

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/308674535>

# Drift Forces – Wingsails vs Flettner Rotors

Conference Paper · October 2016

---

CITATIONS

0

---

READS

6

3 authors:



[Jarle Kramer](#)

Norwegian University of Science and Techn...

3 PUBLICATIONS 0 CITATIONS

SEE PROFILE



[Sverre Steen](#)

Norwegian University of Science and Techn...

51 PUBLICATIONS 205 CITATIONS

SEE PROFILE



[Luca Savio](#)

Norwegian Marine Technology Research In...

15 PUBLICATIONS 42 CITATIONS

SEE PROFILE

# Drift Forces – Wingsails vs Flettner Rotors

Jarle A. Kramer, NTNU, Trondheim/Norway, [jarle.a.kramer@ntnu.no](mailto:jarle.a.kramer@ntnu.no)  
Sverre Steen, NTNU, Trondheim/Norway, [sverre.steen@ntnu.no](mailto:sverre.steen@ntnu.no)  
Luca Savio, Marintek, Trondheim/Norway, [luca.savio@marintek.sintef.no](mailto:luca.savio@marintek.sintef.no)

## Abstract

*When sails produce thrust, they also produce side force, which makes the ship move with a drift angle. The drift angle increases the resistance of the ship, which cancels some of the positive effect from the sails. This paper explores the importance of drift for two different types of sail technologies. A general cargo ship is analyzed, using historical wind data on an example route from Rotterdam to Trondheim, using wingsails and Flettner rotors. The analysis uses CFD and a custom route simulation software.*

## 1. Introduction

Using sails on modern cargo ships, as a way to reduce the fuel consumption, has been suggested many times by both researchers and commercial companies. Recent examples include the “Wind challenger” project from the university of Tokyo, *Ouchi et al. (2013)*, and the work presented in *Traut et al. (2014)* where a Flettner rotor is compared to a kite. There are also many older, but more famous projects such as the Walker Wingsail, *Walker (1985)*, the “turbo sail” developed by the Cousteau foundation, *Charrier et al. (1985)* and the original Flettner rotor ship. Although many solutions exist, the two most popular wind propulsion technologies seem to be wingsails and Flettner rotors. Both of these technologies, create thrust mainly by using “lift”, i.e. the force normal to the incoming wind velocity. A single element symmetric wingsail creates this lift by having an angle of attack relative to the wind, while Flettner rotors are spinning cylinders that create lift by utilizing the Magnus effect. One of the consequences of creating thrust in this way is an unavoidable side force. That is, as long as the thrust from the sails is created by lift, there is no way of pushing the ship forward, without also pushing it sideways. How much the ship is pushed sideways is dependent on the apparent wind direction. If the wind is coming directly from the side of the ship, the only contribution to the side force is from the drag on the sails. However, if the apparent wind is from any other direction, the lift will also contribute to the side force. The result is that in typical conditions, the side force is often many times larger than the thrust. This side force has an effect on the flow around the ship hull. Since the hull is pushed sideways, it starts moving with an increasing drift angle, until the drift induced side force on the hull is equally strong, but with opposite direction to the side force from the sails. The drift angle makes the ship hull into a lifting surface. The problem is, as is the case with all lifting surfaces: with lift, there is also lift induced drag. That is, due to the drift angle, the resistance on the ship hull is increased, which cancels some of the positive effect from the sails. How big of a problem this is, is dependent on several factors, such as the hydrodynamics of the ship hull, the side force to thrust ratio of the sails and the amount of thrust that is generated from the wind. An interesting aspect of modern sails, which are not much studied previously, is the difference in side force to thrust ratio. For instance, Flettner rotors generate very large forces, relative to the sail area. Dependent on the speed of the ship, and the wind direction, the result is often that a Flettner rotor can generate much more thrust than a wingsail, with equal sail area. However, the side force to thrust ratio is also larger, which means that for the same amount of thrust, the ship is also pushed sideways with a much stronger force.

This paper explores two main questions: how big of a problem is the drift-induced resistance for a normal cargo ship with modern sails, and how much difference is there between wingsails and Flettner rotors?

The case study chosen in for these questions is a 120 m long general cargo ship, with 40 m tall sails, on an example route from Rotterdam, Netherlands to Trondheim, Norway. Historical wind data,

Computational Fluid Dynamics (CFD) analysis of both the sails and the ship hull, and a route analysis code, is used to calculate the importance of drift induced effects. The Flettner rotor in this analysis is a simple spinning cylinder, without any end plates or flaps, while the wingsail is a two-element wing, where both elements are of equal length. We limit the study to one aspect ratio for the sails, which is equal to 5. However, the number of sails is varied between 1 and 8, in order to change the amount of thrust produced from the wind. Two different control strategies for the sail are tested: maximum power delivered from the sails, or maximum effective power delivered. The effective power is the power from the sails, minus the added required power due to the sails. In this analysis, the sails generate added resistance due to two main components: the added resistance on the hull, and the added resistance on the rudder. These two are considered to be different effects, as the rudder might be necessary in order to balance the ship hull at the right drift angle. A keel model will also be used to assess the effect of installing a simple keel on a normal cargo ship, with regard to drift-induced resistance. We will also run the analysis with two different assumptions regarding the sail mechanism: one where the sails cannot be retracted, or stowed away when they are not in use, and one where they can. Since the main focus of this study is to evaluate the importance of drift, some simplifications regarding other effects have been made. For instance, we have not calculated added resistance due to waves, interaction effects between sails, or used any form of engine model. The energy savings presented in this paper should therefore be evaluated critically, and the focus should rather be on how including drift changes the results predicted by the simplified model.

All the code used for creating the results in this paper is published on Jarle Kramer's GitHub page, *Kramer (2016)*. This includes a library and scripts used to set up CFD simulations, a ship analysis library, a geometry handling class, a route simulation code, a particle swarm optimization algorithm, a non-linear lifting line code, and weather data analysis code. Everything is written in Python or Cython. Most of the code is written in an object oriented way, with classes that sometimes inherits from each other. Due to page limitations, not everything in this paper is explained in detail, but references will be made to the GitHub page, where the specific code is available for further study, if this is of interest.

## 2. CFD simulations

CFD is used to analyze both the hull and the sails with the open source software library OpenFOAM, version 3.0+, <http://www.openfoam.com>. A custom python library written specifically for OpenFOAM simulation setup is used. This library can be found on GitHub, *Kramer (2016)*, along with example scripts that show how it is used. There is one general library, called "myPyFoam", in addition to three specialized classes, called "TowingTank", "WingSimulation" and "FoilSimulation" which is used to set up simulations of ship hulls, 3D wings, and 2D foil geometries respectively. This approach to CFD simulation setup is based on the idea that, for a specific type of simulation, such as foil simulations, there is a general strategy for setup, that are not very much dependent on details in the geometry. That is, a simulation strategy that works for one foil should also work for another foil, if the Reynolds number and main dimensions are the same. Based on experience developed while running CFD simulations in the past, as well as recommended best practices from different sources, the setup library manages both the meshing process and solver settings automatically, with main dimensions and velocity as input. This scripting approach to CFD ensures that we set up the simulations in a consistent matter, every time.

