NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultet for informasjonsteknologi,
matematikk og elektroteknikk
Institutt for teknisk kybernetikk

HOVEDOPPGAVE

Kandidatens navn: Åsmund Hjulstad

Fag: Teknisk kybernetikk

Oppgavens tittel (norsk): Tilstandsrom-representasjon av konvolusjonsledd for overflatefartøy med foroverhastighet.

Oppgavens tittel (engelsk): State-space representation of convolution terms for surface vessels with forward speed

Oppgavens tekst:

The equations of motion for a surface vessels typically include convolution terms due to memory effects in the radiation problem that is used to derive added mass and potential damping. Such convolution terms can be represented by state-space models of relatively low order as shown in previous work on surface vessels with zero speed. The advantage of this state space representation is that the resulting model structure is suitable for control and simulation purposes.

1. Present the equations of motion for a surface vessel with forward speed. The model should include radiation forces with frequency-dependent added mass and potential.
2. Calculate a low-order state-space representation of the convolution terms in the equations of motion for the surface vessel.
3. Simulate the system with wave excitation.

Oppgaven gitt: 15. februar 2004

Besvarelsen leveres innen: 1. september 2004

Utført ved Department of Naval Engineering, Instituto Instituto Superior Técnico, Lisboa

Trondheim, 25. august 2004

Olav Egeland
Faglærer

Preface

This thesis is written as the final work in the Master of Science program at the Department of Engineering Cybernetics, Faculty of Information Technology, Mathematics and Electrical Engineering at the Norwegian University of Science and Technology (NTNU).The thesis is written in cooperation with the Department of Naval Engineering at Instituto Superior Tecnico, Technical University of Lisbon, Portugal, where the author stayed from March through July 2004.

The author would like to express gratitude for the hospitality at Instituto Superior Tecnico. Nuno Fonseca has provided the strip theory data used as well as being a helpful discussion partner. Serge Sutulo's comments and interest in the problems with regards to the Kramers-Kronig relations was appreciated. Thanks to Cosmin Ciortan for helping me understand Portugal.

The preceding work, done last autumn, resulted in a conference paper, together with PhD student Erlend Kristiansen and Professor Olav Egeland, accepted for presentation at the IFAC conference on Control Application in Marine Systems (CAMS) in Ancona, Italy, this July. The author is gratefull for the financial support from the Centre for Ships and Ocean Structures that made it possible to present the paper at the conference.

Åsmund Hjulstad, July 2004, Lisbon, Portugal

Abstract

The equations of motion for surface vessels typically use matrices with frequency-varying hydrodynamic coefficients; added mass and damping. Whereas this formulation is able to, for many applications, sufficiently describe the motions of the vessel, it is very inconvenient when used for control purposes. These undesirable properties may make it necessary to adopt complicated schemes to overcome the presented difficulties. Furthermore, this formulation has a mass matrix that is non-symmetric at forward speed.

Time-domain descriptions using memory functions, while accurately describing motion without resorting to frequency-varying coefficients, are not easily modelled using standard simulation tools and requires significant care when used with variable-step solvers. Instead of memory functions, this work uses a state-space model to represent these effects, calculated from the memory function using identification and model reduction.

This thesis proposes a six degrees-of-freedom equation of motion specifically geared towards control synthesis and easy simulation. Up to the accuracy of strip methods, forward speed effects are accounted for in the time-domain description.

The link between frequency-domain and time-domain descriptions are restated, while several practical concerns when implementing this link are investigated with some results of interest when implementing the time-domain description.

One interesting topic of further investigation has been identified. Frequency-domain descriptions of causal, linear systems are required to adhere to the Kramers-Kronig relations, relations that link the real and imaginary part of the force coefficients to each other. These relations are required to hold for the radiation problem, a causal system that is linear in the present description. However, they do not immediately hold for several strip theories, including that of Salvesen, Tuck and Faltinsen.

Finally, the equation of motion is implemented in a six degrees-of-freedom simulator in Matlab/Simulink, with forward speed, using a strip-method like approach.

Contents

List of Figures

Chapter 1

Introduction

Predicting the motion of ships in waves has been a field of active research for decades, and many books and articles have been written on the subject. The concepts of added mass and potential damping are well established, providing rudimentary, but often sufficient, insight into the vessel dynamics and wave response. Whereas these methods are limited to the frequency domain, there also exist time domain models used for applications such as simulation of both vessel motion and interaction with mooring systems, drilling equipment and loading operations, among others.

1.1 Problem area

This thesis investigates the equation of motion for a surface vessel. Even though not considered explicitly, the equations will apply equivalently to other vessels in the wave zone, including wholly submerged bodies near the free surface.

The problem of predicting the motion of a surface vessel is divided in two subproblems, each solved separately. First, the forces and moments on a restrained, or fixed, structure in the presence incoming waves are calculated. Thereafter, the forces and moments on the same structure under forced motion without incoming waves are considered. The problem is assumed to be linear, so the forces and moments can be added using superposition. Following the same linearity assumption, the response from irregular sea can be calculated by modelling the sea as a sum of many regular waves.

This work mainly deals with the latter problem. Traditionally, this problem has been solved in the frequency domain, using frequency-by-frequency harmonic motion. The resulting model is a differential equation for frequency-dependent coefficients. These models are thus not valid for time domain simulations, nor are they suited for typical methods of control design.

The excitations from incident waves consist of two parts, the forces and moments resulting from hydrodynamic pressure in the undisturbed wave (the Froude-Krylov forces) and the contribution from the diffraction of waves on the structure (diffraction forces). This distinction is not needed in this work, and the total forces are referred to as wave excitation forces.

1.2 Brief history and related work

Cummins [5] was perhaps the first to point out that a differential equation with frequency-dependent coefficients is indeed a strange creature. He proposed an alternative expression, a time domain formulation with a convolution integral, capturing the same behaviour as the frequency-dependent coefficients, while still comprising a proper differential equation that could describe transient dynamics. The link between the frequency-dependent coefficients and the kernel in the convolution integral was also established. Ogilvie extended and elaborated the results in his review [28]. The relation between the added-mass and damping coefficients are also found in this work. These relations have analogies in other fields (e.g. optics), and are often referred to as Kramers-Kronig relations.

A variety of approaches have been followed in order to capture the dynamics in a form suitable for controller design. Most frequently, the vessel motions due to wave frequency excitations are considered separately to the low frequency vessel motions [2]. The zero frequency added mass is then used in the design of the control law, and the removal of the wave frequency motion from the position measurements is a task of a wave filter. [12, 10]

It is also possible to fit a transfer function to the frequency-dependent coefficients [35, 22, 21]. Alternatively, one can do fitting to wave response spectra [17, 12]. [35] claims to use a state-space model to represent the wave motion, but it is not fully clear to this author how the state-space model is derived.

[24] proposed the use of state-space identification and model reduction as a reliable method of getting a low order state-space model to represent the forces from the radiation potential. This method was later applied to six degrees-of-freedom data for a offshore work vessel and implemented in a simulator in Matlab/Simulink [19]. Further details can be found in [18].

This work builds on these contributions, further refining and extending the model and simulator. In addition, the presented background theory is a coherent summary from various sources, not all easily available.

Calculation of added masses and potential damping — the frequency-varying hydrodynamic coefficients — can be done using commercially available software packages, such as WAMIT. Time domain simulations are offered by several companies, for example the software package VERES from SINTEF. Classification societies, like Det norske Veritas (DNV), also use sophisticated simulations to predict wave loads and motions.

This thesis does not describe new effects that have not been treated elsewhere. Unfortunately, most available implementations have a black-box nature. Although they can predict motions and loads, their inner workings remain hidden. This is problematic in relation to control design. During control synthesis it is desirable to take advantage of key properties in the model in order to simplify design, optimize performance and guarantee robustness.

Furthermore, the existing commercial packages are priced towards major companies in the marine industry, and there is limited availability of good software in academia. This does not imply that there is no good simulation software at universities — where most commercial packages may have originated — but their distribution is limited.

1.3 Contents

This thesis is organized in four parts. The first part presents background theory. Chapter 2 describes mathematical tools and notation used in this work. Background on group theory and the calculus of variations is presented in order to allow for a better understanding of the six degrees-of-freedom kinematics. In chapter 3, the necessary theory to derive added masses and expressions for potential damping is presented. Attention is also given to the restoring forces, resulting in a compact and general expression for these forces and moments. Following these two chapters, in chapter 4.1, the previous results are amalgamated to present the equation of motion, also commenting on several useful properties for control.

Part 2, goes into more detail. Chapter 5 discusses the connection between frequency domain (f.d.) and time domain (t.d.) representations, looking to reconcile the various approaches found in literature that often comprise small variations of each other. The usability of strip-theory coefficients as a starting point for creating time domain representations is investigated. The link between f.d. and t.d. is dependent on causality of the system, a property that at first sight appears to be violated for strip-theory coefficients.

Because of the unavailability of strip theory data for the horizontal modes, a simple approach was used in order to include strip theory-like forward speed effects to 3D data from WAMIT. The argument and justification behind this is presented in chapter 6.

The method used to calculate wave excitation forces is briefly presented in chapter 8. Chapter 9 describes the available data and existing tools used in the work, whereas chapter 10 goes into details regarding implementation. This also includes several factors critical to achieving good accuracy while reducing computational cost.

Several manipulations are carried out on frequency domain data to arrive at the time domain model in the efforts leading up to this thesis. Chapter 12 describes testing set-up and results of how t.d. models match f.d. models. Chapter 13 is devoted to results from time domain simulations of the model.

Finally, chapter 14 presents conclusions and suggestions for future work.

1.4 Contributions

This work is believed to represent the following new contributions.

The understanding of the radiation restoring coefficient from [9, 8] has been improved. In particular, it is now possible to automatically calculate valid values for the coefficient that are applicable over a larger frequency range.

Although nothing but a simple observation, the time domain models with memory functions generated from strip theories can be generalized to slowly varying speeds, extending the usability of the formulation and simplifying simulator design.

A possible problem with the use of strip theory coefficients in time domain simulations has been identified. However, work-arounds have been presented that can provide practical interim solutions.

Finally, a six degrees-of-freedom model structure for vessels on or near the surface is proposed, applicable to forward speeds and arbitrary excitations. This model has properties that should be of significant interest for control design purposes. The model is implemented in a simulator in Matlab/Simulink.

Part I

Theoretical background

Chapter 2

Mathematical tools

This chapter provides brief summaries of some mathematical tools that are used in this thesis. The selection is based on the specific results needed later, and is not intended to be exhaustive. References to books and other sources are included to further assist the unfamiliar reader.

2.1 Some comments on notation

The mathematical notation used in this work differs in some respects from that commonly seen in many textbooks. This section will introduce the differences and the motivation behind them.

Mathematics should be considered a tool for abstract thought. Notation is one means of putting thoughts on paper. A good mathematical text is a blend of definitions, expressions, derivations, textual explanations and prose. Just as a good sentence should not contain too much information, a good mathematical notation should make it possible to focus on the important content. Nor should waste ink. If everything is a vector or a matrix, presenting them all in boldface does not add any substance. In language, common word sequences are contracted, often to form new words. In math, new symbols are used as shorthands for frequently occurring, and more cumbersome, sequences or constellations of symbols.

A small set of preconditions makes it possible to apply the result to a much larger set of problems, thus generating more powerful results. This has consequences for notation. If something can be either a matrix or a scalar, and there is a notational rule dictating that all matrices should be in bold face, what then?

Meaning and representation are sometimes easily confused. A free rotation of an object in three dimensions is just that; a free rotation. A rotation has several possible representations, a study into the different alternatives can easily take a lot of time. Often, however, the choice is of no importance to the task at hand. In implementation, a choice must be made, but if other requirements do not make it necessary, it should be deferred until then. The representation, be it Euler angles or quaternions, is something different from the concept of interest; a free rotation in three dimensions.

Notation should allow this. This text can in some places be considered an experiment in notation. If the result is that statements are ambigous or unclear, then this experiment has failed. If it makes the meaning clearer, reducing clutter and unnecessary symbols, it can perhaps be of use.

Having presented the motivation, it is time to look more closely to the notational details, and where the differences are. Hopefully, the rest of this section will only be necessary as a source of information for unfamiliar notation.

2.1.1 Vectors

A proper definition is postponed to the following section on group theory. For this discussion the familiar notion of a point in space, given by some coordinates, will suffice.

A vector can be represented using many different coordinate systems. A typical coordinate system is given by a set of independent vectors of unit length, and a vector is then represented as a linear combination of these independent vectors. Some texts use a “coordinate-free” form, where vectors are typically written as $\vec{v}$, with an arrow on top. This representation is equivalent to the first mentioned, and neither gives expressive power over the other.

Unfortunately, both come with their set of notational clutter. Arrows waste ink. In addition, they occupy valuable typeface estate that it is very easy to find other uses for, such as dots, thildes, hats and the rest of the decorational repertoire. The alternative notation comes with an armada of transposes that one can easily get wrong, changing the meaning of the expression.

The notation used here takes good parts from both regimes. The validity of this follows trivially from the fact that the two representations are equivalent (or “unique up to an isomorphism” in the language of group theory), and that all operators have possible forms in both representations.

In parts of this work, there is added readability from distinguishing a scalar from a vector. As a result, vectors are in some places set in a bold font, such as this: $\mathbf{v}$.

Three binary operations are frequently used. They are (using a and b as arbitrary placeholders)

Vector sum The sum of two vectors, denoted $a + b$.

Scalar product Also known as dot product, or sometimes inner product, denoted $a \cdot b$.

Cross product The vector cross product is denoted $a \times b$. The cross product is involved in relations between angular velocity and position velocity, as well as between forces and moments.

2.1.2 Operators

Three binary operators over vectors were mentioned in the previous section. Unary operators are also needed. The difference between a unary and a binary operator is the simple fact that whereas a binary operator takes two arguments, a unary operator only takes one. That binary operators are put in between its arguments is only a matter of convenience, to improve readability. The unary operator, taking only one argument, is put on the left side of the argument. Operators also have a right-to-left precedence, so that the expression $ABCx$ is the result of applying C on x, then applying B on the result, and finally applying A to get the final result. One example operator is negation $(-)$, as in $-x$.

Whereas this may seem obvious, it already makes it possible to dispose of some extraneous notation. The partial differentiation operator is denoted

$$\frac{\partial}{\partial \mathbf{x}} \tag{2.1}$$

The operator takes as its one argument a function of one or several variables (with domain D), and returns the derivatives with regards to a subset of the variables. The return value can be considered as a map from D to a subset of the tangent bundle of the function's range. The return value can also be viewed as another operator. This interpretation of notation, as opposed to considering it as some sort of vector, makes any use of transposes irrelevant. They are therefore removed.

The gradient operator, ∇, is the partial differention operator with respect to the spacial coordinates.

Operators can also be created "on-the-fly". In the case of the familiar linear equation $\dot{x} = Ax$, A is a square matrix (possibly of dimension 1×1). It can equally well be considered an operator from the state space to the tangent space, allowing for elegant notational extensions to the nonlinear case. In this case there is no ambiguity between A as a matrix or A as an operator.

In other cases, decorations are used, with the cross product being a notable example. For the vectors a and b, the cross product is written as $a \times b$. If one vector (a) typically remains constant, while the other (b) varies, it is convenient to encapsulate the cross product between a and some other vector as an operator. This operator is denoted $a^{\times}$.

The same decoration is used in the textbook [6] for the skew-symmetric form of a coordinate vector. The difference between the two uses of notation is small, and the interpretations are interchangeable.

2.2 Group theory

The notation and approach used in this chapter is inspired by results and methods of thought from group theory. This section will introduce the concepts in a natural sequence, supplemented with examples, in order to ease an initial learning curve.

Definitions are mainly from Mathworld [37]. Although group theory is considered advanced it is felt to be useful in order to simplify notations and discover patterns in the results. In particular, its concepts are used when deriving the kinematics of a rigid body.

2.2.1 Definitions

A *group* is a set of elements together with a binary operation that together satisfy the four fundamental properties of closure, associativity, the identity property, and the inverse property.

1. Closure: If A and B are elements in the group, the product AB is also in the group.

2. Associativity: The multiplication is associative, i.e., $(AB)C = A(BC)$.

3. Identity: There is an identity element I such that $AI = IA = A$.

4. Inverse: There is an inverse of every element $B = A^{-1}$ that is also in the group, and in addition $AA^{-1} = A^{-1}A = I$

In this context, when using the words “product” and “multiplication”, it is referred to the binary operation associated with the group.

A *field* is a set of elements with two binary operations, addition and multiplication, both satisfying closure, commutativity, associativity, distributivity, has an identity and an inverse.

1. Commutativity: The property $A \cdot B = B \cdot A$.

2. Associativity: $(A \cdot B) \cdot C = A \cdot (B \cdot C)$ and $(A + B) + C = A + (B + C)$

3. Distributivity: The properties $A \cdot (B + C) = A \cdot B + A \cdot C$ and $(A + B) \cdot C = A \cdot C + B \cdot C$.

4. Identity: $a + 0 = a$ and $a \cdot 1 = a$

5. Inverse: $a + (-a) = 0$ and $a \cdot a^{-1} = 1$ (if $a \neq 0$)

The identity for the addition and multiplication operations differ in general, as in the example of the real numbers: $a + 0 = a$ and $a \cdot 1 = a$. Again, addition and multiplication does not need to be the familiar operations on the real numbers, any binary operation will do, as long as they satisify the axioms above. Note also that the identity and inverse are linked with a specific operation.

The rotation matrices form a group under matrix multiplication, but not a field, as there is no addition operation defined. Examples of fields include the real numbers and the complex numbers.

A *vector space* is a set V over a field F that is closed under addition and multiplication with a scalar, that is, $\mathbf{v} + \mathbf{w} \in V$ (where $\mathbf{v}, \mathbf{w} \in V$) and $c * \mathbf{w} \in V$ (where $c \in F$, $\mathbf{w} \in V$). A *vector* is an element in a vector space. The familiar example is the real vector space of the set $\mathbb{R}^n$ over the field $\mathbb{R}$.

An *algebra* is a vector space over a field with a notion of multiplication. Recall that even though a multiplication is defined for a field, it is not defined for a general vector space.

A *Lie algebra* is an algebra with a special kind of multiplication operation, here denoted $[\cdot, \cdot]$. Elements of a Lie algebra satisfy

1. $[f, f] = 0$
2. $[f + g, h] = [f, h] + [g, h]$
3. $[f, [g, h]] + [g, [h, f]] + [h, [f, g]] = 0$

The above conditions also implies $[f, g] = -[g, f]$. One example Lie algebra is the algebra of vectors in $\mathbb{R}^3$ with the vector cross product as the multiplication operation[1].

A formal definition of a *Lie group* starts with the concept of a manifold, a term from topology. Dwelling on the details here does not provide additional insights or tools necessary for this text, so an informal description is presented here. A manifold is a space that is chartable, meaning it is possible to draw a map or chart of the space. The notion of "drawing a chart" is of course extended to higher dimensions than two, and it may be necessary to draw several charts. One could imagine a three-dimensional chart, and in the case of a chart of the earth, it is necessary to have more than one chart to fit the whole planet. A Lie group is then a group with the set of elements given by all the elements on a manifold, that has a well-defined binary operation, and in addition that this operation is differentiable.

In the context of motion control, rotations and transformations form Lie groups.

2.2.2 Maps and functions

The previous section defined a variety of "things". Sometimes it is of use to be able to manipulate these in a manner without considering details that are not directly relevant. This motivates the use of consistent definitions, encapsulating the essential behaviour under discussion.

A *map* is an association of elements in one set to elements in another set. To introduce notation; a map $f : A \mapsto B$ is an association from every element in A to a unique element in B, that is for every $a \in A$ there is a unique element $f(a) \in B$.

[1] Note, however, that this Lie algebra is *not* the one associated with the group $(\mathbb{R}^3, +)$. The algebra associated to this group has a multiplication operation with result identically zero. See also section 2.4.5.

The terms "map" and "function" should be considered synonyms, as their definitions are identical, just replace "map" with "function" and the maps-to arrow $\mapsto$ with a regular arrow $\rightarrow$ typically used in function definitions. When one term is preferred over the other, it can be due to a desire to emphasize certain characteristica of the object.

2.2.3 The exponential map

With every Lie group one can associate a Lie algebra. This Lie algebra captures the local structure of the Lie group. Put differently, for any Lie group, there is a unique Lie algebra that captures the local structure. The transverse does not hold; with one Lie algebra there can be associated several Lie groups.

Some clarifying about the term "unique" is appropriate here. A more precise term would be "excactly one, up to an isomorphism". An isomorphism is here a way to translate back and forth between two algebras while identically preserving all properties during the translation. As such, there can be several algebras that share excactly the same mathematical properties, so that differentiating between the different algebras is of no value.

The exponential map, denoted exp, is a map from a Lie algebra to its Lie group. More precisely, if G is a Lie group and $\mathfrak{g}$ is the associated Lie algebra, $\exp : \mathfrak{g} \mapsto G$. The Lie algebra can be considered as the tangent space of the Lie group at identity. $\exp(v)$ is then defined as the new position in G after a unit time motion with velocity $v \in \mathfrak{g}$. The shortest path between two points in G is mapped by exp from a straight line in $\mathfrak{g}$.

The exponential map is a generalization of the exponential function of real numbers. In the special case, G is the group of positive real numbers with multiplication and $\mathfrak{g}$ is the algebra of all real numbers with the result of multiplication being identically zero.

2.3 Summary of elements used

This section puts familiar elements such as vectors and rotation matrices into the framework presented above.

The tools presented will be used to derive dynamical models of physical systems. First, one can consider the set of possible states for a system, forming a configuration space. An example configuration space is the space of rotations in three dimensions: $\mathbb{RP}^3$. An equivalent space is the set of elements in SO(3), and composition of several rotations can be done using matrix multiplication.

Next, one considers the form of the tangent space and choose a representation. The exponential map maps from straight lines in the tangential space to shortest paths in the configuration space. The definition of the shortest path relies on the choice of metric in the configuration space. In the case of SO(3), a suitable metric is the minimum angle of rotation around an arbitrary vector in order to move from one point to another. Given the metric, the exponential map and the space it maps from can be defined.

The issue of finding suitable expressions for the exponential map still remains. The rotation matrix can be parametrized by an angle θ about a unit vector $\mathbf{k} \in \mathbb{R}^3$ as

$$\mathbf{R} = \cos\theta \mathbf{I} + \sin\theta \mathbf{k}^\times + (1 - \cos\theta)\mathbf{k}\mathbf{k}^\top \tag{2.2}$$

which can be related to the series expression for a matrix exponential [6]

$$\mathbf{R} = \exp(\theta \mathbf{k}^\times) = \mathbf{I} + \theta \mathbf{k}^\times + \frac{1}{2!}\left(\theta \mathbf{k}^\times\right)^2 + \frac{1}{3!}\left(\theta \mathbf{k}^\times\right)^3 + \cdots \tag{2.3}$$

where $(\cdot)^\times$ is the skew-symmetric form of the vector. The domain of this function is $\mathbb{R}^3$, the same space as the tangent space of $\mathbb{RP}^3$. The Lie algebra over $\mathbb{R}^3$ is so(3), with the vector cross-product as the multiplication operation.

2.4 Calculus of variations

Variations is a mathematical concept of great use when analysing mechanical systems. The term *virtual change* is sometimes also used. This presentation is based on [6], and will contain definitions and results used in this text.

2.4.1 The variation of a function

The first step is to define the variation of a continuous and differentiable function $f : \mathbb{R}^n \to \mathbb{R}^m$. The perturbed function $\tilde{f}$ is defined

$$\tilde{f}(x, \alpha) = f(x) + \alpha\phi(x) \tag{2.4}$$

where ϕ is an arbitrary continous and differentiable function $\phi : \mathbb{R}^n \to \mathbb{R}^m$ and $\alpha \in \mathbb{R}$. The variation of f at x is then defined as

$$\delta f(x) = \left.\frac{\mathrm{d}\tilde{f}(x, \alpha)}{\mathrm{d}\alpha}\right|_{\alpha=0} = \phi(x) \tag{2.5}$$

It is straightforward to show that differention and variation commute, that is, the derivative of the variation is the variation of the derivative. The same result holds for integration.

