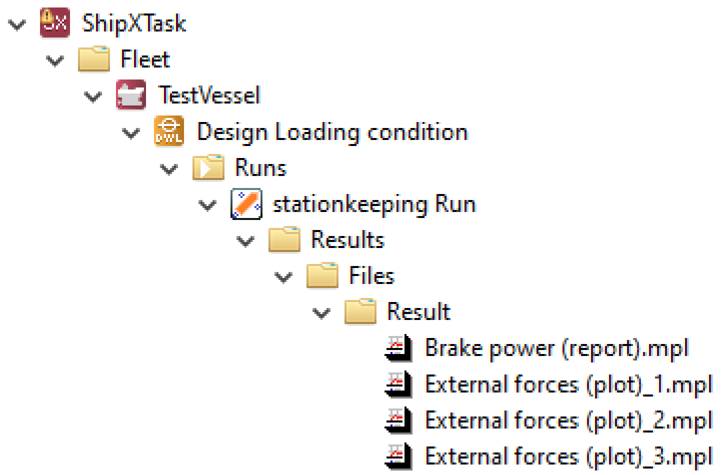
3.9 Run The Calculation

We have now defined all our input for a simple setup and are ready to run the calculation. This is a good time to save the project. The project will auto-save (if you have this feature enabled, see chapter 16.4), but making sure all our work is saved is reassuring, so click the **Save** button to save the project.

To run a calculation, do the following:

- In the **Navigator** window locate the Station Keeping run
- Right-click it and select **Run**

During the calculation the progress can be viewed in the **Progress** tab in the bottom left window.
When the calculation has completed there will be a **Results** - **Files** - **Result** folder under the Station Keeping run containing all results. You may double-click any results to open the plot program.
When you run multiple calculations (like f.ex DPCAP calculations) all results from every calculation will be stored in sub-folders in the Result folder. You can right-click the **Result** folder and click **Explore** to open a file manager into that folder location.



Following the guide in this chapter you now have a basic grasp on how the Station Keeping Plug-In works. In the following chapters we will discuss the software more in-depth.

4 USER INTERFACE

The station keeping dialog window consists of a various amount of input (depending on which calculation mode you have selected. The input is logically divided into groups.

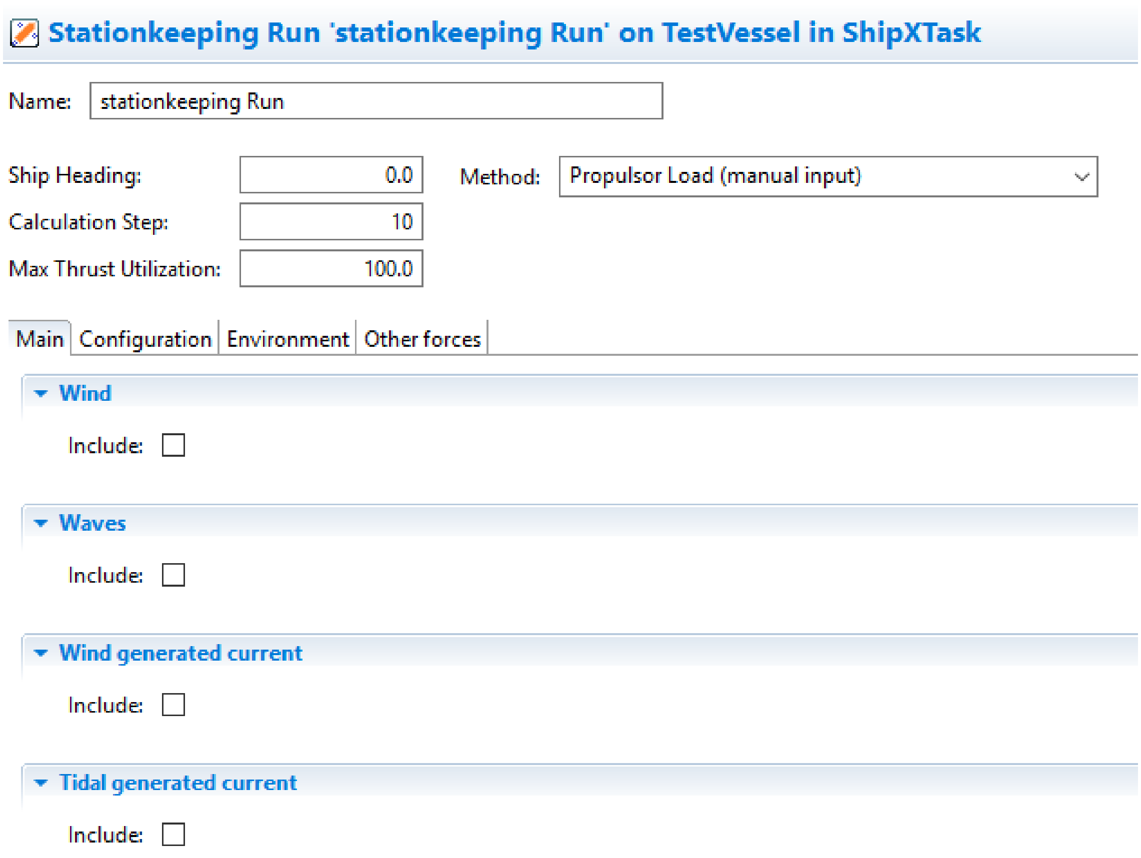


Figure 3, The Main Dialog

In the top part of the plug-in window you define a name for the calculation, define calculation type and other main parameters for the calculation. These inputs influence directly on the amount of output data that are presented in different plots and reports later. You can define the environmental effects to be included in the calculations. When selecting an environmental effect the appropriate input will be enabled in the **Environment** section.

In the **Main** section you can easily switch on/off which environmental effects you want to include in the calculation. The wave direction and the current directions can be set to rotate with the wind (optionally with an offset to the wind direction) or be directionally fixed. Some of the input may be disabled depending on what calculation type and environmental input is selected.

In the **Configuration** section you define your propulsor configuration, your power configuration, the allocation method to use and any automatic thrust loss effects to include in the calculations.

The **Environment** section allows you to specify the environment to use. This information will vary significantly depending on the calculation method selected. The user can select to set input manually or to set the input according to the Beaufort wind scale (wind and wave input only). When setting the input according to the Beaufort scale the data used are gathered from http://en.wikipedia.org/wiki/Beaufort_scale. There are implemented three databases for environmental coefficients. Since the databases are based on different sources the number and types of ships found

therein may vary. You can always select 'Manual Input' and enter, or import, the coefficients of your desire. The file format for import and export is defined in chapter 17.
Note that the wave drift coefficient data is converted to the coordinate system used by this plug-in (as described in chapter 2.2) for presentation both in the plug-in and in the reports/plots.

In the **Other Forces** section you can specify any other external effects to be included in the calculations. When selecting an effect the appropriate input will be enabled. These forces are constant forces. This means they act with a constant force in a constant global direction no matter what happens to the environmental effects. The exception is the "Constant Riser Force" which will rotate in the same direction as the current as would be expected for a force generated by the current.

5 CALCULATION METHODS

The following calculation methods are implemented:

5.1 Propulsor Load (Manual Input)

This method is used for assessing propulsor load for a given environmental setting. The method enables the user to manually define the environmental input including constant or variable direction. If any environmental effects are defined to rotate the calculation is performed through 360 degrees with the calculation step selected.

5.2 Capability Study (Manual Input)

This method is used for assessing environmental capability for the given propulsor configuration. The method enables the user to manually define which environmental forces to be used. The input for wind velocity is not enabled for user interaction since it is being iterated automatically to find the solution. The waves (if selected) can act in a constant or variable direction. The wave height and -period can be automatically set to correspond to the wind velocity according to [7] or can be set manually by the user. The current (if selected) can act in a constant or variable direction, and the current velocity can be constant or relative to the wind velocity. The calculation is performed through 360 degrees with the calculation step selected.

5.3 Capability Study (IMCA M 140 Rev. 1 January 2017)

This method is used for assessing environmental capability for the given propulsor configuration. The method is similar to method 2 but some further restrictions are enforced to make the calculations comply with the given standard. Wind, waves and wind-generated current are the environmental forces included. All environmental forces rotate synchronously. The input for wind velocity and wave height and -period are not enabled for user interaction since they are being iterated automatically to find the solution. The wind velocity is being iterated while the wave height and -period are automatically set to correspond to the wind velocity according to [7]. The current velocity will be constant and not influenced by the iteration process. The calculation is performed through 360 degrees with the calculation step selected. Note that the method for calculating the environmental forces is the same as in the other methods. IMCA requires that methods used for calculating environmental forces comply with “acceptable standards”. A few points concerning IMCA standard calculations are worth drawing notice to when user input is concerned:

- Wind speed is a 1-minute mean at a height of 10 meters.
- Wind areas for the ship should include the hull, superstructure, shielding effects, derricks, cranes and open truss work.
- For some operations a worst-case failure must be investigated in addition to the standard test scenario.

Currently the following points in the plug-in calculation do not comply with the IMCA directive:

- Wind velocity as function of height above water (neglectable for heights below 15.3 meters).

5.4 Capability Study (ERN)

This method is used for assessing the environmental regularity numbers (ERN) as defined by DNV, see [12]. Wind, waves and wind-generated current are the environmental forces included. All environmental forces rotate synchronously. The input for wind velocity and wave height and -period are not enabled for user interaction since they are being iterated automatically to find the solution. The wind velocity is being iterated while the wave height and -period are automatically set to correspond to the wind velocity according to [12]. The current velocity will be constant and not influenced by the iteration process. The calculation is performed through 360 degrees with the calculation step selected. Note that the method for calculating the environmental forces is the same as in the other methods. DNV requires that methods used for calculating environmental forces are “recognized methods”, which our methods are. A few points concerning ERN calculations are worth drawing notice to when user input is concerned:

- A valid ERN calculation is based on a controller mode comparable to the on-board DP controller. Make sure that the on-board DP controller is able to allocate the solutions found when calculating using this plug-in.
- The calculations shall be based on available power. Since not all propulsors output power the user must investigate the solutions founds to verify that the required power is available.
- For the first ERN value all rudders must be locked. The rudders may be used for the other three ERN values if that control mode is included in the on-board DP controller.
- Wind speed is a 1 minute mean.
- The wind speed to wave height relationship is defined in [12] table A1. It is different than the one defined in [7] which is otherwise, by default, used in the capability study calculations.
- The wave period is given as T_z in [12], but this plug-in uses T_p as wave period input so all T_z values have been converted to T_p values using the formula given in section B104 (page 47) in [12].
- For monohull ships the most critical environmental direction is defined (by [12]) to be 90/270 degrees, regardless of the actual worst direction. Make sure the "Use transverse data for evaluating ERN" is selected when calculating for monohull ships. No such restrictions are imposed when calculating for multihull ships.

The ERN number consists of four values, ERN (a, b, c, d). The four values, represented by a, b, c and d, must be found through calculations. The meaning of the values is:

a. Represents the optimal use of the propulsors.
b. Represents minimum effect of single propulsor failure, i.e. the ship capability when removing the least significant propulsor.
c. Represents maximum effect of single propulsor failure, i.e. the ship capability when removing the most significant propulsor.
d. Represents the worst single failure mode, i.e. the worst possible single failure scenario on the ship. Typically, this will be the loss of a switch board with the connected propulsors.

Based on correspondence with DNV a set of guidelines has been defined. All ERN calculations should follow these guidelines:

- All rudders should be locked at zero-degree angle when calculating the first ERN value. The main propellers will be active, but the rudder angles will be locked.
- When inspecting single failure modes to find the last three ERN values all rudders are free to move, and the main propellers are active.
- Conventional propeller/rudder solutions are not considered as “propulsors” and are thus not to be considered when checking the second and third ERN failure scenarios. This means that only tunnel thrusters, pods, azimuths etc are considered when inspecting single failure modes to find the second and third ERN values. The propeller/rudder solutions are always active when checking for the second and third ERN values. When calculating the fourth ERN value the propeller/rudder will fail if it is connected to a switch board which fails.
- Due to locking of the rudder angles when calculating the first ERN value, the second, third and fourth ERN value might be larger than the first ERN value. If either of the second, third or fourth ERN value is larger than the first ERN value the respective value will be set to equal the first value. The second, third and fourth ERN values are not allowed to be larger than the first ERN value.

By default, the environmental calculation range when performing ERN calculations is the same as in the previously described capability methods. In many cases the user is only interested in finding what ERN number the ship will get, he is not interested in finding the actual environmental limits of the design. The user may therefore select to only calculate for environmental scenarios in the range 0-99 ERN. If the ship is capable of keeping station at an ERN 99 environmental scenario, the calculation does not go on to find the real maximum for that case. This may be beneficial when calculating to find for example power consumption in an ERN scenario.

There are several ways of calculating the ERN number as described in the following chapters.

5.4.1 Manual ERN Calculation

Before proceeding read the guidelines from DNV in chapter 5.4 carefully, there are many pitfalls in a manual ERN calculation.
In order to calculate the value represented by 'a' you must select the method for ERN calculation, select manual calculation mode and enable all propulsor units that is to be considered during the calculations, give all necessary input and run a calculation. A report giving the calculated ERN value for each heading is given in the plot/reports dialog. The lowest ERN value is given in the header as 'ERN candidate value'. This is now your 'a' value.
Continuing with calculating the numbers represented by 'b' and 'c' you will have to loop through all relevant propulsors, disabling one propulsor (and one propulsor only) at a time and performing a calculation. Save all results. The 'ERN candidate value' for each run is the possible candidate for the 'b' and 'c' values. The highest 'ERN candidate value' of these runs is the 'b' value; the lowest 'ERN candidate value' of these runs is the 'c' value.
Finally, you must find the 'd' value. To do this you must identify which propulsors are connected to which switch boards. You then loop through all switch boards, disabling the propulsors connected to that switch

board. Note that if a propulsor is connected to multiple switch boards it must not be disabled when disabling one of the switch boards. When all switch board scenarios are calculated the worst switch board failure will give you the 'd' value.

5.4.2 Automatic ERN Calculation

This calculation mode automates the process described in chapter 5.4.1. Select the method for ERN calculation, select automatic calculation mode and enable all propulsor units that is to be considered during the calculations, give all necessary input and run the calculation. The software will now automatically run the required calculations (all propulsor units on, then single propulsor off, looping through all propulsors to consider, finishing with looping through all switch board failures with corresponding multiple propulsor failures). The reports and plots from the different calculations will be automatically saved in a sub-folder for each run. The name of the sub-folder will indicate what type of calculation it contains (all propulsors on, one of the propulsors off or one of the switch boards off). The sub-folders will be stored (as all reports and files) in the Results folder under the run in question (right click the run Result folder and select Explore). Summary reports will be presented at the end of the calculation, giving the complete ERN number directly.

Note that DNV interprets their regulations such that you are only to consider the transverse results when evaluating the ERN numbers, even when the transverse case is not the most critical one. There is therefore a user option to do just this when running the automatic ERN calculation.

Although the use of the automatic calculation method is very convenient and fast the results from all calculations (stored in the resulting sub-folders) should be inspected to verify the results.

5.4.3 Automatic ERN Calculation, Transverse Environment Only

This calculation mode works just like the process described in chapter 5.4.2 with one exception. When running this calculation only transverse environmental directions are considered, as opposed to chapter 5.4.1 and chapter 5.4.2 where the environmental effects are rotated 360 degrees.

5.4.4 Automatic ERN Calculation, Side Environment Only, Simplified 2010

The environmental force models in this calculation scenario are highly simplified compared to a calculation with the full environmental models. This means that the required input is reduced, enabling the user to quickly assess the capabilities of the ship, but it also means that results produced with this calculation method are more questionable compared to a calculation with the full environmental models. When running this calculation mode only transverse environmental directions are considered. *Use this calculation mode with caution and at your own risk.*

5.4.5 Automatic ERN Calculation, Side Environment Only, Simplified 2014

The environmental force models in this calculation scenario are simplified compared to a calculation with the full environmental models. The calculation models are as outlined in Appendix C in [12]. When running this calculation mode only transverse environmental directions are considered as per [12]. There are special propulsor models available for this calculation mode, see chapter 6.1.6.

5.4.6 Use Limited ERN Calculation Range

When performing an automatic ERN calculation, each scenario (loss of propulsor or switch board) will produce an ERN candidate number. There will therefore be multiple ERN candidates to consider when summarizing the calculations into the four ERN numbers. Several of these ERN candidates might be the maximum of 99 (although the ship can be able to keep position for more than the environmental effects representing an ERN candidate value of 99, the maximum ERN candidate value is limited to 99). It may therefore not be practical to use the ERN candidate number as the criteria for, for example, the least significant thruster loss. The default method of selecting the four ERN numbers is therefore to use the withstandable wind velocity for each scenario as the comparison criteria. In this way multiple ERN candidates with 99 values can be sorted based on withstandable wind velocity. When performing a standard automatic ERN calculation this works well.

There is an option in the user interface to only calculate with a limited ERN calculation range. What happens if you switch on this option is that the calculation for a given scenario finishes (and moves on to the next scenario) if the ship is able to keep position for an ERN candidate number of 99. For some ships there may be a lot of scenarios that end up with an ERN candidate of 99. The question is then; how do we rate these results? Sorting them by withstandable wind velocity is not going to work, as they all have the same withstandable wind velocity, as the calculations have not been performed for higher wind velocities than that corresponding to ERN candidate 99. The procedure when rating the results is then as follows:

1. First the results are rated according to their ERN candidate value
2. If two or more scenarios have the same ERN candidate value those scenarios (and only those) will be rated according to the total consumed propulsor power, where of course less consumed power is better.

The resulting rating for automatic ERN calculations with limited calculation range will therefore either be simply by withstandable wind velocity or with a combination of withstandable wind velocity and consumed power. The calculation log will indicate which of the two rating methods are used for the rating.

5.4.7 Use Transverse Data Only For Evaluating ERN

When performing an automatic ERN calculation, the calculations should follow the ERN specification outlined in [12]. This specification is somewhat ambiguous as it states that the calculations are to reflect the worst-case situation, but then goes on to say that for monohull ships the beam environmental direction is to be used, regardless of which direction is worst. This is where the "Use transverse data only for evaluating ERN" option comes in:

- If this option is <u>deselected</u> the ERN candidate from a calculation will be the lowest ERN value for the given calculation regardless of from which environmental direction the value was taken.

- If the option is <u>selected</u> the ERN candidate will be the lowest value from either beam environmental direction, that is, the lowest ERN value from either 90 degrees or 270 degrees environmental direction.
  Note that you may find that the ERN candidate value is lower for other environmental directions than the beam direction. Regardless, the value from the beam environmental direction will be used in the reports.

5.5 Capability Study (SKP, ABS April 2020)

This method is used for assessing the station keeping performance (SKP) as defined by ABS, see [11]. The method is similar to the ERN method by DNV, but some further restrictions are enforced to make the calculations comply with the given standard. Wind, waves and wind-generated current are the environmental forces included. All environmental forces rotate synchronously. The input for wind velocity and wave height and -period are not enabled for user interaction since they are being iterated automatically to find the solution. The wind velocity is being iterated while the wave height and -period are automatically set to correspond to the wind velocity according to the user selected wind-wave relation. The current velocity will be constant but user definable. The calculation is performed through 360 degrees with the calculation step selected. Note that the method for calculating the environmental forces is the same as in the other methods. The user electable wind-wave relation decides the range of environment to be investigated and the resulting probability of non-exceedance.

The SKP number consists of six values, SKP (a, b, c, d, e, f). The first four values, represented by 'a', 'b', 'c' and 'd', must be found through calculations. The last two values refer to the input used. The meaning of the values is:

a. Represents the optimal use of the propulsors, i.e. the probability that the ship can remain on station at the selected site 'f' and current speed 'e' with all thrusters in normal operation conditions.
b. Represents minimum effect of single propulsor failure, i.e. the probability that the ship can remain on station at the selected site 'f' and current speed 'e' with failure of the minimum effect single thruster.
c. Represents maximum effect of single propulsor failure, i.e. the probability that the ship can remain on station at the selected site 'f' and current speed 'e' with failure of the maximum effect single thruster.
d. Represents the worst single failure mode, i.e. the probability that the ship can remain on station at the selected site 'f' and current speed 'e' with failure of the worst-case failure condition. Typically, this will be the loss of a switch board with the connected propulsors.
e. The current speed (in knots) used throughout the calculations.
f. The environment location (i.e. wave spectrum and wind-wave relation) used throughout the calculations. If JONSWAP wave spectrum and IMCA wind-wave relation is selected, the location will automatically be set as North Sea. For any other combination, the location will be set as Unknown. The user can edit the location manually in the report.

A few points concerning SKP calculations are worth drawing notice to when user input is concerned:

- A valid SKP calculation is based on a controller mode comparable to the on-board DP controller. Make sure that the on-board DP controller is able to allocate the solutions found when calculating using this plug-in.
- The calculations shall be based on available power. Since not all propulsors output power the user must investigate the solutions founds to verify that the required power is available.

- The regulations in [11] do not allow a single rudder to be used as a steering device while keeping station. When running with single rudder design, the single rudder will be locked.
- Wind speed is a 1-minute mean at 10 meters above the water surface.
- The SKP numbers a-d are based on the <u>average</u> results from all environmental directions calculated using equal probability for all environmental directions. The ship may therefore in certain environmental directions perform worse than the SKP number for that environmental condition would imply.
- The current interaction on the propulsors must be enabled, see the Tools-Options-Station Keeping menu in ShipX to enable this feature. Note that not all propulsors take notice of this option, i.e. they are not fully 4 quadrant propulsors. To check if the propulsors you have selected implement the current effect click the Info button on the propulsor to view the propulsor summary page.

As always with rules/regulations/guides there are grey areas which do not seem to have been defined well enough. Some interpretations and assumptions must therefore be made in these cases:

- When running with single rudder design the rudder will be locked.
- When running with twin rudder design and one of the rudders fail (when calculating 'b', 'c' or 'd') the single remaining rudder will be locked.
- The resulting number ('a' through 'd') will be an integer.

Note that although the SKP calculation is very similar to the ERN calculation (with respect to calculations performed) the SKP guide [11] specifically demands that propulsor losses are to be included in the calculations. Because of this the calculation time for the SKP calculations may be <u>significantly longer</u> than for a similar ERN calculation.

5.5.1 Manual SKP Calculation

Before proceeding read the guidelines from ABS in chapter 5.5 and [11] carefully, there are many pitfalls in a manual SKP calculation.
To calculate the value represented by 'a' you must select the method for SKP calculation, select manual calculation mode and enable all propulsor units that is to be considered during the calculations, give all necessary input and run a calculation. A report giving the calculated SKP value in the header as an 'SKP candidate value' is generated. This is now your 'a' value.
Continuing with calculating the numbers represented by 'b' and 'c' you will have to loop through all propulsors, disabling one propulsor (and one propulsor only) at a time and performing a calculation. Save all results. The 'SKP candidate value' for each run is the possible candidate for the 'b' and 'c' values. The highest 'SKP candidate value' of these runs is the 'b' value; the lowest 'SKP candidate value' of these runs is the 'c' value.
Finally, you must find the 'd' value. To do this you must identify which propulsors are connected to which switch boards. You then loop through all switch boards, disabling the propulsors connected to that switch board. Note that if a propulsor is connected to multiple switch boards it must not be disabled when disabling one of the switch boards. When all switch board scenarios are calculated the worst switch board failure will give you the 'd' value.
The 'e' value is simply the current velocity (in knots) you used in the calculations.
The 'f' value describes the location you calculated for.

5.5.2 Automatic SKP Calculation

This calculation mode automates the process described in chapter 5.5.1. Select the method for SKP calculation, enable all propulsor units that is to be considered during the calculations, give all necessary input and run the calculation. The software will now automatically run the required calculations (all propulsor units on, then single propulsor off, looping through all propulsors to consider, finishing with looping through all switch board failures with corresponding multiple propulsor failures). The reports and plots from the different calculations will be automatically saved in a sub-folder for each run. The name of the sub-folder will indicate what type of calculation it contains (all propulsors on, one of the propulsors off or one of the switch boards off). The sub-folders will be stored (as all reports and files) in the Results folder under the run in question (right click the run and select Explore). Summary reports will be presented at the end of the calculation, giving the complete SKP number directly.

Although the use of the automatic calculation method is very convenient and fast the results from all calculations (stored in the resulting sub-folders) should be inspected to verify the results.

5.6 Capability Study (DNVGL-ST-0111 July 2016)

This method is used for assessing the station keeping performance as defined by DNVGL, see [13]. Wind, waves and wind-generated current are the environmental forces included. All environmental forces rotate synchronously. The wind velocity is being iterated while the wave height and -period and the current velocity are automatically set to correspond to the wind velocity according to [13]. All propulsors must be of the type outlined in [13]. The calculation is performed through 360 degrees with the calculation step selected by the user (must be 10 degrees or less). The regulations [13] includes forbidden zones and thruster losses. Note that the method for calculating the environmental forces is also specifically outlined in [13].

The standard outlines three levels of DP Capability calculations. Currently only level 1 is supported by this software.

The DP Capability number consists of five parameters, DP capability-LX (A, B, C, D). The four values 'A', 'B', 'C' and 'D' must be found through calculations. The meaning of the values is:

- X The DP Capability level
- A DP Capability number in intact condition with environment direction ±30 degrees on the bow.
- B DP Capability number in intact condition with environment direction 360 degrees.
- C DP Capability number in worst case single failure condition with environment direction ±30 degrees on the bow.
- D DP Capability number in worst case single failure condition with environment direction 360 degrees.

The DP Capability number is the Beaufort number corresponding to the maximum wind velocity (with corresponding waves and current) the ship can sustain. Details are outlined in [13]. There is no interpolation between DP Capability numbers. The user is recommended to thoroughly read the relevant chapters in [13] and go through the example outline in chapter 18.4 in this user manual before performing a DP Capability calculation.

A few points concerning the DP Capability calculations are worth drawing notice to when user input is concerned:

- A valid DP calculation is based on a controller mode comparable to the on-board DP controller. Make sure that the on-board DP controller is able to allocate the solutions found when calculating using this plug-in.
- Forbidden zones on each propulsor (not tunnel thrusters) must be defined by the user. Use the "Automatic" functionality to calculate them automatically.

The environmental forces are calculated according to [13], but when solving the equilibrium of external forces and propulsor forces the external forces will be increased with a "dynamic factor" of 1.25 (for level 1) according to [13].

All reports named "DP Capability…" or "A-xx…" are in the DNVGL coordinate system, hence the directions, angles etc may not correspond with the other Station Keeping reports.

5.7 Capability Study (DNVGL-ST-0111 March 2018)

For Level 1, this method, see [14], is identical to the one outlined in chapter 5.6 with the following exceptions:

- The zone forbidding the propulsor to interact with the skeg is now replaced with a thrust loss model. This thrust loss model will automatically calculate the thrust loss caused by the propulsor-skeg interaction and replaces the requirement for defining a forbidden zone towards the skeg.

When a thruster is flushing a dead thruster, there is a loss region which it may be beneficial to stay away from. Use the "Automatic" mode to calculate the forbidden zones automatically and thus take this region into consideration.

Also make sure that you leave a 10% power reserve when defining the power system to allow for consumers not part of the thruster system (hotel etc).

For Level 2 the requirements to environment forces and propulsor models are relaxed. You can still opt to run with the DNVGL environment, but you can now also select to use a different environment model.

For Level 2 Site you can add other external forces. The environment conditions are even more relaxed, such as user defined wind-wave relation and non-co-linear environment.

For Level 2 and Level 2 Site, the propulsor loss models are the same as when running Level 1.