Three different types of OpenFOAM solvers are used for this case study:

- simpleFoam, which is a steady state incompressible solver that uses the SIMPLE algorithm, *Patankar and Spalding (1972)*. This solver is used for the wing and foil simulations.
- pimpleFoam, which is an unsteady, incompressible solver that uses a mix between the SIMPLE algorithm and PISO algorithm, *Issa et al. (1986)*, to deal with large time steps. The time loop is driven forward with the PISO algorithm, with the option of doing several "inner iterations" using the SIMPLE algorithm. This solver is used for all the simulation classes.

- interFoam, which is an unsteady, incompressible solver, similar to pimpleFoam, but with support for two fluids, such as air and water. The interface between the two fluids are tracked using the Volume of Fluid (VoF) method, *Hirt and Nichols (1981)*. This solver is used to find the wave resistance for the ship hull.

All the simulations in this case study use Reynolds Average Navier-Stokes (RANS) turbulence modelling. The setup library supports several turbulence models, but the  $k-\omega$  SST, *Menter (1994)*, is the default, and has been used for all the simulations for this paper. A continuous wall function is used, which is an implementation of the equation presented in *Spalding (1961)*. The initial conditions for the variables in the turbulence model follows standard practices, with 1% inlet turbulence, [http://www.esi-cfd.com/esi-users/turb\\_parameters/](http://www.esi-cfd.com/esi-users/turb_parameters/). The meshing is done with the OpenFOAM meshing tool “snappyHexMesh”. SnappyHexMesh generates hexahedra and split-hexahedra mesh cells, by iteratively refining and moving a background mesh. The process is controlled by specifying refinement levels, and wall-layers, at the geometry present in the simulation, as well as in optional refinement regions. The size of the cells closest to a geometry is adjusted based on a target  $y^+$  value and a case specific maximum size. The length of the cells corresponding to a certain  $y^+$  value is calculated with a friction line, and the Reynolds number for each simulation. The background mesh is adjusted based on a target size alone. The CFD simulations for ships and 3D wings use wall functions, and general guidelines for wall functions often suggest  $y^+$  values between 30 and 100, which is in the range of the logarithmic law of the wall. Both too small and too large  $y^+$  values can be problematic, as is for instance shown in *Hympendahl and Ciortan (2015)*. The  $y^+$  values chosen by default by the setup library is 60, if wall functions are used, otherwise it is 1. However, depending on the Reynolds number and mesh settings, this can sometimes lead to a too coarse mesh, which in our experience are worse than “wrong”  $y^+$  values. Each case class therefore uses a custom maximum size, for the mesh cells right outside the wall layers. If the target  $y^+$  value suggest that the mesh will be too coarse, the library will first try to alter some mesh settings, such as layer expansion. The maximum layer expansion factor is 1.5, but this is generally reduced to about 1.1-1.3 automatically by the library. If this does not work, a smaller  $y^+$  value will be used. If the  $y^+$  value drops below 30, a warning is generated, so that we can decide if we need to resolve the boundary layer instead. Some important simulation parameters are presented in Table 1. “L” and “U” is reference to the characteristic length dimension (ship length and chord length) and inlet velocity in the simulation respectively. The “max feature cell” size is a reference to the smallest cell size used in the simulation outside the wall layers, which are generated at “features”, or sharp edges in the geometry. All the other cells close to a geometry will be one refinement level less, or twice the size. The number of refinement levels varies, depending on the ratio between the background mesh and the feature cells, but are never larger than 8. The time step in the simulation is adjusted so that the Courant number is never above a maximum limit, which is adjusted automatically by OpenFOAM, in addition to a maximum absolute limit that is proportional to the characteristic length dimension divided by the inlet velocity.

Table 1: Simulation parameters

	Foils	Wings	Ships
Max feature cell size/L	0.001	0.005	0.001
Max background cell size/L	0.1	0.5	0.1
Number cells per refinement level	10	5	5
Number of wall layers	15	5	5
Target $y^+$ value	1	60	60
Approximate number of cells	100 000	10 million	1-5 million
Max Courant number	10	10	10
Max time step $\cdot U/L$	0.005	0.005	0.0025
Simulation time $\cdot U/L$	15	35	14
Max steady state iterations	10000	6000	Not used

In addition to different maximum sizes, different case classes have different refinement regions, which primarily is made to capture the wake in the simulations. This includes refinement in the kelvin wake for ship simulations, tip wake for 3D wings, and a wake that starts at the trailing edge for 2D foil profiles. The ship simulations also use anisotropic refinements only in the vertical direction, in the region where the free surface is located. This is necessary to keep the boundary between water and air relatively sharp. The different meshes used in this analysis can be seen in Fig.1.

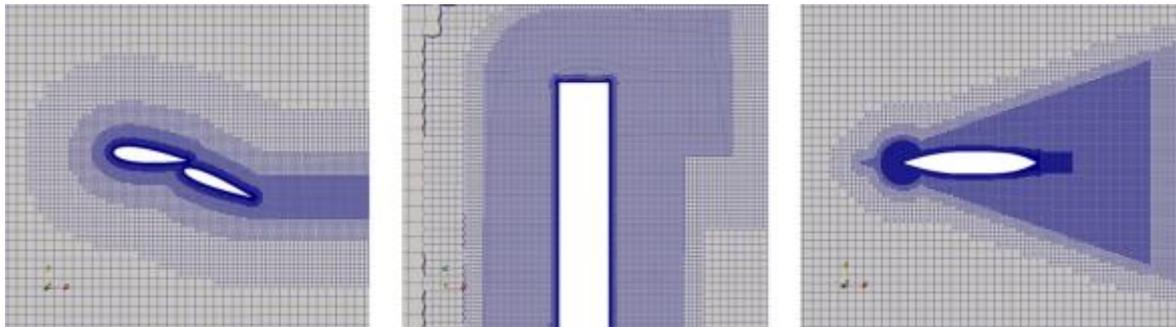


Fig.1: Mesh used for the different simulation cases

The parameters shown in Table 1 are the default settings for each simulation class, corresponding to a “medium” mesh. However, each class also have the option of creating “very coarse”, “coarse”, “fine” and “very fine” meshes. When these settings are activated the length dimensions in the mesh is either multiplied or divided by a factor. A “very coarse” and “coarse” mesh corresponds to a mesh where the maximum length dimension is multiplied with two or the square root of two respectively. For a “very fine” and “fine” mesh, the length dimension is divided by two or the square root of two respectively. This is used to do mesh convergence studies, and the result of such a mesh study for the ship hull can be seen in section 3. The setup library has also been used to perform validation simulations. Some of these validation experiments will be presented along with the numbers for this case study.

### 3. Description of the case study

#### 3.1 Ship

The ship chosen for this case study is a 120 m long general cargo ship. The main dimensions, as well as the service speed, are chosen so that it is similar to a real general cargo ship, and can be seen in Table 2. Both the full-scale values, and the model scale values used in the CFD simulations are shown. A small, relatively slow, cargo ship is considered to be an interesting case study, simply due to the size; we are mostly interested in ships where a significant portion of the total thrust comes from the sails. A very large ship would also need very large sails in order to generate significant amounts of thrust. Very large sails can be problematic, both from a structural point of view, and from practical point of view, due to bridges and cranes in harbors. A smaller ship might need larger sails relative to its own size, as larger ships are more efficient, but the absolute size can still be reduced. It therefore seems more realistic that a small cargo ship can get a large portion of the total thrust from sails, at least in the near future.