The Euler-Lagrange equation can be derived in an elegant manner using variations. Consider the definite integral

$$I = \int_a^b f(y, \dot{y})\,\mathrm{d}t \tag{2.6}$$

where

$$y = y(t), \quad \dot{y} = \frac{\mathrm{d}y}{\mathrm{d}t} \tag{2.7}$$

and y is known at two points in time

$$y(a) = y_a, \quad y(b) = y_b \tag{2.8}$$

Define the perturbed function $\tilde{y}$ as

$$\tilde{y}(t, \alpha) = y(t) + \alpha \phi(t) \tag{2.9}$$

where ϕ is an arbitrary continous and differentiable function that is zero at $t = a$ and $t = b$

$$\phi(a) = \phi(b) = 0 \tag{2.10}$$

The variation of y is defined in the same way as (2.5). Considering the variation of the definite integral (2.6)

$$\delta I = \delta \int_a^b f(y, \dot{y}) \, \mathrm{d}t \tag{2.11}$$

Variation and integration commute

$$\delta I = \int_a^b \delta f(y, \dot{y}) \, \mathrm{d}t \tag{2.12}$$

Substituting the definition (2.5)

$$\begin{aligned} \delta I &= \int_a^b \frac{\mathrm{d}}{\mathrm{d}\alpha} f\left(\tilde{y}(t,\alpha), \tilde{\dot{y}}(t,\alpha)\right)\Big|_{\alpha=0} \, \mathrm{d}t && (2.13) \\ &= \int_a^b \frac{\partial f}{\partial y} \delta y + \frac{\partial f}{\partial \dot{y}} \delta \dot{y} \, \mathrm{d}t && (2.14) \end{aligned}$$

Partial integration of the second term

$$\int_a^b \frac{\partial f}{\partial \dot{y}} \delta \dot{y} \, \mathrm{d}t = \frac{\partial f}{\partial \dot{y}} \delta y \Big|_a^b - \int_a^b \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial f}{\partial \dot{y}}\right) \delta y \, \mathrm{d}t = - \int_a^b \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial f}{\partial \dot{y}}\right) \delta y \, \mathrm{d}t \tag{2.15}$$

where the last equality follows from the fact that the variation (δy) is zero at a and b.

Then,

$$\delta I = \int_a^b \left[\frac{\partial f}{\partial y} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial f}{\partial \dot{y}}\right)\right] \delta y \, \mathrm{d}t \tag{2.16}$$

Since δy is arbitrary for all t, recall that it is defined using the arbitrary function ϕ, $\delta I = 0$ implies the Euler-Lagrange equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial f}{\partial \dot{y}}\right) - \frac{\partial f}{\partial y} = 0 \tag{2.17}$$

2.4.2 The variation of a vector

The following sections will result in definitions of the variations of various elements used in our descriptions of dynamics. As an introductory example, and to show the pattern in the definitions clear, the variation of a vector is presented first. The next two sections deal with the variation of the rotation matrix and the homogoenous transformation matrix.

A vector in a n-dimensional vector space is denoted $\mathbf{r} \in \mathbb{R}^3$. Define a perturbed vector

$$\tilde{\mathbf{r}}(\alpha) = \mathbf{r} + \alpha\epsilon \tag{2.18}$$

The variation of $\mathbf{r}$, that is $\delta\mathbf{r}$ follows from the definition

$$\delta\mathbf{r} = \frac{\mathrm{d}}{\mathrm{d}\alpha}\tilde{\mathbf{r}}(\alpha)\bigg|_{\alpha=0} = \epsilon \tag{2.19}$$

The variation in velocity $(\mathbf{v})$ is

$$\delta\mathbf{v} = \frac{\mathrm{d}}{\mathrm{d}t}\epsilon \tag{2.20}$$

2.4.3 The variation of the rotation matrix

The rotation matrix can express an arbitrary rotation of an object in three-dimensional space. The rotation matrices form a group termed "Special Orthogonal Group of third order" abbrievated SO(3) and defined by

$$\mathrm{SO}(3) = \left\{\mathbf{R} \in \mathbb{R}^{3\times 3} |\, \det\mathbf{R} = 1, \mathbf{R}^\top = \mathbf{R}^{-1}\right\} \tag{2.21}$$

A starting point for a definition of the variation of the rotation matrix is a quantity that is dependant on the original rotation matrix, but still a member of the same group – SO(3). Recall that to define the variation of a function, a perturbed function is introduced. When defining the variation of the rotation matrix, it is necessary to look for the analogy, a perturbed rotation matrix.

A natural choice is then

$$\tilde{\mathbf{R}}(\alpha) = \mathbf{R}\mathbf{R}_{\sigma,\alpha} \tag{2.22}$$

the original rotation and an additional rotation by an angle α about the vector given by $\sigma \in \mathrm{so}(3)$. An equivalent representation is

$$\tilde{\mathbf{R}}(\alpha) = \mathbf{R}\mathrm{exp}\left(\alpha\sigma\right) \tag{2.23}$$

Following the customary definition of variation one finds

$$\delta\mathbf{R} = \frac{\mathrm{d}}{\mathrm{d}\alpha}\tilde{\mathbf{R}}(\alpha)\bigg|_{\alpha=0} = \frac{\mathrm{d}}{\mathrm{d}\alpha}\mathbf{R}\mathrm{exp}(\alpha\sigma)\bigg|_{\alpha=0} \tag{2.24}$$

$$= \mathbf{R}\sigma \tag{2.25}$$

The variation of angular velocity $\delta\omega$ is related to the above by

$$\delta\omega = \frac{\mathrm{d}}{\mathrm{d}t}\sigma + [\omega, \sigma] \tag{2.26}$$

2.4.4 The variation of the homogeneous transformation matrix

The homogeneous transformation matrix can express an arbitrary configuration and translation of coordinate systems or rigid bodies. It is given by

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{0} & 1 \end{pmatrix} \tag{2.27}$$

The group is termed "Special Euclidean Group of third order" with symbol SE(3). Its variation is[2]

$$\delta\mathbf{T} = \begin{pmatrix} \delta\mathbf{R} & \mathbf{R}\delta\mathbf{r} \\ \mathbf{0} & 0 \end{pmatrix} \tag{2.28}$$

which can be written as (and defining $\eta \in \mathrm{se}(3)$ in the process)

$$\delta\mathbf{T} = \mathbf{T}\eta \tag{2.29}$$

with η

$$\eta = \begin{pmatrix} \sigma & \delta\mathbf{r} \\ \mathbf{0} & 0 \end{pmatrix} \tag{2.30}$$

The time derivative of $\mathbf{T}$ is

$$\dot{\mathbf{T}} = \mathbf{T}\mathbf{w} \tag{2.31}$$

with $\mathbf{w} \in \mathrm{se}(3)$ representing velocities in the body frame. The variation of $\mathbf{w}$ is

$$\delta\mathbf{w} = \frac{\mathrm{d}}{\mathrm{d}t}\eta + [\mathbf{w}, \eta] \tag{2.32}$$

2.4.5 Return to the variation of a vector

The presentation of the variation of different configurations started with the variation of a vector, and it is now time for a brief return. First, the variations of vectors, rotations and transformations are repeated

$$\delta\mathbf{r} = \frac{\mathrm{d}}{\mathrm{d}\alpha}\tilde{\mathbf{r}}(\alpha)\bigg|_{\alpha=0} = \epsilon \tag{2.33}$$

$$\delta\mathbf{R} = \frac{\mathrm{d}}{\mathrm{d}\alpha}\tilde{\mathbf{R}}(\alpha)\bigg|_{\alpha=0} = \mathbf{R}\sigma \tag{2.34}$$

$$\delta\mathbf{T} = \frac{\mathrm{d}}{\mathrm{d}\alpha}\tilde{\mathbf{T}}(\alpha)\bigg|_{\alpha=0} = \mathbf{T}\eta \tag{2.35}$$

[2] This expression is different from equation (8.147) in [6], but appears necessary in order to get consistent definitions.

Looking again at the variations of the velocities

$$\delta \mathbf{v} = \frac{\mathrm{d}}{\mathrm{d}t}\epsilon \tag{2.36}$$

$$\delta \omega = \frac{\mathrm{d}}{\mathrm{d}t}\sigma + [\omega, \sigma] \tag{2.37}$$

$$\delta \mathbf{w} = \frac{\mathrm{d}}{\mathrm{d}t}\eta + [\mathbf{w}, \eta] \tag{2.38}$$

the pattern is not perfect, (2.36) appears to miss a term. This is not the case, however. For all configurations, the variation of body velocity is the derivative with respect to time plus the Lie bracket of the velocity and the variation of position. In the case of $\mathbb{R}^n$, however, the corresponding Lie algebra has a bracket that is identically zero. To preserve the common structure of the equations, one could then write

$$\delta \mathbf{v} = \frac{\mathrm{d}}{\mathrm{d}t}\epsilon + [\mathbf{v}, \epsilon] \tag{2.39}$$

an expression that is equivalent.

A geometrical interpretation is to consider the curvature of the configuration space. $\mathbb{R}^n$, the configuration space of a translation, has no curvature, whereas the configuration space of rotations and transformations, are curved. Sometimes one can see the notation $\mathbb{RP}^3$ and $\mathbb{R}^3 \rtimes \mathbb{RP}^3$, used for these spaces. When considering equations (2.33)-(2.35) one should recall behaviour of the exponential map. The exponential map maps between a neighborhood of the zero element in the tangent space and a neighborhood of the identity in the configuration space.

2.5 From mass elements to rigid body

The equations of motion for the vessel will be developed from first principles — Newton's law for a mass element. Taking advantage of the results of calculus of variations, it is possible to develop six degrees-of-freedom equations of motion with a minimum of arithemic excercise. The contents of this section are based on [6].

Starting in the inertial coordinate system, we consider a mass element dm with position $\mathbf{r}$. From Newton's law

$$\ddot{\mathbf{r}} dm = d\mathbf{f} \tag{2.40}$$

where $d\mathbf{f}$ denotes forces acting on the mass element. A rigid body consists of a set of mass elements. Between these elements, there are constraint forces holding the body together. Writing these forces explicitly (redefining $\mathbf{f}$)

$$\ddot{\mathbf{r}} dm = d\mathbf{f} + d\mathbf{f}^{\mathrm{c}} \tag{2.41}$$

and integrating over the rigid body, gives

$$\int_{\mathrm{b}} \ddot{\mathbf{r}} dm - d\mathbf{f} - d\mathbf{f}^{\mathrm{c}} \,\mathrm{d}V = 0 \tag{2.42}$$

Positions and forces are vectors over $\mathbb{R}^n$, and $\cdot$ is the scalar product. This avoids the notational clutter of vector and matrix transposes, giving no additional meaning to the expressions. Note also the difference between the d used to signify forces and mass of an element, and the d used as differential operator and in integrals, set in roman typeface.

The kinetic energy of the body is

$$T = \frac{1}{2} \int_{\mathrm{b}} dm \, (\dot{\mathbf{r}} \cdot \dot{\mathbf{r}}) \, \mathrm{d}V \tag{2.43}$$

whereas the variation is

$$\delta T = \int_{\mathrm{b}} dm \, (\dot{\mathbf{r}} \cdot \delta \dot{\mathbf{r}}) \, \mathrm{d}V \tag{2.44}$$

Virtual work is defined

$$W_\delta \triangleq \int_{\mathrm{b}} (d\mathbf{f} \cdot \delta \mathbf{r}) \, \mathrm{d}V \tag{2.45}$$

representing the work done by the variation $\delta\mathbf{r}$. It is called virtual as the variation does not represent an actual change in value, it is a mathematical experiment saying “what if”.

The derivation continues with taking the scalar product of (2.42) with $\delta\mathbf{r}$.

$$\int_{\mathrm{b}} dm \, (\ddot{\mathbf{r}} \cdot \delta \mathbf{r}) - (d\mathbf{f} \cdot \delta \mathbf{r}) - (d\mathbf{f}^{\mathrm{c}} \cdot \delta \mathbf{r}) \, \mathrm{d}V = 0 \tag{2.46}$$

This is the equivalent of saying that Newton’s law is still valid, even if the position moves around. The product rule for differention gives for the first term

$$\int_{\mathrm{b}} dm \, (\ddot{\mathbf{r}} \cdot \delta \mathbf{r}) \, \mathrm{d}V = \frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathrm{b}} dm \, (\dot{\mathbf{r}} \cdot \delta \mathbf{r}) \, \mathrm{d}V - \delta T \tag{2.47}$$

resulting in

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathrm{b}} dm \, (\dot{\mathbf{r}} \cdot \delta \mathbf{r}) \, \mathrm{d}V - \delta T - W_\delta - \int_{\mathrm{b}} (d\mathbf{f}^{\mathrm{c}} \cdot \delta \mathbf{r}) \, \mathrm{d}V = 0 \tag{2.48}$$

The first term shall now be explored further. The goal is to relate the velocities $\dot{\mathbf{r}}$ to other, more convenient, quantities. Recall that $\dot{\mathbf{r}}$ is the velocity of a single mass element. Often, generalized coordinates are not easily available. Typically, as is the case for Euler angles, there are singularities.

Introducing instead a generalized speed, much more freedom is available in choosing the representation of vessel configuration; position and orientation. The only requirement is that $\dot{\mathbf{r}}$ and $\mathbf{u}$ are affine, or more precisely, there exists an affine transformation between the domains of $\dot{\mathbf{r}}$ and $\mathbf{u}$. In more informal language, for any $\dot{\mathbf{r}}$ there is a unique value of $\mathbf{u}$, and vice-versa, and if one moved a bit, the other would move a bit as well. This requirement fits nicely with the approach of using calculus of variations. The relation between generalized speed ($\mathbf{u}$) and velocity ($\dot{\mathbf{r}}$) is then

$$\dot{\mathbf{r}} = \frac{\partial\dot{\mathbf{r}}}{\partial\mathbf{u}}\mathbf{u} + \frac{\partial\mathbf{r}}{\partial t} \tag{2.49}$$

The variation associated with $\mathbf{u}$ is defined as defined as ξ

$$\xi \triangleq \delta\mathbf{u} \tag{2.50}$$

so that the variation in position, or virtual displacement $\delta\mathbf{r}$ is given by

$$\delta\mathbf{r} = \frac{\partial\dot{\mathbf{r}}}{\partial\mathbf{u}}\xi \tag{2.51}$$

Returning to the first term of (2.48), it can be expressed as

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathrm{b}} dm\,(\dot{\mathbf{r}}\cdot\delta\mathbf{r})\;\mathrm{d}V &= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathrm{b}} \frac{dm}{2}\frac{\partial}{\partial\dot{\mathbf{r}}}\left(\dot{\mathbf{r}}\cdot\dot{\mathbf{r}}\right)\cdot\frac{\partial\dot{\mathbf{r}}}{\partial\mathbf{u}}\xi\,\mathrm{d}V & (2.52)\\
&= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathrm{b}} \frac{dm}{2}\frac{\partial}{\partial\dot{\mathbf{r}}}\frac{\partial\dot{\mathbf{r}}}{\partial\mathbf{u}}\left(\dot{\mathbf{r}}\cdot\dot{\mathbf{r}}\right)\cdot\xi\,\mathrm{d}V & (2.53)\\
&= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathrm{b}} \frac{dm}{2}\frac{\partial}{\partial\mathbf{u}}\left(\dot{\mathbf{r}}\cdot\dot{\mathbf{r}}\right)\cdot\xi\,\mathrm{d}V & (2.54)\\
&= \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial}{\partial\mathbf{u}}\frac{1}{2}\int_{\mathrm{b}} dm\,\left(\dot{\mathbf{r}}\cdot\dot{\mathbf{r}}\right)\;\mathrm{d}V\cdot\xi & (2.55)\\
&= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial\mathbf{u}}\cdot\xi\right) & (2.56)
\end{aligned}$$

using

$$2\dot{\mathbf{r}} = \frac{\partial}{\partial\dot{\mathbf{r}}}\left(\dot{\mathbf{r}}\cdot\dot{\mathbf{r}}\right) \tag{2.57}$$

moving $\partial\dot{\mathbf{r}}/\partial\mathbf{u}$ to the other side of the dot product because $b\cdot Ac = bA\cdot c$ and in front of the paranthesis as $(\dot{\mathbf{r}}\cdot\dot{\mathbf{r}})$ is a scalar. The equation of motion can then be written

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial\mathbf{u}}\cdot\xi\right) - \delta T - W_\delta - \int_{\mathrm{b}} d\mathbf{f}^{\mathrm{c}}\cdot\delta\mathbf{r}\,\mathrm{d}V = 0 \tag{2.58}$$

If the body is rigid, the constraint forces does no work, as the distance between the mass elements remain constant, and the integral term is zero. Assume in addition that the virtual work, defined in equation (2.45), can be expressed using a generalized force τ as $W_\delta = \tau\cdot\xi$. Also, if the kinetic energy T is a function only of $\mathbf{u}$, $\delta T = (\partial T/\partial\mathbf{u})\,\delta\mathbf{u}$

The resulting equation of motion is

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial\mathbf{u}}\right)\cdot\xi + \frac{\partial T}{\partial\mathbf{u}}\cdot\dot{\xi} - \frac{\partial T}{\partial\mathbf{u}}\cdot\delta\mathbf{u} - W_\delta = 0 \tag{2.59}$$

which can be rearranged as

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial\mathbf{u}}\right)\cdot\xi - \frac{\partial T}{\partial\mathbf{u}}\cdot\left(\delta\mathbf{u} - \dot{\xi}\right) - \tau\cdot\xi = 0 \tag{2.60}$$

2.6 Equation of motion

The next step is to get closer to the final equations of motion. Although equation (2.60) can seem somewhat unfamiliar, the steps to the familiar equations of motion, such as the Newton-Euler equation, are short. First, a choice of generalized speeds must be made. Recall that there are many different representations of the tangent space, and one has to choose one. Choosing the velocity of a fixed point in the body and the angular velocity, the generalized speed $\mathbf{u}$ is

$$\mathbf{u} = \mathbf{w} = \begin{pmatrix} \mathbf{v}_{\mathrm{p}}^{\mathrm{b}} \\ \omega^{\mathrm{b}} \end{pmatrix} \tag{2.61}$$

Then, an expression for the kinetic energy must be found. From integrating the intertial frame velocities times mass over the rigid body, and expressing the result using the generalized speeds, the following expression can be found [6, p. 337f][3]

$$T = \frac{1}{2}\mathbf{w} \cdot \begin{pmatrix} m\mathbf{I} & -m\mathbf{r}_{pg}^{\mathrm{b}\times} \\ m\mathbf{r}_{pg}^{\mathrm{b}\times} & \mathbf{M}_p^{\mathrm{b}} \end{pmatrix} \mathbf{w} = \frac{1}{2}\mathbf{w} \cdot \mathbf{M}\mathbf{w} \tag{2.62}$$

All quantities are given in the body frame, denoted by super index b. $\mathbf{M}_p^{\mathrm{b}}$ is the inertia matrix about point p, and $\mathbf{r}_{pg}^{\mathrm{b}}$ is the vector from p to the centre of gravity, and using a single $\mathbf{M}$ (no indexes) to represent the whole matrix. The matrix is symmetrical, giving

$$\frac{\partial T}{\partial \mathbf{w}} = \mathbf{M}\mathbf{w} \tag{2.63}$$

and if $\mathbf{M}$ in addition is time-invariant,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \mathbf{w}}\right) = \mathbf{M}\dot{\mathbf{w}} \tag{2.64}$$

Looking again on (2.60), in the present choice of representation, $\xi = \eta$ and $\delta\mathbf{u} = \dot{\xi} + [\mathbf{w}, \eta]$, so

$$\mathbf{M}\dot{\mathbf{w}} \cdot \eta - \mathbf{M}\mathbf{w} \cdot [\mathbf{w}, \eta] - \tau \cdot \eta = 0 \tag{2.65}$$

that can be rearranged

$$\mathbf{M}\dot{\mathbf{w}} \cdot \eta + [\mathbf{w}, \mathbf{M}\mathbf{w}] \cdot \eta - \tau \cdot \eta = 0 \tag{2.66}$$

$$(\mathbf{M}\dot{\mathbf{w}} + [\mathbf{w}, \mathbf{M}\mathbf{w}] - \tau) \cdot \eta = 0 \tag{2.67}$$

and since η is arbitrary

$$\mathbf{M}\dot{\mathbf{w}} + [\mathbf{w}, \mathbf{M}\mathbf{w}] = \tau \tag{2.68}$$

There are many other, equivalent, formulations. The operation $[\cdot, \cdot]$ can be written on matrix form, resulting in

$$\mathbf{M}\dot{\mathbf{w}} + \begin{pmatrix} \omega^{\times} & \mathbf{0} \\ \mathbf{v}^{\times} & \omega^{\times} \end{pmatrix} \mathbf{M}\mathbf{w} = \tau \tag{2.69}$$

[3] In the reference, equation (8.128) has the wrong signs on the cross-diagonal.

or using the parametrization of [12]

$$\mathbf{M}\dot{\mathbf{w}} + \mathbf{C}(\mathbf{w})\mathbf{w} = \tau \tag{2.70}$$

where it can be shown that **C** can have a skew-symmetrical parametrization.

Chapter 3

Getting wet

The interaction and dynamics of ships and the surrounding water is in its full complexity a really hard problem. Following an approach of "As simple as possible, but not simpler", this chapter will step-by-step present a model that captures a large part of the significant dynamics of interest when designing motion control systems for ships and other marine vessels.

The presentation and resulting model differs in some aspects from several introductory textbooks, such as [12], and should on a first reading be read in its entirety, as certain definitions and assumptions are done differently.

Introductory courses in fluid mechanics present the Navier-Stokes' equation

$$\rho \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = -\nabla p + \mu \nabla \left(\nabla \mathbf{v} \right) + \rho \mathbf{f} \tag{3.1}$$

describing the behaviour of many kinds of fluids[1], sea water being one of them. Body–fluid interactions can then be included by imposing appropriate boundary and initial conditions. Boundary conditions describe physical facts of the system, such as the fact that water cannot go permeate the ship hull or the ocean floor, and the conditions on the free surface between the water and the air above. Initial conditions describe the original position of the ship and possibly other floating nearby, as well as the starting state of the propagating surface waves.

Solving these equations — simulating the dynamic system — is unfeasible with presently available computational power. One is then forced to look at simpler descriptions and split the problem in several parts and look at them separately, often with significant assumptions. The Navier-Stokes equations is a set of coupled, unseparable, second-order nonlinear differential equations, and that is about as hard as equations come. Whether the simplifying approach is a reasonable one can only be verified with experiments. Fortunately, a very large number of experiments have verified that the simplifying steps to be presented indeed do give good results. [5]

[1] The requirement is that the fluid is incompressible and Newtonian

3.1 Potential theory

Initially, inviscid, irrotational and ideal flow is assumed, so that Bernoulli's equation [27]

$$p = -\rho \left(\frac{\partial \phi}{\partial t} + \frac{1}{2} \vec{\nabla} \phi \cdot \vec{\nabla} \phi - gz \right) + p_{\mathrm{a}} \tag{3.2}$$

describes the pressure (p) within the fluid. p_{a} signifies atmospheric pressure, g the gravitational constant, z the vertical coordinate (positive downwards), ρ the fluid density and ϕ the velocity potential. $\vec{\nabla}$ is the gradient operator and $\cdot$ the inner vector dot product. Letting $p_{\mathrm{a}} = 0$ and disregarding the higher order terms, what is left is

$$p = -\rho \left(\frac{\partial \phi}{\partial t} - gz \right) \tag{3.3}$$

This expression for the pressure is linear in velocity potential, allowing us to use superposition, and dividing the potential in several parts. One part results from the ship moving in otherwise still water, causing the surrounding water to move, and creating waves on the surface. This part is commonly termed the radiation potential, denoted by ϕ_{R}. Then the body is assumed fixed and the effects of the incoming waves are evaluated, with the resulting velocity potential denoted ϕ_{W}.