All reports named "DP Capability…" or "A-xx…" are in the DNVGL coordinate system, hence the directions, angles etc may not correspond with the other Station Keeping reports.

5.8 Capability Study (DNV-ST-0111 December 2021)

This method, see [15], is identical to the one outlined in chapter 5.7 with the following exceptions:

- The thruster ventilation method has been changed.

6 PROPULSORS

Individual propulsors are defined on the ship level. The propulsion system is defined as propulsor configuration(s), also on the ship level. This means that both propulsors and propulsor configurations can be defined once and used in multiple plug-ins.

There are a number of propulsors included in ShipX:

- ERN propulsors
- DPCAP propulsors
- Simple Thruster
- Generic Pod
- Generic Tunnel Thruster
- Propeller and Rudder
- Propeller and Flap Rudder

In addition to the abovementioned propulsors, the user can implement their own propulsors and use them in the calculations, see chapter 0 and/or chapter 13 for details.

The Station Keeping plug-in is a static module, so it only requires static propulsor models. The first three propulsor categories above are static propulsors. The other propulsors in the above list are both static and dynamic propulsors.

In order to create a static propulsor, perform the following steps:

- Right-click the ship in the **Navigator** window
- Select **New** – **Static Propulsor** – **<Propulsor type>**
- Give relevant input

If you want to create a dynamic propulsor, you need to use them in their static mode:

- Right-click the ship in the **Navigator** window
- Select **New** – **Dynamic Propulsor** – **<Propulsor type>**
- Check the **Static** input box to enable the required static input for the selected propulsor type
- Give relevant input

Note that the propulsors you define are the different propulsors types. If you, for example, have three identical tunnel thrusters (with different positions on the hull, obviously), you only need to define one tunnel thruster propulsor and use that propulsor three times when defining the propulsor configuration.

6.1 Built-In Propulsor Input Description

The following subchapters describe the static input for the propulsors implemented by SINTEF Ocean. Note that a positive pod- or rudder angle will cause the ship to turn to port.

Some propulsors may be marked with a '(deprecated)' notation in the propulsor name. This means that the propulsor in question has been replaced by a new and improved model. **It is highly recommended to use the new propulsor model in your calculations and discontinue use of the deprecated model.** The deprecated model is kept in the software for an unspecified duration for backwards compatibility with older projects.

6.1.1 'Pod' Input Description

This propulsor models a generic pod unit. This is a combined propulsor, that is, it supports both static and dynamic calculations. You will find this propulsor in the dynamic section of the propulsor menu. Remember to check the **Static** checkbox to define all required input. The input for this propulsor is described in the following.

Propeller Diameter: Defines the propeller diameter.

Propeller Blade Area Ratio: Defines the propeller blade area ratio.

Constant Propeller Pitch Ratio: Defines the constant propeller pitch ratio used throughout the calculation.

Max Angle: Defines the maximum allowed pod angle.

Min Angle: Defines the minimum allowed pod angle.

Max Propeller Revs: Defines the maximum allowed propeller revolutions.

Min Propeller Revs: Defines the minimum allowed propeller revolutions.

Max Propeller Power: Defines the maximum delivered power at the propeller, P_D, for use in reports. This input does not affect the calculation, it is merely for reporting.

6.1.2 'Tunnel Thruster' Input Description

This propulsor models a generic tunnel thruster unit. This is a combined propulsor, that is, it supports both static and dynamic calculations. You will find this propulsor in the dynamic section of the propulsor menu. Remember to check the **Static** checkbox to define all required input. The input for this propulsor is described in the following.

Propeller Diameter: Defines the propeller diameter.

Constant Propeller Pitch: Defines the constant propeller pitch ratio used throughout the calculation.

Tunnel Length: Defines the length of the tunnel centre through the hull.

Hull Angle (front view): Defines the hull angle at the longitudinal position where the centre of the tunnel thruster is positioned. This is the angle between the horizontal axis and the angle of the hull section ("flare") as seen from the front of the ship.

Hull Angle (top view): Defines the hull angle at the longitudinal position where the centre of the tunnel thruster is positioned. This is the angle between the hull section and the longitudinal axis of the ship as seen from above the ship (birds-eye view).

Max Propeller Revs: Defines the maximum allowed propeller revolutions. The minimum propeller revolutions are assumed to be of the same value with negative sign.

Max Propeller Power: Defines the maximum delivered power at the propeller, P_D, for use in reports. This input does not affect the calculation, it is merely for reporting.

6.1.3 'Simple Thruster' Input Description

This propulsor models a standard thrust vector and is as such not a physical propulsor model. The input for this propulsor is described in the following.

Propeller Diameter: Defines the propeller diameter.

Maximum Angle: Defines the maximum allowed thrust vector angle.

Minimum Angle: Defines the minimum allowed thrust vector angle.

Max Thrust: Defines the maximum thrust available.

Min Thrust: Defines the minimum thrust available.

Power at Max Thrust: Defines the maximum delivered power at the propeller, P_D, corresponding to the maximum thrust available defined above. Whenever the thrust required is different from the maximum thrust the corresponding power will be estimated as a function of the required thrust, see Thrust-Power Relationship below.

Power at Minimum Thrust: As described above, but for the minimum thrust available.

Power-Thrust Relationship: When running with a partly loaded thruster the power consumed is calculated as follows:

$$Power \sim Thrust^{ratio}$$

I.e. the power is proportional to the thrust raised to a ratio equal to the power-thrust relationship. By default, the ratio is set to 1.5.

6.1.4 'Propeller And Rudder' Input Description

This propulsor models a generic rudder propeller unit. This is a combined propulsor, that is, it supports both static and dynamic calculations. You will find this propulsor in the dynamic section of the propulsor menu. Remember to check the **Static** checkbox to define all required input. The input for this propulsor is described in the following.

Propeller Diameter: Defines the propeller diameter. See 'H' in Figure 4.

Propeller Blade Area: Defines the propeller blade area.

Constant Propeller Pitch Ratio: Defines the constant propeller pitch ratio used throughout the calculation.

Propeller Rotation: Defines the direction of rotation for the propeller (as seen from behind). Valid input is either:

- -1 for counter-clockwise rotation
- 1 for clockwise rotation

Rudder Height: Defines the rudder height (span). See 'E' in Figure 4.

Rudder Root Chord: Defines the rudder chord (length) of the rudder at the upper end (closest to the hull). See 'D' in Figure 4.

Rudder Tip Chord: Defines the rudder chord (length) of the rudder at the lower end (farthest away from the hull). See 'F' in Figure 4.

Rudder Horn Area: Defines the projected fixed horn area (if any) for the rudder.

Total Rudder Area: Defines the projected total rudder area for the rudder including any rudder horn area (fixed rudder area).

Distance Propeller to Rudder Stock: Defines the longitudinal distance between the propeller plane and the rudderstock. See 'G' in Figure 4.

Distance Propeller Axis to Rudder Top: Defines the vertical distance from the propeller axis to the top of the rudder. See 'C' in Figure 4.

Initial Gap between Rudder and Headbox: Defines the vertical gap between the rudder and the headbox at zero-degree rudder angle. See 'B' in Figure 4.

Max Angle: Defines the maximum allowed rudder angle.

Min Angle: Defines the minimum allowed rudder angle.

Max Propeller Revs: Defines the maximum allowed propeller revolutions.

Min Propeller Revs: Defines the minimum allowed propeller revolutions.

Max Propeller Power: Defines the maximum delivered power at the propeller, P_D, for use in reports. This input does not affect the calculation, it is merely for reporting.

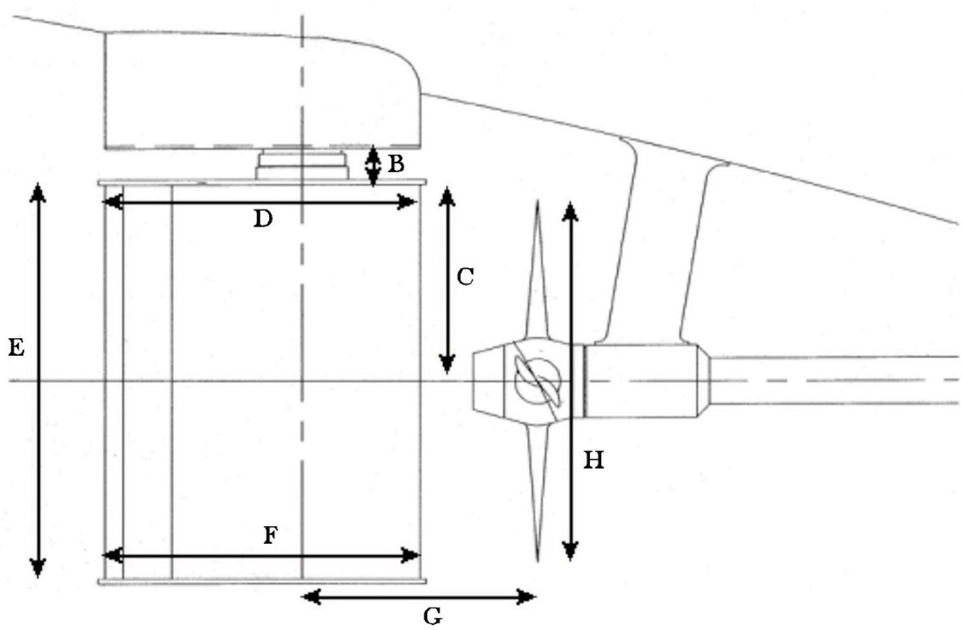


Figure 4, Propeller - Rudder definitions

6.1.5 'Propeller And Flap Rudder' Input Description

The input for this propulsor is identical to the one described in chapter 6.1.4. The only difference between these two propulsors is that the rudder is of a flap type, but no further input is needed. This is a combined propulsor, that is, it supports both static and dynamic calculations. You will find this propulsor in the dynamic section of the propulsor menu. Remember to check the **Static** checkbox to define all required input.

6.1.6 'ERN…' Propulsor Input Description

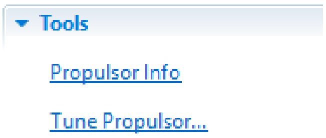
The propulsors named "ERN…." are based on [12]. There are some general propulsors which require all input, and then there are some specialized cases where some of the input has been coded into the module based on the propulsor type. The specialized cases are convenient if you are to calculate for such a propulsor. If not, the generic case, in which you can define the propulsor as you wish, is available. All required input is outlined in detail in [12].
As the thrust loss due to interaction/ventilation/inflow etc is automatically included in these propulsor models (as per [12]), you will not be allowed to run a calculation with the automatic thrust loss feature enabled.

6.1.7 'DPCAP…' Propulsor Input Description

The propulsors named "CPCAP…." are based on [15]. Apart from the difference in model base between [12] and [15], the statements in chapter 6.1.6 also applies here.

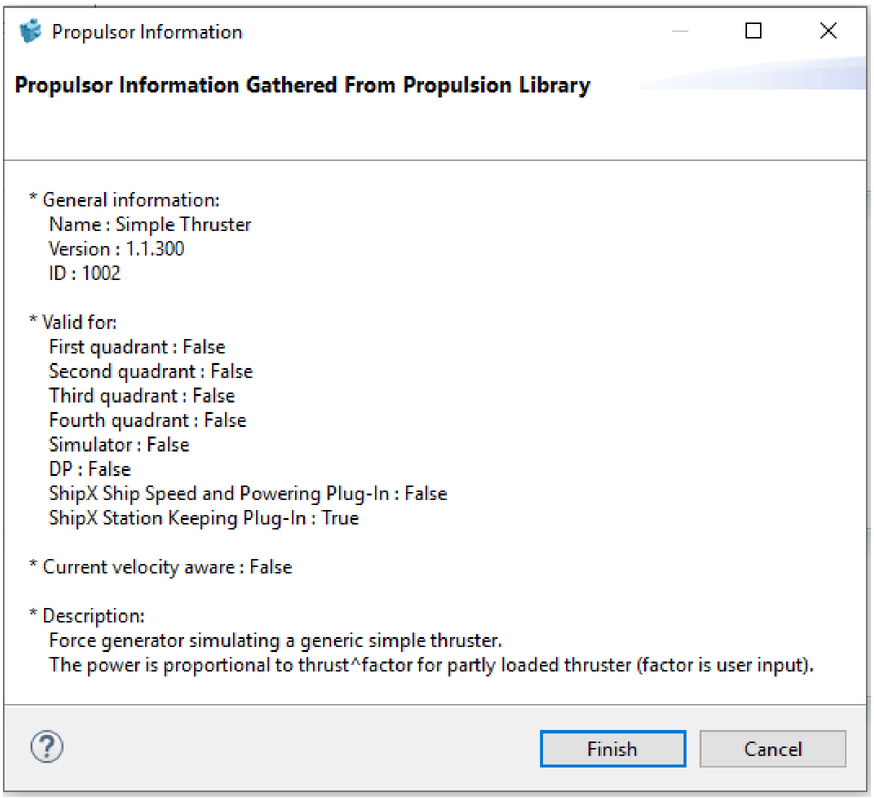
6.2 Propulsor Tools

Depending on which type of propulsor you have defined, there will be some tools available to you in the **Tools** section of the propulsor input dialog:



6.2.1 Propulsor Info

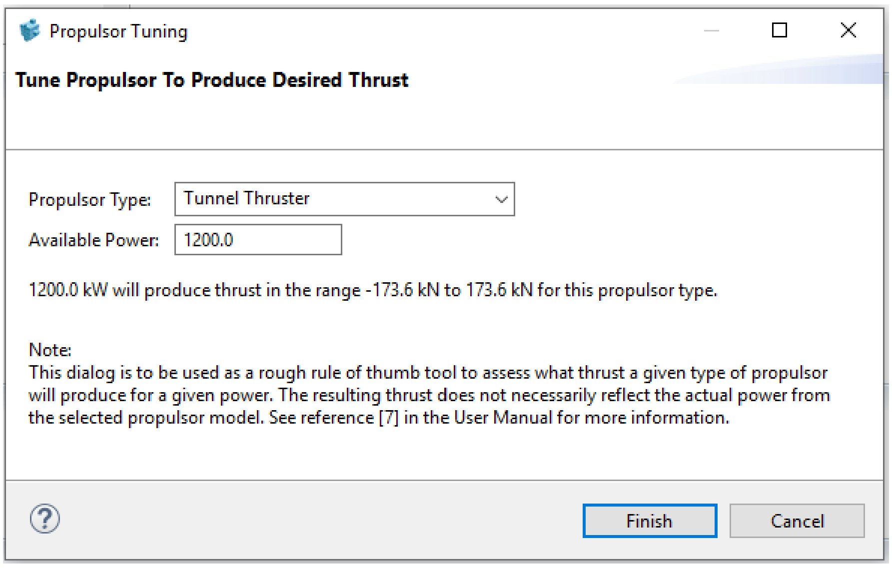
Clicking the propulsor Info link will display a window containing the propulsor info deemed important by the individual implementing the propulsor code.



Among the more important items in this window is the operational range the propulsor is valid for, its version and the propulsor description. It is also important to see if the propulsor supports current velocity, see **Current velocity aware** and chapter 6.3.4.

6.2.2 Tune Propulsor

Some propulsors will allow you to tune the propulsor so that you are sure that the nominal maximum and minimum thrust are to your specifications. The typical workflow is then to define your propulsor input, click the **Tune Propulsor** link, select **Propulsor Type** and **Available Power** in the dialog that appears and click (**Calculate**, if available, and then) **Finish**.



What happens then is that the propulsor you are working on will, for example get a new maximum and minimum revolutions so that the maximum and minimum thrust the propulsor model will deliver matches the data you had in your **Propulsor Tuning** dialog. The propulsor will keep its generic characteristics in your later calculations when partly loaded, but you will be sure that the maximum and minimum values it produces are to your specifications.

6.3 Thrust Restriction- And Thrust Loss Effects

The propulsors may experience interaction effects with the ship hull, with each other and with the free surface (ventilation) causing thrust loss. These effects can be simulated by selecting the appropriate interaction effects. In order to calculate ventilation in waves the ship response must be known. This is defined by a ship response calculation leading to a file of type re1 used for the ventilation calculations. Manual thrust loss effects and forbidden zones can also be defined. The following chapters outline your options.

Note that enforcing thrust loss and/or thrust restriction zone limitations on the allocation algorithm forces it to produce results that may differ significantly from the desired solution. When performing calculations with thrust loss and/or with thrust restriction zones the results found may not be the optimum results obtainable.

6.3.1 Propulsor – Hull Interaction Losses

The propulsor - hull interaction (also called Coanda effect) is caused by a deflection of the propeller race caused by the presence of the hull. The result of this deflection is that the thrust is reduced compared to the case without a hull present. This effect will increase with increasing propeller race distance along the hull, increasing bilge radius, decreasing distance to the hull and decreasing tilt angle. This effect can reduce the propeller thrust by up to approximately 30%.
The waterline shape of the vessel will be used to calculate the distance the propeller race will travel to exit the hull. If no hull geometry file is defined, a simplified box (based on the length and width of the vessel) will be used to define the waterline shape.

6.3.2 Propulsor – Propulsor Interaction Losses

The propulsor - propulsor interaction loss is caused by one propulsor sending its propeller race into another propulsor. Both propulsors might experience a thrust loss caused by this interaction. There are several models of this physical phenomenon implemented in this Plug-In:

- Propulsor in tandem

  In this case the thrust loss of propulsors in tandem is calculated. The propeller race of the front propulsor fans out both vertically and horizontally behind the propulsor. If the hindmost propulsor is positioned inside this propeller race region it will experience a thrust loss. The thrust loss effect will increase with decreasing distance between propulsion units and increasing propeller race exposure on the hindmost propulsion units. This effect can reduce the propeller thrust by up to approximately 60%.

- Main propeller and stern tunnel thruster

  In this case the tunnel thruster in the stern is experiencing thrust loss effects caused by the interaction with the main propellers. This model is only valid for twin screw rudder/propeller main propulsion systems. The thrust loss effect will increase with increasing ratio of main propeller power compared to tunnel thruster power. This effect can reduce the thrust of the tunnel thruster by up to approximately 100%.

- When using the DPCAP thruster interaction models both flushing an active thruster and flushing a dead thruster may cause interaction losses. It is advisable, when running a DPCAP calculation, to include the "avoid flashing dead thrusters" forbidden sectors as this might increase the capability results as opposed to allowing the thruster to operate inside this sector and experience thruster interaction losses.

6.3.3 Ventilation Losses

Thruster ventilation losses occur when the thruster is too close to the surface and starts sucking air, effecting in a less dense medium to operate in (in addition to possible ventilation) and thus reduced thrust. The thrust loss will increase with decreasing distance to the surface, increasing thruster load and increasing thruster vertical motion. This effect can reduce the thrust of the thruster by up to approximately 100%.

6.3.4 Current Velocity Effects

There is an option to include the effect of the current velocity on the propulsors. If this option is selected it means the propulsors will get information about the current velocity and take this velocity into account (if the propulsor is able to do so) when calculating. To check if the propulsor is able to take into account the current velocity, click the "Info" button for the propulsor in question and check the "Current velocity aware" line.
If the "Current velocity effect on propulsors" option is deselected, or the propulsor(s) is unable to use the current velocity information, the propulsor calculations will be performed without current velocity effect.
Given that the current velocity is usually relatively low, the general effect of selecting this feature is:

- The calculation time will be much longer (the current velocity increases the complexity of the calculation and makes it harder to find a good numerical solution)

- The calculated results will be very similar to the ones calculated without including the current velocity, and, in the grand of schemes, well within the inaccuracies of using (for example) generic environmental coefficients

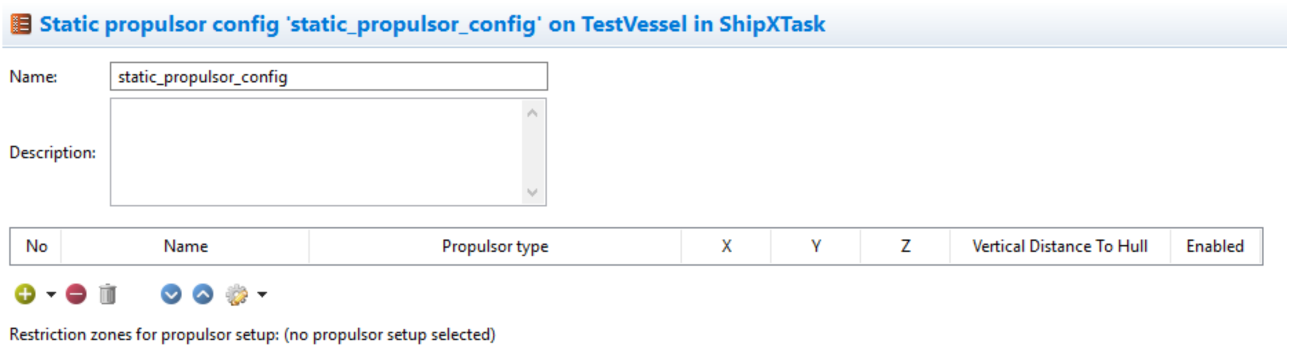
For these reasons, the option to include current velocity in the calculations is by default deselected. At the time of writing, the only calculation that requires this effect to be enabled is the SKP calculations outlined in chapter 5.5.

7 PROPULSOR CONFIGURATION

In the propulsor configuration you define the propulsor setup of your vessel, i.e. you utilize the defined propulsors by positioning them on the vessel and (optionally) define restricted/forbidden operational zones for each propulsor.

To create a propulsor configuration, do the following:

- Right-click the ship in the **Navigator** window
- Select **New** – **Static Propulsor Config**



In order to add propulsors to the configuration:

- Click the + sign on the left
- Select a **Propulsor Type**
- Give it a **Name**
- Provide the position data for the propulsor
- Repeat the process for all propulsors you want to define

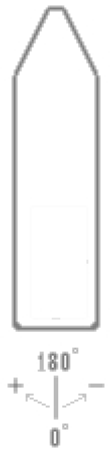
After adding the propulsors to the propulsor configuration you can enable thrust restriction-/forbidden zones if you like.

7.1 Manual Thrust Restriction-/Forbidden Zones

Sometimes the user wants to limit the available thrust in certain propulsor sectors. This can be done by using the thrust restriction zones input. This enables the user to define multiple thrust restriction zones where each have their own sectors, max thrust, thrust loss and dependencies (on any propulsors).

All propulsors (except rudder/propeller units and locked units, like tunnel thrusters) can have restricted-/forbidden zones defined. The propulsor is not allowed to operate in forbidden zones.

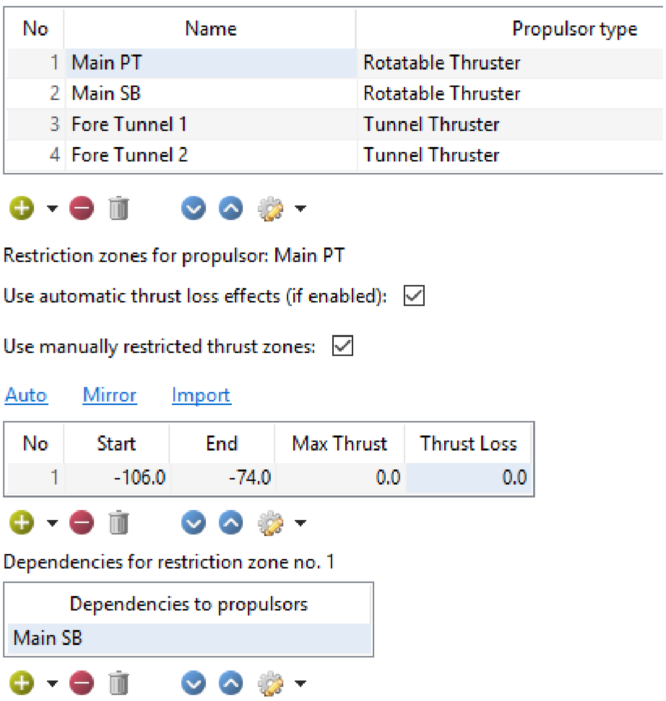
The coordinate system for defining forbidden/restricted zones are as follows:



To define a restricted-/forbidden zone, do the following:

- Click the propulsor name to define the zone for
- Click the **Use manually restricted thrust zones** option
- Click the Forbidden Zones button in order to define the forbidden zones.
- Define your zone using one of the available methods (**Auto**, **Import** or manual)
- Set dependencies for the zone if you want to

As an example, we have defined a forbidden zone for the propulsor "Main PT" to avoid it interacting with the "Main SB" propulsor (note the dependency):



The input in this dialog is used for setting up the thrust loss/restrictions of this propulsor.

Use Automatic Thrust Loss Effects (if enabled): If you have selected to use any of the global thrust loss settings (thruster-thruster, thruster-hull or ventilation) all thrusters will by default be affected by those thruster losses. If you uncheck this input, you tell the program to disregard any such thruster loss effects on this propulsor. *Note that this propulsor will still be able to affect other propulsors (and cause them to have thruster-thruster interaction losses).*

Use Manually Restricted Thrust Zones: By enabling this setting you can define thrust restrictions zones and thus manually setting thrust limitations for a propulsor in given zones. The necessary input for each thrust restriction zone is:

- Start of zone

  Defines the start of the restricted/thrust loss zone. See the coordinate system above for definitions.

- End of zone

  Defines the end of the restricted/thrust loss zone. The end of the zone must be a larger value than the start of the zone.

- Max thrust

  The maximum thrust allowed within this zone. The input is a percentage of the nominal thrust defined by the thruster input. Setting a value here will ensure that the thruster does not exceed the set maximum thrust load.
  Note that if you set a global "Max thrust utilization" the lower of the two values will be the limiting maximum thrust in this sector.

- Thrust loss

  The manual thrust loss enforced on the thruster when operating within this zone. This loss can be added to the automatically calculated losses if the options are selected (see explanation above).

- Dependency

  Defines (optionally) any dependencies for the restricted loss zone. If no dependencies are defined the zone will always be active when the propulsor is enabled. If a dependency is defined the restricted zone will only be active if at least one of the dependencies are active. Multiple dependencies can be defined.
  Example: On a propulsor one can for example define a dependency in a restricted zone to be Propulsor A and Propulsor B. That means that this zone will be active on this propulsor if either Propulsor A or Propulsor B are in use. If neither Propulsor A nor Propulsor B are in use, the zone will not be active in the calculations.

You can define as many thrust restriction zones as you like but remember they must be within the range of the minimum to maximum angle defined for the propulsor.

The thrust restriction zones can overlap. This is very useful as one can define multiple zones depending on thruster scenarios and make each zone dependent on other thrusters. In this way you can define any sector you like given any thruster dependency you like. *If a thruster is operating inside multiple active thrust* *restriction zones simultaneously, the zones will be combined into a larger zone incorporating all the relevant overlapping zones.*

The propulsors generally have an operating range between minimum and maximum angle. Sometimes there may arise needs to limit this operating range further, for example to make sure the propulsor does not operate in a specific range causing interaction with other propulsors or with the hull. In order to limit the operating range in the already defined range, one can define forbidden zones. The way to define forbidden zones is as described above for the thrust restriction zones; the only difference is that you set the max thrust input for the zone to zero.
Note the difference between setting the max thrust to zero and setting the max thrust to "almost zero". If you set the max thrust to zero the zone will be defined as a forbidden zone, and the propulsor will not be allowed to operate in this zone at all. If the max thrust is set to "almost zero" the zone will be perfectly valid for use, but the propulsor will have a very limited thrust in that zone.

If you have defined a thrust restriction zone as a forbidden zone the zone range defines that the propulsor is not allowed to send its propeller race into the forbidden zone.

