Hydrodynamics
Modeling and Simulation

*FloBoS* – A Time Domain Floating Bodies Simulator

**Supervisor**
Prof. Lorenzo Casalino  
Politecnico di Torino

**Student**
Orazio Pinti

**Mentor**
Dr. Marco B. Quadrelli  
NASA Jet Propulsion Laboratory – California Institute of Technology

March 2018
Linearization of the hydrodynamic problem: Strip Theory

The numerical solution of the nonlinear boundary value problem is possible, but very complex and computationally expensive. To simplify the problem resolution, one can carry out the linearization of the boundary conditions. In order to proceed, some restrictions on the nature of the problem are necessary:

- The wet hull must be slender;
- The speed of the ship can’t be too high;
- Waves amplitude must be small;
- Motions amplitude of the ship must be small.

Different combinations of restrictions result on different linear formulations; the adequacy of the simplification depends on the physical problem intended to represent. In our case we will use the Strip Theory.

Indeed, for slender bodies, a good approximation is that the fluid motion field varies much slower in the longitudinal direction of the ship than in the cross-directional plane. Hence, the problem can be reformulated bidimensionally. The principle of strip theory, a low Froude number theory, is to divide the submerged part of the vessel into a finite number of strips and analyze each one separately. After solving a set of bidimensional boundary value problems, one can compute the frequency dependent bidimensional coefficients of added masses and damping for each strip.

Hence, integrating the action of each 2D coefficient along the length of the vessel, one can estimate the overall tridimensional coefficients of the whole hull.

For more details see for example [Faltinsen 1990].

One of the problem of the strip theory is that we can not trace back to the surge force, because is not present in the bi-dimensional problems from which we obtain the tridimensional quantities.

To add the drag force we will use empirical formulas, based on the geometry of the ship, the operational conditions and the characteristic Reynolds number.

Moreover, in FloBoS, the steady part of the potential \( \varphi_0 \) is neglected. The steady perturbation due to the presence of the hull at rest is considered negligible with respect to the sea waves, or of the same approximation degree of the linear hypothesis: this approach is also called of the ghost vessel.

![Figure 1: Strip Theory – discretization example of a 10% scaled container ship S175](image-url)
Part I

Time domain simulation

Time domain models based on frequency domain data are useful both for simulation and control systems design. Furthermore, adding non linear effects for higher accuracy is easier with a time domain formulation.

[...]

Seakeeping simulation

Evaluating seakeeping simulation techniques, Cummins equation expressed in the seakeeping frame is the starting point:

\[
(M_{RB} + A_\infty) \ddot{\xi} + B_\infty \dot{\xi} + \int_0^t K(t - t') \dot{\xi}(t') \, dt' + G\xi = \tau_{FK+Diff}
\]

In order to design and code a simulator, one have to face two problems:

- The convolution term is not efficient for time domain simulations; it implies the storage of a big amount of data to compute the integral at each time step. For this reason, different methods have been proposed in the literature as approximate alternative representations of the convolutions. Because the convolution is a linear operator, different approaches can be followed to obtain an equivalent linear system in the form of either transfer function or state-space models. For an overview on the main methods for replacing the convolutions and a comparison of the different methods in terms of complexity and performance, please see [Taghipour2008] or [PerezFossen2008bis].

- The second problem is to approximate the infinite-frequency added mass and damping matrices \(A_\infty\) and \(B_\infty\), due to the fact that the bidimensional hydrodynamic code PDSTRIP does not provide such estimation.

A method to identify both the convolution term and the infinite added mass matrix in one fell swoop has been proposed by [PerezFossen2008]; we are going to develop a method starting from their work, to find also the matrix \(B_\infty\) and then implement it in our simulator.

Convolution term properties

Before exposing the previously mentioned method of [PerezFossen2008], we will focus on the properties of the convolution term, useful to impose the conditions for building its estimation. The expressions both in time and frequency domain are:

\[
K(t) = \int_0^\infty (B(\omega) - B_{\infty}) \cos(\omega t) \, d\omega
\]

\[
K(j\omega) = B(\omega) - B_{\infty} + j\omega (A(\omega) - A_{\infty})
\]

- For \(t \to 0^+\) it results:

\[
K(0^+) = \int_0^\infty (B(\omega) - B_{\infty}) \, d\omega \neq 0 < \infty
\]

because the functions \(B_{ik}(\omega) - B_{\infty ik}\) are all bounded.

- From the Riemann-Lebesgue Lemma it follows that, for \(t \to \infty\):

\[
\lim_{t \to \infty} K(t) = \lim_{t \to \infty} \int_0^\infty (B(\omega) - B_{\infty}) \cos(\omega t) \, d\omega = 0
\]

That implies the important input-output stability property of the convolution term: indeed, in order to have each term \(\int_0^\infty K_{ij}(t - t') \dot{\xi}_j(t') \, dt'\) bounded for any bounded excitation \(\dot{\xi}_j(t)\), it is necessary that \(\int_0^\infty |K_{ij}(t)| \, dt < \infty\), which holds provided 2.
Using the Riemann-Lebesgue Lemma, it is also verified that for \( \omega \to 0 \):
\[
\lim_{\omega \to 0} K (j\omega) = 0
\]

- For \( \omega \to \infty \), \( B (\omega) \to B_{\infty} \), and from the same Lemma:
\[
\lim_{\omega \to \infty} \omega [A (\omega) - A_{\infty}] = \int_0^{\infty} K (t) \sin (\omega t) \, dt = 0
\]
so it holds also:
\[
\lim_{\omega \to \infty} K (j\omega) = 0
\]

- The last important property is the passivity of \( K (j\omega) \). Passivity establishes that there is no generation of energy within physical system, i.e. the system can either store or dissipate energy. In our case it derives from the fact that radiation forces are dissipative. To reflect this property in the mathematical model, thanks to the linearity, we have just to assure positive realness: the real part of the frequency response function must be non zero. The damping matrix is symmetric and positive-semi definite - \( B (\omega) = B^T (\omega) \geq 0 \), [Newman1997] - so \( K (j\omega) \) is positive real and thus passive. This implies that the diagonal elements of the matrix \( K (j\omega) \) are positive real and the off-diagonal terms needs only to be stable, [Taghipour2008].