The resulting forces and moments on the ship can be found by a surface integral over the ship hull, or body wetted surface (S_{B}), as in

$$\vec{F} = \iint_{S_{\mathrm{B}}} p\vec{n} \, \mathrm{d}A \tag{3.4}$$

$$\vec{M} = \iint_{S_{\mathrm{B}}} p \left(\vec{r} \times \vec{n} \right) \, \mathrm{d}A \tag{3.5}$$

Insertion of (3.3) then gives

$$\begin{aligned} \vec{F} = {} & \underbrace{\rho g \iint_{S_{\mathrm{B}}} z\vec{n} \, \mathrm{d}A}_{} + \underbrace{-\rho \iint_{S_{\mathrm{B}}} \frac{\partial \phi_{\mathrm{R}}}{\partial t} \vec{n} \, \mathrm{d}A}_{} + \underbrace{-\rho \iint_{S_{\mathrm{B}}} \frac{\partial \phi_{\mathrm{W}}}{\partial t} \vec{n} \, \mathrm{d}A}_{} \\ F = {} & \quad \tau_{\mathrm{G}} \quad + \quad \tau_{\mathrm{R}} \quad + \quad \tau_{\mathrm{W}} \end{aligned} \tag{3.6}$$

and similar expressions for the moment integral. The τ_x represent forces and moments, τ_{G} is the hydrostatic forces and moments. In the remaining, generalized forces are used.

Note that a distinction between Froude-Krylov and diffraction forces, often done elsewhere, is not introduced here as it will not be needed.

3.1.1 Radiation potential

Leaving the wave excitation potential and forces for later, the attention now turns towards the radiation potential.The chosen form is that of Cummins [5], letting

ν_k^{p} denote body velocity given in a equilibrium frame.

$$\phi_{\mathrm{R}}(\vec{r},t) = \sum_{k=1}^{6} \nu_k^{\mathrm{p}}(t)\psi_k(\vec{r}) + \sum_{k=1}^{6} \int_{-\infty}^{t} \chi_k(\vec{r},t-\sigma)\nu_k^{\mathrm{p}}(\sigma)\,\mathrm{d}\sigma \tag{3.7}$$

The potential consists of two parts, their contribution illustrated by considering the effects of a velocity impulse; $\nu_k^{\mathrm{p}}(t) = \delta(t)$. The resulting potential is

$$\phi_{\mathrm{R}}(\vec{r},t) = \delta(t)\psi_k(\vec{r}) + \chi_k(\vec{r},t) \tag{3.8}$$

ψ_k represents the instantaneous response to the body motion, whereas χ_k is the potential resulting from surface waves propagating away from the vessel.

The boundary conditions on ψ_k are

$$\psi_k = 0 \quad \text{on } z = 0 \tag{3.9}$$

$$\frac{\partial \psi_k}{\partial \vec{n}} = \begin{cases} \vec{n} \cdot \vec{\nu}_k, & k = 1,2,3 \\ (\vec{r} \times \vec{n}) \cdot \vec{\nu}_k, & k = 4,5,6 \end{cases} \quad \text{on } S_{\mathrm{B}} \tag{3.10}$$

Condition (3.10) is the requirement that fluid does not cross the body boundary. As the surface waves are to be accounted for by χ_k, ψ_k is chosen constant on the mean free surface, using condition (3.9). This boundary value problem is equivalent to the problem of an appropriately chosen double body oscillating in infinite fluid [27, p. 297f].

Returning to the description of the waves on the surface, the water elevation is given by $z = \zeta(x,y,t)$. A water particle on the surface will remain on the surface. Thus, the material derivative of the surface particle equals the material derivative of the surface elevation

$$\frac{\mathrm{D}\zeta}{\mathrm{D}t} = \frac{\mathrm{D}z}{\mathrm{D}t} = w \tag{3.11}$$

where w is the vertical speed of the surface particle. Expanding the material derivative of ζ

$$\frac{\partial \zeta}{\partial t} + \vec{v} \cdot \vec{\nabla}\zeta = w \tag{3.12}$$

and given irrotational flow, such that the speed can be found from a velocity potential $\vec{v} = \vec{\nabla}\phi$, the kinematic boundary condition is

$$\frac{\partial \zeta}{\partial t} + \vec{\nabla}\phi \cdot \vec{\nabla}\zeta = \frac{\partial \phi}{\partial z} \tag{3.13}$$

This is a statement of the fact that water on the surface stays there, following the assumption of irrotational flow.

The dynamic boundary condition originates in Bernoulli's equation

$$\frac{\partial \phi}{\partial t} + \frac{1}{2}\vec{\nabla}\phi \cdot \vec{\nabla}\phi - gz = -\frac{1}{\rho}\left(p - p_{\mathrm{a}}\right) \tag{3.14}$$

On the free surface $z = \zeta$ and the pressure is atmospheric $p = p_\mathrm{a}$, so that

$$\zeta = \frac{1}{g}\left(\frac{\partial \phi}{\partial t} + \frac{1}{2}\vec{\nabla}\phi \cdot \vec{\nabla}\phi\right) \tag{3.15}$$

Insertion of (3.15) in (3.13) gives the somewhat cumbersome boundary condition

$$\frac{\partial^2 \phi}{\partial t^2} + 2\vec{\nabla}\phi \cdot \vec{\nabla}\frac{\partial \phi}{\partial t} + \frac{1}{2}\vec{\nabla}\phi \cdot \vec{\nabla}\left(\vec{\nabla}\phi \cdot \vec{\nabla}\phi\right) = g\frac{\partial \phi}{\partial z} \tag{3.16}$$

that strictly speaking should be satisfied on the time-varying free surface $z = \zeta(x, y, t)$. The expression can be approximated by keeping only the first order terms in (3.15) and (3.13) resulting in

$$\frac{\partial^2 \phi}{\partial t^2} = g\frac{\partial \phi}{\partial z} \tag{3.17}$$

that can be enforced on the plane $z = 0$. [27]

The combined potential ϕ_R of ψ_k and χ_k should satisfy all the boundary conditions presented so far. This gives the following set of boundary conditions for χ_k

$$\frac{\partial^2 \chi_i}{\partial t^2} - g\frac{\partial \chi_i}{\partial z} = 0, \quad \text{on } z = 0 \tag{3.18}$$

$$\frac{\partial \chi_i}{\partial n} = 0, \quad \text{on } S_B \tag{3.19}$$

$$\frac{\partial \chi_i}{\partial t} - g\frac{\partial \psi_i}{\partial z} = 0, \quad \text{on } z = 0 \quad \text{for } t = 0 \tag{3.20}$$

$$\chi_i = 0, \quad \text{for } t = 0 \tag{3.21}$$

Condition (3.18) is (3.17) restated. Necessary boundary conditions on the wetted are allready satisfied by (3.10). Condition (3.19) then ensures that χ_k does not interfere with this. (3.20) and (3.21) gives initial conditions for χ_k on the free surface and the whole fluid. Additionally, a radiation condition ensures that the waves are moving away from the vessel.

3.1.2 Resulting forces and moments

The radiation potential consists of these two separate parts, the instantaneous response of the fluid and the propagation of surface waves. Recalling that the expressions for force and moments are surface integrals of the pressure (equations 3.4-3.5) and that the pressure is the time derivative of the velocity potential (3.3) one can express the resulting forces from the radiation potential can be expressed as

$$\vec{F} = -\rho \iint_{S_\mathrm{B}} \frac{\partial}{\partial t}\sum_{k=1}^{6} \nu_k^\mathrm{p}(t)\psi_k(\vec{r})\vec{n}\,\mathrm{d}A - \rho \iint_{S_\mathrm{B}} \frac{\partial}{\partial t}\sum_{k=1}^{6}\int_{-\infty}^{t} \chi_k(\vec{r}, t-\sigma)\nu_k^\mathrm{p}(\sigma)\,d\sigma\,\vec{n}\,\mathrm{d}A \tag{3.22}$$

Defining added mass as

$$a_{jk} = \rho \iint_{S_\mathrm{B}} \psi_k(\vec{r}) n_j \,\mathrm{d}A \tag{3.23}$$

the first term of the force expression (3.22) can be written as

$$-\sum_{i=1}^{6} a_{jk} \dot{\nu}_k^\mathrm{p}(t) \tag{3.24}$$

Lagrangian kinematics are convenient to use for the rigid body. The viability of such an approach for the ambient water was proven for a submersed body in [4], as referred by [30]. The total kinetic energy of the rigid body and fluid motion from the ψ_k part of the radiation potential is then

$$\begin{aligned} T &= T_\mathrm{RB} + T_\mathrm{A} && (3.25)\\ &= (\nu^\mathrm{p})^\top (M_\mathrm{RB} + M_\mathrm{A}) \nu^\mathrm{p} && (3.26) \end{aligned}$$

with velocities given in a equilibrium coordinate system.

The second term in (3.22) causes forces in the following form [28].

$$-\sum_{k=1}^{6} \int_{-\infty}^{t} K_{jk}(t-\sigma) \nu_k^\mathrm{p}(\sigma) \,\mathrm{d}\sigma \tag{3.27}$$

3.2 Forward speed

In the presence of steady forward speed, the situation changes somewhat. The radiation potential must now include new terms describing this steady flow and the interaction with the ship. Also, the ship is moving, so this interaction will vary. According to arguments by [33] can these interaction effects be accounted for with sufficient accuracy, and represent no major obstacle.

The solution to the resulting boundary problem has a structure similar to (3.7), but with some additional terms [28]. The force and moment expressions can be simplified to a form equivalent to (3.22). This brief discussion suggests that the forces from the radiation potential can be described using added mass and a convolution term with velocities as input.

Strip methods, such as [31], are standard methods presently in use to calculate forces for motion with forward speed [3]. The methods represent the forces using frequency-dependent coefficients, giving predictions for harmonic motion only. Also, the interaction mentioned above is not included at all [3]. Despite its theoretical deficiencies, it produces sufficiently accurate results for many uses, and is widely used in industry.

The coefficients of the Salvesen, Tuck and Faltinsen strip theory (STF, [31]) has forward speed as an explicit parameter. It would be useful if this explicit dependence on forward speed can be carried into the time domain formulations. This appears to be the case.

The STF coefficients has terms involving 2D coefficients of the aft-most section. These terms are not included in this work.

In fact, all manipulations leading from the frequency-dependent coefficients to equations in the time domain are linear transformations and model reductions. This makes it possible to write the forces from the radiation potential with forward speed effects in the following form

$$\begin{aligned}\tau_{\mathrm{R}} = &-\sum_{k=1}^{6}\int\limits_{-\infty}^{t} K_{jk}^{0}(t-\sigma)\nu_{k}^{\mathrm{p}}(\sigma)\,\mathrm{d}\sigma \\ &-\sum_{k=1}^{6}\int\limits_{-\infty}^{t} UK_{jk}^{1}(t-\sigma)\nu_{k}^{\mathrm{p}}(\sigma)\,\mathrm{d}\sigma \\ &-\sum_{k=1}^{6}\int\limits_{-\infty}^{t} U^{2}K_{jk}^{2}(t-\sigma)\nu_{k}^{\mathrm{p}}(\sigma)\,\mathrm{d}\sigma\end{aligned} \tag{3.28}$$

Here U is forward speed. Note that this is not a Taylor-expansion or similar approximation. This carries exactly the same model as the STF strip theory describes into the time domain. The strip theory already assumes constant forward speed, so the speed can be moved outside of the convolution integral.

The various convolution integral kernels can then be calculated directly from the respective terms of the strip theory. The strip theory damping in heave is forward speed independent, so the only term is the constant speed term $K_{33} \equiv K_{33}^{0}$, the rest are zero. The strip theory damping coefficient in pitch is

$$\beta_{55}(\omega) = \beta_{55}^{0}(\omega) + \frac{U^{2}}{\omega^{2}}\beta_{33}^{0}(\omega) \tag{3.29}$$

In this case, the kernels are then

$$K_{55}^{0}(t) = \frac{2}{\pi}\int_{0}^{\infty} \beta_{55}^{0}(\omega)\cos\omega t\,\mathrm{d}\omega \tag{3.30}$$

$$K_{55}^{1}(t) = 0 \tag{3.31}$$

$$K_{55}^{2}(t) = \frac{2}{\pi}\int_{0}^{\infty} \frac{1}{\omega^{2}}\beta_{33}^{0}(\omega)\cos\omega t\,\mathrm{d}\omega \tag{3.32}$$

In the coupling mode heave-pitch, the coefficient is

$$\beta_{35}(\omega) = \beta_{35}^{0}(\omega) + U\alpha_{33}^{0}(\omega) \tag{3.33}$$

and the kernels are

$$K_{35}^{0}(t) = \frac{2}{\pi}\int_{0}^{\infty} \beta_{35}^{0}(\omega)\cos\omega t\,\mathrm{d}\omega \tag{3.34}$$

$$K_{35}^{1}(t) = \frac{2}{\pi}\int_{0}^{\infty} \big(\alpha_{33}^{0}(\omega) - \alpha_{33}^{0}(\infty)\big)\cos\omega t\,\mathrm{d}\omega \tag{3.35}$$

$$K_{35}^{2}(t) = 0 \tag{3.36}$$

There are several significant points to note here. In heave, as well as other modes with similar structure, there is no speed-dependence, and the conversion to the time domain gives no additional problems.

In pitch, the integrand may diverge towards positive infinity as $\omega \to 0$, and evaluation of the integrand at $\omega = 0$ has undefined result. This causes uncertainties in the numerical calculation of K_{55}^{2}. However, near $\omega = 0$, $\cos\omega t \approx 1$. The introduced uncertainty can because of this be approximated by an (unknown) constant added to the impulse response. In light of other simplifications done in the implementation, this can be accounted for in later steps (see section 10.2).

In the coupling mode, the cause of concern is the fact that the added mass does not converge to zero for $\omega \to \infty$. The proposed solution is to subtract the inifinity frequency part of $\alpha(\omega)$, and include this effect in a linear damping coefficient.

These equations for calculating the impulse response, the splitting into speed-independent and speed-dependent parts has been done in all the new calculations in this work, and if not, it is specified in the text.

3.3 Hydrostatic forces

The hydrostatic pressure in the fluid gives rise to restoring forces and moments. The exact expression is for forces

$$\mathbf{F} = -\rho g \iint_{S_\mathrm{B}} z\mathbf{n}\,\mathrm{d}A \tag{3.37}$$

and for moments [27]

$$\mathbf{M} = -\rho g \iint_{S_\mathrm{B}} z(\mathbf{r}\times\mathbf{n})\,\mathrm{d}A \tag{3.38}$$

Gauss's theorem gives for the forces

$$\mathbf{F} = -\rho g \iiint_{V} \mathbf{e}_z\,\mathrm{d}V \tag{3.39}$$

while the corresponding theorem gives for the moments

$$\mathbf{M} = -\rho g \iiint_{V} (\mathbf{r}\times\mathbf{e_z})\,\mathrm{d}V \tag{3.40}$$

where $\mathbf{e}_z$ is a unit vector along the vessel-parallel z-axis (pointing downwards). Equation (3.39) results in a net force acting in the centre of boyancy. At rest, (3.40) is zero, and the gravitational force cancel the hydrodrostatic forces.

For small motions, linearized expressions can be derived using the surface area and moments, as well as the locations of the centre of boyancy and centre of gravity. Denote the hydrostatic forces $\tau_g = g(\eta)$, the linearized restoring force $\tau = \mathbf{G}\eta$, and split the $\mathbf{G}$ matrix in four parts

$$\mathbf{G} = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} \tag{3.41}$$

The WAMIT user guide [38] lists the components individually. In order to better see the structure, the forces can be put on the following form.

$$G_{11} = \rho g \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & S \end{pmatrix}, \qquad S = \iint_{S_\mathrm{B}} \mathrm{d}A \tag{3.42}$$

The only element here is the restoring forces in heave, resulting from the surface area S. The restoring moments from the changes in waterplane area from angular motion are

$$-\rho g \iint_{S_\mathrm{B}} \mathbf{r} \times (\mathbf{r} \times \eta_\mathrm{r})\, \mathrm{d}A = -\rho g \left(\iint_{S_\mathrm{B}} \left(\mathbf{r}^\times \mathbf{r}^\times\right) \mathrm{d}A \right) \eta_\mathrm{r} \tag{3.43}$$

η_r denotes the rotational part of the vessel position. Note the similarity with the inertia matrix.

As the vessel moves, the moments boyancy and gravity no longer cancel, so expressions for these are needed. Hull displacement (Δ) and mass is assumed equal. Assuming the forces still act in their respective centres, the resulting moments in the equilibrium frame are, from boyancy

$$-\rho g \Delta \left(\mathbf{r}_\mathrm{b} \times \mathbf{e}_z\right) \tag{3.44}$$

and gravity

$$\rho g \Delta \left(\mathbf{r}_\mathrm{g} \times \mathbf{e}_z\right) \tag{3.45}$$

For small motion, the relation between body-fixed and vessel-parallel coordinates is

$$\mathbf{r}_\mathrm{b}^\mathrm{p} = \eta_\mathrm{t} + \mathbf{r}_\mathrm{b}^\mathrm{b} + \eta_\mathrm{r} \times \mathbf{r}_\mathrm{b}^\mathrm{b} \tag{3.46}$$

Inserting (3.46) into the expressions for boyancy (3.44) and gravity (3.45) and adding these, will together with the moment term (3.43) give

$$-\rho g \Delta \left(\frac{1}{\Delta} \iint_{S_\mathrm{B}} \mathbf{r}^\times \mathbf{r}^\times \, \mathrm{d}A + \mathbf{e}_z^\times \left(\mathbf{r}_\mathrm{b} - \mathbf{r}_\mathrm{g}\right)^\times \right) \eta_\mathrm{r} \tag{3.47}$$

resulting in

$$G_{22} = \rho g \Delta \left(\frac{1}{\Delta} \iint_{S_{\mathrm{B}}} \mathbf{r}^{\times}\mathbf{r}^{\times}\, \mathrm{d}A + \mathbf{e}_z^{\times} \left(\mathbf{r}_{\mathrm{b}} - \mathbf{r}_{\mathrm{g}}\right)^{\times} \right) \tag{3.48}$$

The elements along the diagonal of

$$\frac{1}{\Delta} \iint_{S_{\mathrm{B}}} \mathbf{r}^{\times}\mathbf{r}^{\times}\, \mathrm{d}A + \mathbf{e}_z^{\times}\mathbf{r}_{\mathrm{b}}^{\times} \tag{3.49}$$

is the metacenter position. The metacenter height is the difference between the metacenter and the height of the center of gravity. [27]

For the remaining two parts in $\mathbf{G}$, the expressions are

$$G_{21}^{\top} = G_{12} = \rho g \iint_{S_{\mathrm{B}}} \mathbf{r}^{\times}\, \mathrm{d}A \tag{3.50}$$

The vessel position η in these expressions is allways given in a vessel-parallel equilibrium frame. As only hydrostatics are considered, this means that surge, sway and yaw in this coordinate system is identically zero. This means that the linear hydrostatic forces can be put on the following compact form

$$-\rho g \Delta \left[\frac{1}{\Delta} \iint_{S_{\mathrm{B}}} \begin{pmatrix} \mathbf{I} & \mathbf{r}^{\times} \\ -\mathbf{r}^{\times} & \mathbf{r}^{\times}\mathbf{r}^{\times} \end{pmatrix} \mathrm{d}A + \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{e}_z^{\times}\left(\mathbf{r}_{\mathrm{b}} - \mathbf{r}_{\mathrm{g}}\right)^{\times} \end{pmatrix} \right] \eta \tag{3.51}$$

This is a linearization of the nonlinear function $\tau_{\mathrm{g}} = g(\eta)$. The nonlinear function can be regarded as describing the forces from motion in a potential field, with several local minima, the minimum of interest given by $\eta_3 = \eta_4 = \eta_5 = 0$. Another minimum is possibly the capsized ship.

3.4 Additional effects

So far, no viscous effects have been included. The viscous effects cause vortices to shed, reducing pressure and thereby causing drag. These, purely dissipative, effects can be modelled using a quadratic damping term. [12] For conventional ships, this additional damping is necessary in order to accurately predict roll motion; in other modes it is of significantly less importance.

In addition, many coupling effects are neglected. This is typically interaction between the various velocity potentials. Furthermore, effects caused by the changing wetted surface are neglected.

Chapter 4

Equations of motion

4.1 Derivation

It is now time to formulate the equations of motion for the rigid body in interacting with the ideal fluid. Following the line of reasoning in the previous chapter, two contributions are considered separately. First, the kinetic energy of the fluid oscillating with the body. Second, the dissipative effect of free surface waves propagating away from the vessel.

4.1.1 Fluid kinetic energy

The kinetic energy of the fluid corresponds to the added masses. There is still an issue as to what velocities should be used. The added masses from programs such as WAMIT are calculated in an equilibrium frame, typically vessel-parallel. The argument that the same added masses can be used in the body frame is as follows. First, the equilibrium frame must be the same as the body frame in equilibrium. The added mass and rigid body mass must then be given in the same frame.

Recall that the added mass in this context is analogous to a specific double body problem in infinite fluid, and there is no kind of frequency-dependence involved. The added mass is a function of body geometry only. The double body is formed by reflecting the body about the free surface. For motion in the horizontal plane, this means that the wetted surface is unchanging, and the double body geometry remains the same. In the other three modes of motion, this is not the case. The added mass for a surface vessel thus changes as the body moves in heave, roll and pitch. However, if the wetted surface does not change significantly, it appears appropriate to at this point use the approximation of constant wetted area, and the resulting constant added mass.

Making sure that the hydrodynamic added mass and the rigid body inertia matrix are given about the same point, in general different from the centre of gravity, the total kinetic energy is then

$$T = \nu \cdot (M_{\mathrm{RB}} + M_{\mathrm{A}})\nu \tag{4.1}$$

The partial derivatives necessary to arrive at the equation of motion are

$$\frac{\partial T}{\partial \nu} = (M_{\mathrm{RB}} + M_{\mathrm{A}})\nu \tag{4.2}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \nu}\right) = (M_{\mathrm{RB}} + M_{\mathrm{A}})\dot{\nu} \tag{4.3}$$

Letting M denote the sum $M_{\mathrm{RB}} + M_{\mathrm{A}}$, the equation of motion, following the derivations from section 2.6 is

$$M\dot{\nu} + [\nu, M\nu] = \tau \tag{4.4}$$

where $[\nu, M\nu]$ is the multiplication in se(3). So far, only added mass is included. The right-hand side τ here includes all other forces.

4.1.2 Dissipative forces from the potential

Next is the convolution term from the propagation of waves along the free surface. These forces are given by equation (3.28) in the vessel-parallel frame, with velocities in the same frame as inputs. When calculating this term, the assumption was made of steady forward speed with the equilibrium frame translating steadily. This assumption breaks down when turning or accelerating/deccelerating. However, the introduced error may not be significant as long as the change in motion occurs slowly compared to the effective length of the memory function.