Click the **Auto** button to start a wizard helping you to define thrust restriction zones based on different standard ways of doing this.
Click the **Mirror** button to mirror all thrust restriction zones for the propulsor in question along the ship centreline.
If you have a predefined set of thrust restriction zones already defined you can import them into the table by clicking the **Import** button below the table. Note that the thruster dependencies are not part of the imported data. The file format for the thrust restriction zones can be found in chapter 17.4.

Note that imposing restrictions as explained above will cause the calculation time to increase significantly, depending on the amount of restrictions imposed. Imposing thrust loss zones may also make the calculated solution less than optimal.

8 POWER CONFIGURATION *

The power configuration enables the user to model an advanced power system and include the power supply limitation in the calculations.

A power configuration consists of four main components:

- **Power Generator**
- **Propulsor**
- **Heavy Consumer**
- **Switch Board**

The functionality of these are as follows:

Power Generator: The power generator is typically an engine/motor producing power into the system. The deliverable power is input. An optional power utilization is also input, where the user may easily reduce the available power without modifying the power input. Each unit can easily be switched on/off by double-clicking the box and selecting/deselecting the include option.

Propulsor: The required propulsor power is calculated and fed into the power distribution automatically. A total efficiency factor can be used as input. Note that this is the total efficiency between the propulsor and the switch board it is connected to. If the propulsor in question also has a, for example, mechanical efficiency as input, the user should be careful not to include the efficiency twice. The total efficiency may therefore be set to 1.0 if the mechanical efficiency is already taken care of by the propulsor input. Manually enabling/disabling a propulsor is done in the propulsor configuration (see chapter 7).

Heavy consumer: The heavy consumers are static power loads to be included in the calculation. They require the consumed power along with an electrical efficiency as input. The heavy consumer can be connected to multiple switch boards or a single power generator. If it is connected to multiple switch boards the power consumed is distributed evenly on these switch boards. Each unit can easily be switched on/off by selecting/deselecting the include input.

Switch Board / connector: This is the glue that connects the power producing units with the power consuming units. Typical connectors may be switch boards, gears etc. The maximum available power (power limit) to flow through the connector along with the electrical efficiency factor are input for each connector. In addition, a power reserved input is available, simply to reduce the available power with a given percentage without editing the installed power. It is mandatory to use the power reserved in for example the DPCAP calculations (see chapter 5.7). Each connector can be defined to take its power from either power generators and/or other switchboards/connectors. A connector can be used to connect other connectors, so it is a versatile object to use. If a connector is to deliver power to other connectors, it will deliver evenly distributed power to the relevant connectors. Each unit can easily be switched on/off by selecting/deselecting the include option

See chapter 3.8 and chapter 18 for more info and example(s) on how to configure the power configuration.

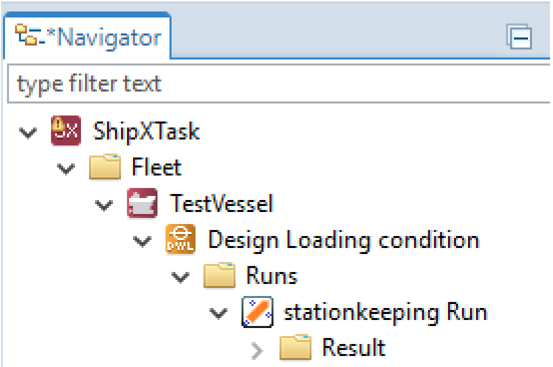
To get full access to all the features it is required to have the proper extended license. The feature limitations can be seen from this table:

Feature	Functionality	Standard license	Extended license
Power Module	Include	Yes	Yes
Power Generator (PG)	Include	Yes	Yes
	Reduced utilization	No	Yes
Switch Board (SB)	Include	Yes	Yes
	Power	Yes	Yes
	Power reserved	No	Yes
	etaE	Yes	Yes
	Powered from PG	Yes, single	Yes, multiple
	Powered from SB	No	Yes, multiple
Heavy Consumers	Include	No	Yes
	etaE	No	Yes
	Powered from SB	No	Yes, multiple
	Powered from PG	No	Yes, single
Propulsors	etaT	Yes	Yes

Note!
The allocation routine in this software will allocate forces/power based on the capabilities reported by the propulsors. If the available power to the propulsors is severely limited by either limiting the power generator power or the switch board available power, the results from the calculation may be poor and extremely conservative.

9 RESULTS PRESENTATION

When a station keeping calculation is completed successfully all the reports and plots will be available in the results folder inside that run.



The number of available plots and reports depend on what type of calculation was performed. Plots and reports can be viewed by double-clicking the one you want, and they will be displayed in the ShipX plot program.

Right-click the Results folder and select the Explore option opens the standard system file manager where you can further work on the reports/plots.

For more information about ShipX plot program features, browse the integrated help menu in the plot program. Here you can, amongst other, read about how to put multiple plots into one, export your plots to into Microsoft Word documents and a variety of other formats, including how to create an animation of a series of plots.

The plots and reports found in the Result folder are mostly self-explanatory. A few comments are in place:

- In many of the plots and reports the environmental direction is given. This is the direction the calculation was performed for. This direction must not be confused with the force direction resulting from the environmental forces. E.g. a wind direction of 30 degrees will not (in general) produce a wind force directed at 30 degrees.
- A propulsor resulting force is the actual force acting on the hull, i.e. corrected for thrust loss.
- A propulsor resulting direction is the angle in which the propulsor resulting force is acting. In general, this will be a different angle than the physical angle of the propulsor.
- The propulsor resulting delivered power, P_D, presents the delivered power for each propulsor.
- The propulsor load presents the actual thrust based load compared to the maximum (or minimum, if negative) thrust for the propulsor.
- The wave drift coefficient data plot is converted from the VERES result file into the coordinate system used in this plug-in.
- In the propulsor result plot the length of the thrust arrows indicates the forces relative to each other. This plot can be animated by performing the following steps:

    1. Close the ShipX Plot Program (or make sure there are no plots or reports shown therein).
    2. Plot the 'Propulsor results' plots from the Station Keeping Plug-In.

3. In the ShipX Plot Program select the 'File – Export – All files to JPEG including HTML animation script' menu. A 'Browse for folder' dialog will appear. Select the folder where you want to save the files and click OK. All plots will now be written to this folder.
4. A dialog will ask if you want to view the animation now. Click 'Yes' to open your browser and view the animation.
5. The animation can be started at a later time by double clicking the 'view_as_animation.html' file found in the folder defined in step 3.

All reports named "DP Capability…" or "A-xx…" are in the DNVGL coordinate system, hence the directions, angles etc may not correspond with the other Station Keeping reports.

10 RUNNING ADVANCED CALCULATIONS

After setting up a Station Keeping calculation run, you can perform different types of calculations:

1) Run the specific Station Keeping run manually
2) Use **Combined Run** to run multiple runs simultaneously
3) Use **Parameter Study** to run a Station Keeping run with variable input
4) Use **Hull Transformation Run** to modify the geometry file and run multiple runs on variations of the geometry
5) Use Workflows to perform batch running / parameter studies on a run

The above list is intended to be sorted based on the required user level where method 1 is the simplest and method 5 the most advanced. See the following chapters for details.

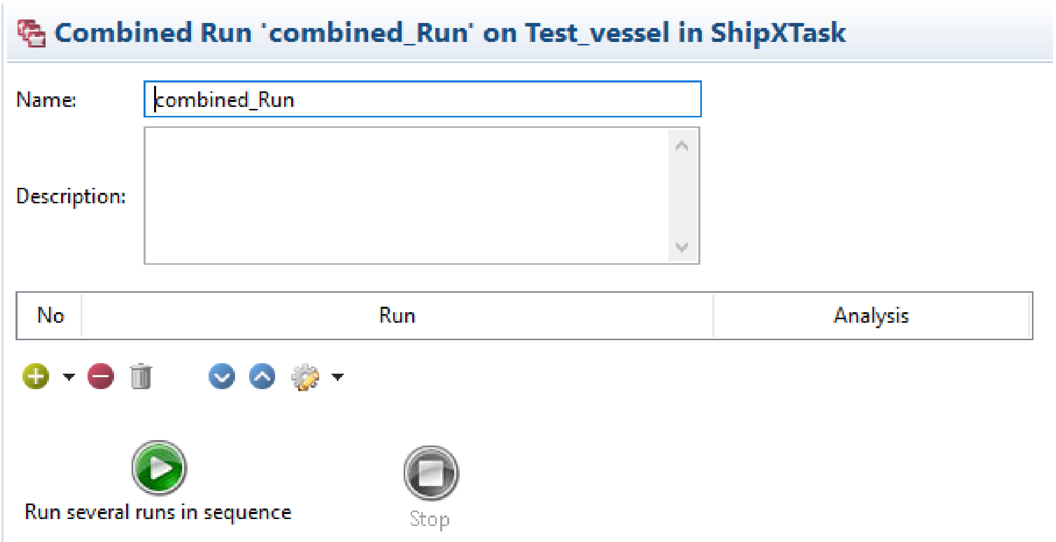
10.1 Manual

To run a Station Keeping calculation manually, simply right-click the run and select **Run** from the pop-up menu, as outlined in chapter 18.

10.2 Combined Run

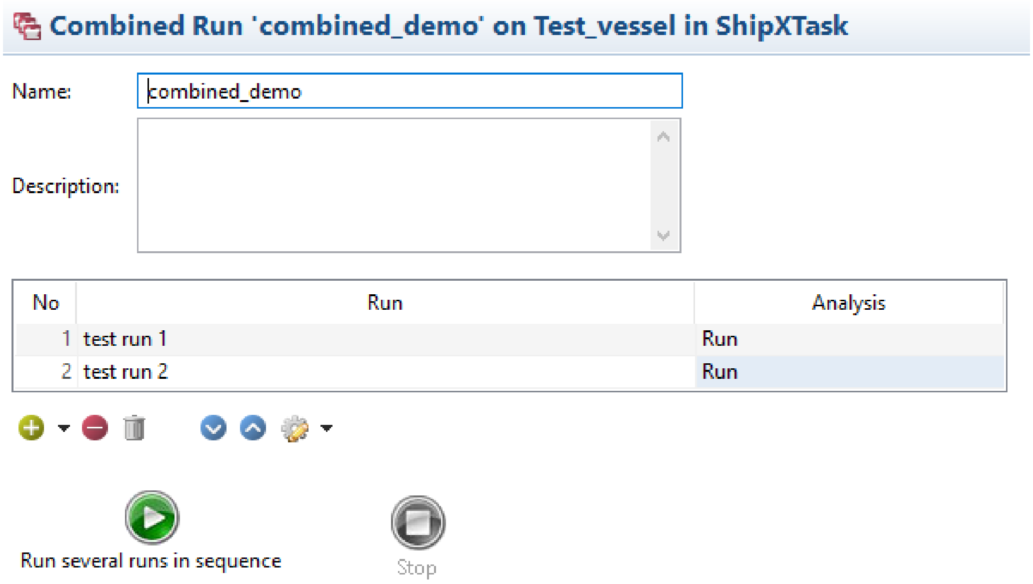
A combined run can be used for running multiple existing runs in batch. The original runs will not be modified in any way, they will simply be run in sequence and the results gathered in the combined run Result folder.

1) To create a Combined Run, right-click the **Runs** folder in a ship loading condition, select **New** and then **Combined Run**.



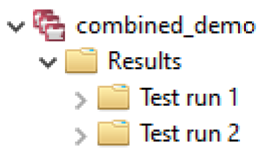
2) The first thing we do is give the run a name, we'll call it **combined_demo**.
3) Then we add existing runs to the list of runs by pressing the + button, clicking in the newly created run-cell and select a run you want to calculate from the dropdown menu. When the run is selected, click in the corresponding analysis-cell to select which type of calculation you want to perform

(some plug-ins support more than one type of calculation). Repeat the process for all the runs you want to batch-calculate, then save your project.



4) When you want to run the calculations, either right-click the **combined_demo** run and select **Run several runs in sequence** or simply press the **Run several runs in sequence** button.

ShipX will go through the list, make a temporary copy of each run, run the calculation and place the results into a subfolder under the Results folder of the combined_demo run.

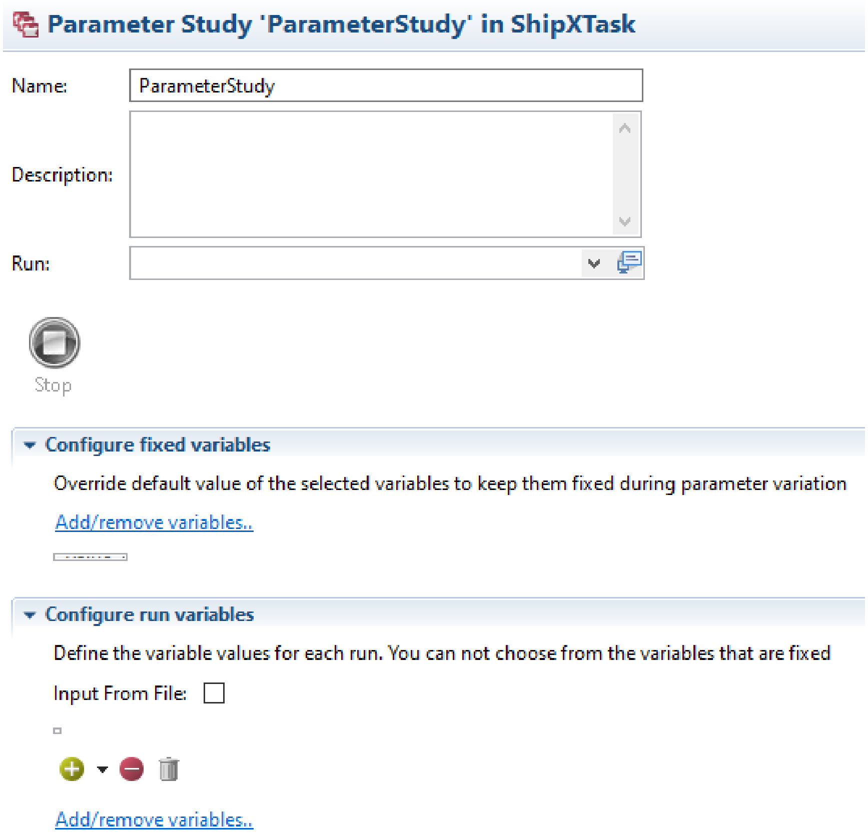


The results can be browsed like any other run results.

10.3 Parameter Study

A parameter study can be used when you have a need to investigate different input variations in a predefined run. For example, you want to perform a calculation with different longitudinal positions for the main propulsors.

1) To create a Parameter Study, right-click the **ShipX task**, select **New** and then select **Parameter Study**.



2) The first thing to do is to give the run a name, we'll call it **study_demo**.
3) Next, we select which run we want to investigate by selecting it from the **Run** dropdown menu.

The input values we want to do a parameter study on must be defined as variables in the run we just selected. We can change the value of an input into a variable and then control the value of the variable in different calculations. More on how to do this later in this chapter.

We now have two sections where we can define variable input:

- **Configure fixed variables:** Here you define any variables from the run that <u>is not to be changed</u> in the calculations. For example, if your run contains a variable for the wave height, but you only want to investigate the variation in the longitudinal position of the main propulsors, you define your wave height variable here and give it a fixed value.

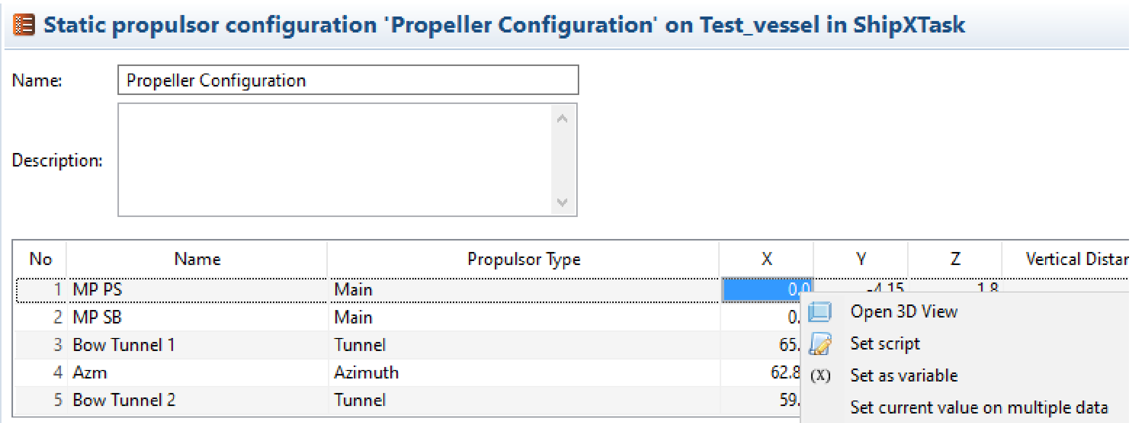
- **Configure run variables:** Here you define any variables from the run that <u>is to be changed</u> in the calculations. In our example we therefore define the variable for the longitudinal position of the main propulsors here.

We can then click Add/remove variables in the two sections, define their value(s) and perform the calculations.

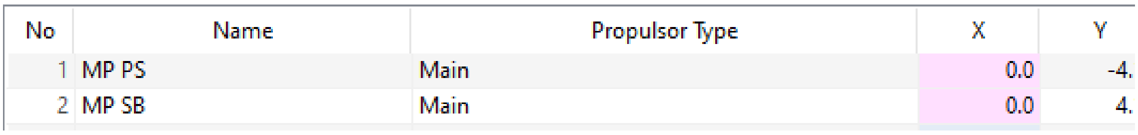
In this example we will use a Station Keeping run already set up to run the calculations. If you haven't created one yet, look to chapter 18 for detailed instructions.

We want to investigate the effect of changing the longitudinal position of the main propulsors. We must therefore make the longitudinal position of the main propulsors input into a variable. The simplest way of doing this is to:

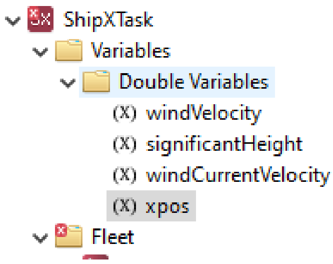
4) Open the Station Keeping run in question, click the Configuration tab, press the Ctrl-key on your keyboard while clicking the name of the Propulsor Configuration name in the selectable dropdown list. This opens the propulsor configuration in question.
5) Inside the propulsor configuration, right-click the X-position of one of the main propulsors and select **Set as variable** from the dropdown menu.



6) A dialog will now appear where you can define the name of the variable, let's call it **xpos**. Repeat the process on the second main propulsor (if you have one) and give it the exact same variable name as we want both to have the same longitudinal position.

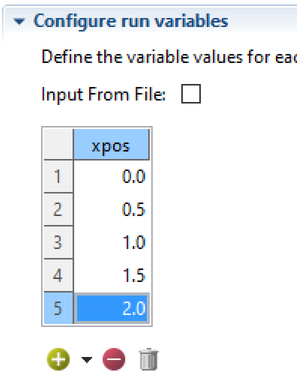


The input boxes now turn pink, indicating that they are controlled by a variable. Hover over the input box to see which variable is controlling them.
If you navigate to the top of the ShipX task, you will find a folder named Variables. Expand that folder to see all variables you have defined (in this case, only the **xpos** is a variable we are interested in).

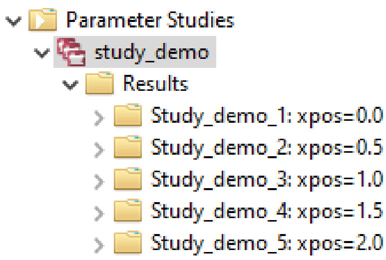


If you double-click the xpos variable, you can define a value for it. If you do so, you can go back to the propulsor configuration and see that the input value has changed accordingly. We are going to do this a bit more cleverly though.

7) Going back to the **study_demo** input. In the **Configure run variables** we click the **Add/remove variables** and select the **xpos** variable and click **Finish**. We now get a table where we can define the input values for **xpos** we want to investigate. Click the + button to add rows to the table. Fill the table with the input values you want to investigate. It may look like this:



If we now right-click the **study_demo** run and select **Run**, or click the **Run** button, all variations of our calculations will run. The results will be gathered in the **Results** folder of the **study_demo** where the results can be studied as usual:



Note: In the above steps, when we defined and selected variables, we could have defined lots more variables. For example, we could add the Y-position, Z-position, propeller diameter, wind speed, wind areas, external forces etc. This would have added more columns to our table of scenarios to investigate, but it would also create a much larger data space to investigate. In this example we used only one variable for simplicity.

10.4 Hull Transformation Run

A hull transformation run can be used for running existing runs in batch on a modified hull. The original runs will not be modified in any way, a temporary copy will be made and later removed. The results gathered in the hull transformation run **Result** folder.

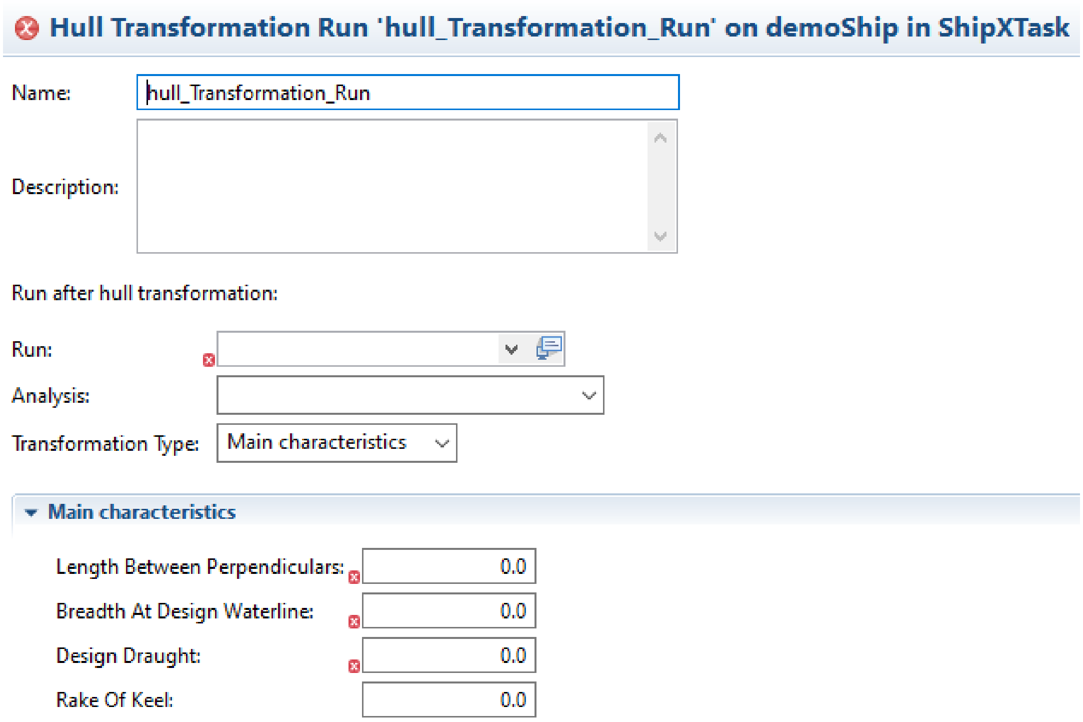
The logical steps are:

- Create a temporary copy of the hull geometry with the prescribed modifications
- Create a temporary copy of the selected run(s) and run them on this modified geometry
- Gather all the results
- Delete the temporary data

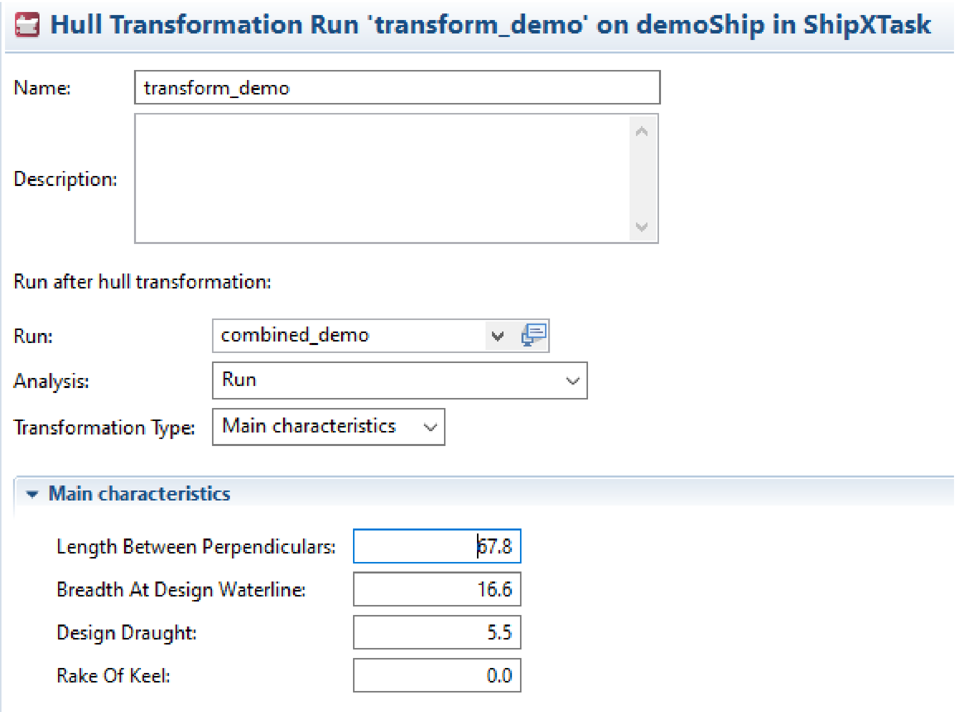
What makes this very flexible is that the run you select to use can be any run, so it can be a normal **Station Keeping Run**, it can be a **Combined Run**, it can be a **Parameter Study** etc. Using these options, you can select to run as many runs you like of the modified hull geometry.

This is how to do it:

1) To create a Hull Transformation Run, right-click the **Runs** folder in a ship loading condition, select **New** and then **Hull Transformation Run**.



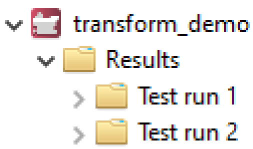
2) The first thing we do is give the run a name, we'll call it **transform_demo**
3) Then we select the **Run** we want to use. We will select the combined run from chapter 10.2, but you can use any existing run here. Remember to also set the **Analysis** to **Run**.
4) Click the **Populate with values from ship** link to gather the existing geometry data and populate the input boxes with them.



We are now ready to transform the vessel geometry and run the calculations on the modified geometry. Remember this does not affect the original vessel, only a temporary copy of it.

5) Select the **Transformation Type** and populate the new hull geometry data input
6) Right-click the **transform_demo** run and select **Run with transformed hull**

The temporary copy of the hull will now be modified according to your settings and the selected runs will be run on the resulting hull geometry. The results will be gathered in the **Results** folder of the **transform_demo** run where the results can be studied as usual:



*Note. The hull geometry run modifies hull dimensions, f.ex Lpp. If your loading condition uses, for example, automatically calculated Lwl, the Lwl will be scaled according to the hull geometry scaling, so all is good. If you, however, have defined a manual Lwl input, this will of course not be modified by changing the hull geometry. To overcome such issues, the Lwl (or any other non-scalable input) should either be set to default (right-click input box and select **Set to default**) or they should be set relative to the corresponding parameter, f.ex the Lwl input value could be set to =0.96*Lpp.*

10.5 Workflow

Using workflows opens up a very advanced tool set of calculations, analysis and reporting. Not all features are yet available/usable for all plug-ins. The example outlined in this chapter may also be performed using the procedure in chapter 10.3 although the workflow approach has potential for more advanced post-processing of the results.