Expression ??, given \( A_{\infty} \) and \( B_{\infty} \), allows to compute the values of the frequency response function \( K (j\omega) \) for a finite set of available data. Thanks to linearity, one can seek a transfer function approximation \( \tilde{K} (s) \), using these values for the identification method:
\[
\tilde{K}_{ik} (s) = \frac{P_{ik} (s)}{Q_{ik} (s)} = \frac{p_r s^r + p_{r-1} s^{r-1} + \ldots + p_0}{s^n + q_{n-1} s^{n-1} + \ldots + q_0}, \quad i, k = 1, \ldots, 6
\]  (3)

In order to carry out a good estimation of the approximations \( \tilde{K}_{ik} (s) \), one have to exploit both the non-parametric data \( K_{ik} (j\omega) \) from ??, and the properties previously listed. The following table from [Perez&Fossen2008] summarizes these properties and their implications on the transfer function 3:

<table>
<thead>
<tr>
<th>Properties</th>
<th>Implication on Parametric Models ( \tilde{K}_{ik} (s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - ( \lim_{t \to 0^+} K (t) \neq 0 &lt; \infty )</td>
<td>Relative degree between ( P_{ik} (s) ) and ( Q_{ik} (s) ) is 1</td>
</tr>
<tr>
<td>2 - ( \lim_{t \to \infty} K (t) = 0 )</td>
<td>Bounded-input, bounded-output (BIBO) stability</td>
</tr>
<tr>
<td>3 - ( \lim_{\omega \to 0} K (j\omega) = 0 )</td>
<td>( P_{ik} (s) ) has zeros in ( s = 0 )</td>
</tr>
<tr>
<td>4 - ( \lim_{\omega \to \infty} K (j\omega) = 0 )</td>
<td>( \text{deg} (Q_{ik} (s)) &gt; \text{deg} (P_{ik} (s)) )</td>
</tr>
<tr>
<td>5 - Passivity of ( K (j\omega) )</td>
<td>( K (j\omega) ) is positive real</td>
</tr>
</tbody>
</table>

Table 1: Properties of retardation functions

Using these informations to set constraints on the model's structure is fundamental to fit a transfer function which will assure a good estimation.

**Frequency domain identification problem**

From property 3 of Table 1, it is known that \( P_{ik} (s) \) must have the form \( s^l P'_{ik} (s) \), so that:
\[
\tilde{K}_{ik} (s) = \frac{s^l P'_{ik} (s)}{Q_{ik} (s)} = \frac{s^l (p_m s^m + p_{m-1} s^{m-1} + \ldots + p_0)}{s^n + q_{n-1} s^{n-1} + \ldots + q_0}
\]
Furthermore using property 1 and 4, the orders of the polynomials must satisfy the relation \( l + m + 1 = n \). Since in general \( A(0) \neq A_{\infty} \), from ?? it follows that there is a unique zero of \( K_{ik}(s) \) in \( s = 0 \), so \( l = 1 \) and \( n - m = 2 \). That also means that the lowest possible order of approximation is \( m = 0 \) and \( n = 2 \). The final form of the rational approximation is:

\[
\hat{K}_{ik}(s) = \frac{\hat{P}_{ik}'(s)}{\hat{Q}_{ik}(s)} = \frac{p_0 + p_1 s + \cdots + p_m s^m}{s^{m+2} + q_1 s^{m+1} + \cdots + q_0} \tag{4}
\]

The problem is now reduced to find the vector of parameters \( \theta = [p_0, \ldots, p_m, q_{m+1}, \ldots, q_0] \), which perfectly defines the \((i, k)\) element of the approximated transfer function \( \hat{K}(s) \).

In order to exploit the available data \( A(\omega_l) \) and \( B(\omega_l) \), which give the exact values \( K(j\omega_l) \) from ??, we can define the auxiliary function

\[
\hat{K}_{ik}(j\omega_l) = \frac{K_{ik}(j\omega_l)}{j\omega_l} \tag{5}
\]

and shape the data in a similar manner of 4, and set the identification problem to find \( P_{ik}'(s) \) and \( Q_{ik}(s) \). One can think to a complex least square curve fitting problem, to minimize the overall displacement of 4 with respect to the points defined by 5. The vector of parameters to choose is the argument of the result of the following minimization problem:

\[
\theta^* = \arg \min_{\theta} \sum_l w_l \left| \hat{K}_{ik}(j\omega_l) - \hat{z}_{ik}(j\omega_l) \right|^2 = \arg \min_{\theta} \sum_l w_l \left| \hat{K}_{ik}(j\omega_l) - \frac{P_{ik}'(s; \theta)}{Q_{ik}(s; \theta)} \right|^2 \tag{6}
\]

The weights \( w_l \) can be chosen in order to emphasize a particular range of frequencies. This kind of parameter estimation is non linear in the parameters: it can be solved with Gauss-Newton algorithms or by linearization, as proposed by [Levy1959].

For other identification methods, both in time and in frequency domain, in order to analyze other ways to estimate \( \hat{K}(t) \) or \( \hat{K}(s) \), [Taghipour2008, Perez&Fossen2008b] provides an exhaustive summary of the state of the art.

[...]  

**Alternative identification method for non-zero forward speed**

Starting from the previous work of [Perez&Fossen2008], we can develop a method to identify also the matrix \( B_{\infty} \): indeed, in the case of a moving vessel, the infinite frequency damping matrix does not vanish.