The dissipative forces can be written (for compactness)

$$\tau^{\mathrm{p}}_{\mathrm{radiation}} = -\mathcal{K}_U \nu^{\mathrm{p}} \tag{4.5}$$

where $\mathcal{K}_U$ represents the convolution operation with speed-varying (U) kernel or memory function. Let J_{p} denote the transformation from the vessel-parallel body frame to the body frame. The forces are then

$$\tau_{\mathrm{radiation}} = -J_{\mathrm{p}}\mathcal{K}_U J_{\mathrm{p}}^{-1}\nu \tag{4.6}$$

The position of the vessel in inertial coordinates is denoted η. Restoring forces are considered as a result of motion in a conservative potential field, the resulting force is

$$\tau_{\mathrm{restoring}} = -J_{\mathrm{i}} g(\eta) \tag{4.7}$$

Adding the kinematic differential equation relating body velocities to change in inertial coordinates, the 6DOF equation of motion is

$$\begin{aligned} M\dot{\nu} + [\nu, M\nu] + J_{\mathrm{p}}\mathcal{K}_U J_{\mathrm{p}}^{-1}\nu + J_{\mathrm{i}} g(\eta) &= \tau \\ \dot{\eta} &= J_{\mathrm{i}}^{-1}\nu \end{aligned} \tag{4.8}$$

where J_{i} is the transformation from inertia frame to the body frame, and τ represents the remaining forces acting on the system (including wave, thruster, mooring etc).

4.2 Properties for control

4.2.1 The symmetric added mass matrix

For zero speed, the added mass matrix is known to be symmetric. This is a very desirable property as it simplifies controller synthesis, especially when using Lyaponov- or energy-based approaches. In the case of non-zero forward speed, the conventional added mass matrix is no longer symmetric. This can be seen in strip theory coefficients [31], referenced in appendix A.1. “Conventional” is here used to indicate that the reference is to the matrix of frequency-dependent added mass coefficients. In [14] the asymmetry of the added mass matrix is circumvented using acceleration feedback to shape the inertia matrix[1].

The model formulation presented in this work has a symmetric added mass matrix regardless of forward speed. This is because the inifinite frequency added mass is used, which is speed-independent and also symmetric. The effects leading to the asymmetries are in the proposed model included in the retardation functions. These functions on the other hand are passive. Passivity is also regardless of forward speed. This property should be useful in controller synthesis.

4.2.2 Energy function and passivity

It is of interest to investigate the stability of the equation of motion. Using an energy function

$$V = \|\nu\|_M + G(\eta) \tag{4.9}$$

where $\|\nu\|_M$ is the kinetic energy and $G(\eta)$ is the potential function such that $g(\eta) = \nabla G(\eta)$. The time derivative along the trajectories of (η, ν) is

$$\dot{V} = -\nu \cdot [\nu, M\nu] - \nu \cdot J_{\mathrm{p}} \mathcal{K}_U J_{\mathrm{p}}^{-1} \nu - \nu \cdot J_{\mathrm{i}} g(\eta) + J_{\mathrm{i}}^{-1} \nu \cdot g(\eta) + \nu \cdot \tau \tag{4.10}$$

The first term is allways zero. The second is a purely dissipative term. The third and fourth term cancel, as $J_{\mathrm{i}}^{-1}\nu \cdot g(\eta) = \nu \cdot J_{\mathrm{i}} g(\eta)$ since the scalar product is independent of coordinate system (J_{i}^{-1} and J_{i} are adjoint linear operators).

From (4.10) one can see that the mapping $\tau \mapsto \nu$ is passive.

4.3 Non-linear observer

A non-linear observer can be constructed for the system (4.8). The system can be written on the form

$$\dot{\eta} = J_{\mathrm{i}}^{-1}(\eta)\nu \tag{4.11}$$

$$\dot{\nu} = A_1(\eta) + A_2\nu + C(\nu) + B\tau \tag{4.12}$$

[1] Referenced in [12].

where

$$\begin{aligned} C(\nu) &= -M^{-1}\left[\nu, M\nu\right] && (4.13)\\ B &= M^{-1} && (4.14)\\ A_1(\eta) &= -M^{-1}J_{\mathrm{i}}g(\eta) && (4.15)\\ A_2 &= -M^{-1}J_{\mathrm{p}}\mathcal{K}_U J_{\mathrm{p}}^{-1} && (4.16) \end{aligned}$$

Following in the same lines as [20], using contraction theory [25], an observer can be chosen using inertial measurements as observer feedback

$$\begin{aligned} \widehat{\dot\eta} &= J_{\mathrm{i}}^{-1}(\eta)\hat\nu + K_1(\eta - \hat\eta) && (4.17)\\ \widehat{\dot\nu} &= A_1(\eta) + A_2\hat\nu + C(\hat\nu) + B\tau + K_2(\eta - \hat\eta) && (4.18) \end{aligned}$$

The subsystems are first considered separately, their jacobians are

$$\begin{aligned} \delta\widehat{\dot\eta} &= -K_1\delta\hat\eta && (4.19)\\ \delta\widehat{\dot\nu} &= \left(A_2 + \frac{\partial C(\hat\nu)}{\partial\hat\nu}\right)\delta\hat\nu && (4.20) \end{aligned}$$

(4.19) can be made uniformly negative definite by selecting K_1 positive definite. As for A_2, $\mathcal{K}_U$ represents a dissipating subsystem, guaranteeing that A_2 is contracting in some metric. $\partial C(\hat\nu)/\partial\hat\nu$ is skew-symmetric, not affecting the contraction property of the system.

The jacobian of the overall system is

$$\begin{pmatrix} \delta\widehat{\dot\eta} \\ \delta\widehat{\dot\nu} \end{pmatrix} = \begin{pmatrix} -K_1 & J_{\mathrm{i}}^{-1}(\eta) \\ \frac{\partial A_1(\hat\eta)}{\partial\hat\eta} - K_2 & A_2 + \frac{\partial C(\hat\nu)}{\partial\hat\nu} \end{pmatrix} \begin{pmatrix} \delta\hat\eta \\ \delta\hat\nu \end{pmatrix} \tag{4.21}$$

which is contracting when

$$K_2 = \frac{\partial}{\partial\hat\eta}A_1(\hat\eta) + J_{\mathrm{i}}(\eta) \tag{4.22}$$

[20] improves on robustness by allowing design freedom in the choice of metric, achieving similar robustness to that achieved using Lyaponov-based techniques [13].

4.4 Control

Here only a suggestive scetch of a control strategy will be presented. There appears, however, to be no significant hurdles for developing a complete array of control strategies for the proposed model, although this is far beyond the scope of this work.

The goal is to design a controller that tracks a reference trajectory with slowly varying forward speed. At constant speed, the damping from the radiation potential is constant. Denote this value D, defined by

$$D\nu \triangleq \lim_{t\to\infty} J_{\mathrm{p}}\mathcal{K}_U J_{\mathrm{p}}^{-1}\nu \tag{4.23}$$

Then, take the control law as

$$\tau = -K_d(\hat{\nu} - \nu_{\mathrm{d}}) - K_p(\hat{\eta} - \eta_{\mathrm{d}}) + D\nu_{\mathrm{d}} \tag{4.24}$$

The task at hand is then to prove that the vessel converges exponentially to the desired trajectory. This can be achieved by substituting (4.24) in the equation of motion (4.8) and recalculating the jacobians. The final steps follow from the separation principle, similarily as in [34]. It then remains to show that the desired trajectory is indeed a solution to the equation, which can be verified assuming constant forward speed (as opposed to slowly varying). The end result is an exponentially stable system.

Part II

Discussion

Chapter 5

From frequency domain to time domain

The link between the frequency domain and time domain descriptions, although first described in early 60's, still are topic of some discussion.

5.1 Forces from the radiation potential and sinusoidal motion

Fonseca lets the forces from the radiation potential include a restoring term, and write the forces on the following form

$$\tau^{\mathrm{R}} = -\left[a\ddot{q}(t) + b\dot{q}(t) + \int_{-\infty}^{t} K(t-\sigma)\dot{q}(\sigma)\mathrm{d}\sigma + cq(t)\right] \tag{5.1}$$

The coefficient b is allowed to be different from zero, as argued for in section 7.5. The value is chosen to be infinity frequency damping (β^{∞}), necessary to make the integral converge. Taking the Fourier transform

$$\mathcal{F}\left\{\tau^{\mathrm{R}}\right\} = -\left[-\omega^2\left(a - \frac{1}{\omega}\mathcal{F}_{\mathrm{s}}\left\{K\right\}\right) + \mathrm{i}\omega\left(b + \mathcal{F}_{\mathrm{c}}\left\{K\right\}\right) + c\right]\mathcal{F}\left\{q\right\} \tag{5.2}$$

and comparing with the harmonic motion expression

$$\mathcal{F}\left\{\tau^{\mathrm{R}}\right\} = -\left[-\omega^2 a + \mathrm{i}\omega b\right]\mathcal{F}\left\{q\right\} \tag{5.3}$$

and then equating the real and imaginary parts of two equations gives in addition to the expression for the memory function

$$K(t) = \frac{2}{\pi}\int_{0}^{\infty}\left[\beta(\omega) - \beta^{\infty}\right]\cos\omega t\mathrm{d}\omega \tag{5.4}$$

an expression for the radiation restoring term c

$$c = \omega^2\left[\alpha^{\infty} - \alpha(\omega)\right] - \omega\int_{0}^{\infty} K(\sigma)\sin(\omega\sigma)\mathrm{d}\sigma \tag{5.5}$$

This expression clearly evaluates to zero at $\omega = 0$. Calculations of c at non-zero frequencies has given values different from zero. As part of the purpose of using memory functions was to eliminate the need for frequency-dependent coefficients, this is an unpleasant result. In addition, when doing time domain simulations, the restoring coefficient appears necessary in order to match time domain simulations to frequency domain solutions.

This paper will argue that the restoring coefficients may be unnecessary, and that they in several cases are a result of errors introduced at an earlier stage in the calculations.

The convolution term in the expression for the forces from the radiation potential (5.1) will for sinusoidal motion, $q(t) = q\cos\omega t$ with $q(t) = 0$ for $t < 0$, be

$$\int_{-\infty}^{t} K(t-\sigma)\dot{q}(\sigma)\mathrm{d}\sigma = \omega q\cos\omega t\int\limits_{0}^{\infty} K_{jk}(\sigma)\sin\omega\sigma\,\mathrm{d}\sigma - \omega q\sin\omega t\int\limits_{0}^{\infty} K_{jk}(\sigma)\cos\omega\sigma\,\mathrm{d}\sigma \tag{5.6}$$

giving

$$\tau^{\mathrm{R}} = -\left[\begin{array}{c} -a\omega^2 q\cos\omega t - b\omega q\sin\omega t + \\ \omega q\cos\omega t\int\limits_{0}^{\infty} K_{jk}(\sigma)\sin\omega\sigma\,\mathrm{d}\sigma - \\ \omega q\sin\omega t\int\limits_{0}^{\infty} K_{jk}(\sigma)\cos\omega\sigma\,\mathrm{d}\sigma + cq\cos\omega t \end{array}\right] \tag{5.7}$$

This is derived using the fact that $K(t) = 0$ for $t < 0$ and the trigonometric identities for cosine and sine of an angle difference. Grouping the terms one gets

$$\tau^{\mathrm{R}} = -\left[\begin{array}{c} -q\cos\omega t\left(a\omega^2 - \omega\int\limits_{0}^{\infty} K_{jk}(\sigma)\sin\omega\sigma\,\mathrm{d}\sigma + c\right) + \\ -q\sin\omega t\left(b\omega + \omega\int\limits_{0}^{\infty} K_{jk}(\sigma)\cos\omega\sigma\,\mathrm{d}\sigma\right) \end{array}\right] \tag{5.8}$$

Alternatively, the forces can be formulated using the expressions for acceleration and velocity

$$\begin{aligned}\int_{-\infty}^{t} K(t-\sigma)\dot{q}(\sigma)\,\mathrm{d}\sigma = -\ddot{q}(t)\left(\frac{1}{\omega}\int_{0}^{\infty} K(\sigma)\sin\omega\sigma\,\mathrm{d}\sigma\right) \\ + \dot{q}(t)\left(\int_{0}^{\infty} K(\sigma)\cos\omega\sigma\,\mathrm{d}\sigma\right)\end{aligned} \tag{5.9}$$

giving

$$\begin{aligned}\tau^{\mathrm{R}} = -\Bigg[a\ddot{q}(t) - \ddot{q}(t)\left(\frac{1}{\omega}\int_{0}^{\infty} K(\sigma)\sin\omega\sigma\,\mathrm{d}\sigma\right) + \\ b\dot{q}(t) + \dot{q}(t)\left(\int_{0}^{\infty} K(\sigma)\cos\omega\sigma\,\mathrm{d}\sigma\right) + cq(t)\Bigg]\end{aligned} \tag{5.10}$$

These expressions are equivalent.

Defining

$$\alpha(\omega) = a - \frac{1}{\omega}\int_0^\infty K(t)\sin\omega t\,\mathrm{d}t \tag{5.11}$$

$$\beta(\omega) = b + \int_0^\infty K(t)\cos\omega t\,\mathrm{d}t \tag{5.12}$$

the forces can be put on the form

$$\tau^{\mathrm{R}} = -\left[\alpha(\omega)\ddot{q}_k + \beta(\omega)\dot{q}(t) + cq(t)\right] \tag{5.13}$$

5.2 Kramers-Kronig relations

The real and imaginary parts of the Fourier transform of the impulse response of any linear causal system satisfies the Kramers-Kronig relations. If the frequency-dependent added mass and damping coefficients are a result of a convolution term in the force, they must also satisify these relations. This section will derive these relations in the context of hydrodynamic coefficients.

Following Ogilvie [28], taking the inverse Fourier sine and cosine transform of equation (5.11) and (5.12), respectively, results in

$$K(t) = -\frac{2}{\pi}\int_0^\infty \omega\left(\alpha(\omega) - a\right)\sin\omega t\,\mathrm{d}\omega \tag{5.14}$$

$$K(t) = \frac{2}{\pi}\int_0^\infty \left(\beta(\omega) - b\right)\cos\omega t\,\mathrm{d}\omega \tag{5.15}$$

For these inverse transforms to exist, $K(t) = 0$ for $t < 0$. Substitution of (5.15) into (5.11) gives

$$\alpha(\omega) = a - \frac{1}{\omega}\frac{2}{\pi}\int_0^\infty\int_0^\infty \left(\beta(\tilde{\omega}) - b\right)\cos\tilde{\omega}t\,\mathrm{d}\tilde{\omega}\sin\omega t\,\mathrm{d}t \tag{5.16}$$

Replacing the limit of the outer integral with a large positive number and taking limits

$$\alpha(\omega) = \lim_{M\to\infty}\left\{a - \frac{1}{\omega}\frac{2}{\pi}\int_0^M\int_0^\infty \left(\beta(\tilde{\omega}) - b\right)\cos\tilde{\omega}t\sin\omega t\,\mathrm{d}\tilde{\omega}\,\mathrm{d}t\right\} \tag{5.17}$$

and then changing the order of integration gives

$$\alpha(\omega) = \lim_{M\to\infty}\left\{a - \frac{1}{\omega}\frac{2}{\pi}\int_0^\infty\int_0^M \left(\beta(\tilde{\omega}) - b\right)\cos\tilde{\omega}t\sin\omega t\,\mathrm{d}t\,\mathrm{d}\tilde{\omega}\right\} \tag{5.18}$$

Applying $2\sin\alpha\cos\beta = \sin(\alpha-\beta)+\sin(\alpha+\beta)$ and evaluating the inner integral gives

$$\alpha(\omega) = \lim_{M\to\infty}\left\{a - \frac{1}{\omega\pi}\int_0^\infty (\beta(\tilde\omega)-b)\left[\frac{\cos\left((\tilde\omega-\omega)M\right)}{\tilde\omega-\omega}\right.\right. \tag{5.19}$$
$$\left.\left. -\frac{1}{\tilde\omega-\omega} - \frac{\cos\left((\tilde\omega+\omega)M\right)}{\tilde\omega+\omega} + \frac{1}{\tilde\omega+\omega}\right]\mathrm{d}\tilde\omega\right\}$$

The Riemann-Lebesque lemma states that if

$$\int_0^\infty f(\tilde\omega)\,\mathrm{d}\tilde\omega \tag{5.20}$$

is absolutely convergent, then

$$\lim_{t\to\infty}\int_0^\infty f(\tilde\omega)\cos\tilde\omega t\,\mathrm{d}\tilde\omega = 0 \tag{5.21}$$

As a result, the terms in the integrand involving M vanish, and what remains is (after propagating the negative sign in front of the integral, and using the Cauchy principal value as the value of the integral)

$$\alpha(\omega) = a + \frac{1}{\omega\pi}\int_0^\infty (\beta(\tilde\omega)-b)\left[\frac{1}{\tilde\omega-\omega} - \frac{1}{\tilde\omega+\omega}\right]\mathrm{d}\tilde\omega \tag{5.22}$$
$$= a + \frac{2}{\pi}\int_0^\infty (\beta(\tilde\omega)-b)\,\frac{1}{\tilde\omega^2-\omega^2}\,\mathrm{d}\tilde\omega \tag{5.23}$$

In a similar fashion one can derive

$$\beta(\omega) = b - \frac{2}{\pi}\int_0^\infty (\alpha(\tilde\omega)-a)\,\frac{\tilde\omega^2}{\tilde\omega^2-\omega^2}\,\mathrm{d}\tilde\omega \tag{5.24}$$

Looking further on the expression for the radiation restoring coefficient (5.5), and inserting the definition of added mass one easily sees that $c = 0$ for $\omega \neq 0$.

To summarize, if the added mass and damping coefficients model a linear causal system, they will adhere to the relations (5.23) and (5.24). If measured or calculated coefficients do not fullfill the relations, this means that either the system is not linear, or that there are errors in the measurements or calculations. The relations can be checked explicitly using (5.23) or (5.24), verifying that $K(t) = 0$ for $t < 0$, or evaluating the expression for the radiation restoring coefficient at selected frequencies.

5.2.1 Alternative derivation

One issue with the above derivation is the changing of integration order which is not proved rigorously. The following derivation is proposed as an alternative.

The definitions and initial manipulations are the same as used in [32], as is the notation. Oscillation frequency $f = 2\pi\omega$ is used in the definition of the Fourier transform,

$$\mathcal{F}\left\{X\right\}(f) = \int_{-\infty}^{\infty} X(t)\mathrm{e}^{-\mathrm{i}2\pi f t}\,\mathrm{d}t \tag{5.25}$$

with the inverse

$$\mathcal{F}^{-1}\left\{Y\right\}(t) = \int_{-\infty}^{\infty} Y(f)\mathrm{e}^{\mathrm{i}2\pi f t}\,\mathrm{d}f \tag{5.26}$$

These definitions of the Fourier transforms makes normalisation unecessary.

To simplify notation, superscript indexes are used to denote Fourier transforms,

$$X^{\mathrm{F}}(f) \triangleq \mathcal{F}\left\{X\right\}(f) \tag{5.27}$$

and later Hilbert transforms

$$X^{\mathrm{H}}(t) \triangleq \mathcal{H}\left\{X\right\}(t) \tag{5.28}$$

Subsequent transforms, ie. taking the Fourier transform of the Hilbert transform are written as

$$X^{\mathrm{HF}}(f) \triangleq \mathcal{F}\left\{\mathcal{H}\left\{X\right\}\right\}(f) \tag{5.29}$$

X without superscript index is allways a function of time.

The Hilbert transform is defined as

$$X^{\mathrm{H}}(t) \triangleq \int_{-\infty}^{\infty} \frac{X(\sigma)}{t-\sigma}\,\mathrm{d}\sigma \tag{5.30}$$

After taking the Fourier transform, and using

$$\mathcal{F}\left\{\int X(\sigma)A(t-\sigma)\,\mathrm{d}\sigma\right\} = \mathcal{F}\left\{X\right\}(f)\mathcal{F}\left\{A\right\}(f) \tag{5.31}$$

and

$$\mathcal{F}\left\{1/t\right\} = -\mathrm{i}\pi\,\mathrm{sgn}(f) \tag{5.32}$$

one finds

$$X^{\mathrm{HF}}(f) = -\mathrm{i}\pi\,\mathrm{sgn}(f)\,X^{\mathrm{F}}(f) \tag{5.33}$$

where sgn is the signum function.

Impulse responses

The impulse response of the radiation potential vanishes for negative time, that is $K(t) \equiv 0\ \forall t < 0$. The even function $K^{\mathrm{e}}(t)$ is given by

$$K^{\mathrm{e}}(t) = \frac{K(t) + K(-t)}{2} \tag{5.34}$$

$K(t)$ can then be expressed in terms of the signum function and $K^{\mathrm{e}}(t)$ by

$$K(t) = K^{\mathrm{e}}(t)\left[1 + \mathrm{sgn}(t)\right] \tag{5.35}$$

The Fourier transform is then ($f * g$ denotes the convolution of functions f and g, and $\mathcal{F}\left\{\mathrm{sgn}(t)\right\} = -\mathrm{i}/\pi f$)

$$\begin{aligned} K^{\mathrm{F}}(f) &= K^{\mathrm{eF}}(f) + \mathcal{F}\left\{K^{\mathrm{e}}(t)\,\mathrm{sgn}(t)\right\}(f) && (5.36)\\ &= K^{\mathrm{eF}}(f) + \mathcal{F}\left\{K^{\mathrm{e}}(t)\,\right\} * \mathcal{F}\left\{\mathrm{sgn}(t)\right\} && (5.37)\\ &= K^{\mathrm{eF}}(f) - \frac{\mathrm{i}}{\pi}\int K^{\mathrm{eF}}(\sigma)\frac{1}{f-\sigma}\,\mathrm{d}\sigma && (5.38)\\ &= K^{\mathrm{eF}}(f) - \frac{\mathrm{i}}{\pi}K^{\mathrm{eFH}}(f) && (5.39) \end{aligned}$$

As the Fourier transform of K^{e} is purely real, given by

$$K^{\mathrm{eF}}(f) = K^{\mathrm{Fr}}(f) \tag{5.40}$$

equating the imaginary parts results in the relationship

$$K^{\mathrm{Fi}}(f) = -\frac{1}{\pi}K^{\mathrm{FrH}}(f) \tag{5.41}$$

Taking the Hilbert transform on both sides, and using the fact that $X^{\mathrm{HH}} = -X$ one finds the second relationsship

$$K^{\mathrm{FiH}}(f) = \frac{1}{\pi}K^{\mathrm{Fr}}(f) \tag{5.42}$$

From (5.41) and

$$K^{\mathrm{F}}(f) = 2\mathrm{i}\pi f\left(\alpha(f) - a\right) + \left(\beta(f) - b\right) \tag{5.43}$$

one finds

$$2\pi f\left(\alpha(f) - a\right) = -\frac{1}{\pi}\int_{-\infty}^{\infty}\frac{\beta(\sigma) - b}{f - \sigma}\,\mathrm{d}\sigma \tag{5.44}$$

β is even, and the integral can be split in two and the two halves evaluated separately. One is expected to arrive at identical results as (5.23) and (5.24), although this has not been verified by this author.

5.3 Coefficients from strip theory

Strip theories such that the one of Salvesen [31] have some expressions for the added mass and potential damping coefficients looking like (omitting terms relating to the aftermost cross-section)

$$\alpha_{55}(\omega) = \alpha^0_{55}(\omega) + \frac{U^2}{\omega^2}\alpha^0_{33}(\omega) \tag{5.45}$$

$$\beta_{55}(\omega) = \beta^0_{55}(\omega) + \frac{U^2}{\omega^2}\beta^0_{33}(\omega) \tag{5.46}$$

where the superscript ^{0} on α and β denote the zero-speed coefficient. A necessary question to ask is whether the coefficient pairs from the strip theory can adhere to the Kramers-Kronig relations.

$$\beta(\omega) - b = -\frac{2}{\pi} \int_0^\infty (\alpha(\tilde{\omega}) - a) \frac{\tilde{\omega}^2}{\tilde{\omega}^2 - \omega^2} \,\mathrm{d}\tilde{\omega} \tag{5.47}$$

For notational simplicity, $\alpha(\omega)$ and $\beta(\omega)$ are redefined so that they are absolutely convergent, and the relation is written as

$$\beta(\omega) = -\frac{2}{\pi} \int_0^\infty \alpha(\tilde{\omega}) \frac{\tilde{\omega}^2}{\tilde{\omega}^2 - \omega^2} \,\mathrm{d}\tilde{\omega} \tag{5.48}$$

Iff the pair (α_1, β_1) is a valid pair, then $(k\alpha_1, k\beta_1)$, $k \in \mathbb{R}\setminus\{0\}$ is a valid pair. Similarily one sees that $(\alpha_1 + \alpha_2, \beta_1 + \beta_2)$ is a valid pair iff (α_2, β_2) is a valid pair.