Table 2: Ship main particulars

	Full scale ship	CFD model ship
Lwl [m]	120	7
Bwl [m]	20	1.167
D [m]	12.5	0.729
T[m]	5.5	0.321
Volume displacement [m <sup>3</sup> ]	7990	1.586
Wetted surface, w.o. rudder [m <sup>2</sup> ]	2591	8.817
Rudder planform area [m <sup>2</sup> ]	11.25	0.0383
Keel planform area [m <sup>2</sup> ]	22.5	0.0766

Service speed [m/s]	7	1.69
Service Froude number	0.204	0.204
Resistance coefficient, $C_T \cdot 10^3$	3.149	4.256
Friction resistance coefficient, $C_F \cdot 10^3$	1.723	3.041
Roughness resistance coefficient $\Delta C_F \cdot 10^3$	0.211	0
Pressure resistance coefficient $C_P \cdot 10^3$	1.215	1.215
Propeller diameter, D [m]	4	0.233
Propeller pitch P/D	0.997	0.997
Propeller number of blades	4	4

The hull geometry is a custom design. The reason for designing a new geometry, rather than using an already existing design, is that most open ship geometries are either very large tankers or very larger container ships. The hull design was created with the goal making a realistic, but simple ship. It does not have a bulb, but instead a straight slender bow. It was made using a Catmull-Clark subdivision surface, *Catmull and Clark (1978)*, in the open source geometry modeling software Blender, <https://www.blender.org>. The subdivision surface representation of the geometry was chosen due to its flexibility with regards to topology. Unlike for instance NURBS based geometry, a subdivision surface can have arbitrary topology, i.e. the entire ship hull can be created as one surface, rather than several individual NURBS patches. Fig.2 shows the hull lines; the 3D model can be downloaded from GitHub.

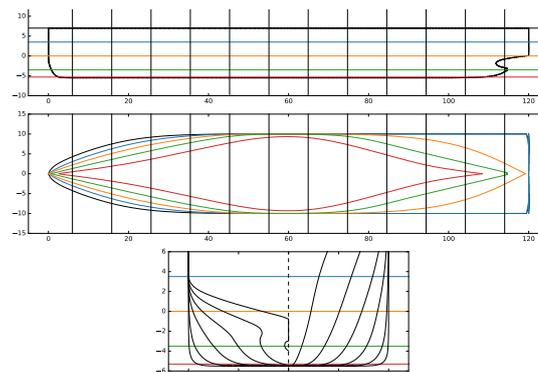


Fig.2: Line drawings of the ship hull

The hydrodynamic forces on the ship are modeled with the “Hull” class in the “Ship” library that can be found on GitHub. This class is initialized with the main dimensions of the ship. Based on the main dimensions, the class estimates the forces that act on the ship hull as function of Froude number, Reynolds number and drift angle, either using simplified theories and empirical models or results from CFD simulation and experiments. For this analysis, CFD is used to compute all the necessary values. When using CFD to estimate the calm-water resistance, the pressure resistance and friction resistance from the simulations are extracted individually. The CFD simulations are performed in model scale, for several Froude numbers. When calculating the full-scale resistance, the pressure resistance is assumed to be independent of Reynolds number, but dependent on Froude number, while the friction resistance is dependent on both. In order to scale the friction resistance to full scale, for a given Froude number, a friction line is used, along with an empirical roughness factor. The scaling factor is the value of the friction line at full scale, divided by the value of the friction line in model scale. The friction line used is a numerical friction line, based on the  $k-\omega$  SST turbulence model, which can be found in *Eca and Hoekstra (2008)*. The reason for choosing a numerical friction line, rather than the more standard ITTC-57 friction line, is based on the work published in *Raven et al. (2006)*. The paper suggests that using the ITTC-57 friction line might not be the best scaling strategy, and that for instance a numerical friction line is a better choice. The CFD values and scaled values for the resistance coefficients in calm water at service speed can be seen in Table 2.

In order to calculate the side force and added resistance due to drift, CFD simulations of the ship hull

with a drift angle, but without free surface modeling is used. The free surface has previously been found to not be very important for estimating the drift-induced forces, and neglecting the free surface simplifies the simulations, *Kramer and Steen (2015)*. The hull is simulated with five drift angles. The data from the simulations are then fitted to second-order polynomials by the ‘‘Hull’’ class, as this is a model that have been found to work well for drift induced forces. The induced drag coefficient is defined as the drag at a specific drift angle, minus the drag at zero drift angle. That is, it is the added resistance due to drift. The computed lift, lift-induced drag, and yaw moment, as a function of drift angle can be seen in Fig.3. The coordinate system is located in the bow of the ship, with the x-axis pointing towards the stern, when the drift angle is zero, and the z-axis pointing up.  $F_x$  is the force in the x-direction,  $F_y$  is the force in the y-direction, while  $M_z$  is the moment around the z-axis. The coefficients are defined as follows, where  $L$  is the ship length,  $T$  is the ship draft,  $U$  is the ship velocity,  $\rho$  is the water density and  $\alpha$  is the drift angle:

$$C_L = \frac{F_y}{0.5\rho L T U^2}$$

$$C_{Di} = \frac{F_x(\alpha) - F_x(0)}{0.5\rho L T U^2}$$

$$C_M = \frac{M_z}{0.5\rho L^2 T U^2}$$

The CFD values are plotted for three different meshes: coarse, medium and fine. This is to show that the result is not very dependent on the mesh resolution. The result for the fine mesh is used in this analysis. The polynomial curve fit is shown as solid lines. In order to validate the CFD simulations, the setup scripts have also been used to generate simulations that reproduce the experiments published in *Kramer et al. (2016)*. In this experiment, a foil-like ship is towed in a towing tank for three aspect ratios, two bottom edge shapes and two Froude numbers. The experimental data shown in Fig.3 is for the lowest aspect ratio, with the rounded bottom edge, and Froude number 0.1.

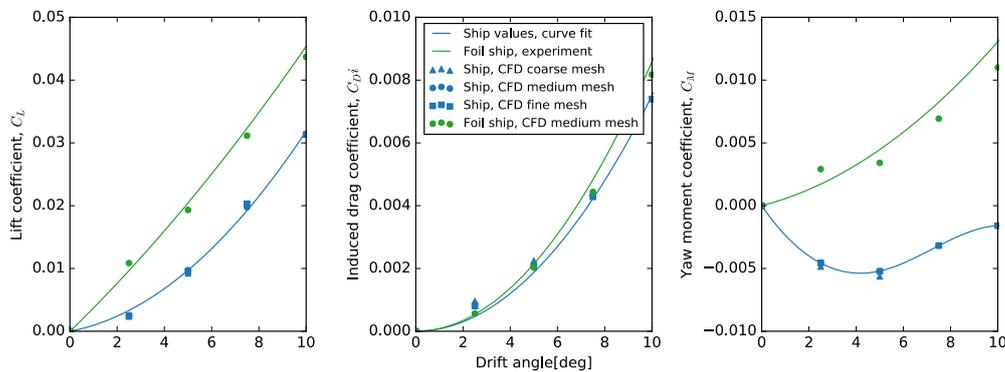


Fig.3: Lift, lift-induced drag and yaw moment coefficients for cargo ship and validation experiment