The Laplace transform of ?? in the general case is:

\[
\tilde{r}_{rad,i}(s) = -A_{\infty,ik}\tilde{\eta}_k(s) - \left[ B_{\infty,ik} + \frac{P_{ik}(s; \theta)}{Q_{ik}(s; \theta)} \right] \tilde{\eta}_k(s) = - \left[ A_{\infty,ik} + \frac{B_{\infty,ik}}{s} + \frac{P_{ik}(s; \theta)}{sQ_{ik}(s; \theta)} \right] \tilde{\eta}_k(s)
\]

We can write the former parenthesis as a polynomial fraction \( \frac{R_{ik}(s; \theta)}{S_{ik}(s; \theta)} \), such that:

\[
\frac{R_{ik}(s; \theta)}{S_{ik}(s; \theta)} = \frac{sQ_{ik}(s; \theta) \hat{A}_{\infty,ik} + P_{ik}(s; \theta) \hat{B}_{\infty,ik} + P_{ik}(s; \theta)}{sQ_{ik}(s; \theta)} \tag{7}
\]

By the means of the coefficients ??, computed using the set of data, we can set the same minimization problem

\[
\theta^* = \arg \min_{\theta} \sum_l w_l \left| \hat{A}_{ik}(j\omega_l) - \frac{R_{ik}(s; \theta)}{S_{ik}(s; \theta)} \right|^2 \tag{8}
\]

in order to find the polynomials of the rational function 7.

Once 8 is solved, we have both \( R_{ik}(s) \) and \( S_{ik}(s) \), and we can trace back to the unknowns \( \hat{A}_{\infty,ik}, \hat{B}_{\infty,ik} \) and \( \frac{P_{ik}(s; \theta)}{Q_{ik}(s)} \), to reconstruct the radiation force \( \tilde{r}_{rad,i}(s) \).

The polynomial function \( Q_{ik}(s) \) is given simply by \( \frac{S_{ik}(s)}{P_{ik}(s; \theta)} \); to continue the identification, we have to exploit the information a priori about \( P_{ik}(s) \) and \( Q_{ik}(s) \). We know that, calling \( n \) the degree of \( Q_{ik}(s) \), then the degree of \( P_{ik}(s) \) is \( n - 1 \), and, obviously, the degree of \( sQ_{ik}(s) \) is \( n + 1 \). That means that we easily find \( \hat{A}_{\infty,ik} \) as the highest degree coefficient of \( R_{ik}(s) \).
The last unknowns to find are the value of $\hat{B}_{\infty,ik}$ and the polynomial $P_{ik}(s)$. We can write the general form of the polynomials as:

\[
R_{ik}(s) = r_{n+1}s^{n+1} + r_ns^n + \cdots + r_1s + r_0,
Q_{ik}(s) = q_ns^n + q_{n-1}s^{n-1} + \cdots + q_1s + q_0,
P_{ik}(s) = p_{n-1}s^{n-1} + p_{n-2}s^{n-2} + \cdots + p_1s + p_0.
\]

From 7 we know that these polynomials are related by:

\[
sQ_{ik}(s; \theta) \hat{A}_{\infty,ik} + Q_{ik}(s; \theta) \hat{B}_{\infty,ik} + P_{ik}(s; \theta) = R_{ik}(s; \theta)
\]

which, extending, turns into:

\[
s \left[ q_ns^n + q_{n-1}s^{n-1} + \cdots + q_0 \right] \hat{A}_{\infty,ik} + \left[ q_ns^n + \cdots + q_0 \right] \hat{B}_{\infty,ik} + \left[ p_{n-1}s^{n-1} + \cdots + p_0 \right] =
\]

\[
= r_{n+1}s^{n+1} + \cdots + r_0
\]

Furthermore we know that $P_{ik}(s)$ has a zero in $s = 0$, which means $p_0 = 0$, and $Q_{ik}(s)$ is monic, so $q_n = 1$. That leads to:

\[
[s^{n+1} + q_{n-1}s^n + \cdots + q_0] \hat{A}_{\infty,ik} + [s^n + q_{n-1}s^{n-1} + \cdots + q_0] \hat{B}_{\infty,ik} + [p_{n-1}s^{n-1} + \cdots + p_1s] =
\]

\[
= r_{n+1}s^{n+1} + \cdots + r_0
\]

Comparing term by term, we reach $n + 2$ relations:

\[
\begin{align*}
  r_{n+1} &= \hat{A}_{\infty,ik}, \\
  r_n &= q_{n-1}\hat{A}_{\infty,ik} + \hat{B}_{\infty,ik}, \\
  r_{n-1} &= q_{n-2}\hat{A}_{\infty,ik} + q_{n-1}\hat{B}_{\infty,ik} + p_{n-1}, \\
  \vdots \\
  r_k &= q_{k-1}\hat{A}_{\infty,ik} + q_k\hat{B}_{\infty,ik} + p_k, \\
  \vdots \\
  r_1 &= q_0\hat{A}_{\infty,ik} + q_1\hat{B}_{\infty,ik} + p_1, \\
  r_0 &= q_0\hat{B}_{\infty,ik}
\end{align*}
\]

where the coefficients $r_j, q_j$ and the value of $\hat{A}_{\infty,ik}$ are all known. Calling $\sum_j r_j = R$, $\sum_j q_j = Q$ and $\sum_j p_j = P$, we can sum the former relations, reaching the new one:

\[
R = Q \cdot \hat{A}_{\infty,ik} + Q \cdot \hat{B}_{\infty,ik} + P
\]

With the two 9 and 10, we can set an iterative process to find both $\hat{B}_{\infty,ik}$ and $P_{ik}(s)$. Indeed the system

\[
\begin{bmatrix}
  [P_{ik}(s; \theta)]_l \\
  [\hat{B}_{\infty,ik}][l+1] \\
  \end{bmatrix}
= R_{ik}(s; \theta) - sQ_{ik}(s; \theta) \hat{A}_{\infty,ik} - Q_{ik}(s; \theta) \hat{B}_{\infty,ik}
\]

\[
= \begin{bmatrix}
  [\hat{B}_{\infty,ik}]_0 \\
  \sum_j [P_j]_l \\
  \end{bmatrix}
= \hat{A}_{\infty,ik} + \frac{R - P}{Q}
\]

can eventually converge to the unknowns. The first step is to initialize a value: due to the fact that we do not have any prior information about the polynomial $P_{ik}(s)$, we have to choose $[\hat{B}_{\infty,ik}]_0$ from the data set of $B_{ik}(\omega_l)$ we can take the value corresponding to the maximum frequency $\omega_{\text{max}} = \max_l \omega_l$ as a good initial approximation of $B_{\infty,ik}$, so $[\hat{B}_{\infty,ik}]_0 = B_{ik}(\omega_{\text{max}})$.