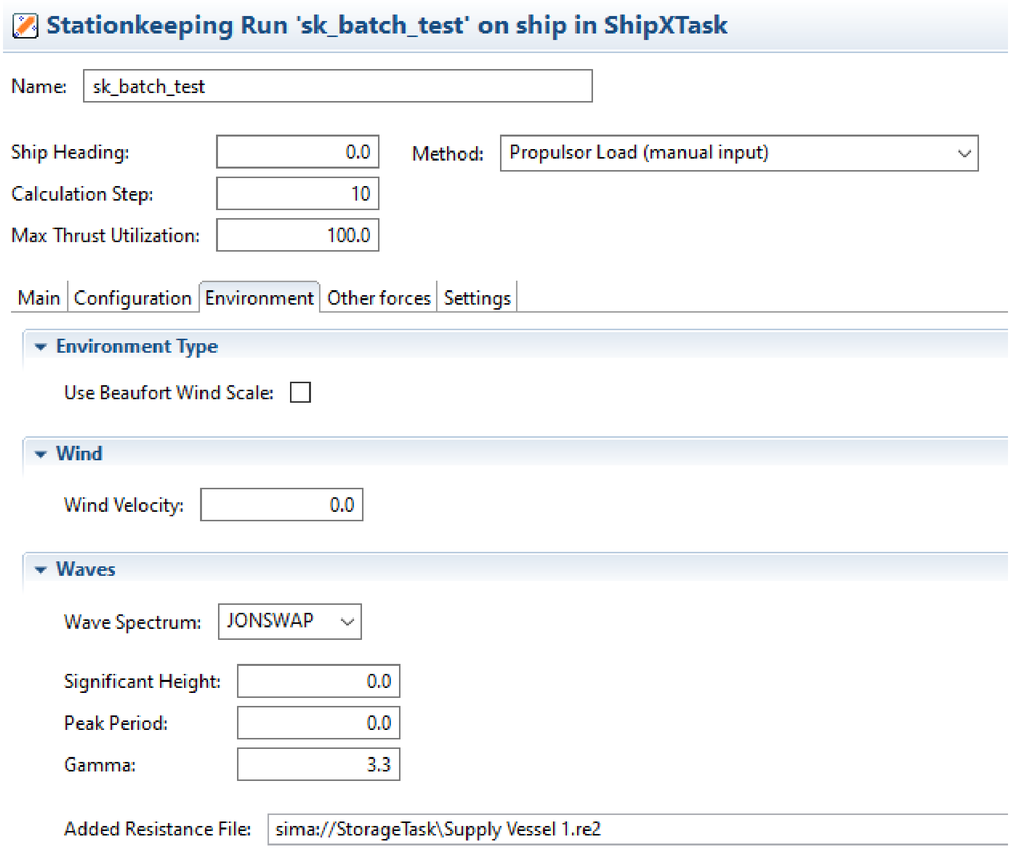
The logical steps are:

- Create a Station Keeping run
- Create a set of input data (for example different wind speeds)
- Run calculations, loop through all defined input (e.g. wind speeds)
- The results will be stored in separate result folders for later investigation.

There are a few steps you must go through to set it all up, but the reward is that you can easily perform multiple calculations. If need be, you can easily change some other input in your Station Keeping run, press a button and perform all the calculations over again with the new input. You can also keep the whole loop infrastructure and replace the initial Station Keeping run to perform the same calculation loops on a different Station Keeping run. This is an extremely powerful tool, so it is well worth learning how to use it. This feature has potential to be a real time saver.

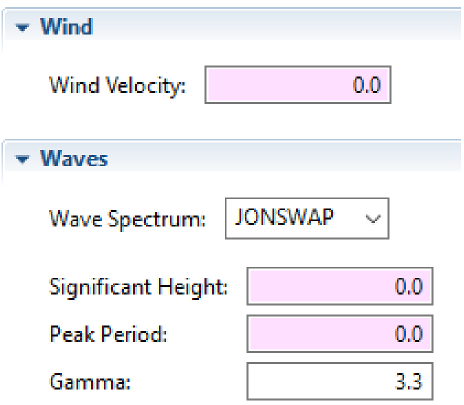
In this chapter we will go through all the steps needed to setup this feature, how to run the calculations and where to find the results. There are a few steps to set this up but follow closely and you will be rewarded.
In this example we will use a Station Keeping run already set up to run a **Propulsor Load (manual input)** calculation. We will then run multiple calculations using different sets of wind speed, wave height and wave period as input. You can create this run any way you like but we will assume you already have created a run that works, as this is outside the scope for this chapter. Make sure you have a run that works, and then follow these steps to set up the automated calculations:

1) You need a Station Keeping run setup to run a **Propulsor Load (manual input)** calculation to follow this example. There may be a few difficult steps along the way, so if you use the same names as in this guide it will be easier to follow. Name the Station Keeping run you are to use **sk_batch_test**. You also need to include wind and waves in the calculation. The environment tab will then look like this:



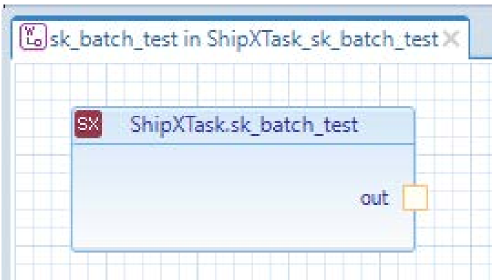
There is no need to define a fixed value for the wind velocity, wave height and wave period as we are going to feed different values into these inputs later.

2) Now we need to create variables (one for each input we want to manipulate) so we can work on these variables later. Simply right-click into the **Wind Velocity** input field and select **Set as Variable** from the pop-up menu. Accept the suggested variable name and click the **OK** button to close the dialog. Repeat this procedure for the **Significant Height** and **Peak Period** input for waves. These three input boxes should now turn purple (indicating they are controlled by a variable) and look like this:



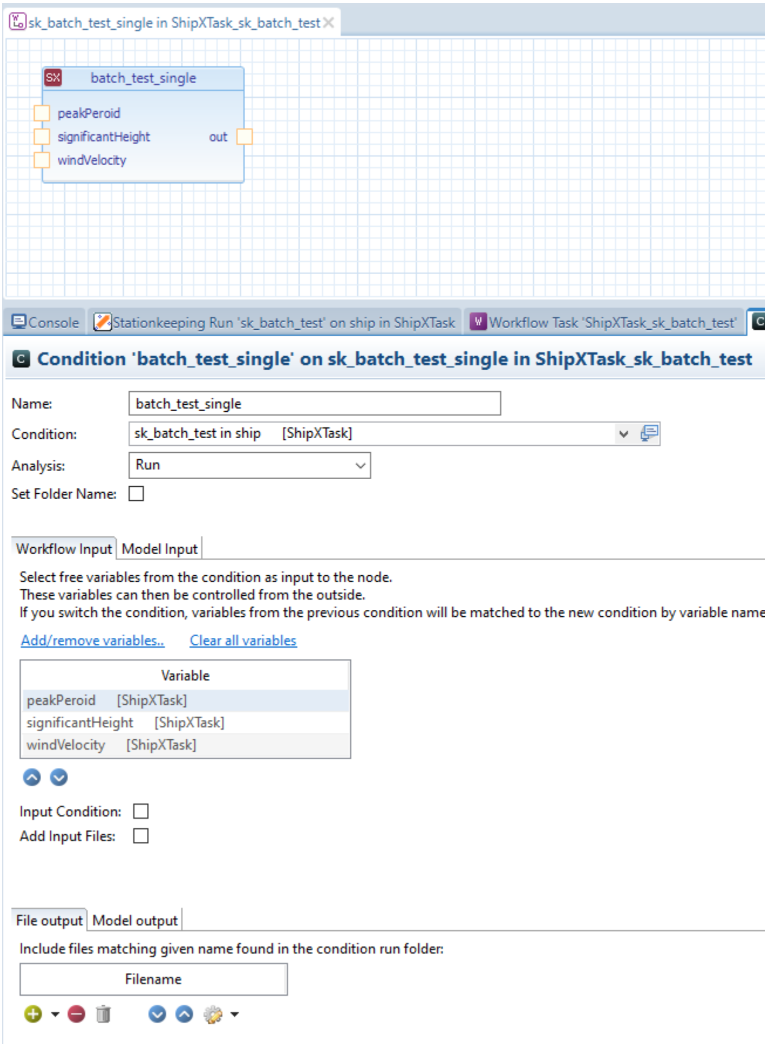
We are now done with the customization in the Station Keeping plug-in and will start to set up the looping of the input and the batch calculations.

3) The next step is to right-click the Station Keeping run in the **Navigator** window and select **Create workflow with this as input** from the pop-up menu. You will now get this window:

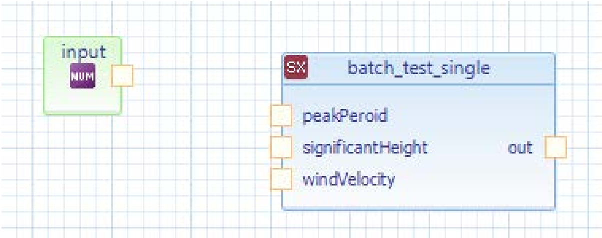


In the **Navigator** window, extend the **ShipXTask_sk_batch_test** item and then right click the **sk_batch_test** item, select Rename and name it **sk_batch_test_single**. Double-click the **ShipXTask.sk_batch_test** box to display its input. Name the input **batch_test_single**. Click the **Add/remove variables** link. A dialog pops up displaying all defined variables. For now, we have only the three we just defined, select all and click **OK**.

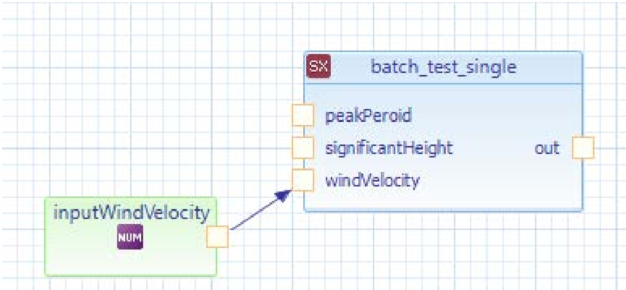
Your input will now look like this:



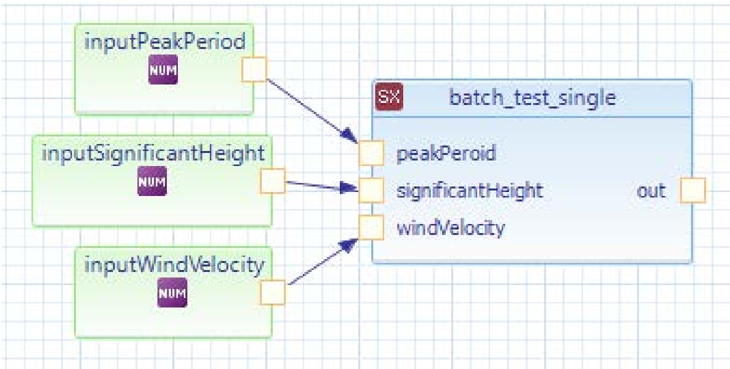
We now need to create a link from the input we are going to loop over into the variables that we have defined. To do this, simply look at the **Palette** in the right top area of ShipX, expand the **Input** section, click the **Number** item, and drag and drop it next to our **batch_test_single** box.



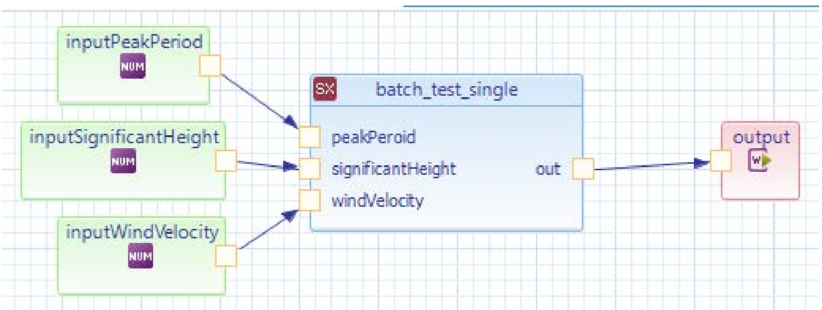
Right-click the **input** box, select **Rename** and name it **inputWindVelocity**. Click in the small orange box in the **inputWindVelocity** box and drag the arrow into the orange box next to **windVelocity** to link them together. Drag the boxes to position them nicely. You should now have this:



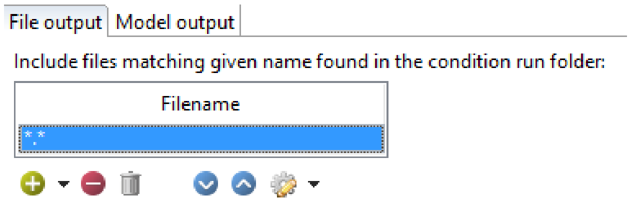
Repeat above procedure for the remaining two variables, so you get something that looks like this:



Now we need to handle the output from the calculations. From the **Palette**, expand the **Output** section, select a **Workflow Output** and add it to the existing design. Link it to the out box, so you get this layout:

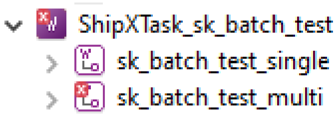


Finally, we need to define which output files from the calculations we are interested in. In most cases we want everything, so double-click the **batch_test_single** box. In the dialog that appears there is a **File Output** section at the bottom. Click the + sign to add a line and type ***.*** in it. It will look like this:

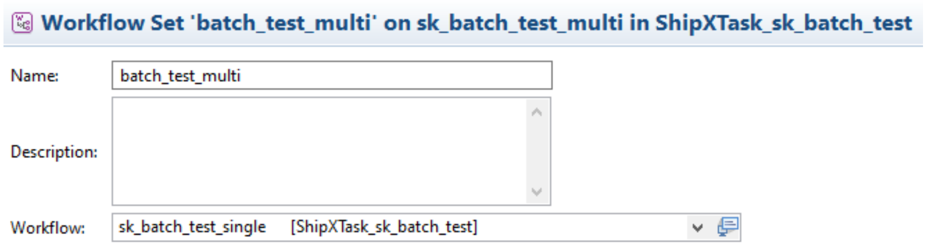


If we, for example, only had been interested in text output files, we could have written ***.txt** instead.
We have now created the setup to run a generalized calculation using the three variables we defined. If we wanted to, we could now double-click each of the three **input*** boxes, set a value for each and run the calculation. We are going to step it up a notch though, we want to add a dataset of input and perform multiple calculations automatically. That is what we will do next.

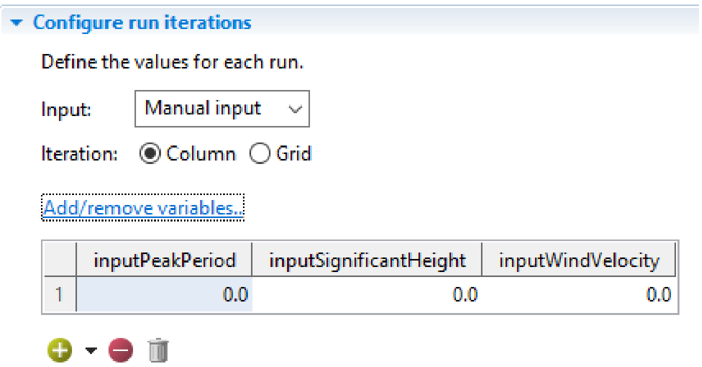
4) Right-click the **ShipXTask_sk_batch_test** in the **Navigator** window, select **New** and **Workflow**. This new workflow will contain the logic we need to run the previous workflow in loops. In the **Navigator** window, right click the new **sk_batch_test_single1** item, select Rename and name it **sk_batch_test_multi**.



Double-click the **sk_batch_test_multi** to open its dialog window. From the **Palette**, expand the **Workflow** item, select the **Workflow Set** and drag it into the workspace. Double-click the **Workflow Set** box to display its input. Name it **batch_test_multi**. In the **Workflow** input dropdown box, select the **sk_batch_test_single** item.



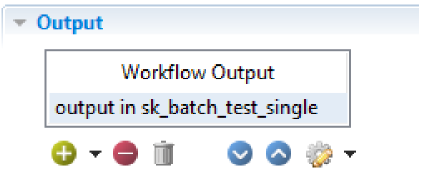
Now we get to define the input data we are going to run/loop with.
In the **Configure run iterations** section, make sure it is set to **Manual input** and **Column**. Click the **Add/remove variables...** link, select all variables and click **OK**. You should now see this:



Click the + sign to add rows, then populate the table with the conditions you would like to investigate. It might look something like this in the end:

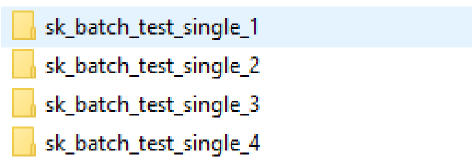
	inputPeakPeriod	inputSignificantHeight	inputWindVelocity
1	8.0	2.0	12.0
2	10.0	1.0	14.0
3	11.0	4.0	15.0
4	12.0	5.0	17.0

Finally, click the + in the **Output** section and select **output in sk_batch_test_single** from the dropdown menu.



We are now ready to perform the calculations. This is a good time to save your workspace.

5) To perform the calculations, right-click in the workspace (next to the **batch_test_multi** box) and select **Run from start**. With above given input data, ShipX will now run 4 calculations and create result folders for each one. To access the result folders, right click in the workspace and select **Explore** from the popup menu. You should now see this:



The number in the folder name corresponds to the row in the input table we provided earlier. You will find a **results** folder in each of the four top folders. The **results** folder will contain all reports and plots as you would expect from a normal calculation. Use the ShipX Plot Program to open the *.mpl result files.

It is a good idea to check the input summary report from each calculation to check that the input used in the calculation corresponds with the input you defined in above input table, just to make sure everything was set up and run correctly.

11 ERROR AND WARNING MESSAGES

This chapter contains the trappable error messages one can encounter when running the ShipX Station Keeping Plug-In.

Error 0: An unspecified error occurred during the calculations.

Cause:

This is an error message for any unspecified error occurring during the calculation.

Solution:

Further investigations need to be performed on the input in order to find a solution to the problem. Check all input thoroughly. If no discrepancies are found, contact SINTEF Ocean to solve the problem.

Error 100: Unable to allocate large enough thrust for this environmental condition.

Cause:

This error may occur when running a propulsor load study because the environmental forces defined are too large compared to the ships propulsor capabilities.

This error may also occur when running a capability study with current or other external forces included. If the external forces are defined to be too large compared to the ships propulsor capabilities the ship will not be able to keep station even with zero wind velocity (which is the iteration parameter).

Solution:

Reduce the user defined external forces in order for the ship to be able to keep station.

Error 101: An error occurred while calculating wave drift forces.

Cause:

This is an error message for any unspecified error occurring during the calculation of wave drift forces.

Solution:

Further investigations need to be performed on the input in order to find a solution to the problem. Check all input thoroughly. If no discrepancies are found, contact SINTEF Ocean to solve the problem.

Error 102: An error occurred while calculating wind forces.

Cause:

This is an error message for any unspecified error occurring during the calculation of wind forces.

Solution:

Further investigations need to be performed on the input in order to find a solution to the problem. Check all input thoroughly. If no discrepancies are found, contact SINTEF Ocean to solve the problem.

Error 103: An error occurred while calculating current forces.

Cause:

This is an error message for any unspecified error occurring during the calculation of current forces.

Solution:

Further investigations need to be performed on the input in order to find a solution to the problem. Check all input thoroughly. If no discrepancies are found, contact SINTEF Ocean to solve the problem.

Error 104: Max number of iterations was reached without getting to a solution.

Cause:

A maximum number of iterations have been defined in order to prevent the calculations to come to a halt. This limit has been set considerably larger than any value found during testing. This error message does however indicate that the limit has been reached.

Solution:

Check all input thoroughly. If no discrepancies are found, contact SINTEF Ocean to solve the problem.

Error 106: Propulsor is of undefined type.

Cause:

When designing your own propulsor code you must remember to define the type of the propulsor you created. See the ShipX Run log for information on which of the propulsors that triggered the error.

Solution:

Define the type of all the propulsors you create. See chapter 12.5.5 for further information.

Error 107: Station keeping process is already running.

Cause:

When running a calculation, the plug-in uses FORTRAN and C dll routines. If some variables in these routines are global or have a 'Save' property they will be shared between all runs. This will cause erroneous calculation results.

Solution:

For the time being it is only possible to run one calculation at a time.

Error 108: Propulsor returned smaller maximum force than minimum force.

Cause:

When communicating with the propulsors throughout the calculation, one of the purposes is to get the maximum and minimum forces available (for a given angle) for any given propulsor. If the maximum force returned is smaller than the minimum force for any propulsor in use this error will occur. See the ShipX Run log for information on which of the propulsors that triggered the error.

Solution:

If the propulsor triggering this error is one of your own, you should debug your code to make sure the forces calculated therein are correct. If the propulsor triggering this error is not one you created, contact SINTEF Ocean to solve the problem.

Warning: The ship is unable to keep station for some of the calculated environmental directions.

Cause:

The ship is not able to keep station given the defined environmental forces.

Solution:

If you run the "Propulsor Load (manual input)" method, you must either reduce the environmental forces or increase the available thrust from the propulsors. If you are running any of the other “Capability” methods the same solution applies, but the typical problem here is that either the current velocity is set too high or that the wave height and -period are set too high (by not linking the waves to the wind) so that the ship is not able to keep station even if no wind is present.

SINTEF

12 IMPLEMENT YOUR OWN PROPULSOR (USING FORTRAN)

This chapter describes how to create your own propulsor and implement it in the ShipX Station Keeping Plug-in. The propulsor code has many tasks, but the main tasks are:

1. Define input required from the user in order to calculate thrust.
2. Return maximum and minimum achievable thrust for a given angle when prompted.
3. Verify that the allocated thrust and thrust direction is achievable.

In addition to developing your propulsor(s) in Fortran, your propulsors can also be implemented in the Propulsion Library using Java code. See chapter 13 for further information.

Contact SINTEF Ocean to get a demo project to help you start creating your own models.

12.1 Typographical Conventions

The following conventions are used in this chapter:

Bold Initial Caps:	Key names, menu names, dialog boxes and items that are selected from menus (for example **File** menu) and submenus.
BOLD CAPS:	Variable names in the source code.
Courier	File names, paths and commands.
Italics	Name of buttons or fields in dialog boxes (for example *Calculate Station Keeping* button). Name of folder descriptions.

12.2 Software Requirements

The development environment used for creating the software described in this chapter is Compaq Visual FORTRAN Professional Edition 6.6B.

12.3 Overview

An example project workspace file named sxudfgen.dsw (called Project from now on) is provided by SINTEF OCEAN. The user is advised to use this Project for implementing your own propulsors. Locate the Project and double-click it to open. Click the **FileView** tab to enable the file view.

The *Interface* folder contains the files common for all defined propulsors in this Project. These files define the propulsor interface towards the ShipX plug-in. You can define a maximum of 20 propulsors in the Project. Each propulsor will have its unique ID identifying it for the ShipX plug-in. It will also have its own name and input. This is all defined in the *Interface* folder of the Project.

The *Propulsors* folder contains the files needed for doing the actual calculations for the propulsor. It is recommended to separate the propulsor files into their own folders under the *Propulsors* folder, identifying the folders with the unique ID in the folder name for clarity.

12.4 How It Works

In order to show what can be achieved with this code, perform the action described in chapter 12.5.1 and in chapter 12.5.2. In chapter 12.5.2 set the variable NUMUNITS to equal 1 and rebuild the project. Make sure that the file named sxudfgen.dll (from now on called User dll) is located in your Windows-System32 folder is the file you just built (the file is automatically replaced with the newly built file if the instructions in chapter 12.5.1 are performed correctly). You will now have told the dll to export one propulsor, namely the demo propulsor. Start ShipX and click the **New-New Station Keeping Data Input** menu. This will start a new run of the station keeping plug-in. Navigate to the **Propulsors** tab and click the **Type** pull-down menu for any propulsor. You can now see the demo propulsor in the pull-down list. Selecting it will bring up the input for this propulsor. All input, including description, default values and number of decimals is defined in the Project. This propulsor (although not correctly representing a physical propulsor) can now be used in station keeping calculations.

What really happened here is that on start-up of the station keeping plug-in, the plug-in asks the Propulsion Library what propulsors are available. The Propulsion Library then relays the request to a main propulsor dll to report back what propulsors are present in the FORTRAN part of the code. This dll is called sxmtfgen.dll (from now on called Marintek dll) and contains the generic propulsors developed by SINTEF Ocean. The SINTEF Ocean dll will then relay the request from the plug-in to the User dll, which in turn will report back its propulsors to the SINTEF Ocean dll. The SINTEF Ocean dll gathers all propulsor descriptions and returns them to the Propulsion Library. The prolusion library also queries existing Java based user models, but this is described in chapter 13. The plug-in receives all propulsor descriptions from the Propulsion Library and displays them in the pull-down menu. When the user selects a propulsor the plug-in asks the Propulsion Library (which in this case relays the request to the SINTEF Ocean dll) for the input description for this propulsor. If the propulsor is described in the SINTEF Ocean dll (defined by its unique ID, see chapter 12.5.4) the propulsor input description is directly returned to the plug-in. If, however, the propulsor is not defined in the SINTEF Ocean dll the plug-in request will be relayed to the User dll which will return the propulsor input description.

The calling procedure is the same when calculating forces from the propulsors. See Figure 5 for a graphical presentation of the dataflow.

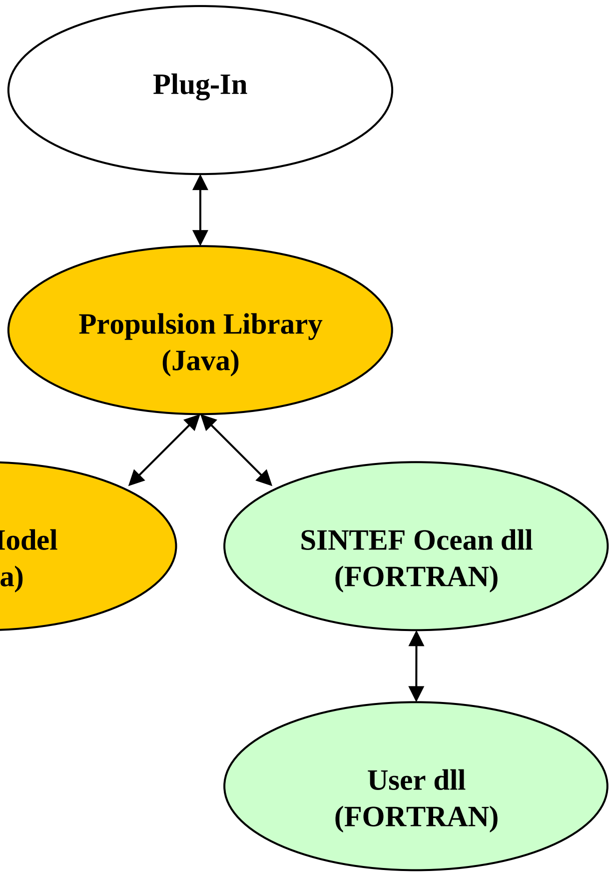


Figure 5, Dataflow between plug-in and propulsor models

The basic idea is that the plug-in only talks to the SINTEF Ocean dll (through the Propulsion Library), which in turn responds directly if the propulsor is defined in the SINTEF Ocean dll. If the propulsor is not defined in the SINTEF Ocean dll the plug-in request will be relayed to the User dll and the reply from the User dll will be relayed through the SINTEF Ocean dll to the plug-in. See also chapter 13 for a description of the Java part of the Propulsion Library.

Chapter 12.5 explains how to implement your own propulsor models into the Project.