Looking again at the strip theory expressions, assuming $(\alpha_{33}, \beta_{33})$ is a valid pair, it follows that $(U^2\alpha^0_{33}/\omega^2, U^2\beta^0_{33}/\omega^2)$ has to be a valid pair in order to $(\alpha_{55}, \beta_{55})$ be a valid pair. Insertion into (5.48) gives

$$\frac{U^2}{\omega^2}\beta^0_{33}(\omega) = -\frac{2}{\pi} \int_0^\infty \frac{U^2}{\tilde{\omega}^2}\alpha^0_{33}(\tilde{\omega}) \frac{\tilde{\omega}^2}{\tilde{\omega}^2 - \omega^2} \,\mathrm{d}\tilde{\omega} \tag{5.49}$$

$$\frac{1}{\omega^2}\beta^0_{33}(\omega) = -\frac{2}{\pi} \int_0^\infty \alpha^0_{33}(\tilde{\omega}) \frac{1}{\tilde{\omega}^2 - \omega^2} \,\mathrm{d}\tilde{\omega} \tag{5.50}$$

$$\beta^0_{33}(\omega) = -\frac{2}{\pi} \int_0^\infty \alpha^0_{33}(\tilde{\omega}) \frac{\omega^2}{\tilde{\omega}^2 - \omega^2} \,\mathrm{d}\tilde{\omega} \tag{5.51}$$

As one side of the pair uniquely defines the other side of the pair, and $(\alpha_{33}, \beta_{33})$ by assumption is a valid pair, the pair $(U^2\alpha^0_{33}/\omega^2, U^2\beta^0_{33}/\omega^2)$ is not valid. The same would apply to all coefficients with a $1/\omega$ term.

It appears then, that one must conclude that the STF strip theory [31] generates coefficients that cannot represent a linear and causal system. As a consequence, one cannot directly use these coefficients to generate impulse responses, as the prerequisites in the derivations of the impulse response are not met.

Also in support of this conclusion are plots of calculated impulse responses in figures 5.1 and 5.2. Either coefficient in a coherent coefficient pair can be used to calculate the impulse response. Let K_α and K_β denote the impulse response calculated from (5.14) and (5.15), respectively. For numerical reasons, it is most convenient to use the damping and thereby avoiding problems caused by the slow convergence of (5.14). $K_\alpha(0)$ will of course allways be zero, but $K_\alpha(\varepsilon)$ will converge to $K_\beta(\varepsilon)$ for all positive ε.

$K(t)$ is by assumption $= 0$ for $t < 0$. This is used in the derivations of (5.14) and (5.15). As the odd and even expansions are used in the derivations, one should find that $K_\alpha(-t) = -K_\beta(-t)$. Numerical investigation in the cases of heave and pitch, as seen in figures 5.1 and 5.2, shows that this is not the case for the strip theory data for forces in pitch.

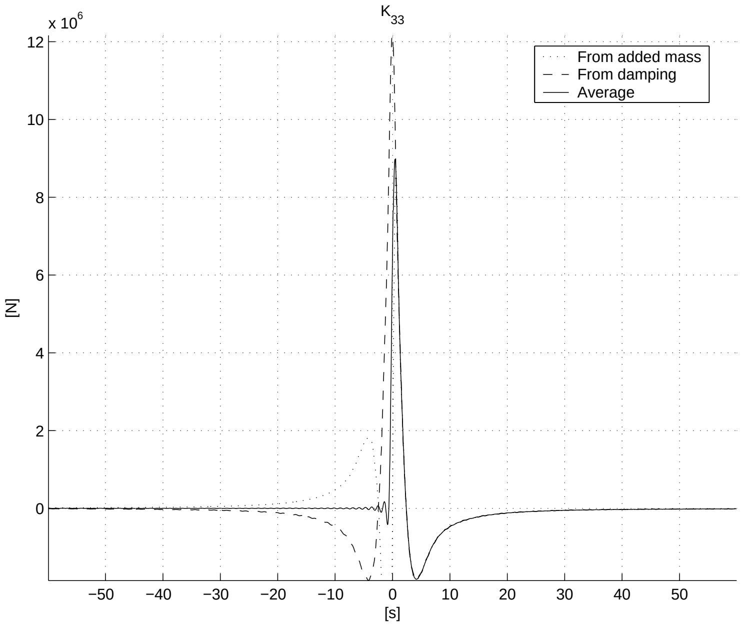

Figure 5.1: Impulse response in heave

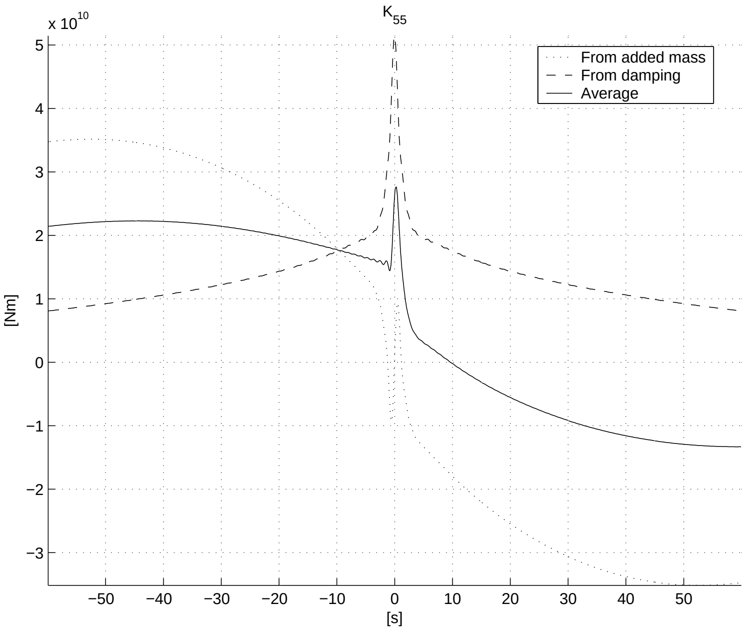

Figure 5.2: Impulse response in pitch

This implies that one is not free to choose whether to use added mass or damping to calculate memory functions.

5.4 Possible explanations

It appears difficult to present a good explanation of the above results and observations. Some thoughts on the subject can be shared, however.

Physical intuition dictates that the radiation problem should have a causal solution. This is also proven to be the case in [36]. This proof applies directly to the case of zero forward speed. It does not extend immediately to the case of forward speed, and in particular, symmetry of the frequency-dependent coefficient matrices is used in the proof.

In [7] the non-causality of two impulse responses involved in the forces from incoming waves is explored. The details are unclear to this author, but some comments can be made. It appears that non-causality is related to the propagation of waves along the hull. This propagation is perhaps similar in nature to the fluid passing along the hull in the forward speed case. Further investigation into this topic is outside the scope of this work, but could still be of interest.

Towards the very end of the work behind this thesis, attention was drawn by Fossen to an article ([1]) that could explain this. It appears that under transformation of the radiation problem to the body frame, the speed-dependence on the added masses and potential damping vanishes.

This result seems promising, although it is somewhat surprising. In the strip-theory derivations many simplifications are made of the hydrodynamic problem, and the choice of simplifications impacts the form of the coefficients. Although the STF strip theory is the most well-known, several other exists. The transformations of [1] only relate to the kinematics. Still, the effects cancel. The underlying reasons for this symmetry is unclear. One explanation might perhaps be found in the derivations of the strip-theory coefficients. The speed-dependent terms have their origin in the requirement of satisfying the body boundary conditions by the steady flow.

An energy-based approach leads to added mass and potential damping coefficients that are not speed-dependent [26]. In this case, however, the radiation restoring coefficients are speed-dependent.

Regardless, some points can be made. The added mass matrix in a model formulation with a convolution integral is always symmetric, positive definite and independent of forward speed. The damping is always speed-dependent, even if the radiation damping is not, viscous damping will be. The restoring coefficient cannot be disregarded, and is non-zero with forward speed. Finally, it appears clearly advantageous to express the radiation damping in the body-fixed frame, making it unecessary to estimate the orientation of the equilibrium coordinate system.

Chapter 6

Strip theory forward speed on 3D data

The developed simulator uses zero-speed coefficients from WAMIT, and applies a strip-theory like approach to model forward speeds. The most important motivation for this may be to have an implementable 6DOF simulator with varying forward speed, and possibly sacrificing some theoretical justification in the progress. Indeed, strip-theories can only provide data for five degrees-of-freedom, excluding surge motion.

The chosen approach is to use zero-speed coefficients from WAMIT and add the forward-speed terms from the strip theory (given in appendix A.1), excluding aft-section terms. The only difference from the 2D approach is that instead of using strips to calculate the zero-speed coefficients, 3D panel methods are used.

This may not be alltogether far-fetched, and has possibly some theoretical justification. All caveats aside, the suggested approach takes as its basis the derivation of the STF strip theory [31], found in appendix 1 of the reference. The oscillatory potential is divided into a speed-dependent and a speed-independent part. The expression for the speed-dependent part can be manipulated using vector relationships and Stokes' theorem. The result expresses the speed-dependent potential as a function of forward speed, oscillation frequency and the speed-*in*dependent potential. As Stokes' theorem is applied on the hull forward of a specific cross-section, the integrated potential around this cross-section also appears. Since vessel motions are of interest, the specific cross-section is chosen to be the aft-most section.

The only assumptions made are that the x-axis lies close to the waterline and the validity of dividing the potential in two parts. No other assumptions of slender body or frequency range are needed. The STF paper then proceedes by using 2D strips to calculate the forces and moments for the 3D body.

In principle, one should therefore be able to use 3D methods to calculate the coefficients at zero forward speed, and use strip theory terms to account for forward speed effects. This will not capture any speed-dependence on the force terms involving surge motion. Hopefully, they are not of major significance.

It is not obvious how to handle the aft-most section terms. For vessel with transom sterns the aft-most section may be significant. The work vessel used in this study does not have a transom stern, so the aft-most section terms can be neglected. Moreover, aft-most section terms were for some unknown reason not included in the available 2D coefficients. These terms are therefore disregarded in this implementation.

One should note, however, that WAMIT can provide as output the radiation potential on the hull. The possibility of using these values to evaluate an integral around a vertical cross-section close aft was initially investigated. The complicating factor was the uneven distribution of panels on the hull. Having no easy way to select what panels, whole or parts, to include in the integral, this idea was not pursued further.

Chapter 7

Comparison with other models

7.1 Models without convolution term

[29] considered the coupling effects between sloshing in liquid storage tanks and ship motions. The sway motion for a two-dimensional strip was studied using experiments and numerical simulations. The motion was found to be near sinusoidal, even in the case of violent sloshing. The sloshing was modelled using a nonlinear model. Still, a convolution term was not necessary to predict the motion. Of course, transient effects cannot be modelled without using the convolution term.

In [9] the vertical motions and loads of a vessel where simulated. Some nonlinearities where included, specifically where the hydrostatic forces and the forces from the incoming wave potential integrated over the exact wetted surface. In this case, a convolution term was necessary in order to predict motions, and in particular loads, with sufficient accuracy. The same results were indicated by the same authors in [8] from experiments.

Appearantly, there are cases where a convolution is necessary, and cases when they are not. Other examples where time-domain models are routinely used is in simulations involving cables and towing. Still, there are cases where only the steady state is of interest, the frequency of motion is known a priori, and the added complexity of the convolution term is unnecessary.

7.2 Other methods of calculating the time domain model

Kaasen and Mo [21] uses least-square curve fitting of a fourth order transfer function in order to calculate the time domain model. The transfer function is fitted to the damping coefficients, or the real part of the complex transfer function from velocity to radiation force. On the assumption that it is not possible to calculate the asymptotic added mass at inifinite frequency, the paper uses the fitted damping curve to calculate these added masses.

The damping coefficients are calculated for 34 frequencies in the frequency range from about $0.2\,\mathrm{rad\,s^{-1}}$ to about $1.6\,\mathrm{rad\,s^{-1}}$. This cut-off frequency is lower then the $6.5\,\mathrm{rad\,s^{-1}}$ used in this work, and would cause greater problems when evaluating the inverse Fourier cosine transforms.

This calculation method has as its main advantage the lower calculation power needed. The models are otherwise equivalent when considering small motions around equilibrium. It is not immediately clear, however, how this method will perform if higher order transfer function are needed, perhaps caused by higher accuracy requirements or unusual dynamics. Looking at the locations of poles and zeros in the present work, there can be many near-cancellations of poles and zeros, a likely source of numerical problems. The canonical form suggested in [21] will also, as all canonical forms, fail to take full advantage of the numerical stability gained from a state-space representation.

If forward speed is introduced, and the damping coefficients do not converge to zero at the low- and high-frequency asymptotes, the method will also need some modifications.

Still, the presented method can be an additional, useful tool to avoid frequency-dependent descriptions of dynamics.

7.3 The Munk moment

The Munk moment is a destabilizing moment in yaw for vessels with forward speed. Hydrodynamic textbooks sometimes present simplistic derivations of this term. The term is, however, nothing different from the added mass part of the Lie bracket term in the kinematics. This can be seen comparing term-by-term the Munk moment expression with the terms from the Lie bracket.

7.4 Time-varying versus frequency-varying

There has been made claims that the frequency-varying property of the coefficient matrices can be adequately represented when considered as time-varying[1]. This claim lacks foundation, as can be seen by considering the simultaneous excitation of two harmonic waves with different frequencies. Then, the coefficient matrices appear to take two different values at the same moment in time, an absurdity.

Still, the coefficients, as well as the memory functions, are indeed time-varying. This is because of the changing wetted surface area, totally unrelated to the dependence on frequency. Moreover, for the limited motions of a surface vessel, the surface may not change significantly, and the matrices and memory functions may be assumed constant.

[1] This claim was made by an unknown member of the audience during the presentation of the CAMS'04 article ([19]).

7.5 The value of b_{jk}

The forces from the radiation potential is often expressed as

$$\tau_j^{\mathrm{R}} = -\left[\sum_{k=1}^{6} a_{jk}\ddot{q}_k(t) + \sum_{k=1}^{6} b_{jk}\dot{q}_k(t) + \sum_{k=1}^{6}\int_{-\infty}^{t} K_{jk}(t-\sigma)\dot{q}_k(\sigma)\mathrm{d}\sigma\right] \tag{7.1}$$

According to King [23], as referred by Fonseca [9], b_{jk} is zero. This appears in general not to be the case, and the following reasoning will show that letting b_{jk} be equal to $\beta(\infty)$ is far more reasonable.

By allowing for $b_{jk} \neq 0$ the expression for $K_{jk}(t)$ according to Cummins [5] is

$$K_{jk}(t) = \frac{2}{\pi}\int_0^{\infty} [\beta_{jk}(\omega) - b_{jk}]\cos\omega t \mathrm{d}\omega \tag{7.2}$$

For $K(t)$ to have a finite value at all t, and particularily at $t = 0$, the integrand has to tend to 0 as $\omega \to \infty$. The requirement that the impulse response is of finite energy also imposes the same restriction on the integrand.

When calculating $K(t)$ numerically, nice behaviour of the integrand is important. Numerical values for $\beta(\omega)$ are typically calculated in a frequency range not very different from $0.1 < \omega < 6$. In addition, asymptotic values at $\omega = 0$ and $\omega = \infty$ may be known. Denoting the lower and upper bounds on the calculated frequency range by L and U, respectively, and using $f(\omega)$ as a shorthand for the integrand, one can write

$$K_{jk}(t) = \frac{2}{\pi}\left[\int_0^{L} f(\omega)\cos\omega t \mathrm{d}\omega + \int_L^{U} f(\omega)\cos\omega t \mathrm{d}\omega + \int_U^{\infty} f(\omega)\cos\omega t \mathrm{d}\omega\right] \tag{7.3}$$

The middle integral can be calculated straightforwardly. The first and last, however, requires some care. Disregarding them – setting them to zero when they are not – introduces oscillations in $K(t)$. This can be seen by assuming $f(\omega)$ constant in the interval, resulting for the first integral in

$$\int_0^{L} f(0)\cos\omega t \mathrm{d}\omega = \left[f(0)\frac{\sin\omega t}{t}\right]_{\omega=0}^{\omega=L} = f(0)\frac{\sin Lt}{t} \tag{7.4}$$

and for the last integral, after in addition replacing the integration limit with a large positive number M,

$$\int_U^{M} f(\infty)\cos\omega t \mathrm{d}\omega = \left[f(\infty)\frac{\sin\omega t}{t}\right]_{\omega=U}^{\omega=M} = f(\infty)\left(\frac{\sin Mt}{t} - \frac{\sin Ut}{t}\right) \tag{7.5}$$

as expressions for the errors added to the impulse response, assuming constant value of f above the cutoff frequency.

Looking again on (7.2), one can see that if $b_{jk} = 0$ and $\beta(\infty) \neq 0$, $K(t) \rightarrow \infty$ as $t \rightarrow 0$. This makes accurate calculations of $K_{jk}(t)$ very difficult. If one indeed argues that $b_{jk} = 0$, the following approach for calculating $K_{jk}(t)$ can be used. $K_{jk}(t)$ is still given by the inverse transform

$$K(t) = \frac{2}{\pi} \int_0^\infty [\beta(\omega)] \cos \omega t \mathrm{d}\omega \tag{7.6}$$

Addition and subtraction of $\beta^\infty = \beta_{jk}(\infty)$ from the integrand gives

$$\begin{aligned} K(t) &= \frac{2}{\pi} \int_0^\infty [\beta(\omega) - \beta^\infty + \beta^\infty] \cos \omega t \mathrm{d}\omega & (7.7) \\ &= \frac{2}{\pi} \int_0^\infty [\beta(\omega) - \beta^\infty] \cos \omega t \mathrm{d}\omega + \frac{2}{\pi} \int_0^\infty [\beta^\infty] \cos \omega t \mathrm{d}\omega & (7.8) \\ &= K_a(t) + K_b(t) & (7.9) \end{aligned}$$

Inserting this into the expression from the forces, and holding on to the assumption that $b_{jk} = 0$, looking only at one mode and omitting indexes

$$\tau^{\mathrm{R}} = -\left[a\ddot{q}(t) + \int_{-\infty}^t K(t-\sigma)\dot{q}(\sigma)\mathrm{d}\sigma\right] \tag{7.10}$$

$$\tau^{\mathrm{R}} = -\left[a\ddot{q}(t) + \int_{-\infty}^t K_a(t-\sigma)\dot{q}(\sigma)\mathrm{d}\sigma + \int_{-\infty}^t K_b(t-\sigma)\dot{q}(\sigma)\mathrm{d}\sigma\right] \tag{7.11}$$

The inverse Fourier cosine transform of a constant is an impulse ($\delta(t)$), giving

$$K_b(t) = \delta(t)\beta^\infty \tag{7.12}$$

which in turn gives the following expression for the forces from the radiation potential

$$\tau^{\mathrm{R}} = -\left[a\ddot{q}(t) + \beta^\infty\dot{q}(t) \int_{-\infty}^t K_a(t-\sigma)\dot{q}(\sigma)\mathrm{d}\sigma\right] \tag{7.13}$$

This shows that setting $b = 0$ “pushes” information into the impulse response, resulting in an awkward expression that is difficult to calculate numerically. Precalculation manipulation results in an expression with equivalent structure, the coefficient has only changed name.

In conclusion, b_{jk} is only zero when the damping coefficient tends to zero at high frequencies. Otherwise, $b_{jk} = \beta^\infty$.

7.6 The radiation restoring coefficient

The exact nature of the radiation restoring coefficient (c) is still somewhat elusive, especially why and how the speed-dependence appears. The various theories have differing expressions, [26] provides a summary.

With zero forward speed, c is always zero.

The vessels equilibrium position changes as speed increases. None of the methods mentioned here accounts for this effect. It is then not obvious what coordinate system to use when calculating these restoring forces. Note also that this is an effect that is linear in speed, unlike convential lift/drag effects that are quadratic in speed.

Both [26] and [1] arrive at added mass and potential damping that are speed-independent. In the first reference, all speed-dependence is found in the restoring coefficient. The latter reference performs calculations in the body frame, making it unsurprising that no restoring forces appear.

Chapter 8

Excitations

The sea state is described using a JONSWAP spectrum with 8 s peak wave period and 2 m significant wave height. Wave elevation and wave loads are calculated using superposition of 1000 wave components, with wave load in mode i given by

$$\tau_i(t) = \sum_j K_{i,j} A_j \sin(\omega_j t - k_j x \cos\theta_j - k_j y \sin\theta_j + \varphi_{i,j} + \varepsilon_j) \tag{8.1}$$

where A_j is the amplitude of component j at frequency ω_j

$$A_j = \sqrt{2S(\omega_j)\Delta\omega} \tag{8.2}$$

k_j is wave number ($k = \omega^2/g$), x and y are vessel position, $K_{i,j}$ and $\varphi_{i,j}$ represent response amplitude and phase, and ε_j is a random phase for each wave component. In addition, $K_{i,j}$ and $\varphi_{i,j}$ vary with the direction of the incoming waves, relative to the vessel.

All coefficients are calculated from WAMIT, using the same frequency interval as the hydrodynamic coefficients, and for 9 evenly spaced wave headings between 0° and 180°. These coefficients are then interpolated linearily to a 1° resolution. Symmetry is used to compute the remaining coefficients.

Part III

Implementation and data

Chapter 9

Existing tools and data

Data from two different sources have been used in this work. A strip model of an S175 containership, the same as used in [9] was used initially. The S175 containership is a design frequently used for model experiments and simulations, in order to easily compare results from different sources. The available strip theory program provided data for vertical motions, and this data has been used to generate state-space models representing forces from the radiation potential. Frequency-varying coefficients were given at 49 frequencies in the range 0.005 to 5.71 rad s^{-1}. The program also provides memory functions. The memory functions had a 0.047 s sample time, and were 40 s long. Main particulars for the ship are summarized in table 9.1.

The program is implemented in-house at Instituto Superior Tecnico, in the FORTRAN programming language. 2D coefficients are calculated using Frank's close fit method [15]. The software can also be used for time-domain simulations of vertical motion and loads, [8] reports on verification with experimental data.

For use in a full six degrees-of-freedom simulation, this model was not suitable. The same vessel data as in [19, 18] was therefore used. This vessel is a 110 m long offshore work vessel, used for various construction purposes, including pipe-lying. The geometry is given using 1700 flat panels to describe one side of the hull. The ship is symmetric about the xz-plane. The hydrodynamic coefficients, as well as exciting forces and moments, were calculated using WAMIT, at many frequencies on a large frequency range.

Length	175 m
Beam	25.4 m
Draught	9.5 m
Displaced volume	24140 m^2
Froude number	0.25

Table 9.1: S175 main particulars

Chapter 10

Implementation

10.1 Interpolation of hydrodynamic coefficients

For accurate calculation of the impulse response from equations such as (5.14) or (5.15) it is necessary to have a smooth description of $\alpha(\omega)$ or $\beta(\omega)$ over the integration interval. As the calculation of the coefficients is somewhat time-consuming, it is customary to interpolate the calculated values in some manner. In [24] and [19] linear interpolation is used. In the first case, a limited number of values were available, in the latter case linear interpolation was believed to be sufficient as the coefficients were calculated at a large number of frequencies.