The complete algorithm becomes:
At the end we find $\hat{A}_{\infty,ik}, \hat{B}_{\infty,ik}$ and $K(s) = \frac{P_{ik}(s)}{Q_{ik}(s)}$. The evaluations about the transfer function order and the model quality check are the same described in the previous sections, exception for that, if the speed is not zero, the inverse relations to re-compute the approximated $A_{ik}(\omega)$ and $B_{ik}(\omega)$ are:

$$\hat{A}_{ik}(\omega) = \sum_{k=0}^{N-1} R_{n+k}(s; \theta) S_{ik}(s; \theta)$$

Once we have the approximated transfer functions of $K_{ik}(s)$, we can find the memory effect term in the time domain, exploiting the relation ??%. Indeed, by taking the Laplace transform of

$$\mu(t) = \frac{1}{\omega} Im \left\{ \hat{K}_{ik}(j\omega) \right\}$$

we obtain

$$\mu_i(s) = K_{ik}(s) \hat{\xi}_k(s) \approx \frac{P_{ik}(s)}{Q_{ik}(s)} \hat{\xi}_k(s) \tag{11}$$

but the Laplace transform of the state-space formulation

$$\mu(t) \approx \begin{cases} \dot{x} = A_s x + B_s \hat{\xi} \\ \mu = C_s x \end{cases} \tag{12}$$

leads to

$$\mu(s) = C_s (sI - A_s)^{-1} B_s \hat{\xi}(s) \tag{13}$$

Hence, a correlation between 11 and 13 can be made to find the constant matrices $A_s$, $B_s$ and $C_s$ of the state-space formulation, starting form the frequency domain transfer functions $\frac{P_{ik}(s)}{Q_{ik}(s)}$, then, $A_s$, $B_s$ and $C_s$ can be used in 12 for the time domain approximation of $\mu(t)$.

**Cummins equation resolution**

Once we found a time domain approximation of the memory effect term $\mu(t)$ and the constant matrices $A_\infty$ and $B_\infty$, we can proceed with the numerical resolution of the Cummins equation

$$(M_{RB} + A_\infty) \hat{\xi} + B_\infty \hat{\xi} + \mu(t) + G \xi = \tau_{FK+Diff}$$
From potential theory we can compute the dynamic pressure due to incident and diffracted waves and, after the integration over the wet surface of the vessel, we can find the Froude-Krilov and diffraction force vector $\tau_{FK+Diff}^s$.

Writing the equation as

$$\ddot{\xi} = -(M_{RB} + A_\infty)^{-1} G \xi - (M_{RB} + A_\infty)^{-1} B_\infty \dot{\xi} + (M_{RB} + A_\infty)^{-1} \left( \tau_{FK+Diff}^s - \mu (t) \right)$$

we can reduce the order of the problem defining the following state vector

$$y (t) = \begin{bmatrix} \xi (t) \\ \dot{\xi} (t) \end{bmatrix}$$

Indeed, the time derivative of $y (t)$ can be written in a linear system form as:

$$\dot{y} (t) = W \cdot y (t) + l (t) = f (t, y (t))$$

(14)

where $W$ is the constant matrix

$$W = \begin{bmatrix} 0_{6 \times 6} & I_{6 \times 6} \\ -(M_{RB} + A_\infty)^{-1} G & -(M_{RB} + A_\infty)^{-1} B_\infty \end{bmatrix}$$

and $l (t)$ the time dependent vector

$$l (t) = \begin{bmatrix} 0_{6 \times 1} \\ (M_{RB} + A_\infty)^{-1} \left( \tau_{FK+Diff}^s - \mu (t) \right) \end{bmatrix}$$

As we can see, in order to integrate the motion equation 14, we need the initial condition $y (0)$, so the starting position and the velocity of the ship with respect his equilibrium position. For the memory effect term, a parallel update can be performed separately each time step, initializing the state vector $x$ to zero $x (0) = 0$.

$$\begin{cases} \dot{x} (t) = A_s x (t) + B_s \dot{\xi} (t) \\ \mu (t) = C_s x (t) \\ x (0) = 0 \end{cases}$$

(15)

Finding the excitation forces, by integration of the pressure on the hull, implies the computation of $\tau_{FK+Diff}$ at each step; indeed, the computation must be carried out on the actual position of the hull, so it’s possible only after updating it, integrating the motion equation.

We can apply several numerical methods to integrate both 14 and 15, for example a Runge-Kutta method.

The general algorithm to proceed with the integration is:

$$\xi (0), \dot{\xi} (0) \rightarrow y (0)$$

$$\xi (0) \rightarrow \tau_{FK+Diff} (0)$$

$$x (0) \rightarrow \mu (0)$$

$$\begin{cases} y (0) \\ \tau_{FK+Diff} (0) \rightarrow y (\Delta t) \\ \mu (0) \end{cases}$$

$$\xi (k\Delta t) \rightarrow \tau_{FK+Diff} (k\Delta t)$$

$$x (k\Delta t) \rightarrow \mu (k\Delta t)$$

$$\begin{cases} y (k\Delta t) \\ \tau_{FK+Diff} (k\Delta t) \rightarrow y ((k+1)\Delta t) \\ \mu (k\Delta t) \end{cases}$$

The possibility to add a forward speed $U$ is implemented.