12.5 Step-By-Step Implementation Procedure

This chapter gives a step-by-step introduction of implementing your own propulsor into the ShipX Station Keeping Plug-in. Completing the steps in this chapter correctly will lead to a built dll named sxudfgen.dll located in your Windows-System32 folder.

12.5.1 Step 1 - Make Initial Changes To The Example Project

Open the Project, go to the **Project-Settings** menu and click on the **Post-build step** tab. Check that the *Copy* command correctly reflects the location of the Projects Release folder as well as the System32 folder on your system. Edit the *Copy* command if necessary.

12.5.2 Step 2 - Define Number Of Propulsors

Open the fgennums.f90 file in the *Interface* folder. Locate the variable NUMUNITS and set it to the number of propulsors defined in the Project. This will define the total number of propulsors from this Project to present for the plug-in.

Example: With only the existing demo propulsor in use the number of propulsors should be 1.

12.5.3 Step 3 - Define Name

You must decide on a propulsor array index (from now on called Index) and use it consistently throughout the Project.
Open the fgennames.f90 file in the *Interface* folder. Locate the DESCRIPTION variable and write the propulsor name on the appropriate Index.
Note that the propulsors are used from the first index and up, meaning that if you set number of propulsors to 3 the propulsors presented in the plug-in will be DESCRIPTION array index 1,2 and 3.

Example: The existing demo propulsor name is defined in Index 1.

12.5.4 Step 4 - Define Unique ID

Open the fgenuniqueids.f90 file in the *Interface* folder. Locate the UNIQUEID variable and identify the unique ID of the propulsor based on the Index. The unique ID will be used in the following steps.
Note that the unique ID numbers are reserved in the following ranges:

- From 1000 to 1999 is reserved for SINTEF Ocean propulsors in the sxmtfgen.dll.
- From 2000 to 2999 is reserved for Client propulsors in the sxudfgen.dll.
- From 3000 to 3999 is reserved for Client propulsors in the Java part of the Propulsion Library.
- From 4000 to 4999 is reserved for SINTEF Ocean propulsors in the Java part of the Propulsion Library.

Example: The existing demo propulsor has a unique ID of 2000.

12.5.5 Step 5 - Define Type

Open the fgentypes.f90 file in the *Interface* folder. Locate the Case sentence containing the unique ID for your propulsor. Set the appropriate propulsor code. See the information above the CASE sentence for guidance on what propulsor type code to use.

Example: The existing demo propulsor is of propulsor type 0.

12.5.6 Step 6 - Define Input

Open the fgeninput.f90 file in the *Interface* folder. Locate the Case sentence containing the unique ID for your propulsor. The statements found in here define the input displayed in the plug-in.

Note that the first four inputs are mandatory. You can change the value of the NUMDEC, MAXVALUE, MINVALUE and DEFAULTVALUE variables, but NOT the sequence and meaning of the first four inputs.

To add your own input simply copy for example the code block starting with I=4, change it to I=5 and define the input description, limits, default values, number of decimals etc. Input limits related to the actual hull can be defined by using special code limits (see comment above the SELECT CASE code).

Variables are transmitted from FORTRAN to the plug-in interface with datatype Real. To define an Integer datatype simply set number of decimals to zero and convert the variable returned from the plug-in to Integer before using it in your code (see next chapter).

Example: The existing demo propulsor has defined a number of inputs in the CASE(2000) code as it has a unique ID of 2000.

12.5.7 Step 7 - Define Calculation Model

Rename the folder corresponding to your propulsors unique ID under the *Propulsors* folder to a more describing name, for example from *ID 2001 - Not in use* to *ID 2001 – Tunnel thruster* (from now on called working folder) if the unique ID for the propulsor is 2001. Add the file(s) containing the code for calculating the propulsor thrust to this folder.

Example: The existing demo propulsor is defined in the demofgen.f90 file in the *Propulsors / ID 2000 - Demo propulsor* folder. This code gives a very simple example of a propulsor code.

Note that the input through the subroutine call is completely standard, it contains no evidence of it being used in a ShipX plug-in manner. It is therefore easy to implement existing code since no rewriting of the calculation model code is needed.

12.5.8 Step 8 - Define Calculation Interface

Create a new file in the working folder and for simplicity give it the same name as the file containing the propulsor code but add an extra 'sx' at the end of the filename indicating that the file has connections to ShipX.

Example: The demofgen.f90 file in the *Propulsors / ID 2000 - Demo propulsor* folder contains the propulsor code. When defining the propulsor interface we then use a file named demofgensx.f90.Copy the contents of the demofgensx.f90 file to your propulsor interface file.

This file is used for interpreting the data received from the plug-in to the data needed in the propulsor code. The contents of the file should be pretty self-explanatory, but a few items are important to note:

- The INPUTREAL array contains the input received from the plug-in. It is important to maintain the index sequence identical to the input index sequence defined in the fgeninput.f90 file.

- If a variable received through the INPUTREAL array is supposed to be INTEGER you will have to cast it to integer before assigning it to a local variable.
- A maximum of 26 inputs can be defined and are received from the plug-in. INPUTREAL array index 27 and above are dedicated to general input such as water density etc.

The WHATTODOFLAG indicates what action to be performed.

- WHATTODOFLAG = 1 indicates that the plug-in is asking for maximum and minimum thrust the propulsor can give. This is in relation to the allocation routine.
- WHATTODOFLAG = 0 indicates that the plug-in is asking if the propulsor really can produce this thrust. This is relevant when the propulsor is unable to produce the same thrust for all angles. A simple iteration routine is implemented to check if the propulsor can produce the required thrust. You may freely copy and use this routine when implementing your own propulsors. If the propulsor can produce the required thrust at the required angle this routine also returns what parameters are needed for giving this thrust (propeller pitch, revolutions etc). This is in relation to the reporting of the results. This reporting feature is currently not in use.

12.5.9 Step 9 - Define Output For Reports

The file defined in chapter 12.5.8 contains arrays named OUTPUTTEXT, NUMDEC and OUTPUTUNIT. These arrays contain the parameters you would like to have in the final reports from each calculation. The maximum number of outputs is currently limited to six where propulsor thrust and propulsor angle are mandatory as the first two.

12.5.10 Step 10 - Define Version

Open the fgenversions.f90 file in the *Interface* folder. Locate the Case sentence containing the unique ID for your propulsor. The statements found in here define the propulsor version displayed in the plug-in.
When changing the propulsor calculation model (for example after bug fixes) it is important to also change the version number. This will enable the users to quickly see what version of the propulsor model they are running and will ease any later debugging issues.

12.5.11 Step 11 - Relay Input To Correct Propulsor

The final step is to route the data received from the plug-in to the correct propulsor model based on the unique ID of the propulsor. Copy the subroutine call from the <filename>sx.f90 file and paste it into the fgencalc.f90 file at the correct unique ID in the CASE sentences.

12.5.12 Step 12 - Final Actions

If you have performed the above steps you should now be ready to build the Project. Upon completion of building the Project make sure that the freshly built dll named sxudfgen.dll is located in your Windows-System32 folder. You are now ready to start ShipX and test your propulsor code.

13 IMPLEMENT YOUR OWN PROPULSOR (USING JAVA)

This chapter describes how to implement your own propulsor in the ShipX Station Keeping Plug-in.

Contact SINTEF Ocean to get a demo project to help you start creating your own models.

13.1 Typographical Conventions

The following conventions are used in this chapter:

Bold Initial Caps:	Key names, menu names, dialog boxes and items that are selected from menus (for example **File** menu) and submenus.
BOLD CAPS:	Variable names in the source code.
Courier	File names, paths and commands.
Italics	Name of buttons or fields in dialog boxes (for example *Calculate Station Keeping* button). Name of folder descriptions.

13.2 Software Requirements

The development environment used for implementing the software described in this chapter is Java. The example outlined in the following will use the standard distribution from Sun.

13.3 Overview

In order to be able to implement your propulsor routine in ShipX Station Keeping Plug-In you must follow a few simple steps. The following chapters will walk you through these steps one by one.
All propulsor models are gathered in a Propulsion Library. This library consists of one file (PropulsionLibrary.jar) developed by SINTEF Ocean. Additional jar files can be added to the library from which the user may select propulsor models in the plug-in. In order for your propulsor model to be included into the Propulsion Library it must follow certain rules (known as interface) defined in Java. The following chapters will walk you through the required steps to perform this task. When you have completed these steps your propulsor model will be shown and used in the plug-in on a same level as the original methods implemented by SINTEF Ocean.

13.4 How It Works

On start-up of the station keeping plug-in, the plug-in asks the Propulsion Library what propulsors are available. The Propulsion Library then relays the request to both the Java and the FORTRAN part of the Propulsion Library. The Java parts of the Propulsion Library are implementing a Java interface and thus support the standard requests relayed from the Propulsion Library. All Java models will then report back its propulsors to the Propulsion Library. The Propulsion Library also queries existing FORTRAN based user models, but this is described in chapter 0. The plug-in receives all propulsor descriptions from the Propulsion Library and displays them in the pull-down menu. When the user selects a propulsor the plug-in asks the Propulsion Library (which relays the request to the appropriate propulsor model) for the input description for this propulsor.
The calling procedure is the same when calculating forces from the propulsors. See Figure 5 for a graphical presentation of the dataflow.

See also chapter 0 for a description of the FORTRAN part of the Propulsion Library.

13.5 Download And Install The Required Software

As mentioned earlier the software required for this task is Java. What you need is the Java Development Kit (JDK), which includes both the Java runtime files and a Java compiler. First navigate to java.sun.com and download the "JDK with NetBeans" with file name "jdk-8u25-nb-8_0_1-windows-i586.exe" or a newer version if available. Remember to download the 32 bit version (x86), the 64 bit version (x64) will not work. A newer version should also work; just make sure it is a JDK version with NetBeans. This manual refers to the above-mentioned Java version.

After downloading the file, install it using the default settings.

13.6 Step-By-Step Implementation Procedure

This chapter gives a step-by-step introduction of implementing your own propulsor into the ShipX Station Keeping Plug-in.

13.6.1 Step 1 - Setting Up Your Project

Perform the following steps to set up this project.

1. Start NetBeans by double-clicking the appropriate icon on your desktop.
2. Select the **File – New Project** menu. Under Categories select **Java** and under Projects select **Java Class Library**. Click **Next**. Enter the Project Name. In this example we will call it "MyPropulsor". Select Project Location. For this example we will use "C:\Example". Verify that the Project Folder now is "C:\Example\MyPropulsor". Click **Finish**.
3. Select the **Files** tab, right click the root item (named "MyPropulsor") and select **Properties**. Under Categories select **Libraries**. Select the **Add JAR/Folder** button, browse to "C:\Program Files (x86)\ShipX\lib" and select the "PropulsionLibrary.jar". Click Open to add it to the Compile-time Libraries list. Click **OK** to close the dialog.
4. Use the Windows Explorer and browse to "C:\Example\MyPropulsor\src". Create the appropriate folder structure according to the Java standard. In this example we will create the following folder structure "C:\Example\MyPropulsor\src\no\client\propulsor".
5. Copy the file "DemoPropulsor.java" file received by SINTEF Ocean into the folder defined in step 4. In NetBeans, expand the tree view (click the + icons) of MyPropulsor in the Files tab to locate the "DemoPropulsor.java" file. Double-click the file to open it.

The methods required by the interface are already implemented in demo class. What you need to do is to edit them to facilitate your needs.

The "DemoPropulsor.java" class file contains many methods that are not really relevant for implementing your Station Keeping thruster. The chapters to follow describe the changes you must do in order to personalize the demo class to your needs. The rest of the class can be left untouched.

Note that when you add propulsors to the Propulsion Library using the FORTRAN method (see chapter 0) they can only be used in the ShipX Station Keeping Plug-In. When you add them using Java (as described in this chapter) the propulsors can be used in more general applications as well. The validity of the propulsor model is defined by the implementer of the model as can be seen in the following chapters.

13.6.2 Step 2 - Define Correct Package

If you created a different folder structure than the one described in chapter 13.6 step 4 you must edit the first line of the class file to reflect the appropriate folder location.

13.6.3 Step 3 - Define Unique ID

Every propulsor model needs to have its own unique ID. This is the parameter that ShipX uses to differentiate the propulsor models. The range from 3000 to 3999 has been reserved for client propulsor models. Edit the **getUniqueID()** method to an unique ID. In this example we will use a unique ID of 3000.

13.6.4 Step 4 - Define Name

The name of the propulsor model is displayed in the drop-down box in the ShipX Station Keeping plug-in. Enter a name for your propulsor method in the **getName()** method. In this example we will use the name “Client Demo Propulsor Model”.

13.6.5 Step 5 - Define Version

The version number of the propulsor model is displayed in the **Info** box in the ShipX Station Keeping plug-in. Edit the **getVersion()** method to set this propulsor models version number. In this example we will use version 1.0.000 as it is the initial version of the propulsor model. When changing the propulsor model (for example after bug fixes) it is important to also change the version number. This will enable the users to quickly see what version of the propulsor they are running and will ease any later debugging issues.

13.6.6 Step 6 - Define Info

The propulsor model info is displayed in the **Info** box in the ShipX Station Keeping plug-in. The version info may contain information regarding this propulsor model, such as limitations, on which background theory it is based etc. The information is purely for the benefit of the user, it has no practical meaning on the use of the propulsor model. Edit the **getInfo()** method to set the appropriate propulsor model info. In this example we will simply use the existing text.

It is good practice to leave the existing text in the example and just edit the description part. In this way, all propulsors will report the basic data (name, version, ID, validity etc) in the same format.

13.6.7 Step 7 - Define Propulsor Type

The propulsor type roughly sorts the propulsors into categories. These categories give information to the logic surrounding the propulsors in the plug-in. To define the propulsor type edit the **getTypeShipXDP()** method. A description of the different categories is given in the comments to the method. The propulsor we are making in this example will be a freely rotating unit, thus the propulsor type is FGEN_TYPE_GENERAL.

13.6.8 Step 8 - Define Model Validity

As described earlier, the propulsors implemented using Java can be used in other applications as well. The Propulsion Library consists of many models, all which have their own model limitations (some are valid for DP only; others are full four quadrant models etc). In order for applications to know which of the available models can be used in the application the model validity is defined in each model in the following methods:

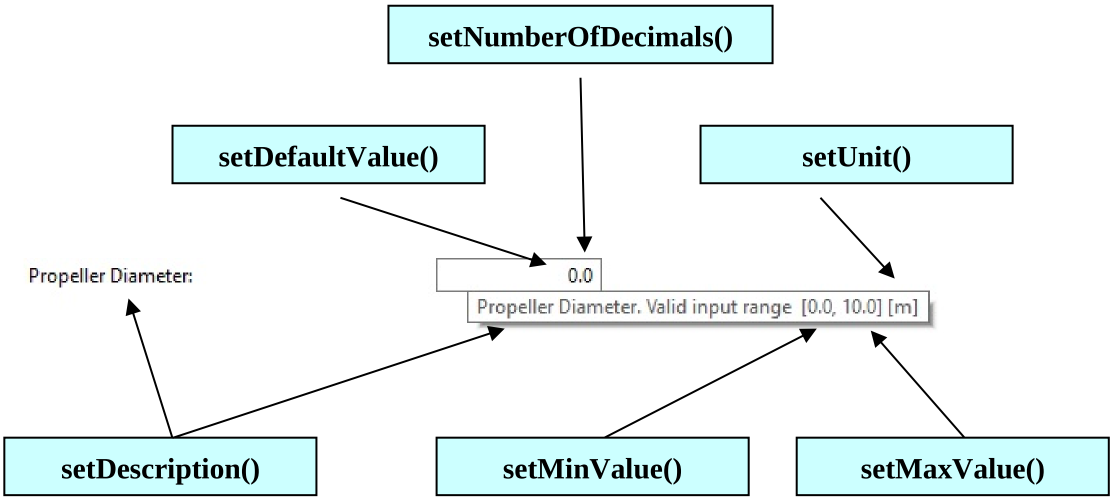
- **getValidForDP()**
- **getValidForQ1()**
- **getValidForQ2()**
- **getValidForQ3()**
- **getValidForQ4()**
- **getValidForSimulator()**
- **getConfiguredForShipXDP()**

The meaning of each method is described in the example java class file. In this manual we are only interested in implementing propulsors in the ShipX Station Keeping Plug-In. We therefore only focus on the **getConfiguredForShipXDP()** method. Make sure this method returns **true**.

13.6.9 Step 9 - Define Static Input

It is now time to define which input the propulsor model requires. These are divided into static- and dynamic input. The static input is input that doesn't change with time, for example propulsor position, propeller diameter etc. These are defined in the **getStaticInputDescriptionShipXDP()** method. As can be seen from this method the position of the propulsor and the maximum and minimum working range of the propulsor are mandatory. If you want to edit some of the other input add them to the end of the list. Each input parameter is defined in an input class named **PropulsorInputElement()**. This class contains the required methods for defining the input parameter. The methods are:

- **setDescription()** defines the label the input will have when displaying it in the plug-in, for example "Propeller Diameter".
- **setName()** defines a short name that can be used within the plug-in for input management, for example "prop_diam".
- **setUnit()** defines the unit displayed in the plug-in, for example "m".
- **setNumberOfDecimals()** defines the number of decimals the input value should be formatted with, for example 3.
- **setMaxValue()** defines the maximum allowable value this input may have, for example 10.0. The plug-in will not allow input values exceeding this value.
- **setMinValue()** defines the minimum allowable value this input may have, for example 0.0. The plug-in will not allow input values lower than this value.
- **setDefaultValue()** defines the default value this input will have when displayed in a newly created run, for example 0.0. The default input value must be in the range between the minimum value and the maximum value.



The description of the input has now been defined. In addition we must edit the **setStaticInputDataShipXDP()** method in order to store the input when it is transmitted from the plug-in to this class. The order of the input will always be kept, so that for example the propeller diameter described as item 7 in the **getStaticInputDescriptionShipXDP()** method above will also be the seventh item when the data are transmitted from the plug-in. In the **setStaticInputDataShipXDP()** we then simply extend the case sentences to match the order of the input we have added. As any propulsor model may have a maximum of 26 inputs the input range from 27 and upwards are reserved for application specific input.
Note that the Java language uses zero as base, so for example item number 7 will have an array index of 6.

13.6.10 Step 10 - Define The Propulsor Model

The propulsor thrust calculations are performed in the **performCalculationShipXDP (int mode)** method. The function of this method is twofold, governed by the "mode" input flag.

- Mode = 1. It will return the maximum- and minimum thrust the propulsor can produce for a given propulsor angle.
- Mode = 0. It will return the propulsor settings required for delivering the requested thrust and thrust direction.

An iteration routine has been implemented and can be freely used in new projects. If no changes are made to the model input the only change required to define the propulsor is to edit the **calculateThrust(float propellerRevs)** method. This method now utilizes the input defined earlier to calculate a thrust. An external call to a dll or class or whatever solution you prefer can be implemented here.

13.6.11 Step 11 - Build The Project

Select **MyPropulsor** from the **Run** - **Set Main Project** menu. Select **Build Main Project** from the **Run** menu (or press F11) to build the project. The “MyPropulsor.jar” file is now automatically created and placed in the “C:\Example\MyPropulsor\dist” folder.

13.6.12 Step 12 - Prepare For Use In ShipX

Perform the following two steps to prepare your project for use in ShipX.

1. Copy the file “C:\Example\MyPropulsor\dist\MyPropulsor.jar” to the folder “C:\Program Files (x86)\ShipX\lib”.
2. Open the file “C:\Program Files (x86)\ShipX\lib\propulsors.xml” in a text editor (for example Notepad). Add the following line to this file (just above the </propulsionlibrary> line):
   <includejar filename="C:\Program Files (x86)\ShipX\lib\MyPropulsor.jar" />
3. Save the file.

13.6.13 Step 13 - Test In ShipX

If you have successfully performed the previous steps, you are now ready to test your propulsor model in ShipX. Start ShipX, tight-click a ship, select **New** – **Static Propulsor** – **Propulsion Library Propulsor**. The dropdown list will now display the propulsor you just created.

Select the propulsor and click **Finish**. In the propulsor input window that appears, you get all the input you defined (except positional input which is automatically handled by the Propulsor Configuration). Click **Propulsor Info** to get the info window. The info you defined in chapter 13.6.6 should be displayed along with the version number, unique ID etc.

The propulsor model is now ready to be used.

13.7 Define A Propulsor For Use In Other Applications

If you want to use the propulsor in other applications than the ShipX Station Keeping Plug-In you must use the other methods found in the class described in this chapter. The other methods are documented in the class itself and might be self-explanatory. Contact SINTEF Ocean if you have problems with the implementation.

14 THEORY AND METHODS

This chapter will briefly describe the theory that this application is based on. The basic purpose of the program is to balance the external environmental forces with the defined onboard propulsors. The user defines the propulsor characteristics, while the environmental forces are defined by a limited number of inputs. The following chapters describe how these limited number of inputs result in environmental forces.

14.1 Wind Forces

The static wind forces and moments are calculated as follows:

Surge: $$F_x = C_x(\alpha)\cdot\frac{1}{2}\cdot\rho_{air}\cdot U^2\cdot A_{transverse}$$

Sway: $$F_y = C_y(\alpha)\cdot\frac{1}{2}\cdot\rho_{air}\cdot U^2\cdot A_{lateral}$$

Yaw: $$M_n = C_n(\alpha)\cdot\frac{1}{2}\cdot\rho_{air}\cdot U^2\cdot A_{lateral}\cdot L_{OA}$$

The symbols are defined in the following table.

Symbol	Description	Unit	Input
F_x	Wind force in surge for the relative wind heading.	N	No
F_y	Wind force in sway for the relative wind heading.	N	No
M_n	Wind moment in yaw for the relative wind heading.	Nm	No
C_x	Wind coefficient in surge for the relative wind heading.	-	Yes
C_y	Wind coefficient in sway for the relative wind heading.	-	Yes
C_n	Wind coefficient in yaw for the relative wind heading.	-	Yes
α	Wind direction relative to ship heading.	deg	Yes
U	Wind velocity.	m/s	Yes
ρ_{air}	Density of air. Defined to be 1.225.	kg/m^3	No
$A_{transverse}$	Transverse area of ship superstructure.	m^2	Yes
$A_{lateral}$	Lateral area of ship superstructure.	m^2	Yes
L_{OA}	Length overall. Defined in the ShipX workbench.	m	No

Wind coefficients are gathered from reference [1].

14.2 Current Forces

The static current forces and moments are calculated as follows:

Surge: $$F_x = C_x(\alpha) \cdot \frac{1}{2} \cdot \rho \cdot U^2 \cdot B \cdot T$$

Sway: $$F_y = C_y(\alpha) \cdot \frac{1}{2} \cdot \rho \cdot U^2 \cdot L_{PP} \cdot T$$

Yaw: $$M_n = C_n(\alpha) \cdot \frac{1}{2} \cdot \rho \cdot U^2 \cdot L_{PP}^2 \cdot T$$

The symbols are defined in the following table.

Symbol	Description	Unit	Input
F_x	Current force in surge for the relative current heading.	N	No
F_y	Current force in sway for the relative current heading.	N	No
M_n	Current moment in yaw for the relative current heading.	Nm	No
C_x	Current coefficient in surge for the relative current heading.	-	Yes
C_y	Current coefficient in sway for the relative current heading.	-	Yes
C_n	Current coefficient in yaw for the relative current heading.	-	Yes
α	Current direction relative to ship heading.	deg	Yes
U	Current velocity.	m/s	Yes
ρ	Density of water. Defined in the ShipX workbench.	kg/m^3	No
B	Beam at midships. Defined in the ShipX workbench.	m	No
T	Draught at midships. Defined in the ShipX workbench.	m	No
L_{PP}	Length between ship perpendiculars. Defined in the ShipX workbench.	m	No

The wind generated current velocity vector and the tidal generated current velocity vector are added and the resultant velocity vector is used when calculating current forces.
Current coefficients are gathered from reference [2].

14.3 Wave Forces

The static wave drift forces and moments are calculated as follows:

Surge: $$F_x = 2 \cdot \frac{\rho \cdot g \cdot B^2}{L_{PP}} \cdot \int_0^\infty S(\omega) \cdot C_x(\omega,\alpha) \cdot d\omega$$

Sway: $$F_y = 2 \cdot \frac{\rho \cdot g \cdot B^2}{L_{PP}} \cdot \int_0^\infty S(\omega) \cdot C_y(\omega,\alpha) \cdot d\omega$$

Yaw: $$M_n = 2 \cdot \rho \cdot g \cdot B^2 \cdot \int_0^\infty S(\omega) \cdot C_n(\omega,\alpha) \cdot d\omega$$

Where

$$C_x(\omega,\alpha) = \frac{\overline{F}_x(\omega,\alpha)}{\dfrac{\rho \cdot g \cdot B^2 \cdot \zeta^2}{L_{PP}}}$$

$$C_y(\omega,\alpha) = \frac{\overline{F}_y(\omega,\alpha)}{\dfrac{\rho \cdot g \cdot B^2 \cdot \zeta^2}{L_{PP}}}$$

$$C_n(\omega,\alpha) = \frac{\overline{M}_n(\omega,\alpha)}{\rho \cdot g \cdot B^2 \cdot \zeta^2}$$

The symbols are defined in the following table.

Symbol	Description	Unit	Input
F_x	Wave drift force in surge for the relative wave heading.	N	No
F_y	Wave drift force in sway for the relative wave heading.	N	No
M_n	Wave drift moment in yaw for the relative wave heading.	Nm	No
C_x	Wave drift coefficient in surge for the relative wave heading.	-	Yes
C_y	Wave drift coefficient in sway for the relative wave heading.	-	Yes
C_n	Wave drift coefficient in yaw for the relative wave heading.	-	Yes
α	Wave direction relative to ship heading.	deg	Yes
S	Wave spectrum.	-	Yes
ρ	Density of water. Defined in the ShipX workbench.	kg/m^3	No
g	Gravity. Defined in the ShipX workbench.	m/s^2	No
ζ	Wave amplitude.	m	Yes
B	Beam at midships. Defined in the ShipX workbench.	m	No
L_{PP}	Length between ship perpendiculars. Defined in the ShipX workbench.	m	No

It is important to notice the following if using a wave drift coefficient file from a similar ship:

- These coefficients are only valid for ships with same length over beam ratio.
- Small changes to the ship length over beam ratio by changing the ship beam should be taken care of by the squared beam term in the scaling of the resulting coefficients.
- Small changes to the ship length over beam ratio by changing the ship length will cause a change in the transfer functions as the changed ship length will shift the ship to different relative wave frequencies in the transfer functions. This change is taken care of by the transfer functions. This means that the wave drift coefficients will be different, and one cannot look at the formulas above to easily find how the force will scale.
- Small changes to the ship length over beam ratio may be acceptable. If your ships length over beam ratio differs from the ratio in the wave drift coefficient file you should consider performing a proper calculation of the wave drift coefficients by use of, for example, the ShipX Ship Responses Plug-In.
- Scaling a ship, keeping the length over beam ratio constant, is taken care of by scaling the wave frequencies in the coefficient file to the correct ship length.

All the changes outlined above are performed automatically by the Plug-In.

14.4 Allocation

The main allocation philosophy and theory is described in this chapter. The text refers to azimuth thrusters and tunnel thrusters. In the implementation this is generalized to propulsors. A tunnel thruster is for example a general propulsor with a fixed thrust angle of 90 degrees.