We are also interested in the effect of rudder and a keel. The rudder was present in all the CFD simulations performed for the ship hull, but only with zero rudder angle. The effect of setting the rudder angle to something other than zero is modeled with the ‘‘Rudder’’ class in the ship library. This is a simple model of a lifting surface, based on a simplified rudder model suggested in *Bertram (2012)*. The exact flow around a rudder is a complicated phenomenon, with very high Reynolds number, presence of a propeller slip stream and interaction from the ship hull. The details of this flow has been neglected. Rather, steady state CFD simulations of the rudder geometry is performed, where the rudder is standing on a symmetry plane in order model the presence of the ship hull. The CFD simulations are performed for a Reynolds number equal to  $2E6$ , but the since the rudder will actually be experiencing a Reynolds number more close to  $15E6$ , the friction resistance on the rudder is scaled in the same way as for the ship hull. Only rudder angles well below stall is simulated. Rudder stall is not directly modeled in the route simulation, but the magnitude of the rudder angle is evaluated to assess whether stall is a likely problem or not. The values for lift and drag from the CFD simulations

are then used to construct polynomial models. The lift is assumed to be linearly dependent on the rudder angle, while the lift-induced drag is assumed to be a second order polynomial. The rudder geometry is a spade rudder, with NACA 0018 foil profile, aspect ratio of 2.22 and taper ratio of 0.83. Fig.4 shows the computed lift and drag coefficients.

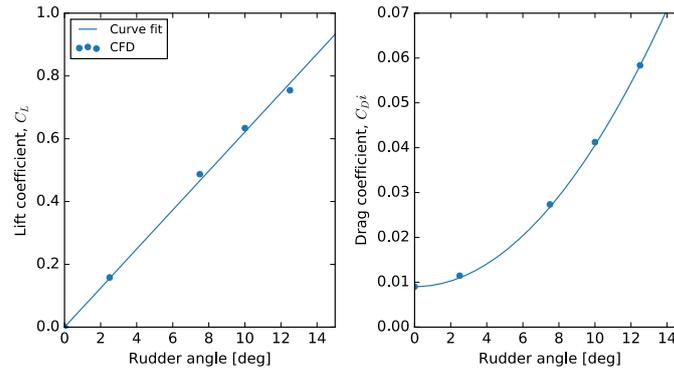


Fig.4: Lift and drag coefficients for the rudder and the keel

The area of the rudder,  $A_R$ , is calculated from a recommended formula in *Bertram (2012)*, as follows:

$$\frac{A_R}{L \cdot T} \geq 0.01 \left( 1 + 25 \left( \frac{B}{L} \right)^2 \right)$$

The effect of the propeller slip stream is treated by adding lift,  $\Delta L$ , and drag,  $\Delta D$ , as a function of thrust,  $T$ , to the lift and drag calculated by the coefficients from CFD. The formulas are taken from *Söding (1998)*.  $C_{Th}$  is the thrust coefficient for the propeller, and  $\alpha$  is the rudder angle.

$$\Delta L = T \left( 1 + \frac{1}{\sqrt{1 + C_{Th}}} \right) \sin \alpha$$

$$\Delta D = T \left( 1 + \frac{1}{\sqrt{1 + C_{Th}}} \right) (1 - \cos \alpha)$$

The incoming velocity to the rudder is assumed to be following the ships center line, as the rudder is located in the ship and propeller wake. That is, the lift and drag from the rudder is in a ship fixed coordinate system, and must be rotated when they are added to the global forces. The keel is modeled in the same way as the rudder, only with twice the area, and with the assumed incoming velocity to be in the ship traveling direction. The yaw moment from both the rudder and the keel is calculated by multiplying the force normal to the ship centerline with the distance from the bow to the rudder/keel. The rudder is located at the stern of the ship, so the distance is  $0.95 \cdot L$ , while the keel is located in the middle of the ship, or  $0.5 \cdot L$ , which is also the assumed mean center of pressure for the sail. Global forces and yaw moment on the ship hull as function of drift and rudder angle, with and without keel, at service speed can be seen in Fig.5.

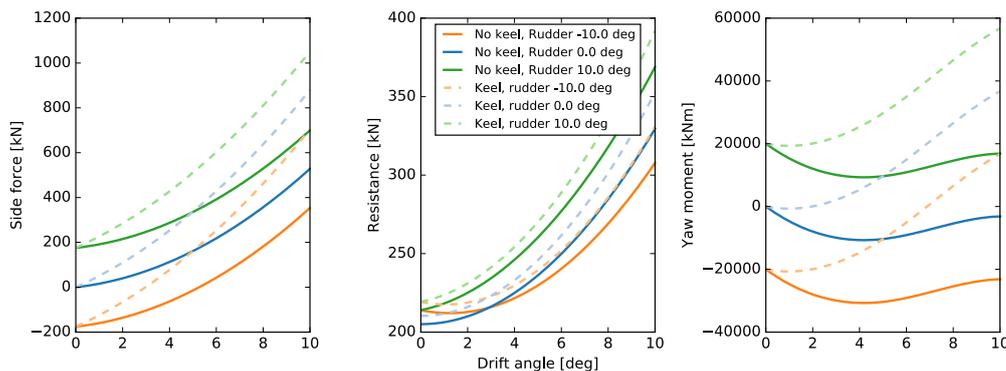


Fig.5: Forces on the ship as a function of drift and rudder angle, with and without keel

### 3.2 Sails

Two different types of sails are modeled in this paper: a two-element wingsail and a Flettner rotor. Both sails have a geometric aspect ratio of 5 but they are assumed to stand on a large deck structure, so that the effective aspect ratio is 10. That is, we assume that the deck acts as a symmetry plane. The Flettner rotor analyzed is a spinning cylinder, with a constant diameter along the span of the rotor. The wingsail is assumed to have a taper ratio of 0.4. The leading element of the wingsail is based on the NACA 0020 profile, while the trailing element is based on NACA 0015. Both elements are of equal length. The maximum flap angle is 15 degrees. The hinge point of the flap is at the quarter chord of the foil as a whole, or halfway into the first element. The sails are modeled with the “Sail” class in the “Ship” library published on *Kramer (2016)*. This class consists of methods for calculating lift, drag, thrust and side force, as well as a method that can optimize the sail control parameters based on an arbitrary input objective function. The forces are determined from force coefficients. More specifically, the input parameters used to initialize the sail class are the area of a single sail, the height of the sail, the number of sails in total, the lift and drag coefficients for a single sail, along with the corresponding control parameters. For the Flettner rotor, the power coefficient is also needed, which tells us how much input power is required in order to spin the Flettner rotor at a given speed. The control parameter can be either the spin ratio (Flettner rotor) or the angle of attack and flap angle (wingsail). The coefficients are defined as follows, where A is the sail planform area and U is the wind velocity:

$$C_{L/D/x/y} = \frac{\text{Lift/ Drag/ Thrust/ Side force}}{\frac{1}{2} \rho A U^2}$$
$$C_p = \frac{\text{Sail input power}}{\frac{1}{2} \rho A U^3}$$

The optimization of the sail control parameters can be done with several methods: brute force, built in optimization methods from the SciPy library, <http://scipy.org>, or a custom written optimization method, based on the particle swarm method, *Eberhart and Kennedy (1995)*. For the wingsail in this analysis, the particle swarm method is used, while the Flettner rotor is optimized with brute force.