[...]
Part II

FloBoS - An overview

The simulator assembles all the theoretical and technical apparatus mentioned before, implementing it in Matlab. The first part of SeaBoS and ManBoS has been written starting from the floating frequency domain Ship Response Simulator of [Musci2015]: the data pre-processing section of it has been adapted to new needs and recoded to make it more easily transferable in Python. Then, the time domain features has been added, as the optimization algorithms, the time integration of the motion equations, the manoeuvring simulator (possible only in time domain) with the related equations and the graphic part.

Starting from the shape of the vessel, given as a series of cross sections defined by points, every needed geometrical property is extrapolated: length of each strip element, mean and maximum draft, beam and freeboard, actual wet hull, normal and tangent unit vectors, each section’s area, water plane area, area moments, buoyancy center, fluid displaced volume etc. To accomplish these tasks, the work of [Musci2015] has been adapted.

The software PDSTRIP solves a bidimensional boundary value problem for each $j$th cross section, which are globally $n_{sec}$ finding its bidimensional added mass and potential damping matrices, $a_j(\omega_k)$ and $b_j(\omega_k)$, for each of 52 frequencies $\omega_k$ that PDSTRIP chooses for each vessel, based on its dimension.

After an integration along the ship length, the tridimensional frequency dependent matrices of the whole hull, $A(\omega_k)$ and $B(\omega_k)$, are computed.

Once the vessel is completely defined with all the needed geometrical data, a static stability analysis is carried out: both longitudinal and transverse metacentric stability are evaluated and the restoring coefficients are computed.

To proceed with the real time domain simulation, the infinite frequency matrices $A_\infty$ and $B_\infty$ must be computed, as the definition of the state-space model matrices for the fluid memory effect term $\mu(t)$. In order to do that, specifying what simulator is intended to run, SeaBoS or ManBoS, is needed. The differences are in the algorithm to find $A_\infty$ and $B_\infty$ and in the motion equation integrated during the simulation.

Moreover, to start a manoeuvring analysis with ManBoS, a set of target points must be inserted: the vessel will be controlled with the rudder, by the means of the PD controller, with the objective to reach the target positions in the selected order.

Concerning the sea state, the waves harmonics can be defined manually or by uniform sampling of a Torsethaugen spectrum, which requires as inputs $H_{m0}$, $T_p$ and $\theta_0$; with the harmonics data, the computation of the excitation forces can be made during each time step of the simulation.

After the simulation, performed until the final time specified by the user, a post-processing phase starts. The dynamical responses are plotted and a vessel animation is shown.

In what follows we will give a more detailed description of FloBoS, taking a look in each of his part.

A scheme of how SeaBoS works is presented:
Figure 2: SeaBoS block diagram
Code Logic description

A brief summary of the simulator code is now presented. Only SeaBoS is considered; indeed, for a qualitative description, the two simulators structures are similar, except for small details. SeaBoS is composed by 16 sections; we are going to quickly analyze each one.

1. Create the Sea Spectrum
   This first part of the code is dependent on the variable Man Spec: if its value is 0, WAFO routines will be called to generate the sea spectrum, by the means of the variables specified in the section Sea State - Waves spectrum of FloBoS; then the discrete simple harmonics will be defined by a sampling. Otherwise the harmonics can be defined in this section by the user. Independently by Man Spec, at the end of this first section the vectors zeta, T and beta will be assigned and therefore available to represent the approximated sea.

2. Time simulation Data
   The variables concerning the simulation time are created: the size of the time step is set by default to 0.05 seconds; so the number of steps is computed as the ratio between the final time imposed by the user and this time step. A time vector is also defined, containing all the values from zero to the final time, all spaced by the size of the time step.

3. Prepare Input Data
   Geometry data are extracted by the file namegeom and the first variables concerning the vessel are created. There is, of course, the dependence of the file format, that is if the file is structured for PDSTRIP or the if it's .mgf. Having the length of the ship, a check about the Froude number can be effectuated: the value should not be greater than 0.4.

4. Geometrical Properties of the Ship
   The few geometrical data inserted by the user are processed to deduce all the other needed quantities. After all the other mentioned operations (like computing the strips spacing, the vessel draft, beam and center of buoyancy, the elements lengths, area moments and unit vectors) a deck is also added and the actual wet hull is obtained. A first plot of the vessel is showed, where the mean sea surface is visible too.

5. Run PDSTRIP
   The software PDSTRIP is launched and the bidimensional added mass and damping matrices are found. If the file namegeom format is .mgf, a conversion must be carried out first. The results are elaborated by the function SectionResults and made ready to be elaborated.

6. 2D Added Mass and Damping Coefficients
   The data from the 2D matrices are saved in dedicated variables. The matrices \( a_j(\omega_k) \) and \( b_j(\omega_k) \), \( j = 1, \ldots, n_{sec} \) and \( k = 1, \ldots, 52 \), are all \( 6 \times 6 \) and sparse matrices; that’s why it’s convenient to save these data in structures containing the same element of each matrix, for all the vessel sections \( n_{sec} \) and for all the 52 frequencies.

   For example, a non zero element of the matrices \( a_j \) is the \( (2, 2) \) one: we can create a matrix \( a_{22} \) which contains all these elements (the added mass term in the second degree of freedom, the sway, due to a sway oscillation) for the all the strips and for all the simulated frequencies, so that \( a_{22} \in \mathbb{R}^{n_{sec} \times 52} \).

7. 3D Added Mass Coefficients
   The vessel overall matrices are now computed. That is possible by integrating the contribution of the single sections ones along all the vessel length: in this way we will found one Added Mass matrix \( A(\omega_k) \) for each frequency. The formulas used for the integration, dependent also by the speed of the vessel, are the ones proposed by [Faltinsen1990].