Using the available strip theory data, this approach was problematic, as high resolution data was not available. This was in particular problematic at low frequencies, where the coefficients appeared to diverge near $\omega = 0$. In addition, zero frequency asymptotic values of $\alpha(\omega)$ were not known. It is also desirable to avoid manipulations that are likely to affect the coherence of the (α, β) pair. The investigation was triggered by the numerical calculation of radiation restoring coefficients different from zero in several modes.

Figures 10.1 through 10.3 show the frequency-dependent coefficients in three modes. The large changes in function value near $\omega = 0$ can indicate that the choice of interpolation method affects the values of the functions near $\omega = 0$. The lack of asymptotic values at $\omega = 0$ can also cause problems.

The coefficients are calculated using a strip theory approach with linear and quadratic speed dependent terms included. These terms can easily be calculated from the zero-speed coefficients. The desired end-result of the calculations at hand is to generate impulse responses using inverse Fourier transforms of the frequency-dependent data. As the transforms are linear, this process can be done separately on the terms in the strip-theory formulation.

Figure 10.4 shows added mass and damping in the coupling mode from pitch to heave at zero speed. Comparing the data to the Froude number 0.25 in figure 10.2 one sees that the damping coefficient in the zero speed case the approaches zero at zero frequency, as expected. The strip-theory expression for $\beta_{35}(\omega)$ (without the

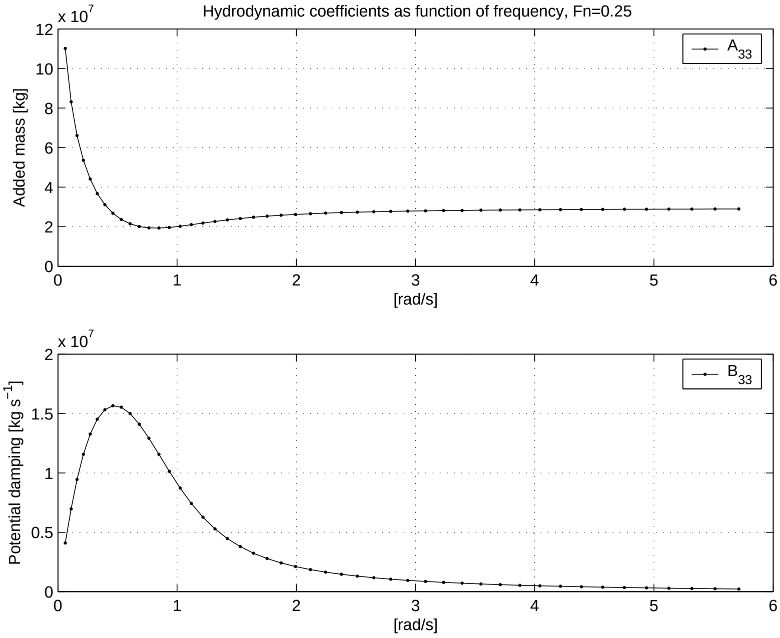

Figure 10.1: Frequency-dependent coefficients

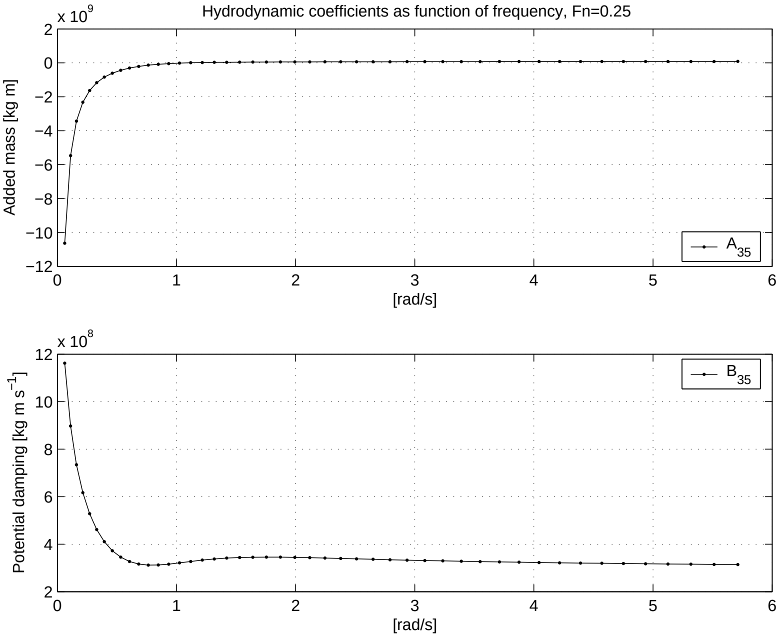

Figure 10.2: Frequency-dependent coefficients

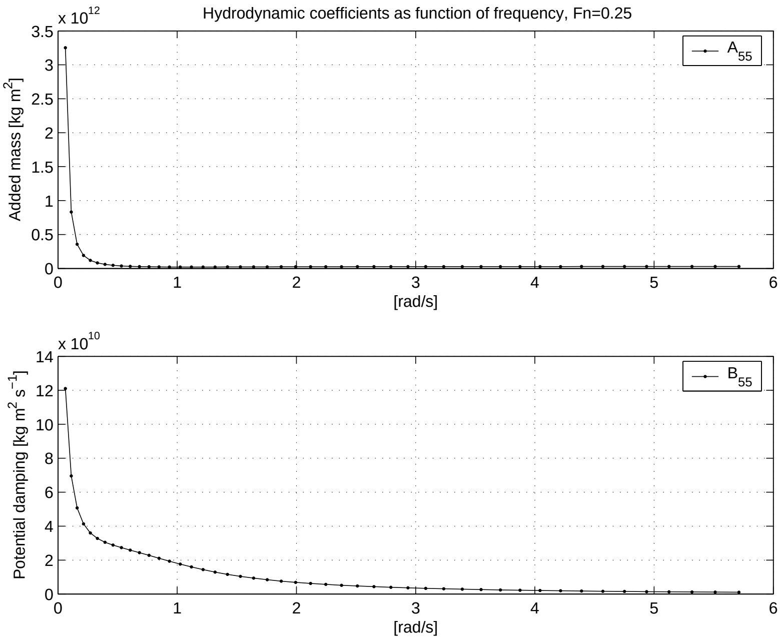

Figure 10.3: Frequency-dependent coefficients

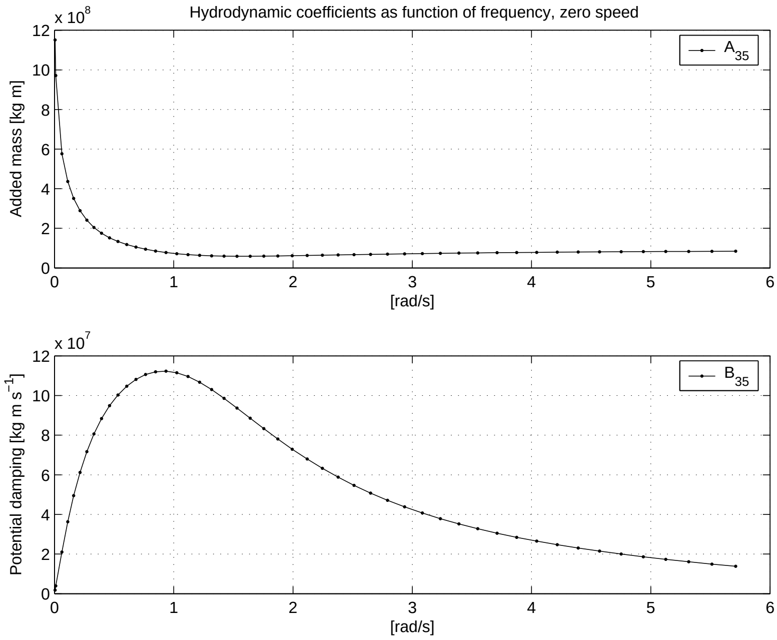

Figure 10.4: Frequency-dependent coefficients, from pitch to heave, zero speed

end terms) is

$$\beta_{35}(\omega) = \beta^0_{35}(\omega) + U\alpha_{33}(\omega) \tag{10.1}$$

Evaluating the inverse Fourier cosine transform of the last term

$$\frac{2}{\pi}\int_0^\infty (\alpha_{33}(\omega) - \alpha_{33}(\infty)) \cos\omega t \,\mathrm{d}\omega \tag{10.2}$$

needs some care, as the added mass in heave changes rapidly close to $\omega = 0$. Using the calculations of the radiation restoring coefficient as a metric, different interpolation methods can be tested against each other.

Motivated by a need to handle the coefficient pair in a consistent manner, one can seek ways of treating the pair as one unit. The strip-theory software used calculates the coefficients from the complex force coefficent T resulting from

$$\tau = -T\dot{q} \tag{10.3}$$

for harmonic motion, $q(t) = \mathrm{e}^{-\mathrm{i}\omega t}$. T relates to the added mass and damping coefficients by

$$T(\omega) = \omega^2\alpha(\omega) - \mathrm{i}\omega\beta(\omega) \tag{10.4}$$

T is computed (10.4) at the known frequency points, and interpolated using cubic spline interpolation in the complex plane. This makes it possible to use the known asymptotic values

$$T(0) = 0 \tag{10.5}$$

$$\lim_{\omega\to\infty} T(\omega) = \omega^2\alpha^\infty \tag{10.6}$$

when interpolating. Infinite frequency data is represented by placing points three points on the real axis for frequencies 10, 11 and 12 rad s^{-1} above the highest frequency data point (5.7 rad s^{-1} in the used data set). The coefficients can thereafter be regained from (10.4). Figure 10.5 shows the resulting difference in coefficients between this method and straightforward cubic spline interpolation. Note the scale on the x-axis. The lowest data point of the original dataset is 0.005 rad s^{-1}. Even though differences can be seen, they occur over a very small frequency range. If, however, as in the first available data, the lowest data point is at 0.11 rad s^{-1}, the difference is easily seen in the impulse response.

10.2 Calculating impulse responses

With fine-grained values available, either from calculations at many frequencies or interpolation, the various memory functions can be calculated using the inverse Fourier transforms. As the integral has as its limits between 0 and ∞, something has to be done in order to efficiently get numerical values. It is possible to use an analytical extension of the hydrodynamic coefficients at high frequencies [16, 29],

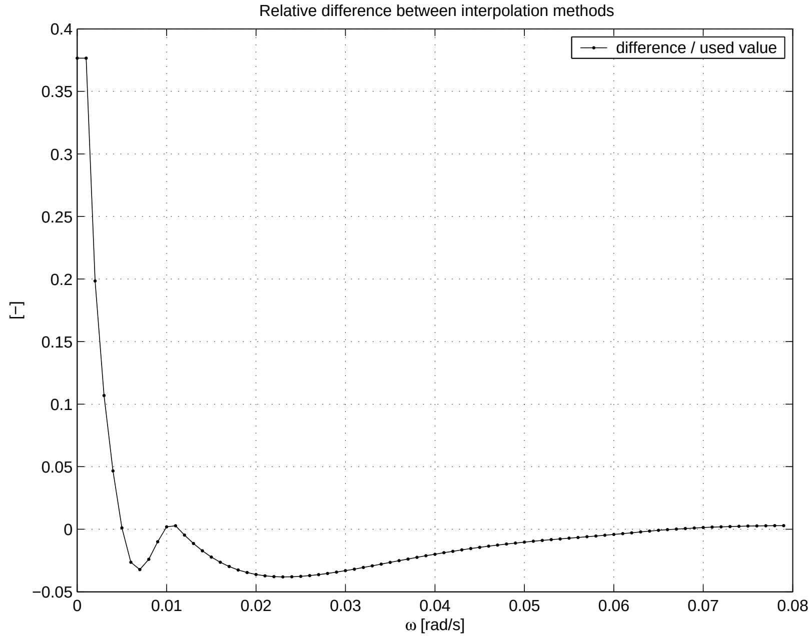

Figure 10.5: Results from different interpolation methods, added mass in heave

and integrate the high-frequency part analytically. In these references, only zero-speed, 2D data is used. In the present application this is not sufficient. Also, the analytical extensions are possibly different for each coupling mode.

In this work it was chosen for simplicity reasons to calculate the coefficients for a large frequency range, making the added accuracy offered by analytical extensions of the remaining part of the integral no longer needed.

The memory function will eventually be represented using a reduced order state-space model. Because of this, it is only necessary to calculate the memory function at a limited number of points, smaller then would typically be the case if the memory function where to be used for simulation.

The error introduced by truncating the transform is more significant when calculating K near $t = 0\,\mathrm{s}$ than for larger values of t. This is because $\cos \omega t \approx 1$ for $t \approx 0$, causing the calculated value to be lower than the actual value. Looking at the plots, this can be the reason for the flattening of K near $t = 0\,\mathrm{s}$. The effect has not been quantified in this work. If it is significant, it might be corrected for by using interpolated values of K for low t, or perhaps only at $t = 0\,\mathrm{s}$.

When using strip theory coefficients, the added masses diverge towards infinity near $\omega = 0$. The integral may still exist analytically. Uncertainty near $\omega = 0$ is similar to adding a constant to the impulse response. The rapid increase near $\omega = 0$ is also seen in the time domain as a very slowly decaying memory function. For simulation purposes, the memory function must be known for all the time until it converges to zero. A slowly converging memory function makes it necessary to calculate the function at many points, increasing simulation cost. The computational cost of creating the reduced-order state-space model is also heavily dependent on the number of points in the impulse response. In some modes, the calculation appears infeasible on a regular office computer.

The impulse response $K(t)$ for $t > 80\,\mathrm{s}$ contains information about the behaviour at frequencies $\omega < 0.08\,\mathrm{rad\,s^{-1}}$. In this frequency range hydrostatic forces dominate. It therefore seems reasonable to truncate the memory function, and replace the truncated part with a coarse approximation. The simplest method, and the one used in this work, is to only calculate K up to $t_{\max} = 80\,\mathrm{s}$. In general, $K_{\text{end}} \triangleq K(t_{\max}) \neq 0$. The original K is then replaced by $\tilde{K}(t) = K(t) - K_{\text{end}}$. The impulse response $\tilde{K}$ now reaches zero within the calculated interval, and the subtracted constant can be treated individually. This is of course a crude and pragmatic approach. However, numerical simulations show that the resulting models show the desired behaviour in the interesting frequency range.

The equivalent of a constant impulse response in the s-plane is a pure integrator. The effect of the subtracted K_{end} can thus be modelled using a radiation restoring coefficient; a constant times the velocity integrated. Alternatively, the pole can be shifted slightly to the left.

10.3 Generating reduced order model

The reduced order state-space model is generated from the memory functions using functions from the Matlab Robust Control toolbox: `imp2ss` and `schmr`. In this process, there are several potential pit-falls and implementation details that are important. These will be described in this section.

The notion of an impulse response in Matlab is different for a continous time and a discrete time model. In the continous time case an impulse is the Dirac delta function (δ), whereas in the discrete time it is a unit pulse lasting one step length. The Dirac delta has area 1, the discrete time unit pulse has area Δt. The input to `imp2ss` is an impulse response given at discrete time intervals, and is interpreted to be the result of a unit pulse, not a Dirac delta. This is done regardless of whether a step length is specified as an argument to `imp2ss`.

Because of this, the impulse response calculated from the inverse transform must be scaled according to time step length, or equivalently, the generated state-space model must be scaled similarily. This has been a cause of some trouble, the solution, but not the cause, being also presented in [24].

Another possible cause of trouble relates to the specifics in the conversion from discrete to continous time. In this work a Tustin transformation is used

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right) \tag{10.7}$$

where T is sample period, using standard Matlab functions. The original impulse response is given in continous time. When calculating it at a finite number of points in time, and using this as input in the model identification, for thereafter to use a Tustin transformation to convert back again to continous time, errors are introduced. To compensate for this effect, the calculated impulse response is shifted left one half time step.

Previously, no significant model reduction was done in the function call to `imp2ss`, all reduction was subsequently done starting with the large state-space model. This was because the truncation tolerance options in `imp2ss` were not found to be suitable, the only option being the H^∞-norm of the error between the approximate and exact realizations. Doing the model reduction on a large state-space model is costly, involving matrix operations, meaning rapidly growing computational cost as the number of states increase.

Specifying reasonable values for the H^∞-norm of the error is much easier if the model is scaled in advance. The impulse response is therefore scaled so that $K(0) = 1$.[1] The H^∞-norm of the error is then specified to be in the interval between 0.01 and 0.1. The ability to freely specify the tolerance in this manner proved to be very helpfull, especially when debugging the code, significantly shortening computational time.

If a model of a specific order is of interest, one can then reduce the model further using `schmr`. Typical model orders pr coupling are in the area between four and eight.

Previously, in [18, 19], the method `balmr` was used instead of `schmr`. According to the Matlab documentation, `schmr` has numerical advantages when a balanced realization is not needed. `balmr` returns a balanced realization, `schmr` does not. As the input-output-behaviour of the resulting models are identical, `schmr` is used in this work.

10.4 Finding the vessel-parallel coordinate system

Online determination of the vessel-parallel (VP) coordinate system poses some challenges. If the vessel was following a specified trajectory, one could use the Serret-Frenet frame[2] as the vessel-parallel coordinate system. Requiring that the desired or planned trajectory is known by the simulator creates an unwanted dependence of the simulator on a trajectory planning system. Moreover, trajectory planning has not been a topic in this work.

The VP system must be an equilibrium frame. As centre of boyancy and centre of gravity align (by assumption), the horizontal planes of the VP system and the inertial system are allways parallel. The VP system will in this work only be used to do the coordinate transformation of forces, moments and velocities. The

[1] It may in general be better to scale so that $\max K(t) = 1$, but the difference is not believed to be of significance.

[2] For explanation of the Serret-Frenet frame see [6, p. 253ff].

location of the VP origin is thus not of interest. Following this argument, the VP transformation is calculated by low-pass filtering the yaw angle with a first order filter $(1/(1+Ts))$ with time constant $T = 25\,\mathrm{s}$, and using the filtered yaw angle as input to modified coordinate transformation blocks from the GNC toolbox [11]. The time constant was chosen, after some experimenting, to be three times as long as the peak wave period.

10.5 Differences from the previous model

As the developed simulator takes as its basis previous work, it is of interest to highlight some differences between them. Forward speed effects are in addition to the ones mentioned below.

Support for centre of gravity different from coordinate system origin The kinematics module has been revamped in order to allow for differing coordinate system origin and centre of gravity. Furthermore, according to WAMIT documentation, the inertia matrix is supposed to be given in the body coordinate system. In the supplied dataset, the inertia matrix is diagonal, but the centre of gravity is not in the body coordinate origin. It is assumed that this is an error in the dataset, and corrected for by translating the inertia matrix using (3.217) from [12].

Vessel-parallel calculation of radiation, wave and hydrostatic forces The derivation of radiation, wave and hydrostatic forces is done in an equilibrium frame. In the previous work, the assumption of small motion was invoked, and the forces were applied as if they were given in the body frame. This is now corrected for, using proper transformations between the two coordinate systems.

Support for short-crested sea The calculation of wave forces now allows for a short-crested wave spectrum.

10.6 Program structure

While developing the software, it quickly became evident that program structure would be important for the quality of the results. The algorithms used have a number of parameteres that significantly affect accuracy and computational cost. Moreover, the complete calculation consists of many steps, each with temporary, intermediate results.

The trust in the produced results relies on the ability to consistently reproduce them. This should include preliminary findings, initial investigations as well as the final results. Developing the software can be a very exploratory process, and as such the program code is very evolving. The requirement to be able to continously reproduce previous results makes it necessary to tackle this problem with some care. A specific program structure has been crafted with this in mind. This section is devoted to describing this structure.

The presented solution is not believed to be very optimal. In many ways it is made necessary by inherent limitations in the Matlab programming language. It is also very memory-consuming in its current implementation. However, the usefullness of Matlab as a prototyping language makes it still worthwhile to adapt to its shortcomings.

Four ideas are pivotal to the design, summarized below.

Option structures, like the ones frequently found in the Optimization toolbox, are used everywhere where there is a tuning parameter. Each subroutine has sensible defaults provided, but all can be overridden. This makes it possible to externalize and automate the code that tries several different options, to see how the results are affected.

Data structures represent the state during calculation. As state is considered indata, intermediate and final results. Storing these in a single structure makes it possible to have consistent snapshots of processing state that can be stored, and later recalled.

A graph stores all previous data and option structures. The process is divided into a series of steps performed in sequence. At the end of each step, the option and data structures are written to the graph, together with a name. Similarily, each step begins with referencing the predecessor and fetching the previous option and data. The graph thus consists of the results of all previous computation steps.

Highly automated plotting of diagrams makes it easy to recreate them in case something changes. All generated plots are allways written to disk in several formats. Easy viewing is facilitated by the PNG graphics format[3], Postscript is used for the publication quality graphics. In the case subsequent editing of the plot is necessary, it is also stored in the native Matlab figure format. Furthermore, no plotting code is included in the algorithm subroutines themselves. One small function is written for each kind of plot, describing legends, labels and what data should go where.

To illustrate the usability of this approach, creating plots of all modes, comparing any number of different calculation approaches, is done with a single line of code. The generated plots are stored in a structured system of directories, making them easy to include in the final work. As only links to the graphics are used in the report, regenerated graphics are automatically included.

[3] Portable Network Graphics: http://www.w3.org/Graphics/PNG/

Chapter 11

Simulator

The simulator is implemented in Simulink. The implementation builds in parts on previous work of the author [18]. The rigid body kinematics of [18], originally from [39], has been partially rewritten to allow for centre of gravity different from the origin of the coordinate system. The forward speed effect in the radiation potential is accounted for by modelling the forces using the state-space representation of (3.28).

Forces from the radiation potential, as well as restoring forces, are modelled in the vessel-parallel frame. The orientation of this frame is calculated as described in section 10.4. For the radiation forces, velocities in the body frame are transformed to the vessel-parallel frame before entering the state-space models, and transformed back afterwards. The varying speed is multiplied *before* the state-space model. This choice, as opposed to after or on both sides, was taken without special considerations, and is rather arbitrary.

Wave forces are calculated using superposition of any number of sinusoidal wave components with arbitrary frequency, amplitude and propagation direction. These forces are also calculated in the vessel-parallel frame, although instantenous positions and orientations are used.

To test the simulator with (very) simple maneuvering, a course-keeping autopilot is used. The design goal for the autopilot was only to generate plausible control input to the simulator, and because of this, many simplifications are made. Most importantly, it only works between $-180°$ and $180°$. The autopilot consists of two monovariable controllers, using deviations in sway and yaw to generate thrust commands in the same modes. The wave affected position measurements are filtered using a notch filter as found in [12], with peak frequency $\omega = 1.1(2\pi/8\,\mathrm{s})$, and relative damping $\zeta = 0.1$.

The control rule is a limited PD

$$K_p \frac{Ts+1}{\alpha T s + 1}, \quad \alpha < 1 \tag{11.1}$$

where K_p is chosen as $(M_{22} + A_{22})/40$ in sway and $(M_{44} + A_{44})/20$ in yaw, $\alpha = 0.1$ and $T = 15\,\mathrm{s}$. These choices were arrived at after a few tuning iterations.

The forward control thrust is predifined, starting from zero and approaching a fixed value. This is implemented as a low-pass filtered step input, $T = 20\,\mathrm{s}$. No thruster modelling is included.

Part IV

Results

Chapter 12

Verification of radiation model

Calculations using frequency-domain and time-domain models should match in the case of harmonic motion. In this work, it is the model of the radiation forces that has went through significant processing, and as such has the most need for verification. This chapter is devoted to results from comparing frequency-domain and time-domain models, investigating sensitivity to methods and parameters in the implementation. All data used are from the 2D strip-theory model of the S175 containership.