The force coefficients for the wingsail are analyzed using a combination of 2D CFD and a non-linear numerical lifting line. The non-linear numerical lifting line uses the basic principle of the traditional lifting line, *Prandtl and Tietjens (1934)*, but with linear foil theory exchanged with a non-linear viscous 2D lift coefficient, as well as an iterative method to solve the equations. This approach for analyzing 3D wings in general is for instance described in textbooks such as *Anderson (2005)*, but has also been used specifically for modelling two-element wingsails in the scientific literature, *Graf et al. (2014)*. Details of the algorithm can be found in *Anderson (2005)* chapter 5, while the implementation used specifically for this analysis can be viewed in the “LiftingLine” code on *Kramer (2016)*. The work presented in *Graf et al. (2014)* shows that the method works well for predicting the lift and drag on the sail while the flow is attached. The method can also work for stalled wings, which is shown both in *Anderson (2005)* and *Graf et al. (2014)*. However, *Graf et al. (2014)* show that the maximum lift coefficient can be over predicted compared to 3D CFD, and when the maximum lift coefficient is very large, there is sometimes problems with convergence for the iterative solver. The wing used in this analysis has a large maximum lift coefficient. From the 2D analysis, the maximum lift coefficient is 2.26, which happens with an angle of attack of 12.5°, and a flap angle of 15°. In order to avoid the convergence problems with the method, we have used the lifting line method for angles of attack almost up to stall for the largest flap angle, but not above. A stalled wing is in general of little interest, as this will only be useful when there is a tail wind, with a speed that is higher than the ship speed. The maximum angle of attack used with the lifting line method is 13.5°, which gives a lift coefficient of 2.04 for a flap angle of 15°. Larger angles of attack caused convergence problems for the largest flap angle, and based on the lift coefficient, this is fairly close to stall. The benefit of the method is calculation time. Since the wingsail is a two element wing, the forces depend on both the angle of

attack and the flap angle. The number of simulations that must be performed in order to get a complete picture of the forces on a wingsail can quickly become large. For instance, in this case, 4 flap angles have been simulated with at least 18 angles of attack each, giving more than 72 CFD simulations. 2D CFD allows for simulations with a higher resolution relative to the chord length, at a much shorter time, compared to the 3D case. The resulting lift and drag coefficients used in this analysis for the wingsail can be seen in Fig.6.

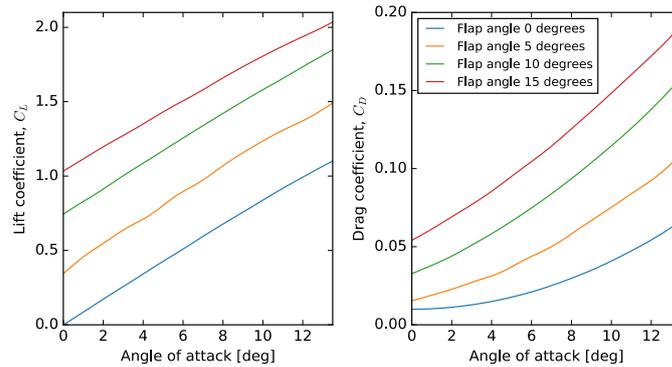


Fig.6: Lift and drag coefficients for the wingsail as function of angle of attack and flap angle

In order to get values for the lift and drag for the Flettner rotor, 3D CFD simulations have been used. The non-dimensional value for the spin velocity is called the spin-ratio ( $\alpha$ ), and is calculated as the velocity of the outer surface of the cylinder, divided by the incoming wind velocity. The flow around a Flettner rotor can be both steady and unsteady, depending on the spin-ratio, and both aspect ratio and Reynolds number have an effect on the resulting forces. The little experimental data that is available is only for very small Reynolds numbers, well below realistic conditions for a Flettner rotor on a cargo ship. It is therefore hard to say much about the uncertainty of the forces we have calculated. Many study this phenomenon using very high fidelity simulations, with many cells, small time steps and LES turbulence models. However, this is very time consuming, and only practical for smaller Reynolds numbers. The work presented in [Zhang et al. \(2013\)](#) show fairly good agreement between experimental values and steady state CFD values for both lift and drag, with meshes with less than 10 million cells, and a Reynolds number of 40000. The difference between simulation and experiments are between 1-15% depending on spin ratio, number of cells and turbulence model. The same approach was used to analyze the Flettner rotor in this paper, as it is both practical and relatively accurate. We have also simulated the case presented in [Zhang et al. \(2013\)](#), with the same setup script as used for our case. The “WingSimulation” class applies slightly different settings due to the low Reynolds number, for instance for the wall functions, but the overall rules for setting up the mesh are the same. The difference between simulations and experiment for the low Reynolds number case, as well as the values for the lift, drag and power coefficients for our high Reynolds number case can be seen in Fig.7.

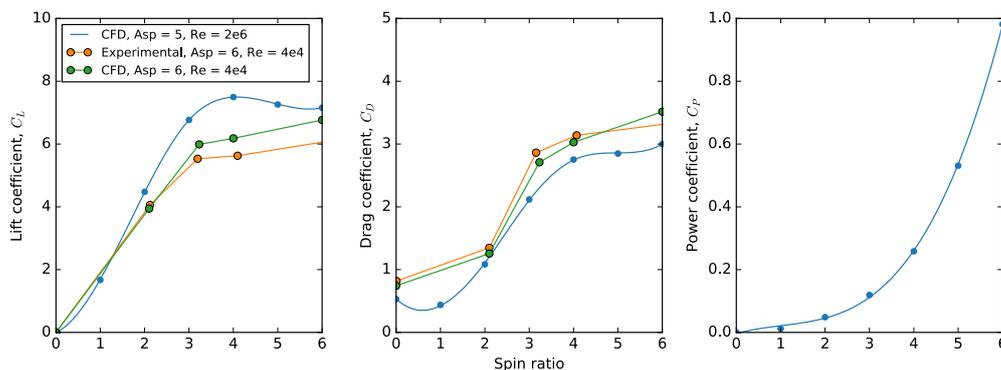


Fig.7: Lift, drag and power coefficient for the Flettner rotor

The lift and drag coefficients is used to calculate thrust and side force. Fig.8 shows the calculated thrust coefficient and side force to thrust ratio for the two different sails, with different ship speed to wind speed ratios, as a function of true wind direction.  $0^\circ$  are head wind,  $90^\circ$  are wind directly from the side and  $180^\circ$  are tail wind. The thrust coefficient is made non-dimensional with the wind velocity, so an increase in ship speed can actually increase the thrust coefficient for the wingsail. This is not the case for the Flettner rotor, which has a lower lift to drag ratio. The figure also shows the difference between the wingsail and the Flettner rotor when it comes to the amount of side force relative to the thrust. In general, the Flettner rotor has significantly higher side force, for the same amount of thrust.