8. 3D Damping Coefficients
   For the Damping matrices the same integration process is carried out, finding the global ship matrices \( B(\omega_k) \).
9. Moments of Inertia
In this section, the moments of inertia in roll $I_{44}$, pitch $I_{55}$ and yaw $I_{66}$ are computed. The evaluation is made by the means of gyration radii, following the approximated formulas of [Faltinsen1990]. Indeed, the values of $I_{jj}$ are computed as $I_{jj} = mr_{jj}^2$, where

$$r_{44} = 0.35 \cdot L_{pp}$$
$$r_{55} = 0.25 \cdot L_{pp}$$
$$r_{66} = 0.25 \cdot L_{pp}$$

10. Transverse Metacenter – Static Stability
To evaluate the static stability in roll, the hull is tilted until a maximum of 10 degrees, by steps of 0.5 degrees. At each iteration, the static pressure is integrated all over the wet surface, which changes as the vessel rolls, and the resulting reaction moment $K$ is computed. For the small angles range considered during this process, the hydrostatics moment is basically linearly dependent on the roll angle $\phi$. After the evaluation of $K$ at each roll angle $\phi$, the derivative $K_\phi$ in $\phi = 0$ is found by an interpolation of the computed curve $K(\phi)$. Thanks to the linear behaviour in this range, this value could also be found simply by a ratio. If this value is positive, then the vessel is unstable and the simulation is interrupted. Otherwise, the value $K_\phi < 0$ is also equal to the opposite of the term $G_{44}$.
From the derivative $K_\phi$, it is also possible to find the value of $GM_T$, as we saw. The inverse procedure is not really feasible, because the transverse second water plane area moment $J_T$ is not easy to calculate: indeed the discretization in strips is carried out in the longitudinal direction. The direct tilting simulation to compute the roll moment at each angle is simpler.

11. Longitudinal Metacenter – Static Stability
Concerning longitudinal static stability, the geometrical data make possible a easy calculation of the longitudinal second moment:

$$J_L = \int_{A_w} x^2 dA \approx \sum_{j=1}^{n_{sec}} B_j x_j^2 \Delta x_j$$

where $B_j$ is the beam, i.e. the length along $y$ direction, of the $j$th section, $x_j$ is its distance from the centroid of the water plane and $\Delta x_j$ is the spacing along $x$ with the $(j+1)$th section. Then, by exploiting the relations of the hydrostatics physics:

$$GM_L = (z_b - z_g) + \frac{J_L}{\Omega}$$

The longitudinal static stability is basically always assured for conventional slender vessels.

12. Restoring Coefficients
The coefficients of the restoring matrix $G$ are computed following the formulas already seen:

$$G_{33} = \rho g A_{wp}$$
$$G_{35} = \rho g \int_{A_{wp}} x \cdot dA_{wp} \approx \rho g \sum_{j=1}^{n_{sec}} B_j x_j \Delta x_j$$
$$G_{44} = \rho g GM_T$$
$$G_{53} = G_{35}$$
$$G_{55} = \rho g GM_L$$

For slender moving vessels, a particular correction have to be made about yaw stability. If a yaw perturbation $\psi$ appears, with respect to the equilibrium state, of course no hydrostatics reaction is manifested. Nevertheless, if the vessel has a forward speed $U$, and the velocity is not aligned with the $x$ body axis, a stabilizing moment shows up. Indeed, in the seakeeping reference frame, a yaw angle $\psi$ coincides also with a sideslip angle $\delta$.

The former statement is completely true if the main direction of waves is zero. By contrast, if the waves are crashing on the vessel hull with a certain heading angle $\beta$, the correct approach would be to evaluate the sideslip angle $\delta_0$ by a vectorial sum of the vessel velocity $U$, which is along the $x$ seakeeping direction, and
the sea one $u_{sea}$. Then, the effective sideslip angle $\delta$ would be the sum of $\delta_0$ and the eventual yaw angle $\psi$.

We can see that if the sea velocity is aligned with the $x$ direction of the seakeeping frame, $\delta_0 = 0$ and we come back to $\delta = \psi$.

To simplify the analysis, we assume that $u_{sea}$ is always negligible with respect to $U$, and the approximation $\delta \approx \psi$ is assumed. Indeed, from potential theory we know that the sea surface speed $u_{sea}$, which exponentially decreases with the depth, is limited by the value $\zeta \omega_0$:

$$ u_{sea} = Re \left\{ \sqrt{\left( \frac{\partial \varphi}{\partial x} \right)^2 + \left( \frac{\partial \varphi}{\partial y} \right)^2} \right\} = \zeta \omega_0 Re \left\{ \sqrt{-\cos^2 \beta e^{2jk(x \cos \beta \sin \beta + \beta \omega t)} - \sin^2 \beta e^{2jk(x \cos \beta - y \sin \beta + \beta \omega t)} \right\} \leq \zeta \omega_0 $$

Neglecting $u_{sea}$ amounts to saying $\zeta \omega_0 \ll U$, that for small amplitude waves with peak periods long enough can be a good approximation. In future versions, that can be treated with higher accuracy.

For considering yaw stability and to point out the opposite reaction moment $N$ and the cross force $Y$ caused by the presence of a non-zero sideslip angle $\delta$, two coefficients are taken into account. Indeed, in order to compute the exact values of $N$ and $Y$, one should solve the potential flow around the ship water plane section. But, to simplify, we will use the results of [Lee & Shin 1998, Inoue et al. 1981] on this argument; defining a overall vessel aspect ratio $k_L = \frac{B}{L}$ and the block coefficient $C_b = \frac{\Omega}{\rho \psi d^2}$, the following approximated derivatives are evaluated

$$ N_\delta = -0.0024 - 1.0272 \cdot k $$
$$ Y_\delta = \frac{\pi}{2} k + 1.4 \cdot C_b \cdot \frac{B}{L} $$

The reaction moment in yaw is stabilizing, so the derivative is negative; on the other hand, if a sideslip angle appears, the sway force has its same sign. From these values, we can compute the following terms

$$ G_{16} = -\frac{1}{2} \rho U^2 L_{pp} d \cdot Y_\delta $$
$$ G_{55} = -\frac{1}{2} \rho U^2 L_{pp} d \cdot N_\delta $$
Let us remember that the matrix $G$ is defined positive on the first side of the Cummins equation: that explains the opposite sign with respect the real forces. Furthermore, it is important to notice that these former two terms are not restoring coefficients due to hydrostatics physics, but they are dynamical coefficients, which appear in presence of a forward speed. They are just linearly dependent on an element of the generalized position vector, $\psi \approx \delta$, so that they can be incorporated in $G$.