12.1 Testing set-up

The purpose of the test is to see whether the time-domain models generate the same output as the frequency-domain models. First, the absolute value of the complex force coefficient

$$T(\omega) = \omega^2\alpha(\omega) - \omega\beta(\omega) \tag{12.1}$$

is calculated over the frequency-range of interest. The coefficients used are from the fixed forward speed dataset (Froude number 0.25, approx $10\,\mathrm{m\,s^{-1}}$).

For each frequency and mode coupling, simulations are run using the various time-domain formulations; memory functions and state-space models. The simulations are open-loop simulations of a forced motion, considering the resulting radiation forces as output. The time-domain expression used is, in the case of the convolution integral formulation

$$F = a\ddot{q}(t) + b\dot{q}(t) + \int_{-T}^{t} K(t-\sigma)\dot{q}(\sigma)\,\mathrm{d}\sigma + cq(t) \tag{12.2}$$

and the equivalent for the state-space formulation. b and c are in some cases zero. The impulse response is of finite length, so T is chosen to be the length of the impulse response. The simulation runs for one wave-length of excitation after the convolution integral kernel is “filled”. The maximum calculated force is then used

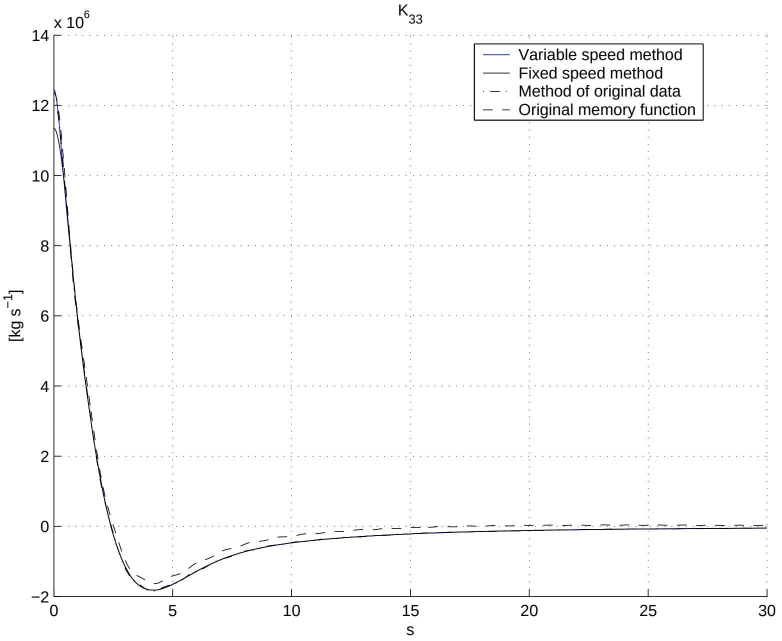

Figure 12.1: Impulse response, S175, heave, verification of calculation methods

for comparison with the frequency-domain coefficients. Two implementations were made, one using a custom S-function in Simulink, written as a Matlab function, and doing the entire simulation in a Matlab script. For efficiency reasons, it is easier to write memory efficient convolution evaluations in Matlab then model them in Simulink, the latter approach was chosen for the presented results.

In the state-space case, simulations are run in a similar manner. Calculating the forces from the state-space model was done using `lsim`. One can of course argue that one could get the same results by looking at the frequency-response of the state-space model, using i.e. Bode plots. For ease of comparison with the convolution integral results, and to use as similar code as possible for the two cases, this option was not considered further.

12.2 Initial comparisons

A large variety of methods of calculating impulse responses have been tested. In the work of Fonseca and Soares ([9]), b was set to zero. This is in contrast to Kristiansen ([24]), where b was chosen to be the infinite frequency damping. This approach was used both for the fixed-speed data-set, as well for forward speed calculated from zero-speed, such as in equation (3.28). For the data from Fonseca, the original, supplied, memory functions is also compared with a supposedly equivalent calculation using the fixed-speed coefficients.

Figure 12.1 shows the impulse response in heave. The original memory function

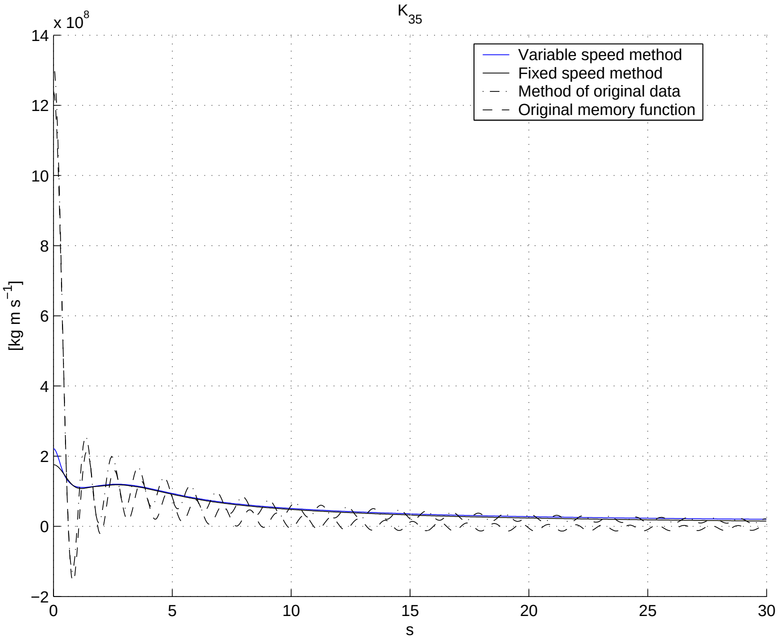

Figure 12.2: Impulse response, S175, pitch→heave, verification of calculation methods

differs from the others. This is believed to be caused by the truncation of the inverse Fourier transform at a low, non-zero frequency, as opposed to extrapolating the coefficients towards zero frequency and including these in the integral.

In figure 12.2 the coupling mode pitch to heave is plotted. The coupling coefficients here have a non-zero damping at infinite frequency. Not subtracting this value when evaluating the integral leads to large values at $t = 0$ and oscillatory behaviour. The coupling in the other direction shows similar results.

Finally, figure 12.3 shows the pitch mode. The pitch damping approaches infinity for low frequency (due to the $1/\omega^2$ term), making the error introduced when truncating the integral very large.

To verify the calculations of impulse responses, the added mass and potential damping can be calculated using Fourier transforms on the memory functions. Some results are shown in figures 12.4 through 12.5.

The effect of truncating the integral at low frequencies are clearly seen in both plots. Note also in figure 12.5 how the damping falls to zero a bit after $5\,\mathrm{rad\,s^{-1}}$, an artifact from failing to handle infinite frequency damping properly. The method used in this work compares very well with the original data, indicating no numerical problems in the impulse response calculation in these modes.

In pitch, matters are more complicated. In this mode, the memory function is not the same whether they are calculated from the added mass or the damping. Also, there appear to be some numerical problems near $\omega = 0\,\mathrm{rad\,s^{-1}}$.

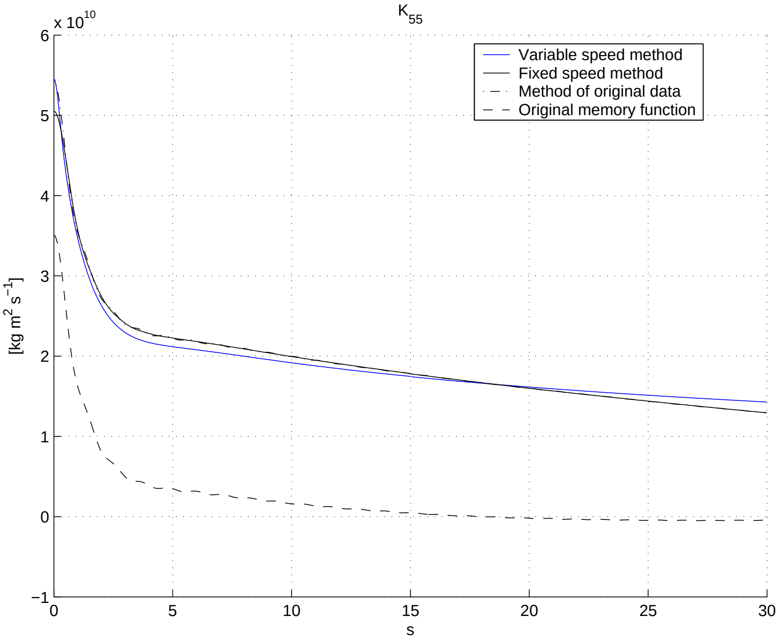

Figure 12.3: Impulse response, S175, pitch, verification of calculation methods

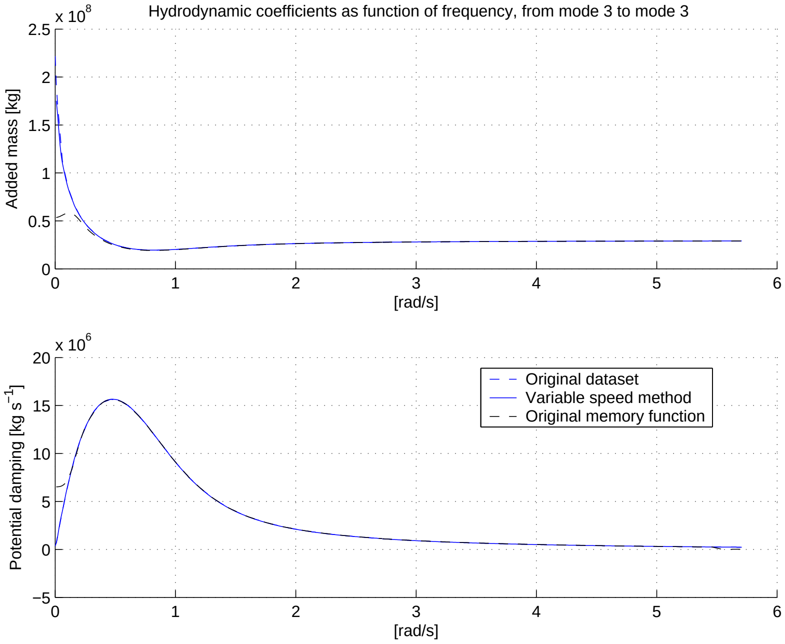

Figure 12.4: Recalculated hydrodyn. coeff., heave, S175

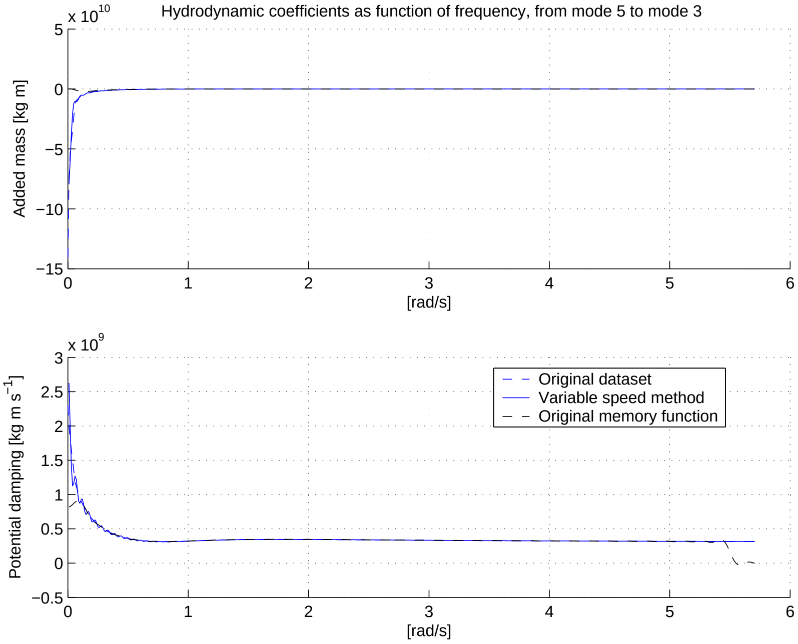

Figure 12.5: Recalculated hydrodyn. coeff., pitch→heave, S175

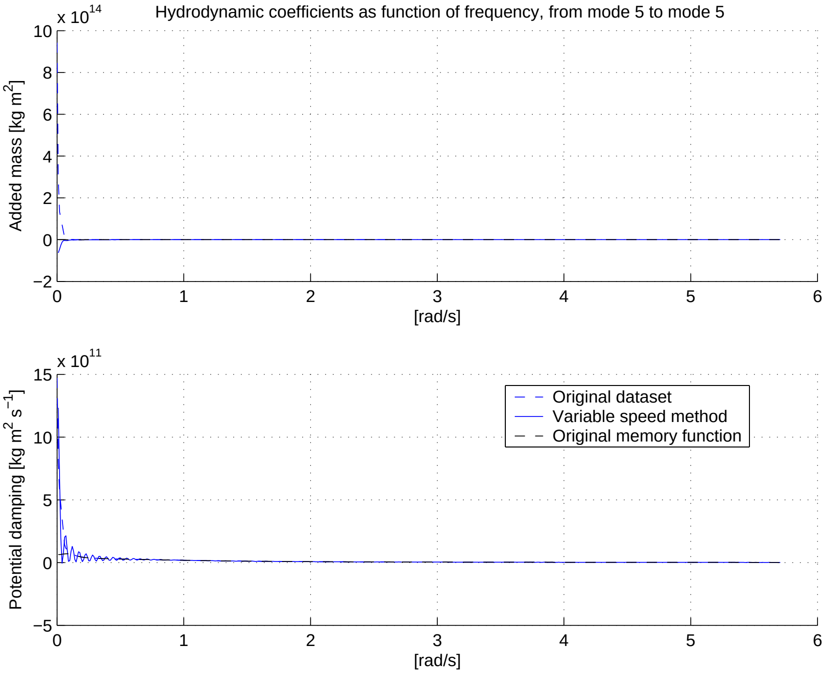

Figure 12.6: Recalculated hydrodyn. coeff., pitch, S175

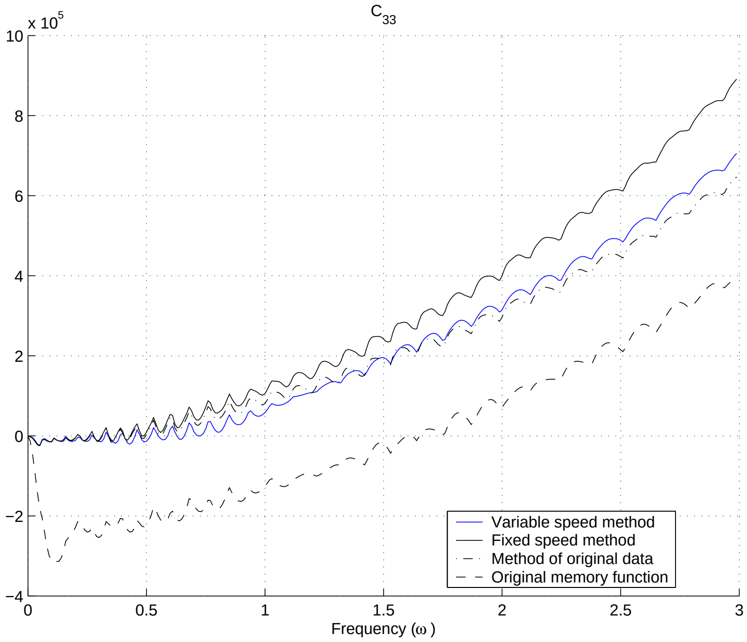

Figure 12.7: Radiation restoring coefficient, pitch

Radiation restoring coefficient

Recall that the radiation restoring coefficient was introduced to reconciliate frequency-domain and time-domain data. The coefficient is calculated for the same modes as above, shown in figures 12.7–12.10.

Turning the attention first to the coupling modes, figures 12.8 and 12.9, it is evident that it is necessary to allow for $b \neq 0$ in order to get good results for c. The apparent problem in figure 12.7 is in reality of no importance, noting the scaling of the graph. The observed phenomenon is the same as seen to a varying extent in all graphs, a slight upwards curving. This is believed to be caused by errors introduced by using the trapezoidal method in evaluating the integral. In fact, the amount of curving is significantly affected by the step length.

Time-domain simulation

Results from time-domain, forced-motion tests are in figures 12.11 through 12.13. On the y-axis is the relative error between f.d. and t.d. amplitude

$$\frac{\text{t.d.} - \text{f.d.}}{\text{f.d.}} \tag{12.3}$$

In heave, results compare well, except at very low frequencies. This is of no practical consequence, as hydrostatic forces dominate. There are no other