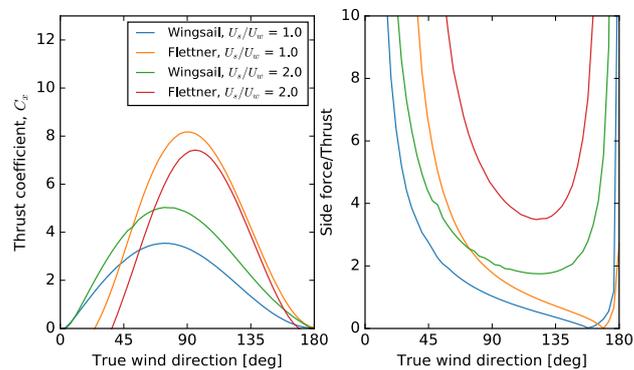


Fig.8: Thrust coefficient and side force to thrust ratio for the sails, at different ship speed to wind speed ratios

### 3.3 Route and wind

The wind data used in this analysis is taken from the European Centre for Medium-Range Weather Forecasts (ECMWF) ERA-interim reanalysis dataset *Berrisford et al. (2011)*. This dataset includes the wind velocity 10 m above the surface, covering the entire globe with a spatial resolution of  $0.75^\circ$ , and four time instances per day. Data from the beginning of the year 2000 until the end of 2015 is used in this analysis. The discrete points making up the route traveled by the ship is created by manually mapping out rough waypoints, and then calculating the great circle lines between the waypoints with the “Route” class located on *Kramer (2016)*. The distance between each discrete point is set to be 50 km. The route is plotted on top of the world in Fig.9, with the average wind velocity for the used dataset as a color map in the background. In order to find the wind velocity on a specific point and a specific time, cubic spline interpolation is used, with the help of the SciPy library. Details can be found in the “Wind” class on GitHub.

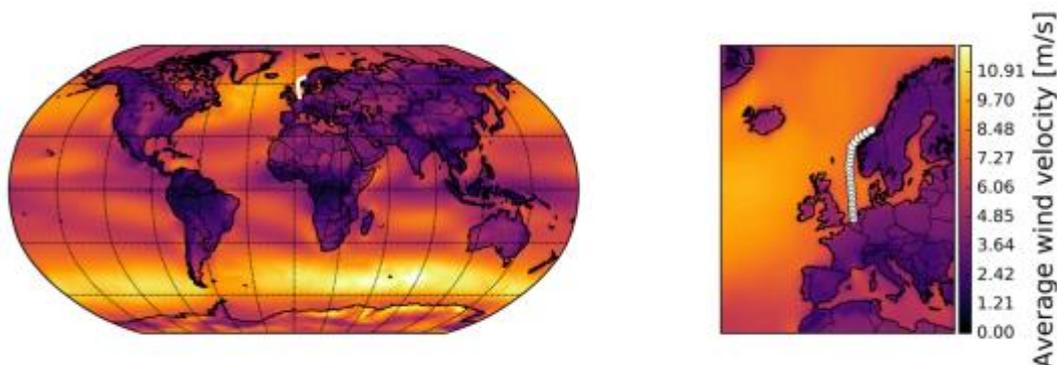


Fig.9: Example route plotted on maps with average wind speed color-mapped to the background

A histogram plot of the wind direction and velocity for this specific route can be seen in Fig.10. The wind direction is relative to the ships center line, where  $0^\circ$  is head wind,  $90^\circ$  side wind, and  $180^\circ$  tail wind. The wind data is only shown for  $0-180^\circ$  due to symmetry. In total, the number of individual discrete points with wind data for this route is 1,262,304. In order to decrease the computational time,

the dataset used in the simulation is reduced by randomly picking 10 000 points from the overall dataset. This reduction is not expected to alter the overall statistics. Both the reduced dataset and the full dataset is shown in Fig.10.

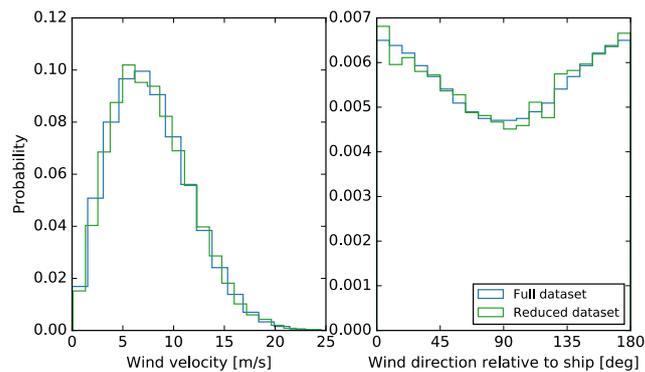


Fig.10: Wind statistics

#### 4. Route simulation

The route simulation is based on steady state evaluation of the ship at each discrete point in the route and weather data. The resulting statistics will therefore tell us how the ship will perform if it is located at a random place on the route, at a random time. Details on how exactly the ship is moving, i.e. when it is located at a specific point, is neglected, as it is considered to not be relevant for this study. Using the data presented in section 3, the following steps are performed in order to evaluate the ship and sails:

- The performance of the ship without sails is evaluated for the given ship speed. This includes wave resistance, friction resistance, with roughness, and propeller characteristics, such as efficiency.
- For a given wind speed, direction and sail loading, the forces on the sails are computed. That is, both thrust, side force and yaw moment.
- The necessary drift angle is found numerically using Newton's method, from the SciPy library. The input function to the numerical solver is a function that returns the side force from the sails, minus the side force from the hull, keel and rudder, with a drift angle as input. For a given drift angle, the rudder angle is calculated such that it balances the yaw moment. However, for an arbitrary sail loading, it is not guaranteed that there is a drift angle that provide balance both in terms of side force and yaw moment. In addition, the function that gives side force as function of drift, with the rudder always balanced might have local maxima/minima, which can be problematic for the numerical solver. In order to handle this problem, several initial values for the drift angle can be used. First,  $5^\circ$  are tried as default. If this does not lead to a solution, random values between  $0^\circ$  and  $30^\circ$  are tried, either until the maximum number of tries are reached, or a solution is found. The maximum number of tries is set to be 10. If the algorithm cannot find a solution, the hull drift angle is set to a very large value ( $90^\circ$ ), which causes the added resistance due to drift to become so large that the sail control algorithm will avoid the specific sail loading.
- When the necessary drift and rudder angle is found, the forces on the ship is recalculated, and the effective thrust is found by subtracting all the added resistance that is caused due to drift and rudder angles. The necessary power to the propeller is found by multiplying the total resistance on the ship hull, keel and rudder with the ship velocity, and dividing it with an estimated propeller efficiency.

In order to decide the sail control parameters for each discrete point in the route simulation, the built in sail optimization method is used, as described in section 3.2. The objective function in the sail optimization will be delivered power to the propeller, calculated with and without drift-induced

effects. When drift-induced effects are not considered, the wingsail will deliver maximum thrust, independent of what the consequences of this strategy will be, while the Flettner rotor will deliver maximum power. The power from the Flettner rotor is calculated as the thrust multiplied with the ship velocity, minus the required input power. When drift-induced effects are included, the sail loading might decrease in order to reduce added resistance due to drift and rudder angles. Even when drift-induced effects are not included in the optimization of the sail, there will always be an explicit check of how the performance of the ship would be without the sails “turned on”. That is, either how the ship would be without sails altogether, if the sails are retractable, or how it would be with the control parameters set to zero, if the sails are not retractable. If the control parameters from the sail optimization gives worse performance than a sail in “off position”, the sail control algorithm will choose to turn it off. This is to model a situation where the captain on board the ship will decide to turn of the sails, if he detects that the sail control program increases the fuel consumption.