13. Infinite Frequency Matrices $A_\infty$ and $B_\infty$ and Transfer Function

The optimization problem is solved for a certain range of orders of the memory effect term approximated transfer function - from 2 to 20, and for each pair $(i, k)$ of degrees of freedom separately; then the order choice will fall on the best trade off between a good fitting and a limited order.

Fixing the pair $(i, k)$, for every order in the range $\{2, \ldots, 20\}$, the optimization problem will find $\tilde{K}_{ik}(s)$, $\tilde{A}_{\infty,ik}$ and $\tilde{B}_{\infty,ik}$; in order to do that, the minimization algorithm is chosen in function of the speed - if it is zero or not, and according to what simulator is running, *SeaBoS* or *ManBoS*.

We saw how, after solving the problem for a certain approximation order, it is possible to carry out a quality control to evaluate the fitting; that is made by comparison of the data set discrete values for the 52 frequencies $\omega_l, A_{ik}(\omega_l)$ and $B_{ik}(\omega_l)$, with the approximated ones:

$$\hat{B}_{ik}(\omega) = \hat{B}_{\infty,ik} + Re \left\{ \tilde{K}_{ik}(j\omega) \right\}$$

$$\hat{A}_{ik}(\omega) = \hat{A}_{\infty,ik} + \frac{1}{\omega} Im \left\{ \tilde{K}_{ik}(j\omega) \right\}$$

This weighted comparison is effectuated by computing both the mean relative distance of the points of the data set with the correspondent values of $\hat{A}_{ik}(\omega_l)$ and $\hat{B}_{ik}(\omega_l)$, and the difference of the trend - i.e. the first derivative:

$$err_A = \frac{1}{52} \left(0.4 \cdot \sum_i \left| \frac{\hat{A}_{ik}(\omega_l) - A_{ik}(\omega_l)}{A_{ik}(\omega_l)} \right| + 0.6 \cdot \sum_i \left| \frac{\hat{A}_{ik}(\omega_{l+1}) - \hat{A}_{ik}(\omega_l)}{A_{ik}(\omega_{l+1}) - A_{ik}(\omega_l)} \right| \right)$$

$$err_B = \frac{1}{52} \left(0.4 \cdot \sum_i \left| \frac{\hat{B}_{ik}(\omega_l) - B_{ik}(\omega_l)}{B_{ik}(\omega_l)} \right| + 0.6 \cdot \sum_i \left| \frac{\hat{B}_{ik}(\omega_{l+1}) - \hat{B}_{ik}(\omega_l)}{B_{ik}(\omega_{l+1}) - B_{ik}(\omega_l)} \right| \right)$$

The overall error is computed as the sum of the two former terms, multiplied by the order of the fitting:

$$err = (err_A + err_B) \cdot \text{order}$$

That will weight both the precision and the degree of the fitting function: for low orders a greater distance $err_A + err_B$ is found, but for higher orders the heaviness of the transfer function leads to other disadvantages. The algorithm will choose the order with the lowest accumulative error $err$.

If the variable *Man_order* is off, the simulator will proceed with all the degrees of freedom. Otherwise the user will be demanded each time for checking the fitting and the algorithm choice.

14. Time Simulation

This is the simulator core: after each time step $\Delta t$, the equation integration leads to the generalized position and velocity vector update. At first, there is a routine called *force_exc.m* responsible of the excitation forces computation; given the time and the vessel position and velocity, Froude-Krilov and diffraction forces are calculated on the body fixed frame, and then rotated in the seakeeping one.

We know that the motion equation can be written in the following linear form:

$$y(t) = W \cdot y(t) + l(t) = f(t, y(t))$$

where $y(t) = \begin{bmatrix} \xi(t) \\ \dot{\xi}(t) \end{bmatrix}$, the constant matrix $W$ is:

$$W = \begin{bmatrix} 0_{6 \times 6} \\ -(M_{RB} + A_\infty)^{-1} G^T \\ - (M_{RB} + A_\infty)^{-1} B_\infty \end{bmatrix}$$

and the time dependent vector $l(t)$ is:

$$l(t) = \begin{bmatrix} \begin{bmatrix} 0_{6 \times 1} \\ (M_{RB} + A_\infty)^{-1} \tau_{FK+Diff} - \mu(t) \end{bmatrix} \\ \end{bmatrix}$$
To integrate this equation, a standard 4th order Runge-Kutta method (RK4) is applied; calling \( t_j = j \Delta t \) and \( y_j = y(j \Delta t) = y(t_j) \), this method operates as follows:

\[
\begin{aligned}
k_1 &= \Delta t \cdot f(t_j, y_j) \\
k_2 &= \Delta t \cdot f\left(t_j + \frac{\Delta t}{2}, y_j + \frac{k_1}{2}\right) \\
k_3 &= \Delta t \cdot f\left(t_j + \frac{\Delta t}{2}, y_j + \frac{k_2}{2}\right) \\
k_4 &= \Delta t \cdot f(t_{j+1}, y_j + k_3) \\
y_{j+1} &= y_j + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
\]

We can notice that in order to apply this method, \( l(t_j + \frac{\Delta t}{2}) \) and \( l(t_{j+1}) \) must be computed, which means to evaluate \( \tau_{FK+Diff}^s \) and \( \mu \) at the instants \( t_j + \frac{\Delta t}{2} \) and \( t_{j+1} \). The variable \( \mu \) is updated by the means of another independent RK4 method, about the state variable \( x(t) \), and it is difficult to couple both integration: the approximation \( \mu(t_j + \frac{\Delta t}{2}) \approx \mu(t_{j+1}) \approx \mu(t_j) \) is used.