14.4.1 Configuration

Let the commanded forces in surge and sway be denoted τ_x and τ_y and the commanded moment in yaw be denoted τ_N. These are stacked in a vector of commanded generalized forces $\tau = (\tau_x, \tau_y, \tau_N)^{\mathrm{T}}$. The allocation problem for a ship with *s* azimuth thrusters and *r* tunnel thrusters can be formulated as the calculation of pairs (α, f) satisfying

$$\tau = C(\alpha) f$$

where $\alpha \in \Re^{s+r}$ denotes the orientation of the actuators, $f \in \Re^{s+r}$ is a vector of control forces, and $C(\alpha) \in \Re^{3\times(s+r)}$ is a configuration matrix with the *i*-th column given by

$$C_i(\alpha) = \begin{bmatrix} \cos\alpha_i \\ \sin\alpha_i \\ -l_y^i \cos\alpha_i + l_x^i \sin\alpha_i \end{bmatrix}$$

The location of the *i*-th thruster is at (l_x^i, l_y^i).

14.4.2 Optimization Criterion

Minimize with respect to (α, f, δ) the criterion

$$J(\alpha, f, \delta) = \sum_{i=1}^{N} W_i |f_i|^{3/2} + \delta^{\mathrm{T}} \Omega \delta + (\alpha - \alpha_0)^{\mathrm{T}} Q (\alpha - \alpha_0)$$

subject to the following constraints

$$\begin{aligned} &\delta = \tau - T(\alpha) f \\ &f_{\min} \le f \le f_{\max} \\ &\alpha_{\min} \le \alpha \le \alpha_{\max} \\ &\Delta\alpha_{\min} \le \alpha - \alpha_0 \le \Delta\alpha_{\max} \end{aligned}$$

One objective is to minimize power consumption, represented by the first term in the criterion. The second term penalizes the error δ between the commanded and achieved force. This is necessary in order to guarantee that the optimization problem has a feasible solution for any τ. The tuning weights in matrix $\Omega > 0$ is chosen such that the solution $\delta \approx 0$ whenever possible.

The control allocation for ships equipped with azimuth thrusters is in general a non-linear optimization problem that is non-convex and hard to solve. If we require the azimuth angles α to change slowly from one sample to the next, a linear approximation to the constraints given above will be reasonable. The non-convex nonlinear program may therefore be replaced by a convex quadratic program that can be solved with high efficiency and reliability, without much loss of performance.

14.4.3 Quadratic Programming (QP) Reformulation

In order to get a convex quadratic approximation, we approximate the first term (power consumption) by a quadratic term, and we let $\alpha = \alpha_0 + \Delta\alpha$:

$$J(\alpha, f, \delta) = (f_0 + \Delta f)^{\mathrm{T}} W_0 (f_0 + \Delta f) + \delta^{\mathrm{T}} \Omega \delta + \Delta\alpha^{\mathrm{T}} Q \Delta\alpha$$

subject to

$$\delta + T(\alpha_0)\Delta f + \frac{\partial}{\partial\alpha}(T(\alpha) f)_{\alpha_0, f_0} \Delta\alpha = \tau - T(\alpha_0) f_0$$
$$f_{\min} - f_0 \le \Delta f \le f_{\max} - f_0$$
$$\alpha_{\min} - \alpha_0 \le \Delta\alpha \le \alpha_{\max} - \alpha_0$$
$$\Delta\alpha_{\min} \le \Delta\alpha \le \Delta\alpha_{\max}$$

If thruster *i* is not operative one may require $f^i_{\min} = f^i_{\max} = 0$, and for a tunnel thruster one can set $\alpha^i_{\min} = \alpha^i_{\max} = 90$ degrees.

14.4.4 Static Control Allocation

For a static control allocation problem, the QP problem defined in the previous section is solved a finite number of steps. In the first step we have to specify initial angels α_{init}, and set $\alpha_0 = \alpha_{init}$. For the next steps α_0 is equal to the angles at the previous step. An algorithm to choose initial angles is presented in the following.

Consider a ship with *s* azimuth thrusters and *r* tunnel thrusters. Define the extended thrust vector by

$$t = \left[t_x^1, t_y^1, \ldots, t_x^s, t_y^s, t_y^{s+1}, \ldots, t_y^{s+r}\right]^{\mathrm{T}}$$

where the first *2s* elements correspond to the azimuths *x*- and *y*-forces. The last *r* elements correspond to the tunnel thrusters *y*-forces. The thrust vector τ is given by

$$\tau = AWt$$

where $\tau = (f_x, f_y, M)$, and $W(\alpha_c) \in \Re^{2s+r \times 2s+r}$ is a positive semi definite diagonal weight matrix. $A \in \Re^{3 \times 2s+r}$ is given by

$$A = \begin{bmatrix} 1 & 0 & \cdot & 1 & 0 & \cdot & 0 \\ 0 & 1 & \cdot & 0 & 1 & \cdot & 1 \\ -l_y^1 & l_x^1 & \cdot & -l_y^s & l_x^s & \cdot & l_x^r \end{bmatrix}$$

For azimuth *i*, the initial angle $\alpha_{init,i}$, is given by

$$\alpha_{init,i} = \mathrm{atan2}\left(t_i^y, t_i^x\right), \quad \mathrm{i} = 1,\ldots,\mathrm{s}.$$
$$\alpha_{init,i} = 90, \quad \mathrm{i} = \mathrm{s}+1,\ldots,\mathrm{s}+\mathrm{r}.$$

If this algorithm orders a forbidden angle, the closest zone limit is selected.

15 ASSUMPTIONS AND SIMPLIFICATIONS

This chapter contains simplifications and assumptions used in this ShipX plug-in.

15.1 Assumptions

In the internal allocation routines, the following assumptions have been made (the list may not be complete):

- If propulsors of 'rudder and propeller' type are used, they must be positioned as propulsor number 1 and 2 (if applicable). This restriction does not apply to any other propulsor type.
- Absolute value of the minimum force must not exceed the absolute value of the maximum force for any propulsor.

The user may freely disable any propulsors and may also freely limit the working angles of the propulsors.

15.2 Simplifications

The following simplifications are made (the list may not be complete):

- The theory behind the thrust losses caused by propulsor interaction with other propulsors assumes that the thrust is directed along the same axis. Most of the time this will not be true but based on estimated reasoning the results are valid approximations. The method assumes that the propulsors are in the same horizontal plane, i.e. the method is 2 dimensional. A correction of the thrust loss is performed based on the vertical overlap of the propulsors in question. Any thrust angle relative to the horizontal plane is currently disregarded.
- The theory behind propulsor interaction with the hull assumes that the propeller race travels along a flat plate (defined by the water plane area of the hull). Skegs and other “obtrusions” are not used in the calculations. If no hull geometry file is available, the water plane is estimated by use of the length and beam of the ship.

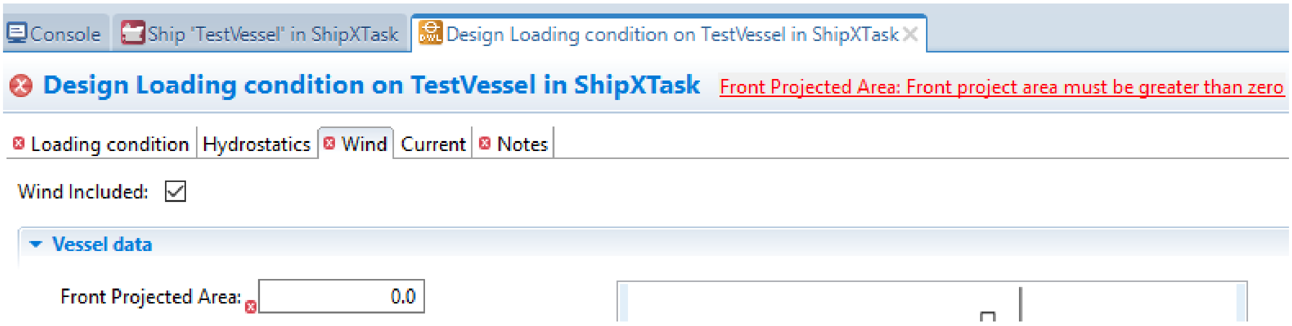
The allocation routines may return a seemingly reasonable answer even if these guidelines are not met, but the results will need further inspection.

16 TIPS AND TRICKS

This chapter contains tips and tricks for easing the use of this ShipX plug-in.

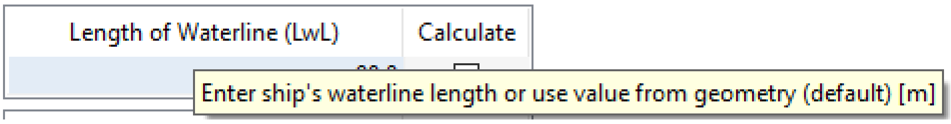
16.1 Input Error Check

ShipX automatically investigates the input you've given and notifies you if the input is wrong and/or missing. On top of the input window you will find a summary of the input errors, and each input error will also be marked with a red cross. See below image indicating an input error for the wind **Front Projected Area**.



16.2 Tooltips

If you point the mouse cursor on objects in the user interface (and wait a second) you will see a tooltip (as in the image below) explaining more about the object you are pointing at.



16.3 Find Input In Navigator

It may often be difficult and/or tedious to find the location of the input you are working on in the Navigator tree view. If you want to open the Navigator tree view to the location where the plug-in you are editing is located, press the purple square in the plug-in top right corner:

This will expand the Navigator tree view and select the appropriate run for you.

16.4 Auto-Save

You can have ShipX automatically save your project at certain intervals. To change this behaviour, go to the **Tools** – **Preferences** menu, click **General** and locate the **Minutes between automatic save** setting. Change the value to your liking.

16.5 Manually Open mpl-files

When you double-click an mpl file (report/plot) inside the **Navigator** window in ShipX, ShipX will make sure that the ShipX Plot Program is started (if it wasn't already) and will open the selected result file in it. If you want to use the Windows Explorer to open mpl-files, you will need to tell Windows which program to use, i.e. associate the mpl file the with the ShipX Plot Program. This is how you do it:

- In Windows Explorer, double-click an mpl-file
- In the dialog that appears, asking you which program to use to open the file, click **More apps,** scroll down to the bottom of the list and select **Look for another app on this PC**
- Browse to this location: <ShipX install folder>\plugins\no.sintef.ocean.sxplot.ui_<version>\sxplot
- Select the **PlotX.exe** file, click **Open**
- Select the **Always use this app to open .mpl files** checkbox and then click **OK**

In some instances (unknown for what reason), even when following above procedure, the ShipX Plot Program is not started by Windows when double-clicking an mpl file. The trick is then to manually start the ShipX Plot Program and then continue with manually opening the mpl-files by double-clicking them.

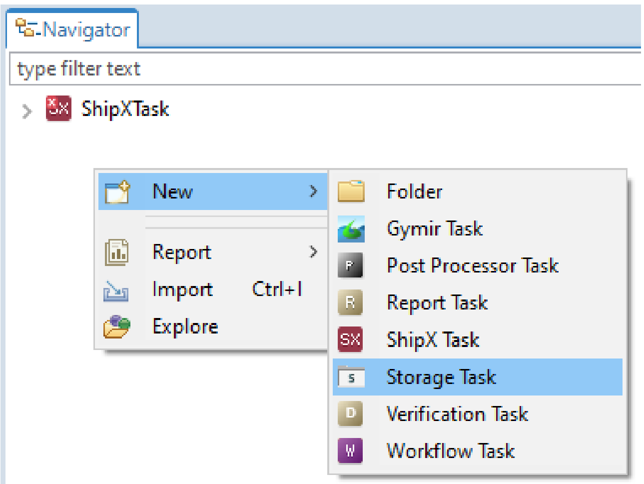
16.6 Create a Storage Task for The Wave Drift Coefficient Files

The Station Keeping distribution comes with a set of wave drift coefficient files. These files are in a subfolder deep inside the folder structure. It is highly recommended to make a storage task for these files, because:

- It is much easier to locate files within a storage task than having to browse the folder structure every time you need the file.
- A storage task is a "shortcut" to a folder, so every file inside the defined folder will easily be available in browse dialogs (see below for further explanation)
- Files within a storage task will by default be exported along with any run you export, so the recipient of the exported run will be able to use the same file from the storage task as you do.

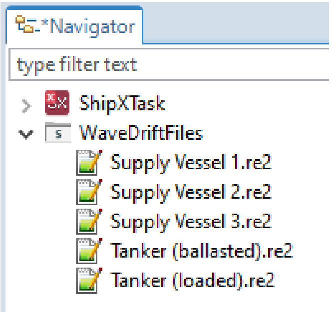
Let us set up a storage task for the built-in wave drift coefficient files:

- Right-click the Navigator window
- Select **New** – **Storage Task**



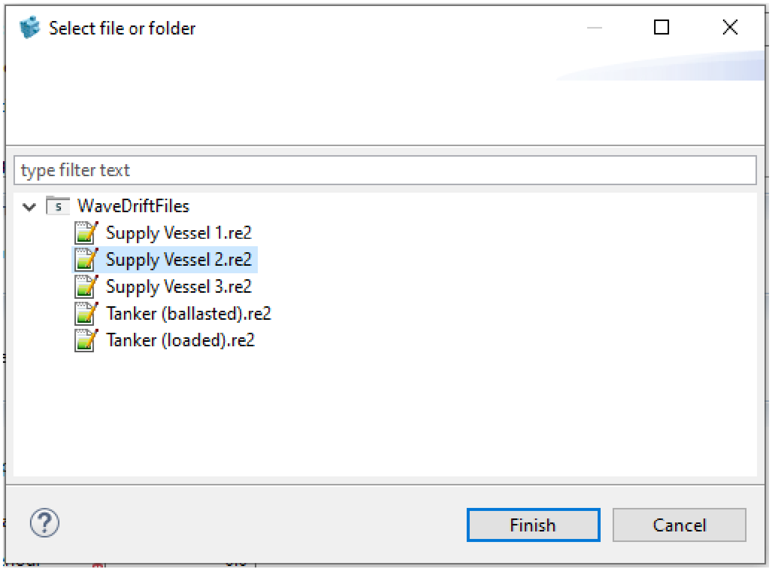
SINTEF

- In the wizard that appears, give the task a name (f.ex WaveDriftFiles) and click **Finish**.
- Double-click the WaveDriftFiles storage task in the **Navigator** window
- For the **Root** input, click **Browse** and locate the path for the built-in wave coefficient drift files (typically located at "<install folder>\plugins\no.sintef.ocean.stationkeeping.resources_nnn\re2", where nnn is the actual version of the plug-in release)
- Leave the **Include files when exporting** option checked
- You have now defined this storage task and you can close the storage task window

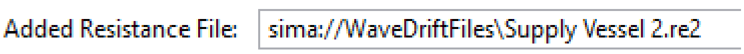


When you need either of the wave drift coefficient files as input, simply click the **Refer to local storage** button:

And then select the correct file from the storage task you just defined:



You will the see that the file path defined as input will be set to a relative path inside the workbench:



That is all there is to it. The storage task files can be used throughout the workspace and will be exported along with all the other input if need be, ensuring the recipient of the data will be able to run calculations using the same input files as you have.

Note that you can create multiple storage tasks and of course also include your own files into them, thus creating your own small database of resources.

16.7 Quickly Jump To Objects

Many places the input is defined using dropdown boxes, for example when defining a run to use, a propulsor configuration to use, a power configuration to use etc. It is very easy to to jump directly into that object. Simple press the Ctrl-button on your keyboard and while keeping it pressed, point to the object in question (the mouse pointer changes into a finger pointing) and click on it. The object will now open in the editor.

17 FILE FORMATS

This chapter contains a description of the file formats used for importing data into the plug-in.

17.1 Wind Coefficients File

The wind coefficients file is of file type *.skw. The file is in plain ASCII and is formatted as described below.

Text info
Ship Info
N *:Number of headings*
Heading_1 WindCoeffX_1 WindCoeffY_1 WindCoeffN_1
.
.
Heading_N WindCoeffX_N WindCoeffY_N WindCoeffN_N

The wind coefficients are as defined in chapter 14.1. The file must contain 19 headings, where Heading_1 is 0 deg (head wind) and Heading_N is 180 deg (stern wind). The heading step should be 10 deg. All columns must be separated with spaces (no tabulators).
Note that the definition of the skw file is somewhat changed since the old version of ShipX. Both file versions are supported in this version of ShipX.

17.2 Current Coefficients File

The current coefficients file is of file type *.skc. The file is in plain ASCII and is formatted as described below.

Text info
Ship Info
N *:Number of headings*
Heading_1 CurrentCoeffX_1 CurrentCoeffY_1 CurrentCoeffN_1
.
.
Heading_N CurrentCoeffX_N CurrentCoeffY_N CurrentCoeffN_N

The current coefficients are as defined in chapter 0. The file must contain 19 headings, where Heading1 is 0 deg (stern current) and Heading_N is 180 deg. The heading step should be 10 deg. All columns must be separated with spaces (no tabulators).

17.3 Wave Drift Coefficients File

The wave coefficients file is of file type *.re2 (as exported from ShipX Ship Responses Plug-In) so the coordinate system in this file differs from the rest of the input files. The coordinate system of this file is located at CG, with positive X-axis astern, positive Y-axis to starboard and positive Z-axis up. Wave heading is zero degrees for head sea, 90 degrees for beam seas from port and so on. The file is in plain ASCII and is formatted as described below.

Text info
Ship Info
Text2
Text3
Text4
Text5
WaterDensity Gravity [WaterDepth]
Lpp Beam Draught
nSpeeds nHeadings nFrequencies [nCurrentVel] [nCurrentDir]
[Loop CurrentVelocities]
 [Loop CurrentDirections]
  Loop Speeds
   Loop Headings
    Speed Heading [CurrentVelocity] [CurrentDirection]
    Loop Frequencies
     Frequency WaveCoeffX WaveCoeffY WaveCoeffN
    End Loop Frequencies
   End Loop Headings
  End Loop Speeds
 [End Loop CurrentDirections]
[End Loop CurrentVelocities]

The wave coefficients are as defined in chapter 0. Data in brackets [] are part of the new file format and are optional. They may not be read by all software.

17.4 Thrust Restriction Zones File

The thrust restriction zones file is of file type *.trz. The file is in plain ASCII and is formatted as described below.

Text info
Ship Info
Number of restricted zones
Loop Zones
 Start Stop MaxThrust Dependency
End Loop Zones

All columns must be separated with spaces (no tabulators). To make sure you have the correct format of the file, export a file from the program and edit it afterwards.

17.5 Wind-Wave Relation File

The wind-wave relation input file is of file type *.windwave. The file is in plain ASCII and is formatted as described below.

File version
Text info
Ship Info
N   *:Number of rows*
$Wind\_vel_1$ $Wave\_Hs_1$ $Wave\_Tp_1$ $PnonExc_1$
.
.
$Wind\_vel_N$ $Wave\_Hs_N$ $Wave\_Tp_N$ $PnonExc_N$

The PNonExc is the probability of non-exceedance of the environmental state for that row. This value is used, amongst others, when calculating SKP numbers. If no PNonExc data is available, just set the value to 0.0. This means you will not be able to use the wind-wave relation file for calculations where the PNonExc value is required. All columns must be separated with spaces (no tabulators). To make sure you have the correct format of the file, export a file from the program and edit it afterwards.

17.6 Post-Processing Output Files

The Station Keeping plug-in may generate output files for use with specific third party tools. These files are not for the common ShipX user and can generally be neglected. For the purpose of documenting these file formats for the relevant users, the file formats are described in the following subchapters.

17.6.1 pp_power.txt

This file contains a summary of the environment and the propulsors power used for each environment direction.

File version
Text info
N *:Number of rows*
P *:Number of propulsors*
H_s T_p V_w V_c dir $prop1_1$ $prop2_1$ … $propP_1$
.
.
H_s T_p V_w V_c dir $prop1_N$ $prop2_N$ … $propP_N$

Where

H_s = significant wave height [m]
T_p = wave peak period [s]
V_w = wind velocity [m/s]
V_c = current velocity [m/s]
dir = environment direction [deg]
$propi_n$ = propeller power [kW]

18 EXAMPLES

This chapter will walk you through creating Station Keeping runs from scratch. If you follow these steps, you will be up and running in no time. This chapter is divided into multiple sub-chapters. The first sub-chapter will guide you through creating a ship from scratch. The subsequent chapters will walk you through creating Station Keeping runs for different types of calculations.

18.1 Create A Ship In ShipX

In order to perform a Station Keeping calculation you must first define your ship and then define the Station Keeping input. This chapter deals with creating your ship and giving the required hull input needed for a Station Keeping run. There are several ways to create your ship in ShipX; normally it is done by importing a geometry file. The Station Keeping plug-in does, however, not require a geometry file; it only requires the proper main dimensions to scale the environmental force coefficients into forces and moments. In this chapter we will guide you through creating a ship in ShipX without using a geometry file.

If you haven't created a workspace yet, follow the instructions in chapter 3.2 to do so.
The procedure for creating a ship is outlined in detail in chapter 3.3. Follow the instructions found therein.

When we have finished any such "milestone" with regards to input, we make a habit of saving the input so we won't need to do the work again should something bad happen:

- Save the database input by using the **File** - **Save** menu. Alternatively, you can use the shortcut Ctrl + S or click the leftmost button in the button row in ShipX.

We have now input all the main dimensions of the ship needed to perform a Station Keeping calculation. We now need to create a Station Keeping run to define the type of calculation we are to perform, the propulsor input and the environmental input. Continue with the appropriate sub-chapter below, depending on what type of calculation you want to perform.

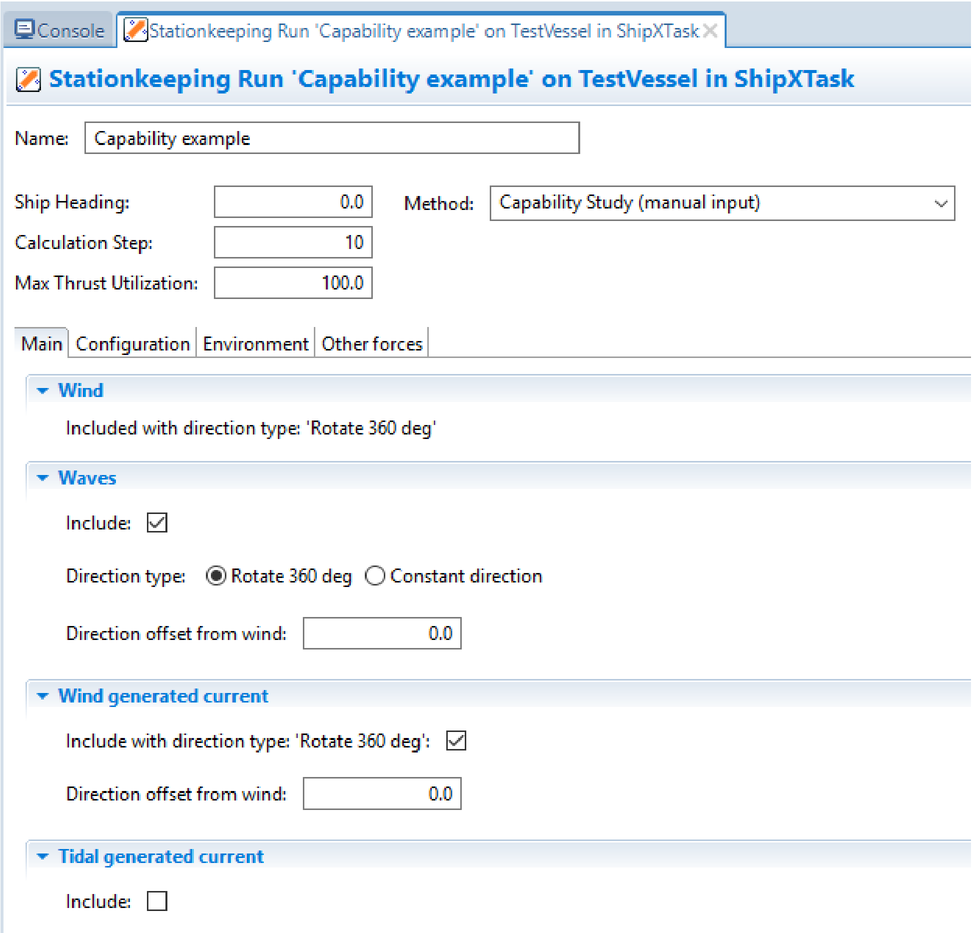
SINTEF

18.2 Capability Calculation

In order to perform a capability calculation, the input must be defined similar to how it is done in this chapter. Before continuing in this chapter, make sure you have created the ship according to the instructions in chapter 18.1. Use the following guide to create a capability calculation from scratch:

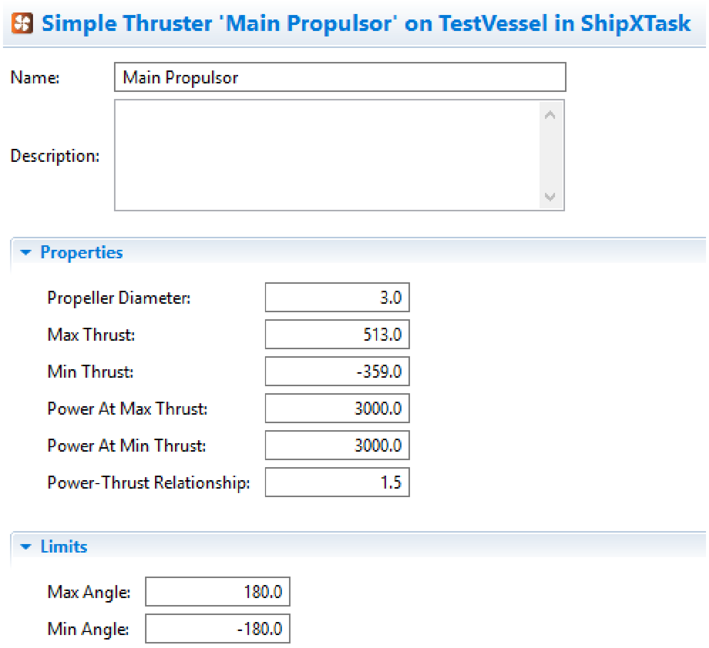
1) Start the Station Keeping Plug-In by right-clicking the **Runs** folder found under the loading condition of your vessel, select **New** and then **Stationkeeping Run**.