## 5. Results

Fig. 11 and Fig. 12 show the predicted reduction in delivered power to the propeller, due to the sails, as a percentage of the necessary power in calm water without sails. Fig. 11 is the data for the non-retractable sails, while Fig. 12 is the data for the retractable sails. The power reduction is shown as a function of number of sails, as well as with and without drift-induced effects, with and without rudder and keel, and with and without hydrodynamics in the sail control.

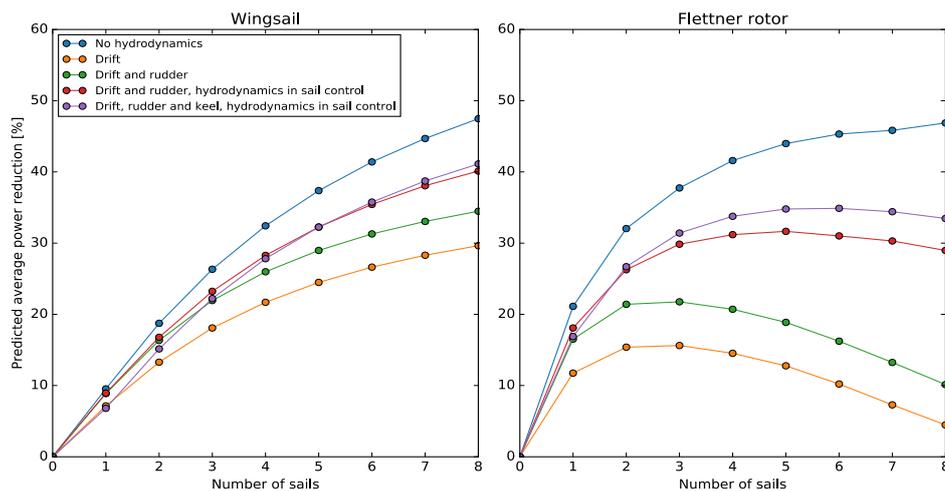


Fig. 11: Average power reduction, with non-retractable sails

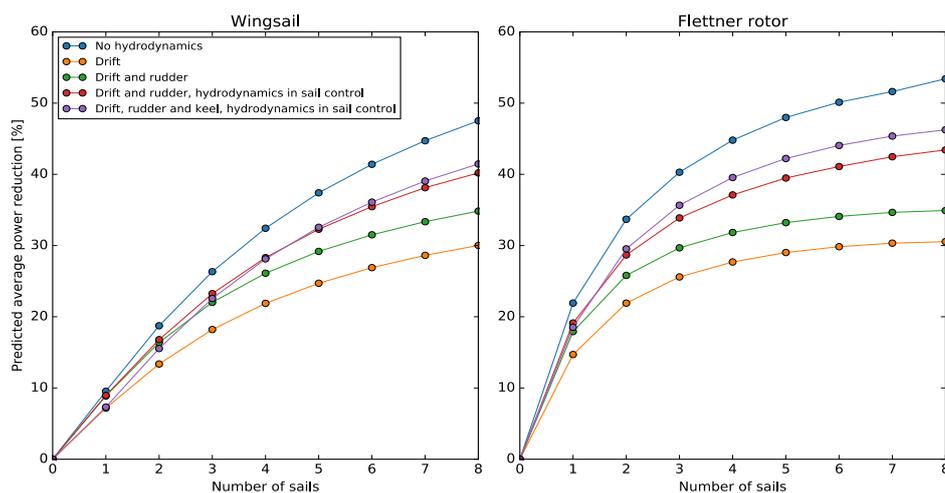


Fig. 12: Average power reduction, with retractable sails

A lot of data is generated in the route simulation, regarding the details of the ship as a system. Fig. 13 is used to represent some of this data. It shows histogram plots, and mean values of the drift and rudder angle, for the case with 6 non-retractable sails. Similar patterns can be generated for all the other cases as well, only with smaller/larger values, depending on how many sails there are. This figure is included, as it shows an important result, which is discussed in section 6.

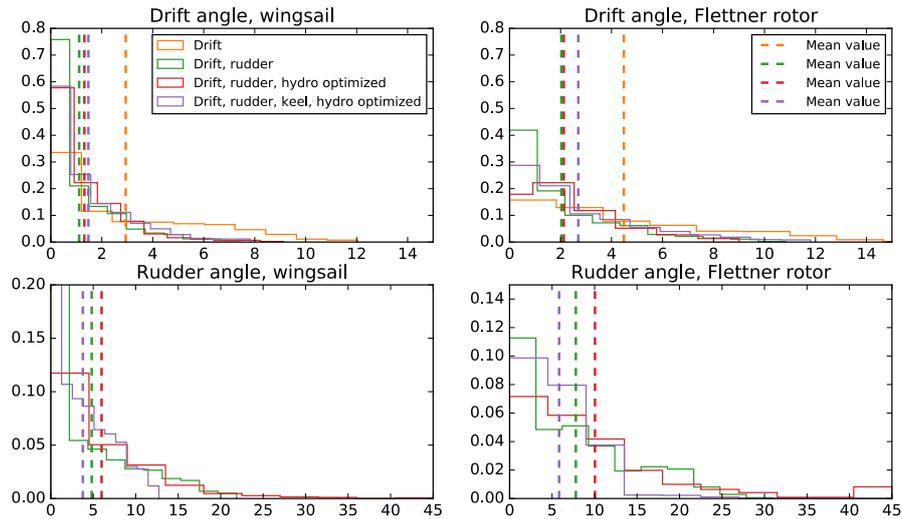


Fig. 13: Drift and rudder angle statistics, for 6 non-retractable sails

## 6. Conclusion and discussion

Whether drift is an important effect or not is dependent on the sail type, the amount of thrust generated from the sails, the sail control strategy, the sails ability to be stowed away, and of course the hydrodynamics of the ship hull. When just one sail is used, there is only a small reduction in the energy savings due to drift. When more sails are used, and the amount of thrust from the wind increases, the hydrodynamic effects get more and more important, which is not very surprising. What is interesting is the effectiveness of including the hydrodynamics into the control algorithm of the sails. For instance, with the most extreme example, which is the case with 8 non-retractable Flettner rotors, the energy savings due to sails are increased from roughly 10% to almost 30%, by just including the information about the hydrodynamics in the sail control. When the hydrodynamics of the ship hull is considered, the loading of the sail, or amount of thrust produced, is sometimes reduced. That is, sometimes it is better to produce less thrust from the sails, in exchange for less added resistance. Another consequence of considering hydrodynamics is that the sails can be used more often. In the route simulation, there is a very basic “captain model”, that will always turn off the sails, if having the sails on is worse than having them off. When we look at the mean values for drift angles in Fig. 13, we can see that it is slightly larger for the case with hydrodynamics in the sail control, than it is for the case without. Considering that drift causes added resistance, this might seem strange. If drift is the problem, how can a larger mean drift angle cause more power reduction for the ship? The reason for this is simply that the captain will allow the sails to be turned on more often when the hydrodynamics are included in the control algorithm. That is, without hydrodynamics in the control algorithm, the sails will sometimes produce so much side force that all the thrust, and more, is lost to drift-induced resistance. This will cause the captain to turn off the sails, which results in no thrust from the sails at all, but also smaller drift angles. By including hydrodynamics in the control algorithm, the sails will operate at a lower loading, ensuring that they actual produce positive effective thrust, but also a larger drift angle compared to the sails in off position. There is a clear difference in the importance of drift between wingsails and Flettner rotors. Flettner rotors, which have larger side force to thrust ratios in general, have more added resistance due to drift, for the same amount of thrust. This is true both for the retractable sails and the non-retractable sails, although the pattern is more clear for the non-retractable sails. For the Flettner rotor, the difference between retractable and non-retractable sails is large. This is explained by the relatively large drag coefficient in off-position.