Concerning the vector \( \tau_{FK+Diff}^s \), the routine \texttt{force.m} allows to compute it giving as input only the time \( t \) and the vector \( y \). To compute \( \tau_{FK+Diff}^s \) at \( t_j + \frac{\Delta t}{2} \) and \( t_{j+1} \), the following process is carried out: approximating \( \dot{y}_{j+1} = y_j + \Delta t \cdot f(t_j, y_j) \), \( \tau_{FK+Diff}^s(t_{j+1}) \) is calculated giving \( (t_{j+1}, \dot{y}_{j+1}) \) as input to \texttt{force.m} then

\[
\tau_{FK+Diff}^s \left(t_j + \frac{\Delta t}{2}\right) = \frac{1}{2} \left( \tau_{FK+Diff}^s(t_j) + \tau_{FK+Diff}^s(t_{j+1}) \right)
\]

The iterations go on until the final time \( T_{fin} \) is reached, that is after \( \frac{T_{fin}}{\Delta t} = 20T_{fin} \) steps.

15. **Post-Processing**

   After completing all the iterations, the dynamical responses are plotted: position, attitude, velocities and angular velocities in function of time.

16. **Plot Vessel Dynamics – Time animation**

   To give a better physical sense of the results and of the vessel behavior, a 3D animation is available. It shows the vessel dynamics, while floating on the wavy fluid free surface

**Examples**

Now we will show some examples of results of FloBoS. More than standard simulations, we will also try to stress the code, pushing to the limit the cases concerning the validity range of the hypothesis.

**Container S175 - SeaBoS**

We present a set of simulation performed with the container ship S175, scaled differently in each case to decrease the computational cost. The data of the container and the hull geometry are presented below:

- Length \( L_{pp} = 175 \) m
- Beam \( B = 25 \) m
- Draft \( d = 7.5 \) m
- Free board \( FB = 3.5 \) m
Non-zero forward speed

[...] 

Zero speed – green water

Trying to evaluate the simulator reaction to a rough sea condition, we imposed a spectrum with $H_{m0} = 2 \text{ m}$ and $T_p = 8 \text{ s}$, corresponding to a sea state Beaufort number 4, and main waves direction of $\beta = 20^\circ$. The container, at zero speed $U = 0$, has been scaled at 33%:

- Length $L_{pp} = 57.75 \text{ m}$
- Beam $B = 8.38 \text{ m}$
- Draft $d = 2.47 \text{ m}$
- Free board $FB = 1.15 \text{ m}$

After simulating one minute, the results are:

Figure 4: Container S175 with water free surface

Figure 5: Position and attitude with respect to the seakeeping frame
Angles are small enough to consider them in the validity range of small oscillations theory. We can notice a drift both in surge (for the same reason as the first case) and in sway, because of the heading angle $\beta$. On the other hand, yaw angle increases in time, also if $\beta > 0$: in FloBoS a yaw stabilizing moment appears only if there is a speed $U > 0$, because the action of the sea is neglected – also if in this case of zero speed, it can not be treated as negligible. Therefore, the only effect acting on yaw is due to the excitation forces: this can be confirmed by noticing the oscillatory trend of the yaw response. The integration of the dynamic pressure on the hull leads, evidently, to an increasing yaw angle.

For the velocities, as before, the most accentuated is the pitch one; the heading angle is too small to lead to important roll angular rate.

![Surge velocity](image1.png) ![Roll velocity](image2.png)

![Sway velocity](image3.png) ![Pitch velocity](image4.png)

![Heave velocity](image5.png) ![Yaw velocity](image6.png)

**Figure 6:** Velocities with respect to the seakeeping frame

The computed excitation forces are:

![Excitation force in surge](image7.png) ![Excitation moment in roll](image8.png)

![Excitation force in sway](image9.png) ![Excitation moment in pitch](image10.png)

![Excitation force in heave](image11.png) ![Excitation moment in yaw](image12.png)

**Figure 7:** Waves excitation forces

Four frames of the animation are showed below, to relate the sea state to the vessel dimensions:
Figure 8: Container S175 scaled at 33% in a sea state Beaufort number 4
Container S175 - ManBoS

A similar simulation can be performed to evaluate how efficient is a solo rudder maneuver. The vessel is now scaled at 25% and the rudder geometric properties are:

- Surface equal to 5% of the longitudinal $L_{pp}d$;
- Aspect ratio of 2.5;
- Maximum angle of deflection $\delta_{\text{max}} = 30^\circ$

With relatively calm sea conditions, $H_m = 0.1$ m and $T_p = 5.5\, s$, the vessel will try to reach two target points, one after the other: $P_1 = (150, \, 25)$ and $P_2 = (300, \, -15)$.

![Figure 9: Container S175 with the first target point](image)

After simulating, we can see how the rudder controls the vessel and it consequently follows and reach the target points. The results are good and the PD controller is functional; however the physical model of the controller is not realistic: it does not take into account neither the operational time of the actuator nor the hydrodynamics feedback. In spite of this, we can say that the PD controller design, based on the computation of the heading angle error, can be considered a good solution.

Looking at the dynamical responses, both the velocities and the positions are affected by the discontinuity after about 30 seconds, because of the sea lane change. The responses are pretty smooth, but after the second maneuvering action, with the purpose of reaching the second point, the roll response increases forcefully, remaining however in an acceptable range.

![Figure 10: Vessel path](image)
Figure 11: Position and attitude with respect to the seakeeping frame

Figure 12: Velocities with respect to the seakeeping frame

The computed excitation forces are:

Figure 13: Waves excitation forces
References