2) In the plug-in window we must enter the name of the run and select the correct calculation method. In this calculation we will include co-linear wind-, wave- and current effects, so these inputs must be selected and be set to rotate:



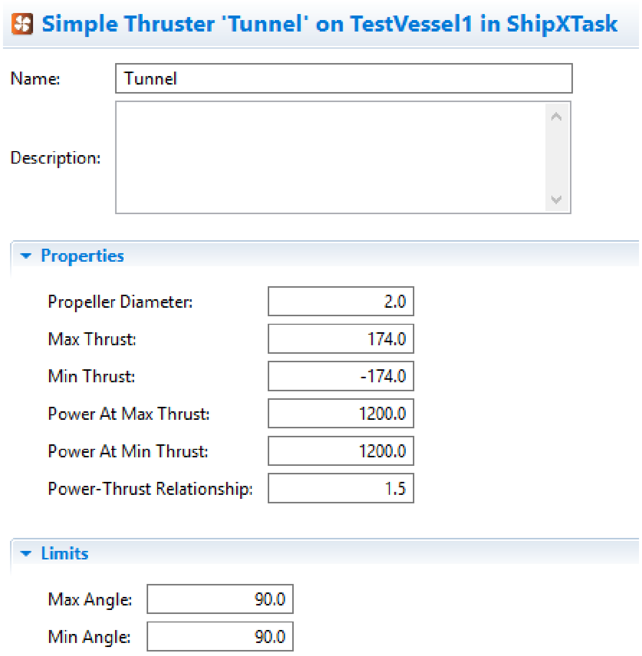
3) Moving on to the *Configuration* tab. Here we must define our propulsor configuration. In order to create a propulsor configuration, we need to create some propulsors first. The propulsors can be defined using simple thruster models or by using more advanced models. For simplicity we will use the "Simple Thruster" model to define our propulsors in this example. We are going to add 4 propulsors, two main propulsors units and two tunnel thrusters in the bow. Since both main propulsors are the same, and both tunnel thrusters are the same, we only need to create one of each and then use them as basis in the propulsion configuration:

a. Right-click the ship, select **New** – **Static Propulsor** – **Simple Thruster**
In the simple thruster input dialog that appears, enter the following input:

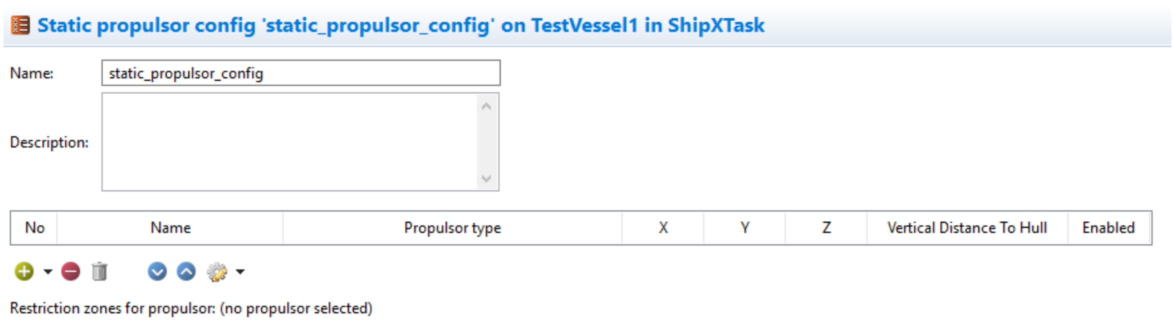


For some propulsors the thrust and power input can also be defined by using the *Tune* button. Try it and see how easy it is to estimate a thrust based on the available power.

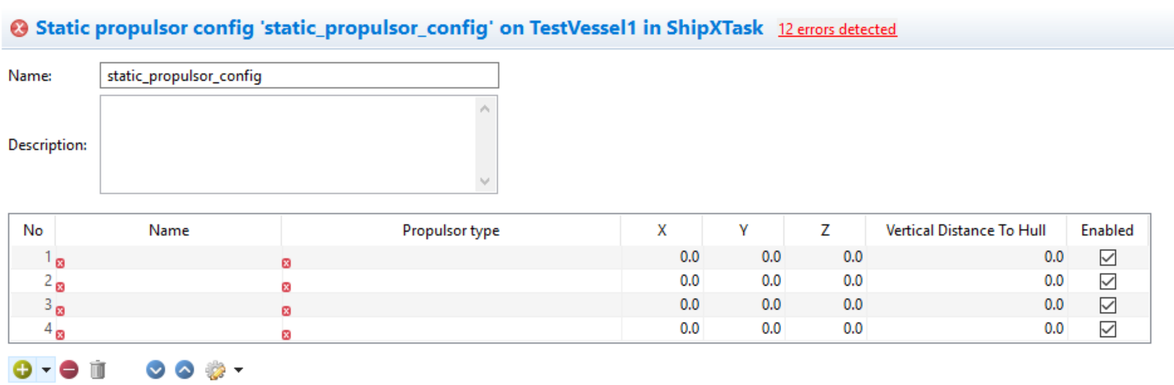
b. Next we will make a tunnel thruster. In order to make a tunnel thruster out of a simple thruster model, the rotation must be locked to 90 degrees. Right-click the ship, select **New** – **Static Propulsor** – **Simple Thruster**
In the simple thruster input dialog that appears, enter the following input:



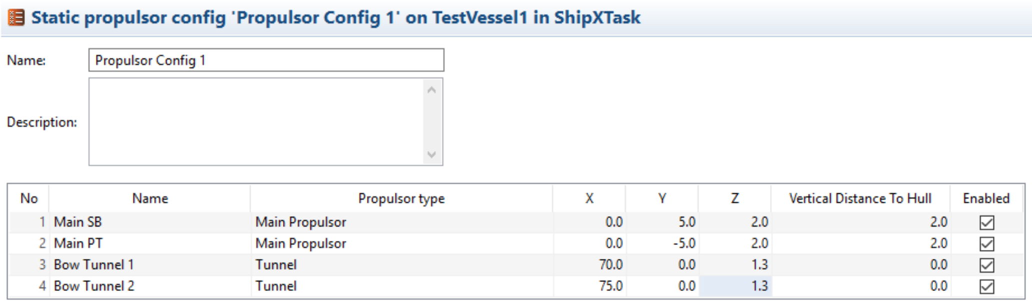
c. We have now created the two propulsor types we need. It is now time to build the propulsor configuration. To create a new propulsor configuration, right click the ship, select **New – Static propulsor config**:



To populate this propulsor configuration with the four propulsors we need, click the bottom left + sign four times:



For each of the rows, type in the propulsor name, select the propulsor type and enter the position data. The result should look like this:

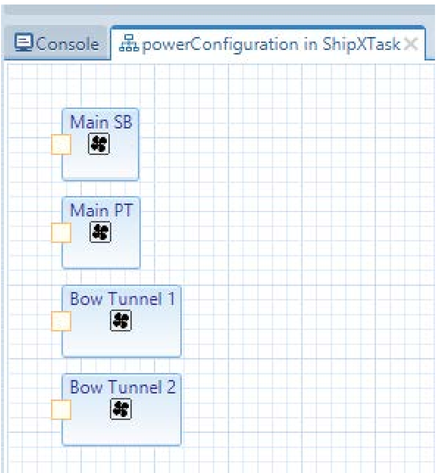


No	Name	Propulsor type	X	Y	Z	Vertical Distance To Hull	Enabled
1	Main SB	Main Propulsor	0.0	5.0	2.0	2.0	☑
2	Main PT	Main Propulsor	0.0	-5.0	2.0	2.0	☑
3	Bow Tunnel 1	Tunnel	70.0	0.0	1.3	0.0	☑
4	Bow Tunnel 2	Tunnel	75.0	0.0	1.3	0.0	☑

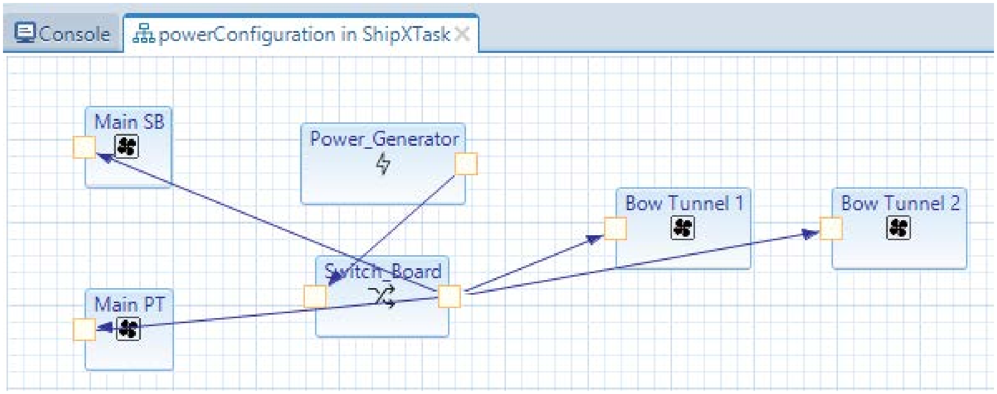
d. We have now created the propulsors we need and also created the propulsor configuration to use. The last step is to set the propulsor configuration input in the station keeping run to the one we just created:

4) The next thing we need to do is define a power configuration. The power configuration defines the available power in the system. In this case we are not interested in the power, that is, we want to run the calculation without power restrictions. Nevertheless, we need to build a power configuration. We will therefore build a minimum power configuration, making sure that all propulsors have enough power available to run as they like.

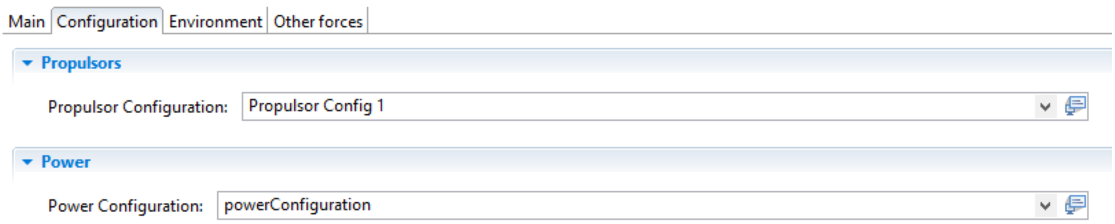
   a. Right-click the ship, select **New** – **Power Configuration.** Note that the power configuration input opens in the top window. It can be dragged and dropped into the bottom window if you want all input dialogs gathered there; simply drag and drop the power configuration tab to where you want it located.
   b. In the Navigator view, find the propulsor configuration and drag and drop it into the power configuration window.



c. The boxes can be reorganized as you like. See chapter 3.8 and chapter 8 for further details.
d. Add a power generator and a switch board, both with available power 10000 kW (to make sure there is enough power in the system for the propulsors to perform as they like. Connect all up as below:

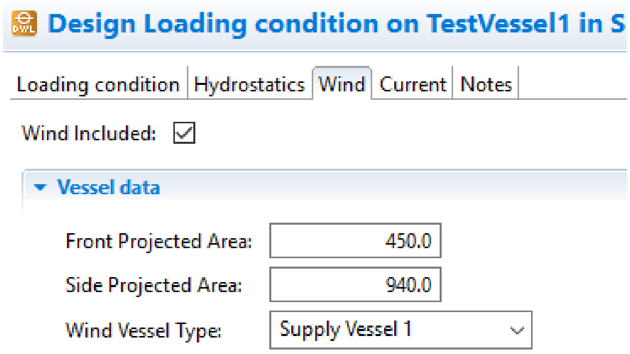


e. The last step is to define the power configuration input in the Station Keeping run:

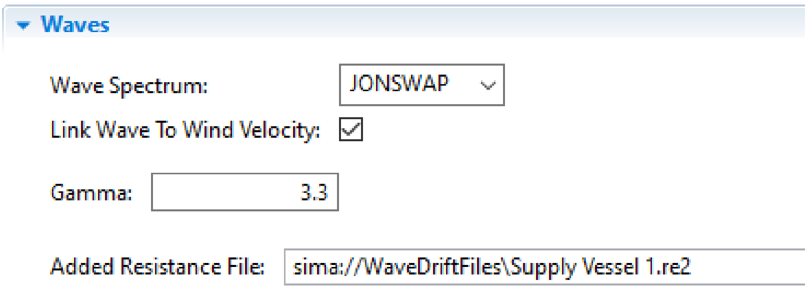


5) The only input left to be defined now is the environmental input. To define this, we click the *Environment* tab.

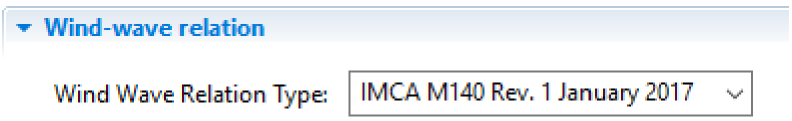
a. In the wind input section, note that we cannot define a wind velocity. This is of course because we are to perform a capability analysis, and the wind velocity will be iterated so we can find the highest wind velocity where the ship can hold position.
The front- and side projected wind areas of the superstructure along with the wind coefficients for the ship are defined in the loading condition input. Double-click the loading condition to inspect the input data, and make sure it is defined like this:



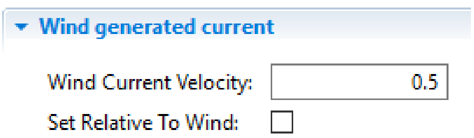
b. In the wave input section, we can select if we want the wave height and -period to be linked to the wind velocity, or if we want to define a fixed wave height and -period. In this case we will keep the waves linked to the wind speed. We must then define the relationship between the wind and the waves. We must also select a set of wave drift coefficients; we can use either an existing dataset or a dataset calculated specifically for our ship by, for example, the ShipX Vessel Responses Plug-In. In this case we will use an existing dataset, and we should select a dataset that most closely resembles our ship, both in ship type and in length over beam and beam over draught ratio.
ShipX comes with a predefined set of example re2-files. See chapter 16.6 for how to access them. Your input should look like this:



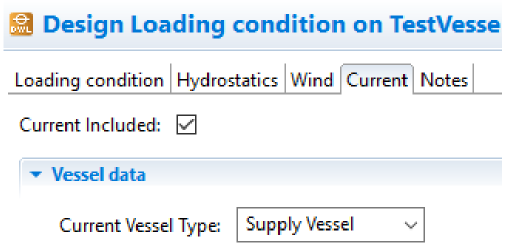
The wind-wave relation, defining how the waves will build up relative to the wind velocity should in this example be defined like this:



c. In the current input section, we can define the current velocity, or we can define that the current velocity should be a percentage value of the wind speed. We will set a fixed current velocity, like this:

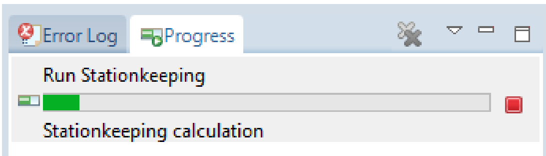


We must also define the current coefficients, this can be found in the loading condition for the ship. In this example we will use one of the built-in datasets for the current coefficients:

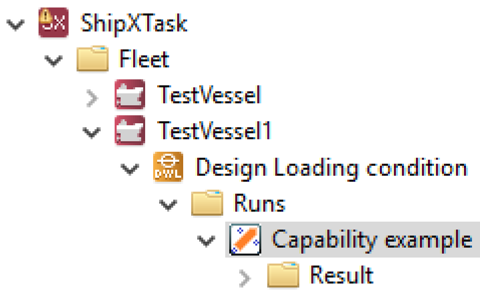


6) We have now defined all required input to perform a capability calculation. To start the calculation, right-click the Station Keeping run in the **Navigator** window and select **Run Stationkeeping**.

   a. The calculation will take a few seconds. While calculating you can see the progress in the bottom left **Progress** window of ShipX.



   b. When the calculation has finished all reports and plots will be created in the **Result** folder under the station keeping run in the **Navigator** window. You can expand the folder and double-click on the reports and plots you want to look at. You can also right click the folder, select **Explore** and look at the result folder in the file browser of your choice.



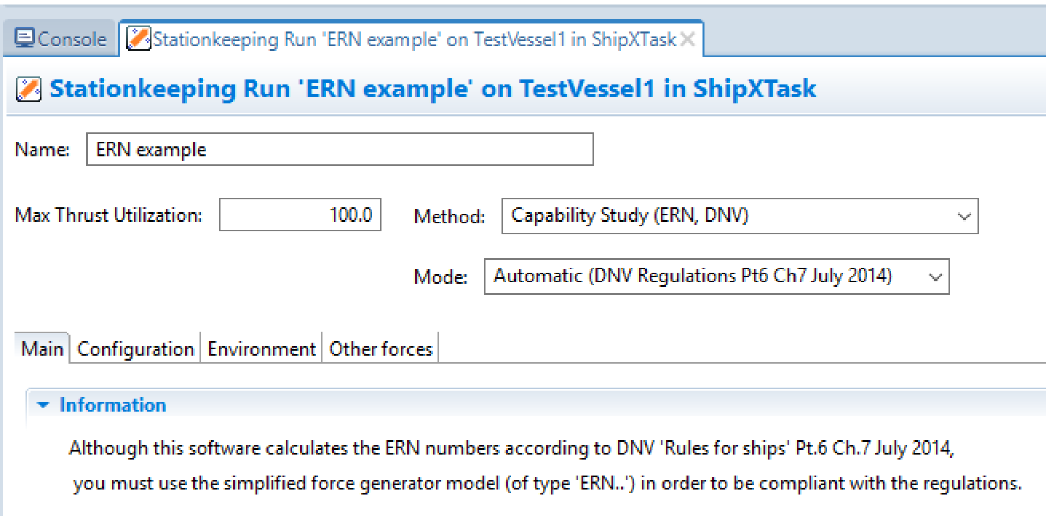
   c. After plotting the reports and plots you like you can export them from the *ShipX Plot Program* to, for example, *Microsoft Word* to include into a report.

18.3 ERN 2014 calculation

In order to perform an ERN station keeping calculation there are many considerations to take into account while creating the input and performing the calculations (see chapter 5.4 and chapter 5.4.5 for details). The Station Keeping Plug-In has been developed to aid the user as much as possible in order to minimize the chance of user errors.
Before continuing in this chapter, make sure you have created the ship according to the instructions in chapter 18.1. The following guidelines assume you have completed chapter 18.2, and will therefore be a little briefer in explaining all the steps. Use the following guide to create an ERN 2014 calculation from scratch:

1) Start the Station Keeping Plug-In by right-clicking the **Runs** folder found under the loading condition of your vessel, select **New** and then **Stationkeeping Run**.

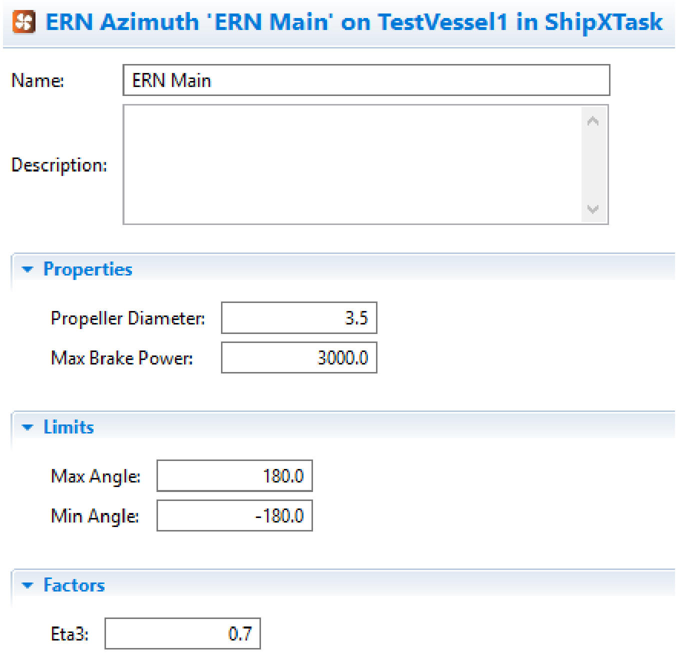
1) In the main window we must enter the name of the run and select the correct calculation method and -mode:



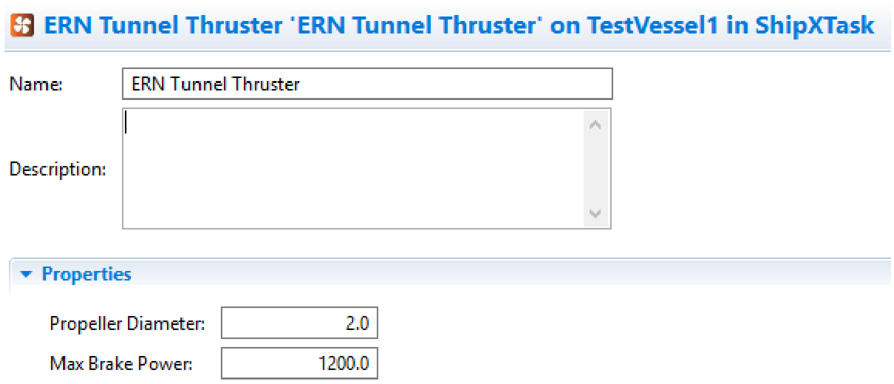
When the ERN simplified 2014 calculation mode has been selected, the user interface will change to only allow relevant input.

2) Moving on to the *Configuration* tab. Here we must define the propulsor configuration, hence we must first define our propulsor types. The ERN 2014 standards are strict about which propulsor types of can be used. SINTEF Ocean have implemented propulsors based on the DNV regulations [12]; they all have names starting with *ERN*. If you try to select any other thrusters, you will not be allowed to perform an ERN 2014 calculation. We are going to add 4 propulsors, two main propulsion units and two tunnel thrusters in the bow. As before, the two main propulsors are of same type, as are the two tunnel thrusters. We therefore create two new propulsors:

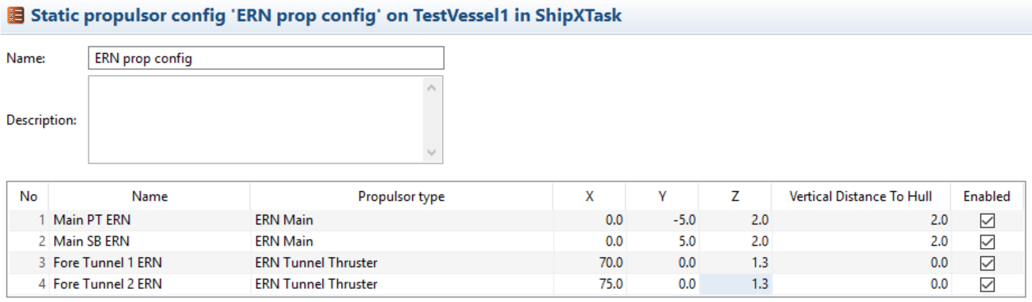
   a. Create the main propulsor type:
      - Right-click **Propulsors** in the **Navigator** window, select **New** – **Static Propulsor** – **ERN Azimuth**
      - Set the input like this:



b. Create the tunnel thruster propulsor type:
   - Right-click **Propulsors** in the **Navigator** window, select **New** – **Static Propulsor** – **ERN Tunnel Thruster**
   - Set the input like this:



c. We have now defined our propulsor types. Let's create the propulsor configuration:
   - Right-click **Propulsor Configurations** in the **Navigator** window, select **New** – **Static propulsor config**
   - Add four lines and add the four propulsors we want, like this:



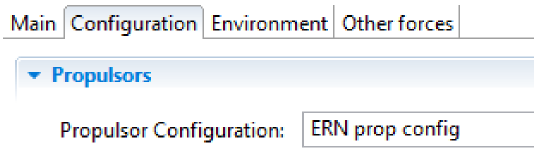
Static propulsor config 'ERN prop config' on TestVessel1 in ShipXTask

Name: ERN prop config

Description:

No	Name	Propulsor type	X	Y	Z	Vertical Distance To Hull	Enabled
1	Main PT ERN	ERN Main	0.0	-5.0	2.0	2.0	☑
2	Main SB ERN	ERN Main	0.0	5.0	2.0	2.0	☑
3	Fore Tunnel 1 ERN	ERN Tunnel Thruster	70.0	0.0	1.3	0.0	☑
4	Fore Tunnel 2 ERN	ERN Tunnel Thruster	75.0	0.0	1.3	0.0	☑

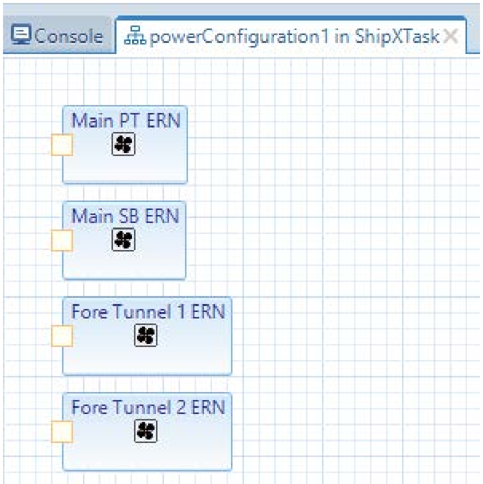
d. Define the propulsor configuration in the station keeping run



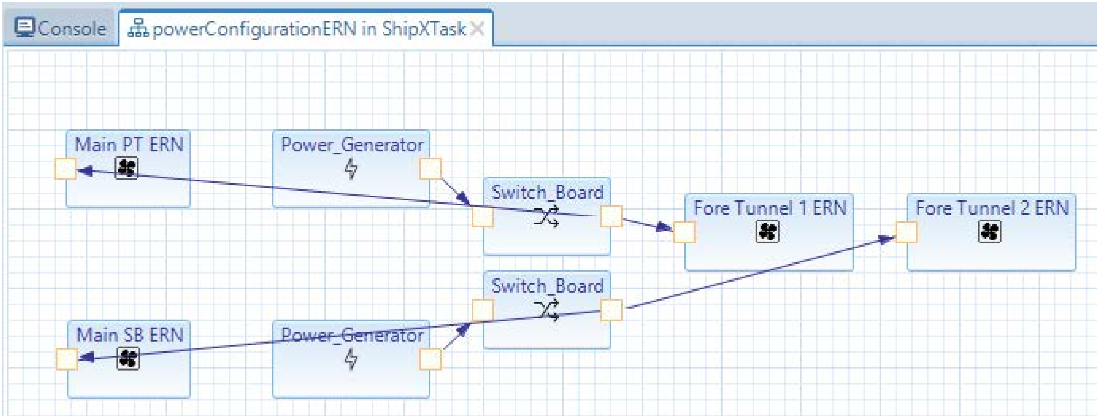
3) The next thing we need to do is define a power configuration. The power configuration defines the available power in the system. In this case we are not interested in any power limitations, but we want to run the calculation with two logical switch boards to look at failure scenarios.

a. Right-click the ship, select **New** – **Power Configuration.**

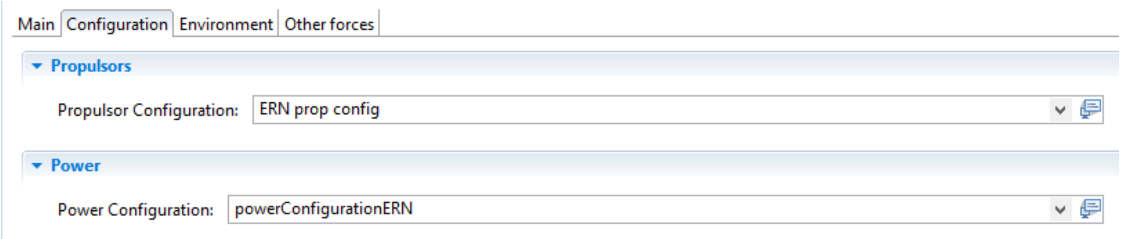
b. In the Navigator view, find the propulsor configuration and drag and drop it into the power configuration window.