Not only can a non-retractable sail generate drag by itself, it can also push the ship sideways, whenever the wind is coming from the side. It seems that a Flettner rotor in off position is a large source of added drift-induced resistance. The performance of the wingsail is much less affected by the ability to retract when not in use, as the drag coefficient in off-position is very small. It is interesting to see how the rudder is greatly increasing the performance. At first, one might think that the only purpose of a rudder is to balance the yaw moment from the sails. However, in doing so, the side force on the ship as a whole is greatly increased. As the rudder is an effective lifting surface, it is much better to produce side force with the rudder than it is to produce it with the ship hull. The fact that the keel has a very small effect on the overall performance of the ship can also be explained by this. Since the keel is increasing the stabilizing yaw moment on the ship hull, there is less need for the rudder. However, the rudder was not actually the problem. Since the rudder is less needed for balancing the yaw moment, it produces less side force, which must instead be balanced by the ship hull and keel. This is part of the reason why the mean drift angle is increased when the keel is added. Another reason is that the sail control algorithm allows larger drift angles, as the keel improves the drift characteristics of the ship hull. The influence of the rudder could change significantly if the balance of the hull was different. For instance, if the yaw restoring moment from the hull was larger than the yaw moment from the sails, the rudder would have to be turned in the opposite direction, in order to balance the ship. This would produce a side force in the same direction as the sails, which would increase the drift angle. Since the yaw moment from the keel affects how the rudder is used, the position of the keel can probably be optimized to give better results. From a purely steady state perspective, where rudder stall is not an issue, the more optimal position would be further forward, so that it generates less stabilizing yaw moment. However, this could be problematic from a maneuvering perspective, as the necessary rudder angle for turning might increase. Maneuvering and hull balance is in fact already an issue. The necessary rudder angle calculated by the route simulation code is sometimes larger than 30 degrees, which would probably cause the rudder to stall in reality. That is, some of the events that happened in the simulation is not realistic, and in reality, the sail loading would need to be reduced in order to avoid rudder stall. This would further reduce the predicted energy savings due to the sails. The rudder angle is in general larger when Flettner rotors are used, than it is when wingsails are used, which means that this problem is more severe for the Flettner rotor. Moving the keel further back should help the rudder stall problem. Even if the rudder is not stalling in steady state condition, it might stall if a turning maneuver is necessary, which is an argument for putting the keel further back.

## Acknowledgements

We would like to thank the Research Council of Norway, DNV GL and Rolls-Royce Marine AS for funding this research through the research program called “Low Energy and Emission Design of Ships” (LEEDS) (Grant no. 216432/O70) at the Department of Marine Technology, NTNU.

## References

- ANDERSON, J.D. (2005), *Fundamentals of Aerodynamics*, McGraw-Hill
- BERRISFORD, P. et al. (2011), *The ERA-Interim archive Version 2.0*, ERA Report Series 1, ECMWF, Shinfield Park, Reading
- BERTRAM, V., 2012. *Practical ship hydrodynamics*. Oxford: Elsevier.
- CATMULL, E.; CLARK, J. (1978), *Recursively generated B-spline surfaces on arbitrary topological meshes*, *Computer-Aided Design* 10(6), pp.350-355
- CHARRIER, B. et al. (1985), *Foundation Cousteau and windship propulsion*, *J. Wind Eng. and Industrial Aerodynamics* 20, pp.39-60
- EBERHART, R.C.; KENNEDY, J. (1995), *A new optimizer using particle swarm theory*, 6<sup>th</sup> Symp.

Micro Machine and Human Science, New York, pp.39-43

ECA, J.; HOEKSTRA, M. (2008), *The numerical friction line*, J. Marine Science and Technology 13(4), pp. 328-345

GRAF, K.; HOEVE, A.; WATIN, S. (2014), *Comparison of full 3D-RANS simulations with 2D-RANS/lifting line method calculations for the flow analysis of rigid wings for high performance multihulls*, Ocean Engineering 90, pp.49-61

HIRT, C.W.; NICHOLS, B.D. (1981), *Volume of fluid (VOF) method for the dynamics of free boundaries*, J. Computational Physics 39(1), pp.201-225

HYMPENDAHL, O.; CIORTAN, C. (2015), *Systematic assessment of model errors in CFD ship resistance simulations*, NuTTS, Cortona

ISSA, R.I.; GOSMAN, A.; WATKINS, A. (1986), *The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme*, J. Comp. Physics 62(1), pp. 66-82

KRAMER, J. (2016), GitHub profile, <https://github.com/jarlekrmer>

KRAMER, J.; STEEN, S. (2015), *Importance of the free surface for the drift-induced forces on a ship-like foil*, NuTTS, Cortona

KRAMER, J.A.; STEEN, S.; SAVIO, L. (2016), *Experimental study of the effect of drift angle on a ship-like foil with varying aspect ratio and bottom edge shape*, Ocean Eng. 121, pp.530-545

MENTER, F.R. (1994), *Two-equation eddy-viscosity turbulence models for engineering applications*, AIAA J. 32(8), pp.1598-1605

OUCHI, K.; UZAWA, K.; KANAI, A.; KATORI, M. (2013), *“Wind Challenger” the Next Generation Hybrid Sailing Vessel*, 3<sup>rd</sup> SMP, Launceston, pp.562-567

PATANKAR, S.V.; SPALDING, D.B. (1972), *A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows*, Int. J. Heat and Mass Transfer 15(10), pp.1787-1806

PRANDTL, L.; TIETJENS, O.G. (1934), *Fundamentals of Hydro- and Aeromechanics*, Dover Publ.

RAVEN, H.C.; PLOEG, A.v.d.; ECA, L. (2006), *Extending the benefit of CFD tools in ship design and performance prediction*, 7<sup>th</sup> Int. Conf. Hydrodyn., Ischia, pp.573-580

SÖDING, H. (1998), *Limits of potential theory in rudder flow predictions*, Ship Technology Research 45(3)

SPALDING, D. (1961), *A single formula for the “law of the wall”*, J. Applied Mech. 23(3), pp.455-458

TRAUT, M. et al. (2014), *Propulsive power contribution of a kite and a Flettner rotor on selected shipping routes*, Applied Energy 113, pp.362-372

WALKER, J.G. (1985), *A high performance automatic wingsail auxiliary propulsion system for commercial ships*, J. Wind Eng. and Industrial Aerodynamics 20, pp.83-96

ZHANG, W.; BENSOW, R.; GOLUBEV, M.; CHERNORAY, V. (2013), *Flow past a rotating finite length cylinder: numerical and experimental study*, AIAA, Texas