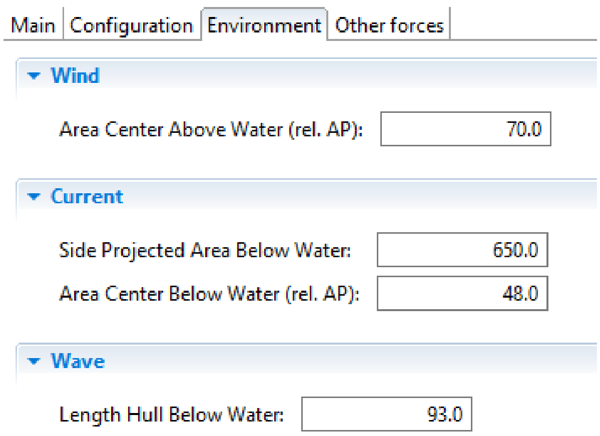
c. Add two power generators and two switch boards, all with available power 5000 kW. Connect them all as below:



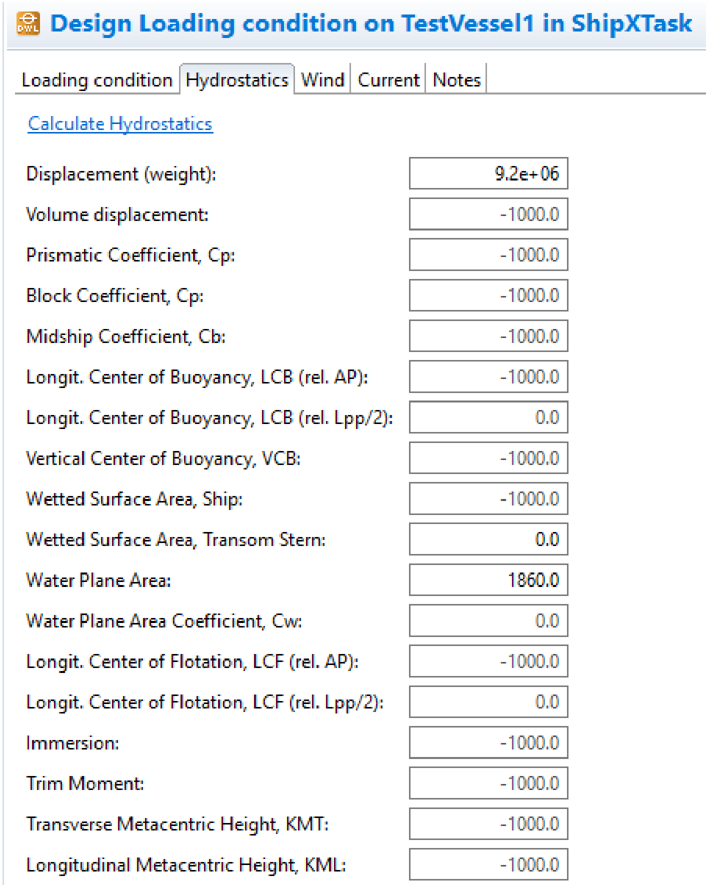
d. The last step is to define the power configuration input in the Station Keeping run:



4) The next input to be defined is the environmental input. To define this, we click the *Environment* tab. Enter the input values so that the input dialog finally looks like this:

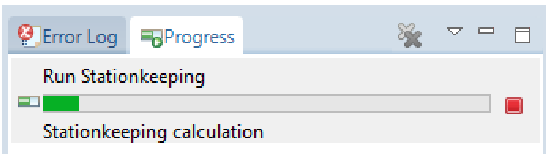


5) Had we used a hull geometry file as input when defining our ship, we would have all the input we needed now, and would have been ready to perform the calculation. Unfortunately, we used only main dimensions as input for defining our ship, so some of the input required by the ERN calculation method is still not defined. Trying to run the calculation now will give error messages for those specific input. If we go to the loading condition and press the **Hydrostatics** tab, we must manually create the input like this:

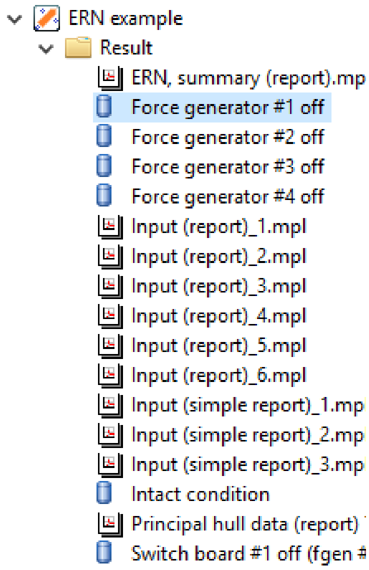


6) We have now defined all required input to perform a calculation. To start the calculation, right-click the Station Keeping run in the **Navigator** window and select **Run Stationkeeping**.

   a. The calculation will take a few seconds. While calculating you can see the progress in the bottom left **Progress** window of ShipX.



   b. When the calculation has finished all reports and plots will be created in the **Result** folder under the station keeping run in the **Navigator** window. You can expand the folder and double-click on the reports and plots you want to look at. You can also right click the folder, select **Explore** and look at the result folder in the file browser of your choice.
   c. After plotting the reports and plots you like you can export them from the *ShipX Plot Program* to, for example, *Microsoft Word* to include into a report.
   d. If you want detailed information about every sub-calculation comprising the complete ERN number, you can find them by right-clicking **Result** folder and selecting the **Explore** button, or simply double-click one of the sub-folders denoted by the blue barrel-icon:

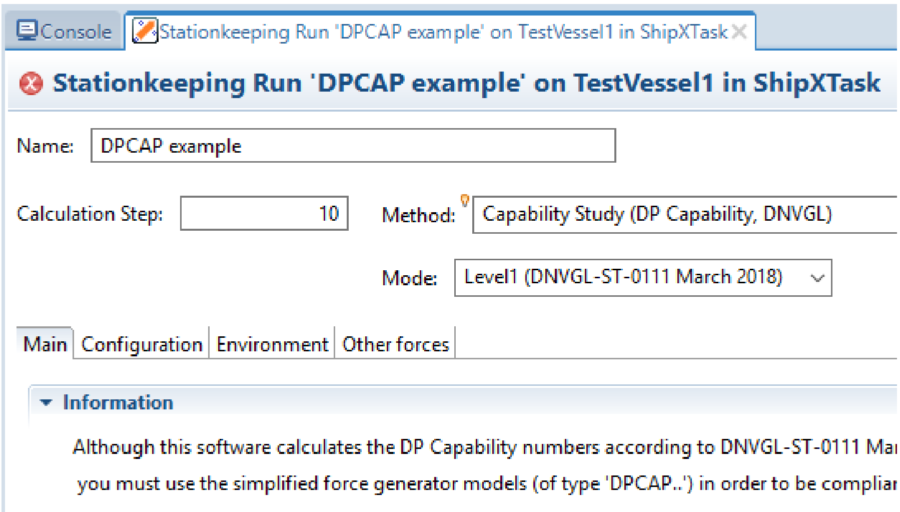

ERN example
Result
ERN, summary (report).mp
Force generator #1 off
Force generator #2 off
Force generator #3 off
Force generator #4 off
Input (report)_1.mpl
Input (report)_2.mpl
Input (report)_3.mpl
Input (report)_4.mpl
Input (report)_5.mpl
Input (report)_6.mpl
Input (simple report)_1.mp
Input (simple report)_2.mp
Input (simple report)_3.mp
Intact condition
Principal hull data (report)
Switch board #1 off (fgen #

18.4 DP Capability Calculation

In order to perform a DP Capability station keeping calculation according to [13]-[15] there are many considerations to take into account while creating the input and performing the calculations (see chapter 5.6 and 5.7 for details). The Station Keeping Plug-In has been developed to aid the user as much as possible to minimize the chance of user errors.
Before continuing in this chapter, make sure you have created the ship according to the instructions in chapter 18.1, 18.2 and 18.3. The following guidelines assume you have completed chapter 18.2, and will therefore be a little briefer in explaining all the steps. Use the following guide to create a DP Capability calculation from scratch:

1) Start the Station Keeping Plug-In by right clicking the **Runs** folder found under the loading condition of your vessel, select **New** and then **Stationkeeping Run**.

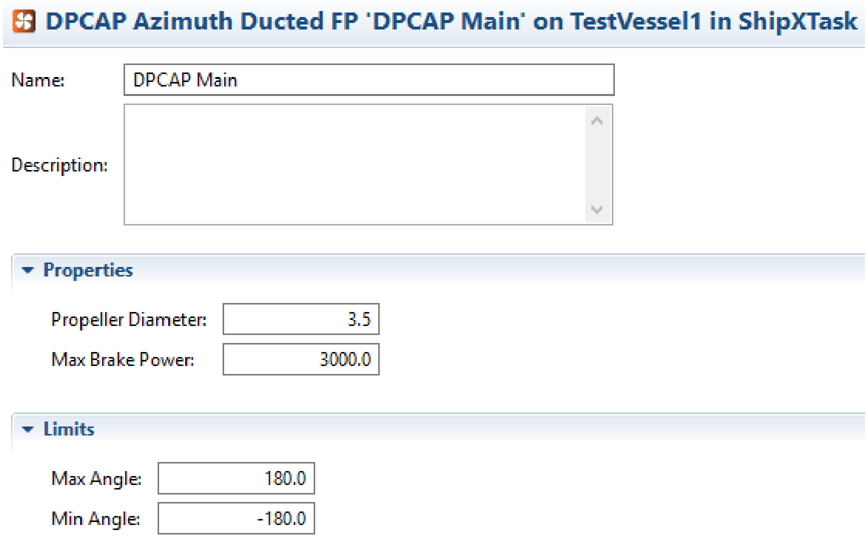
2) In the main window we must enter the name of the run and select the correct calculation method and -mode:



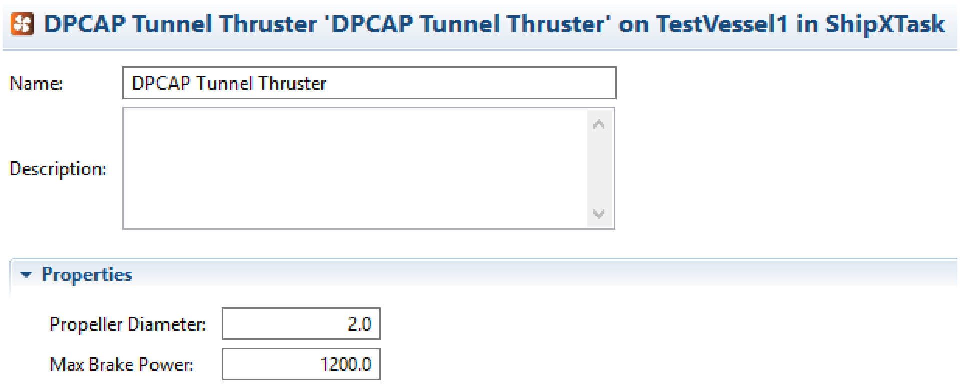
When the correct calculation mode has been selected, the user interface will change to only allow relevant input.

3) Moving on to the Configuration tab. The DP Capability standards are strict about what types of propulsors can be used. We have implemented propulsors based on [13]-[15]; they all have names starting with *DPCAP*. If you select any other propulsor types, you will not be allowed to perform a DP Capability calculation. We are going to add two types of propulsors, one main propulsion type and one tunnel thruster type in the bow:

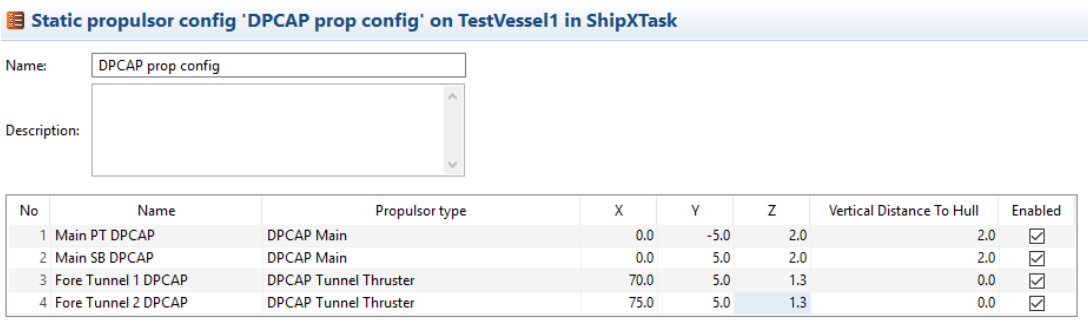
    a. Create the main propulsor type:
        - Right-click **Propulsors** in the **Navigator** window, select **New** – **Static Propulsor** – **DPCAP Azimuth Ducted FP**
        - Set the input like this:



b. Create the tunnel thruster propulsor type:
  - Right-click **Propulsors** in the **Navigator** window, select **New** – **Static Propulsor** – **DPCAP Tunnel Thruster**
  - Set the input like this:

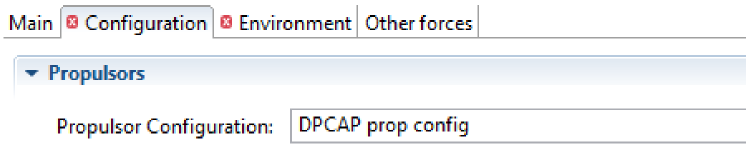


c. We have now defined our propulsor types. Let's create the propulsor configuration:
  - Right-click **Propulsor Configurations** in the **Navigator** window, select **New** – **Static propulsor config**
  - Add four lines and add the four propulsors we want, like this:



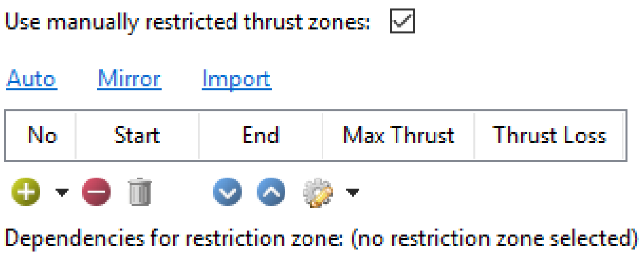
No	Name	Propulsor type	X	Y	Z	Vertical Distance To Hull	Enabled
1	Main PT DPCAP	DPCAP Main	0.0	-5.0	2.0	2.0	☑
2	Main SB DPCAP	DPCAP Main	0.0	5.0	2.0	2.0	☑
3	Fore Tunnel 1 DPCAP	DPCAP Tunnel Thruster	70.0	5.0	1.3	0.0	☑
4	Fore Tunnel 2 DPCAP	DPCAP Tunnel Thruster	75.0	5.0	1.3	0.0	☑

d. Define the propulsor configuration in the station keeping run

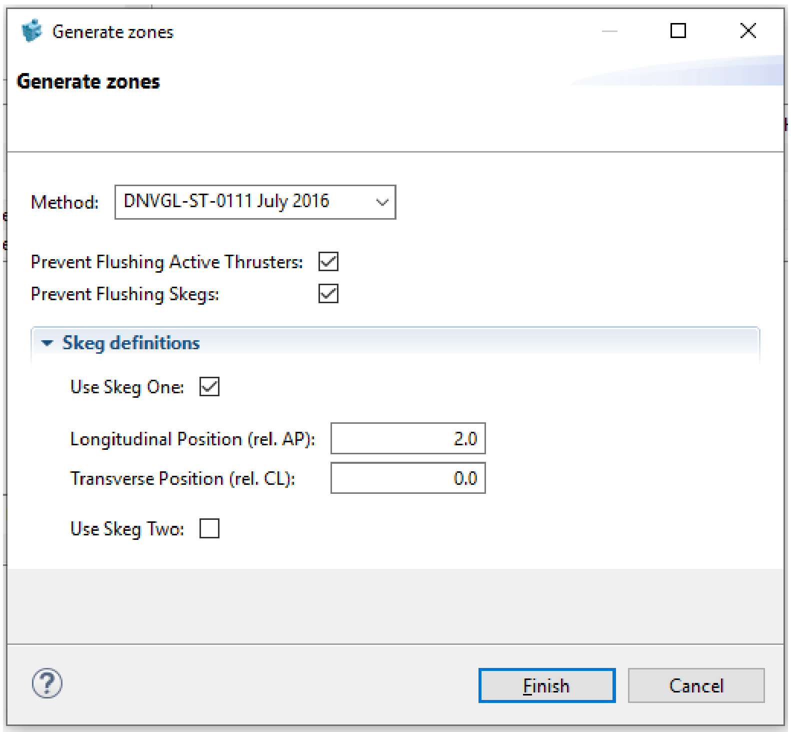


4) To finalize the propulsor input we must now define the forbidden thrust zones. In the 2016 version of the rules [13] this includes a zone for each skeg and zone(s) to avoid flushing any other live propulsor. In the 2018 version of the rules [14] the skeg zone(s) has been replaced by a loss model (built into the software).
We must define these forbidden zones for all propulsors. There is a very good reason that we waited until now to do this; to define the forbidden zones, we need to know where all the propulsors and skegs are. That also means that if you move a propulsor or skeg on your ship you will need to perform the following steps again to redefine the forbidden zones!

a. Open the propulsor configuration
b. Click the *Main PT DPCAP* row.
c. Check the *Use manually restricted thrust zones checkbox*
d. You will now see a new dialog:

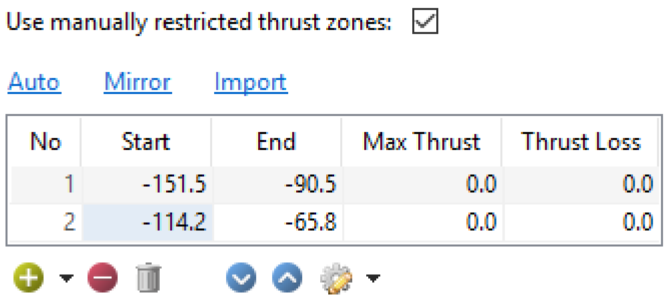


This is the dialog we can use to, amongst others, define manually forbidden zones. See chapter 7.1 for the input coordinate system for forbidden/restricted zones. We could click the plus sign button and add the zones manually, but the zone definition in [13]-[15] requires some manual work to calculate correct zones. We have therefore added an *Auto* button that does this for you. This is also why we need to know where all the other propulsors and skegs are, so that we can calculate the forbidden zones automatically. Click the *Auto* button. You will now see a new dialog. Set the input as seen below and then click **Finish**.

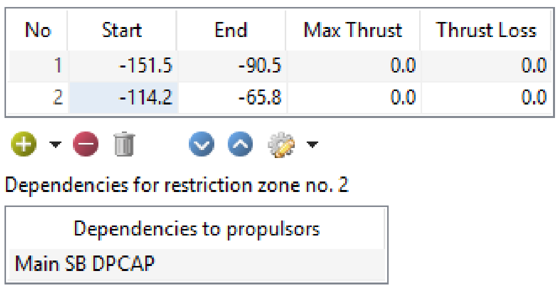


Note that since we are running a 2018 type DPCAP calculation in this example, we <u>should</u> have selected the 2018 method in above screen. We have in this example on purpose selected the 2016 method to illustrate the different types of forbidden zones.

You will now see the generated forbidden zones:



Click on each row to display the dependencies for that zone.
As you can see from the screenshot above, forbidden zone number 2 has a dependency of the **Main SB DPCAP** propulsor.



This means that the zone is there to prevent the **Main PT DPCAP** propulsor flushing the **Main SB DPCAP** propulsor. When the **Main SB DPCAP** propulsor is dead (as it will be sometimes during the failure calculations), this forbidden zone will be switched off automatically. Forbidden zone number 1 does not have any dependencies, so this forbidden zone will always be used. This zone is there to avoid the **Main PT DPCAP** propulsor flushing the skeg. You can also see that the zones overlap between -90.5 deg and -114.2 deg. This is not a problem, and if both zones are in use at the same time, they will be automatically merged in calculation to form a larger forbidden zone covering both zones.

Do the exact same thing on the **Main SB DPCAP** propulsor to produce the forbidden zones for that propulsor. As expected, the forbidden zone on for the **Main SB DPCAP** propulsor will be mirrored those of the **Main PT DPCAP** propulsor.

No	Start	End	Max Thrust	Thrust Loss
1	65.8	114.2	0.0	0.0
2	90.5	151.5	0.0	0.0

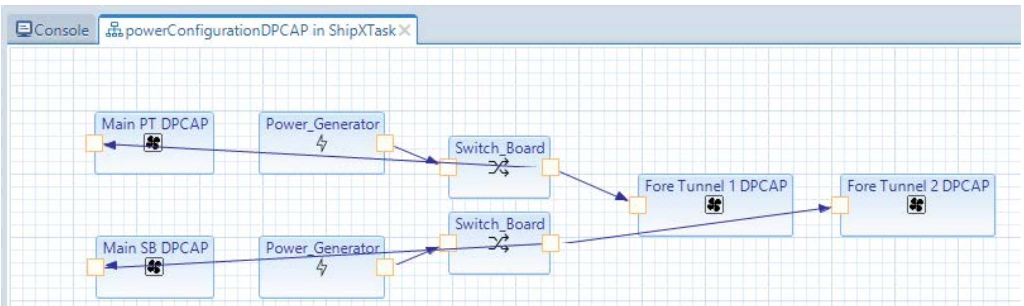
Dependencies for restriction zone no. 1

Dependencies to propulsors
Main PT DPCAP

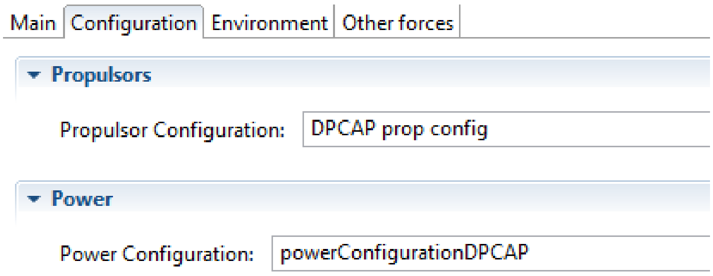
You should now perform the same action of defining forbidden zones on the rest of the propulsors. As the rest of the propulsors we have define are tunnel thrusters, they don't have any forbidden zones.

5) The next thing we need to do is define a power configuration. The power configuration defines the available power in the system. In this case we want to run the calculation with two logical switch boards to look at failure scenarios.

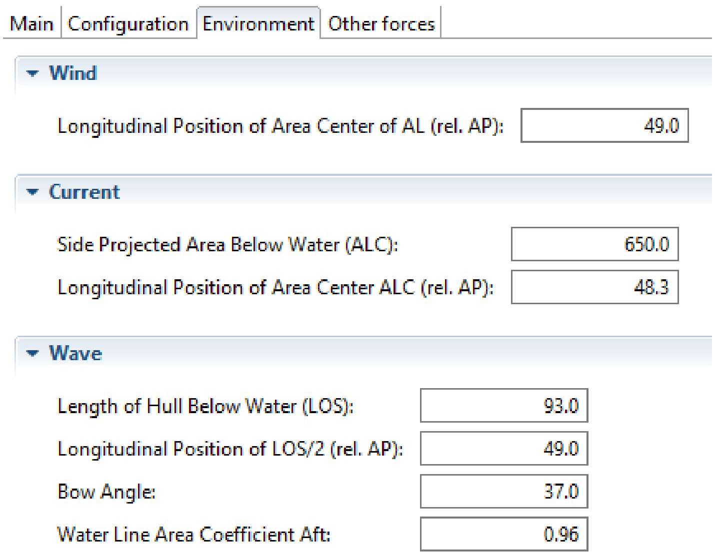
   a. Right-click the ship, select **New** – **Power Configuration.**
   b. In the Navigator view, find the propulsor configuration and drag and drop it into the power configuration window.
   c. Add two power generators and two switch boards, all with available power 5000 kW. Set a 10% power reserve on the switch boards if you have the extended power module license. If you don't have the extended power module license, reduce the available power on each switch board by 10%. Connect them all as below:



d. The last step is to define the power configuration input in the Station Keeping run:



6) The next input to be defined is the environmental input. To define this, we click the *Environment* tab. Enter the input values so that the input dialog finally looks like this:

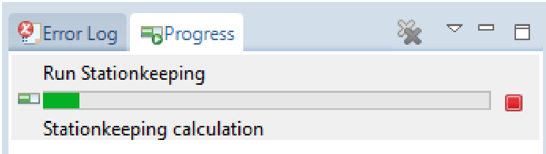


Note that the definition of the bow angle is outlined in chapter 3.1.7 in [13] or chapter 3.7 in [14]-[15] and must be calculated manually by the user.

7) We must also define any skeg input. Even though in later DPCAP versions the skeg does not enforce a forbidden zone (as seen when defining forbidden zones for the propulsors) the skeg still causes hull interaction losses for the propulsors. To define the skeg, go to the Miscellaneous tab and enter your data. In this example we don't have any skeg data to input, so we leave both skeg checkboxes deselected.

8) We have now defined all required input to perform a calculation. To start the calculation, right-click the Station Keeping run in the **Navigator** window and select **Run Stationkeeping**.

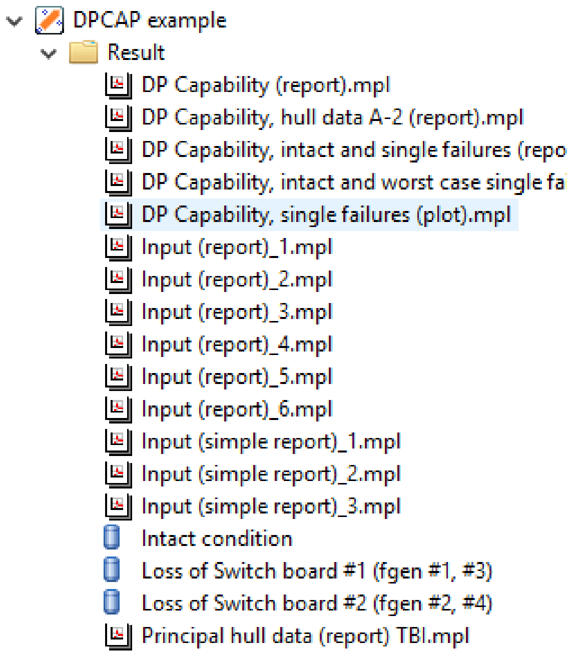
e. The calculation will take a few seconds. While calculating you can see the progress in the bottom left **Progress** window of ShipX.



f. When the calculation has finished all reports and plots will be created in the **Result** folder under the station keeping run in the **Navigator** window. You can expand the folder and double-click on the reports and plots you want to look at. You can also right click the folder, select **Explore** and look at the result folder in the file browser of your choice.

g. After plotting the reports and plots you like you can export them from the *ShipX Plot Program* to, for example, *Microsoft Word* to include into a report.

If you want detailed information about every sub-calculation comprising the complete ERN number, you can find them by right-clicking **Result** folder and selecting the **Explore** button, or simply double-click one of the sub-folders denoted by the blue barrel-icon:



SINTEF

19 BIBLIOGRAPHY

[1] J. Brix
"Manoeuvring Technical Manual"
Seehafen Verlag, 1993

[2] Ir U. Nienhuis
"Simulations of Low Frequency Motions of Dynamically Positioned Offshore Structures"
Royal Institute of Naval Architects, Volume 129, 1987

[3] Thor I. Fossen
"Marine Control Systems"
Marine Cybernetics, 2002

[4] Svein Peder Berge
"Nonlinear Way-Point Tracking Control and Docking of Ships"
Report 99-5-W
Department of Engineering Cybernetics
Norwegian University of Science and Technology, 1999

[5] O. M. Faltinsen
"Sea loads on ships and offshore structures"
Cambridge University Press, 1990

[6] T. A. Johansen, T. I. Fossen, Svein P. Berge
“Constrained Nonlinear Control Allocation with Singularity Avoidance using Sequential Quadratic Programming”
IEEE Trans. Control Systems Technology, 2003

[7] The International Marine Contractors Association
"Specification for DP Capability Plots"
IMCA M 140 Rev. 1, January 2017

[8] Det Norske Veritas
"Capability Plots and Environmental Regularity Numbers"
Pt.6 Ch.26 Sec.7 – Page 23
Rules for Ships, July 2010

[9] Wojciech Kauczynski
"Numerical model for estimation of thrust and torque losses in waves"
MARINTEK Report 512001.01.02 (87-0088), March 1987

[10] Kourosh Koushan
"Prediction of thruster-thruster and thruster-hull interaction effects"
MARINTEK Memo, 2004

[11] American Bureau of Shipping
"Guide for Dynamic Positioning Systems"
ABS, April 2020

[12] Det Norske Veritas
"Environmental Regularity Numbers"
Pt.6 Ch.7 Sec.7
Rules for Ships, July 2014

[13] DNVGL
"Assessment of station keeping capability of dynamic positioning ships"
DNVGL-ST-0111, July 2016

[14] DNVGL
"Assessment of station keeping capability of dynamic positioning ships"
DNVGL-ST-0111, March 2018

[15] DNV
"Assessment of station keeping capability of dynamic positioning ships"
DNV-ST-0111, December 2021