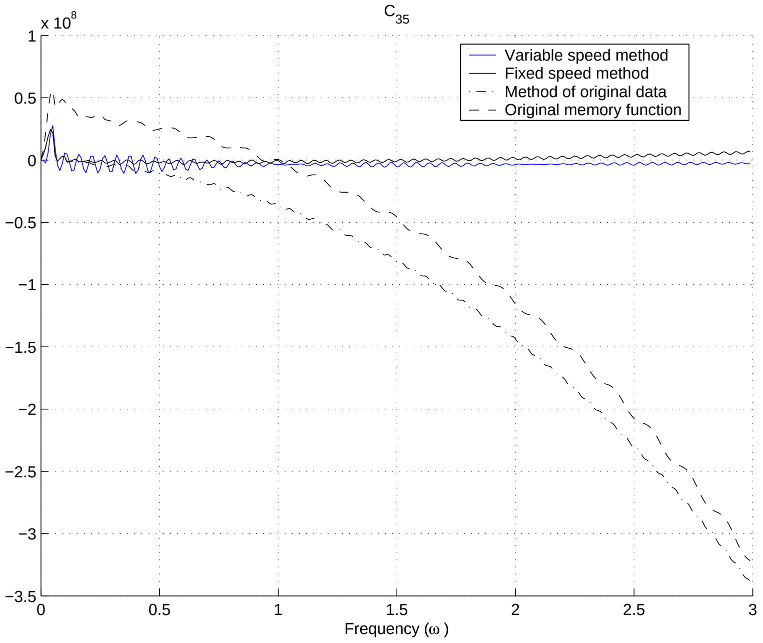

Figure 12.8: Radiation restoring coefficient, heave→pitch

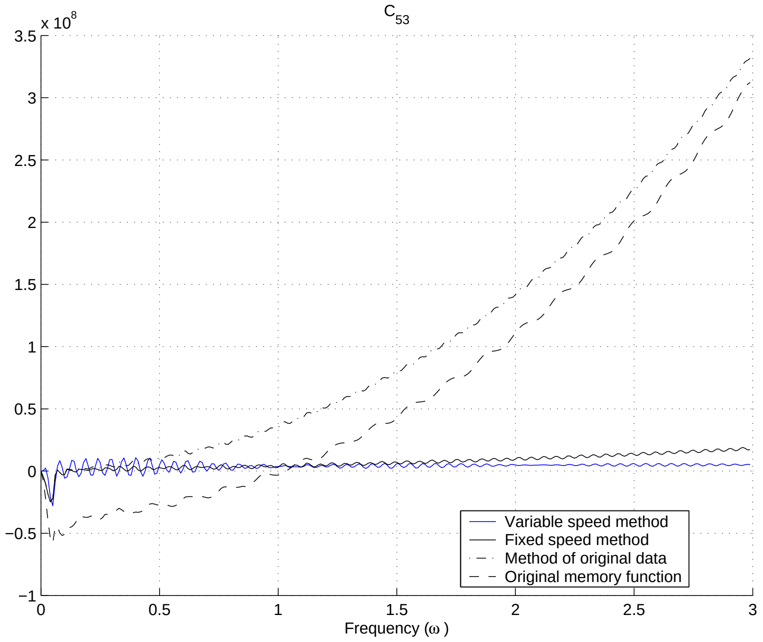

Figure 12.9: Radiation restoring coefficient, pitch→heave

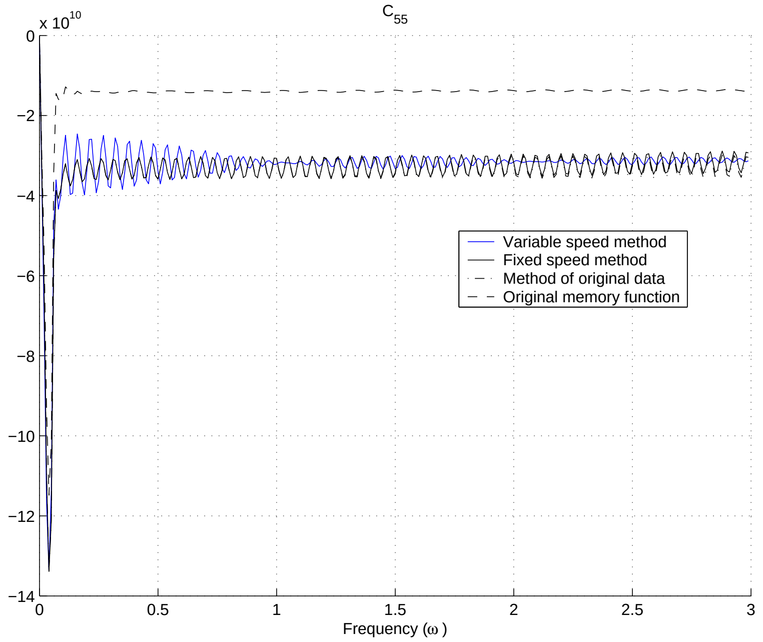

Figure 12.10: Radiation restoring coefficient, heave

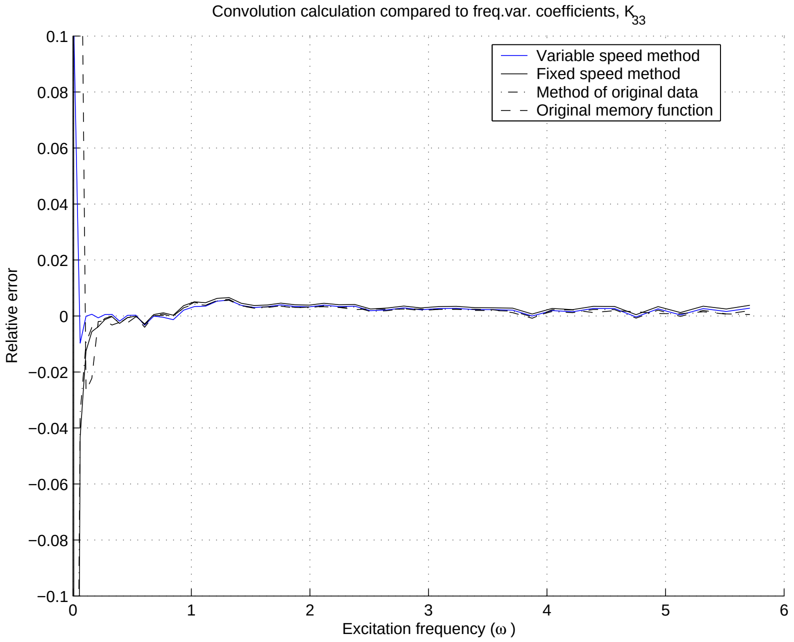

Figure 12.11: Time domain, frequency domain comparison, heave

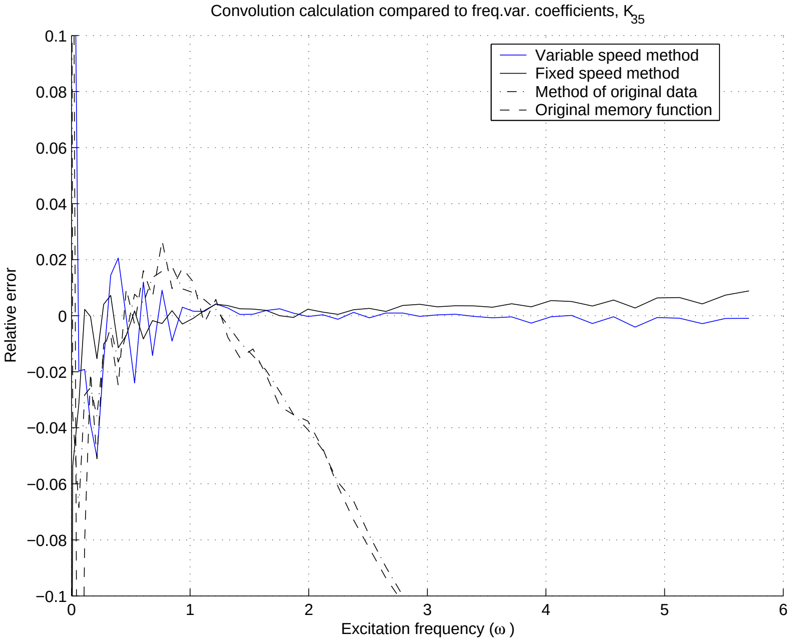

Figure 12.12: Time domain, frequency domain comparison, heave→pitch

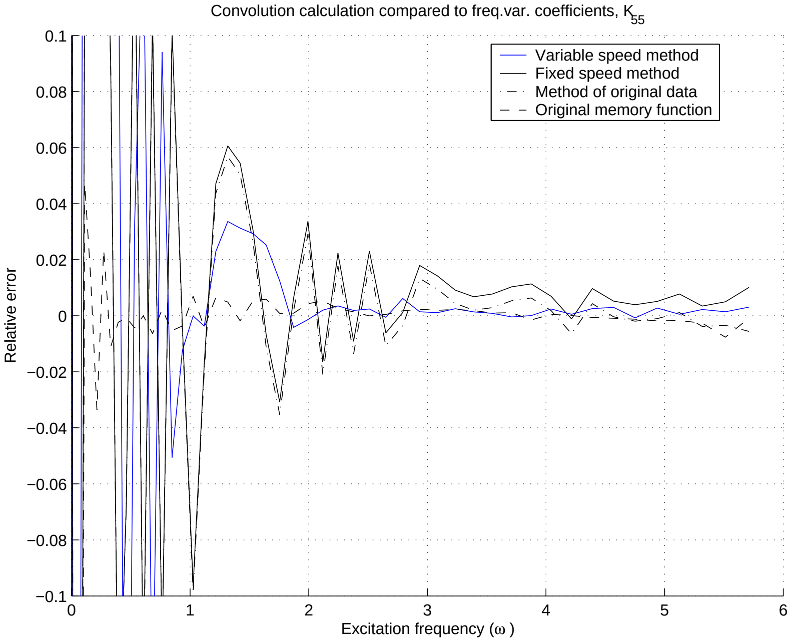

Figure 12.13: Time domain, frequency domain comparison, pitch

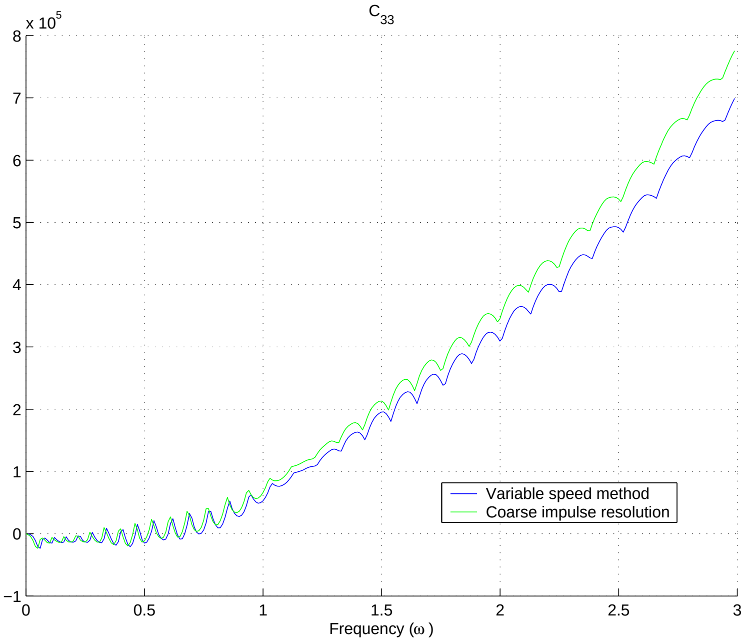

Figure 12.14: Radiation restoring, calculated with coarse resolution memory function.

differences between the methods. In the coupling modes, as in figure 12.12, failing to account for non-zero b leads to large errors in the calculated forces at higher frequencies. The effect of this may not be significant, especially since there is little vessel motion at these frequencies. However, the major benefit is the easy selection of a good value of c.

Variations are seen around $0.2 < \omega < 0.8\,\mathrm{rad\,s^{-1}}$. These are caused by stopping the memory function before it reaches zero, an effect that will be removed in later sections. This is also the reason for the undesired behaviour dominating in figure 12.13, for pitch motions.

12.3 The effect of memory function resolution

The importance of the number of points in time that the memory function is evaluated was investigated by changing the time step from 0.05 s to 0.1 s. No significant effect was seen, except in the calculation of the restoring coefficient, supporting the previously mentioned hypothesis that integral step-size causes some upwards curving (figure 12.14).

The effects on the calculated forces where neglible, illustrated by figure 12.15.

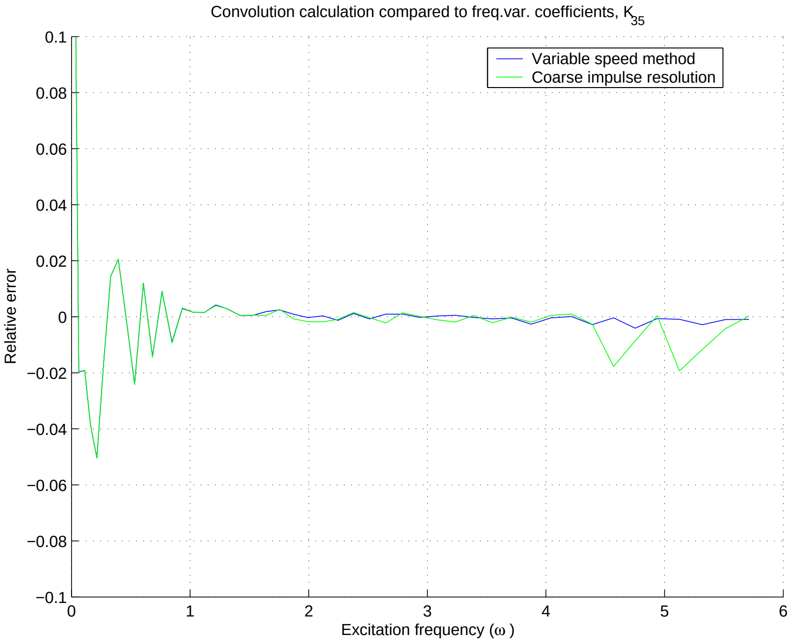

Figure 12.15: Effect of coarse impulse on force calculation

12.4 Memory function truncation

The number of sample points in the memory function is best kept low by keeping the memory function as short as possible. It also gives little value to calculate memory functions for longer than $t = 2\pi/\omega_{\text{lowest}}$, where lowest means the lowest frequency of the f.d. data. Initially, it can appear reasonable to calculate the memory function until it converges to zero. In the case of pitch motion, this is not feasible, as it converges very slowly.

As explained in section 10.2, the memory function is then truncated, both "to the right" and "below", resulting in a memory function that is zero at the end. Figure 12.16 compares the t.d. to f.d. calculations with two different memory function lengths, 30 s and 300 s. For comparison, the best results from figure 12.13 are included as well. As can easily be seen, this approach significantly improves the accuracy of the results. The effects are similar, but not as pronounced, in the other modes, caused by faster convergence to zero of the memory function.

Note also that with this truncation method, a 30 s memory function gives better results than a 80 s function using the previous method.

12.5 State-space model

The final step is the creation of state-space models from the memory functions.

Figures 12.18 through 12.21 show the results from the force comparisons. The

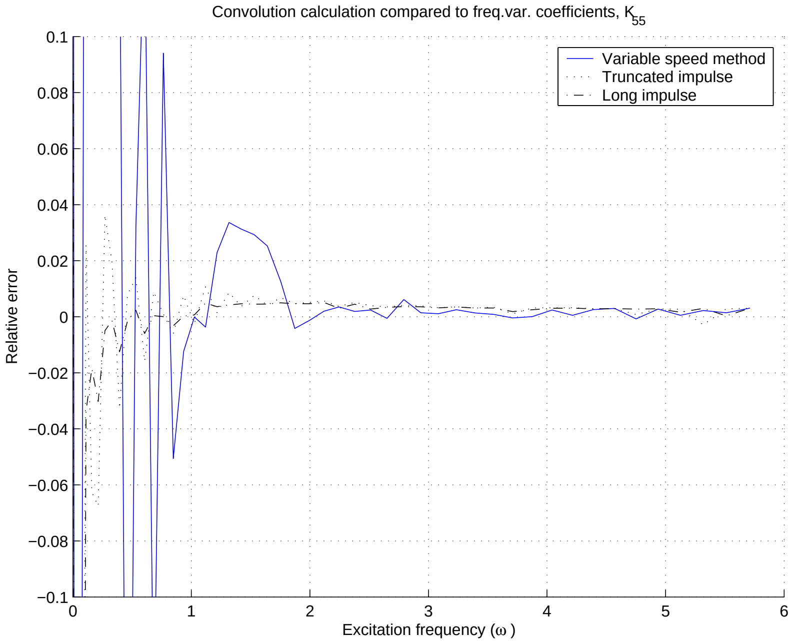

Figure 12.16: Force calculation, pitch motion, varying memory function length

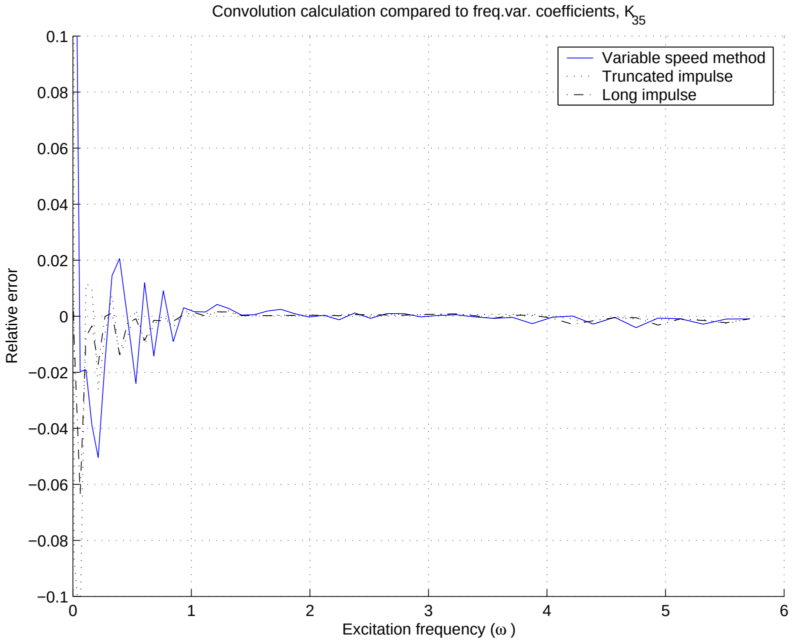

Figure 12.17: Force calculation, heave→pitch, varying memory function length

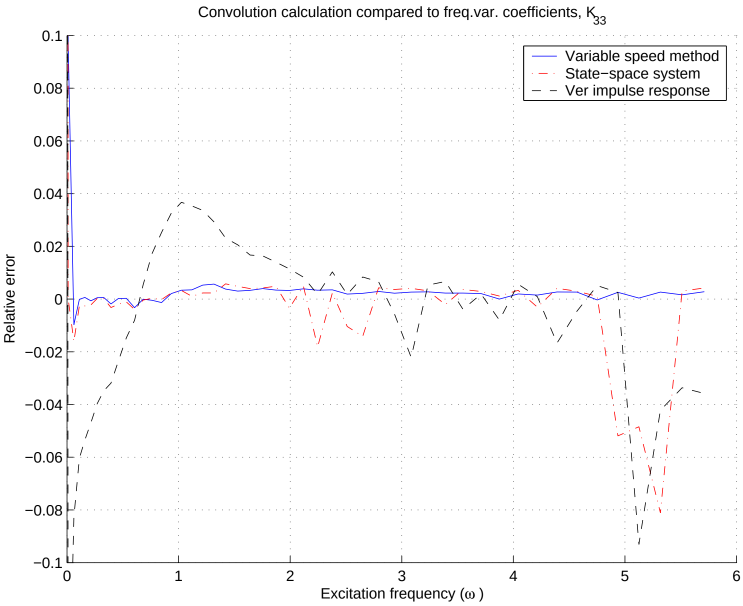

Figure 12.18: Force calculation, heave, state-space model

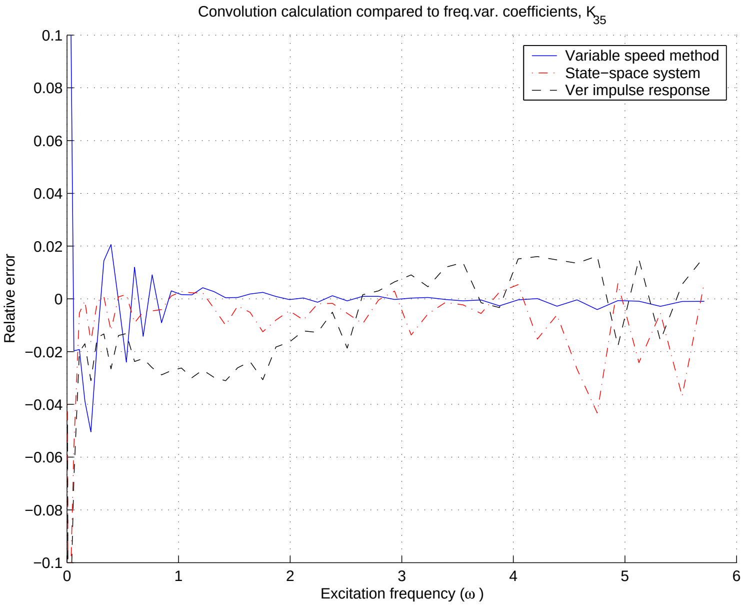

Figure 12.19: Force calculation, pitch→heave, state-space model

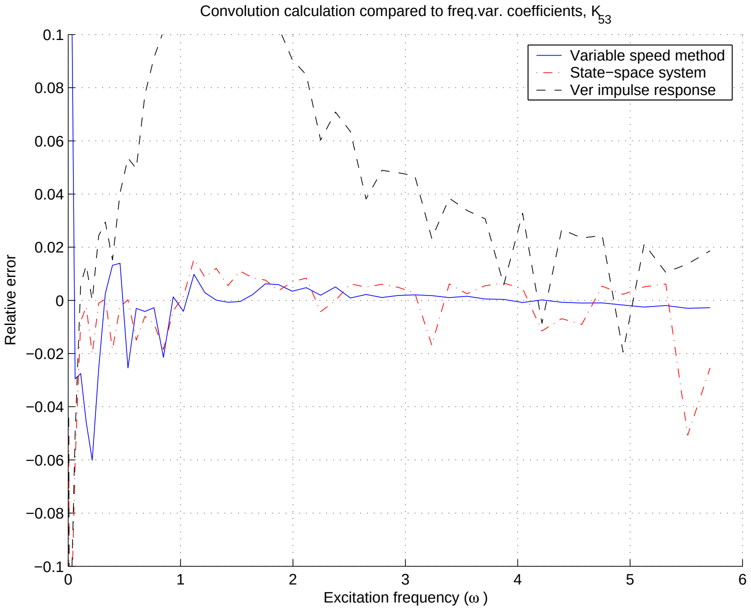

Figure 12.20: Force calculation, heave→pitch, state-space model

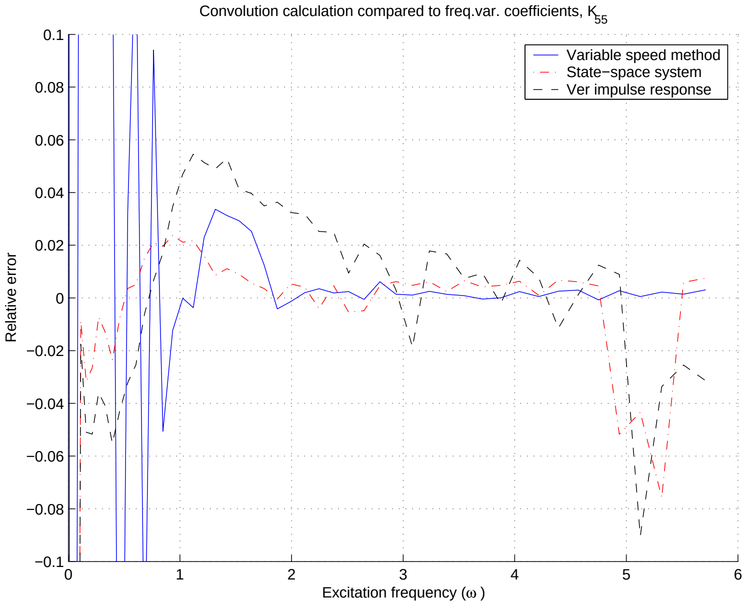

Figure 12.21: Force calculation, pitch, state-space model

dash-dotted line represents the results using the state-space model whereas the solid line is the original results, also included in the previous plots. After generation of the state-space models, impulse responses were extracted and these were also used for force calculations. These are seen to differ from the state-space force calculations. This may be because of the half-step shifting of the impulse response prior to computing the state-space model.

The disrepancies around $\omega = 5\,\mathrm{rad\,s^{-1}}$ may be because this is near the cutoff frequency. The somewhat systematic disrepancies at low frequencies are of more interest. They are expected to be related to the somewhat abrubt truncation of the memory function at $t = 30\,\mathrm{s}$.

Chapter 13

Simulation results

The vessel is simulated with three different forward thrusts, zero, $5 \cdot 10^5$ and $1 \cdot 10^6$. The sway and yaw setpoints are both zero. The waves are quartering and long-crested[1]. In all plots, the units along the primary axis are seconds.

Figure 13.1 shows the vessel motions without thrust in the surge direction. Attention is first drawn to the unrealistic roll motion. As mentioned earlier, no viscous effects are included, resulting in large errors in the prediction of this mode. Note also the constant displacement in surge, for which no simple explanation can be provided. There are no higher order wave forces included in the simulation, which could easily have explained the observed behaviour.

Next, forward thrust is added. The maximum thrust is in this case $5 \cdot 10^5$ N, increasing from zero starting at simulation time 100 s.

Figure 13.2 shows the surge motion, whereas figure 13.3 shows the various force contributions. The apparent wave frequency increases, caused by the vessel forward speed. This is the first time this phenomenom is included, needed or observed in this work. Thus, the concept of encounter frequency only appears as an effect in the simulation, and is not needed in the preceeding calculation. In particular, the encounter frequency is irrelevant to the calculation of radiation potential damping.

The vessels reaches significant speeds within a few minutes. This can again be argued to be unphysical, caused by not modelling drag. Towards the end of the simulation, the vessel becomes unstable. This is most likely because of the destabalizing effect of the Lie product term (including the Munk moment).

Doubling the forward thrust makes this phenomenon more visible. The resulting motion (figures 13.4 and 13.5) is clearly unphysical. There is a clear transition between two different types of motion, and this transition occurs at similar speeds to that seen in the previous simulation run.

[1] The simulation could just as easily have been with short-crested sea.

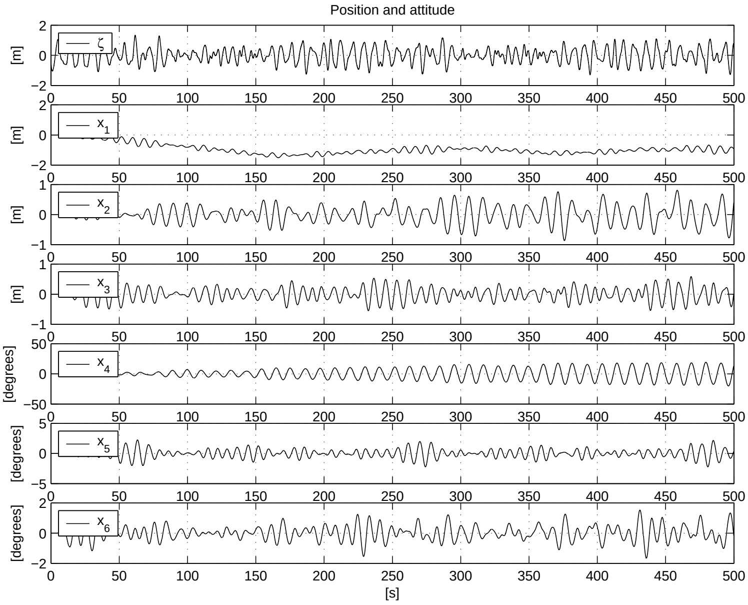

Figure 13.1: Vessel motion, zero forward thrust

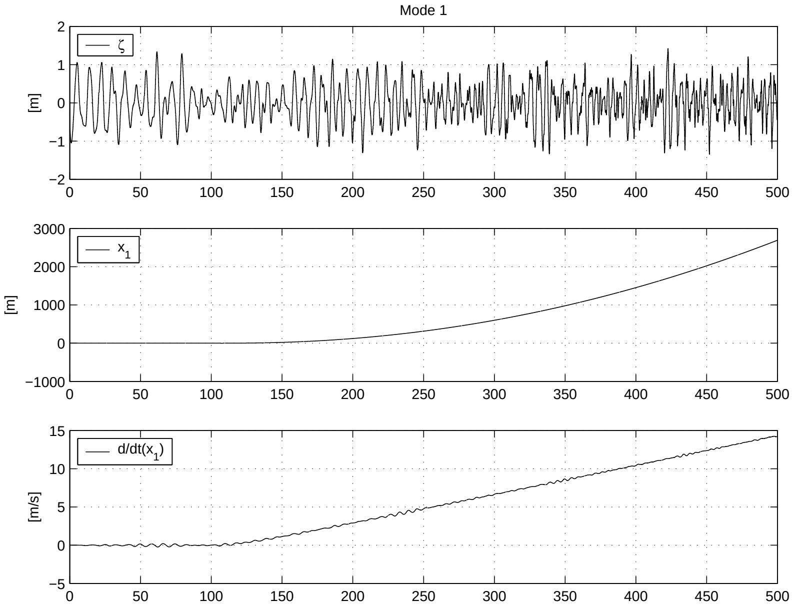

Figure 13.2: Surge motion, with forward thrust

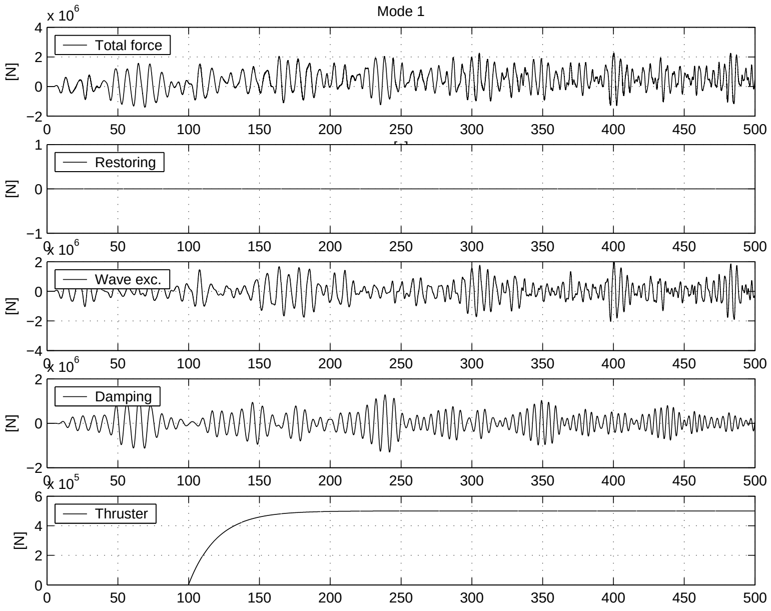

Figure 13.3: Forces in surge, with forward thrust

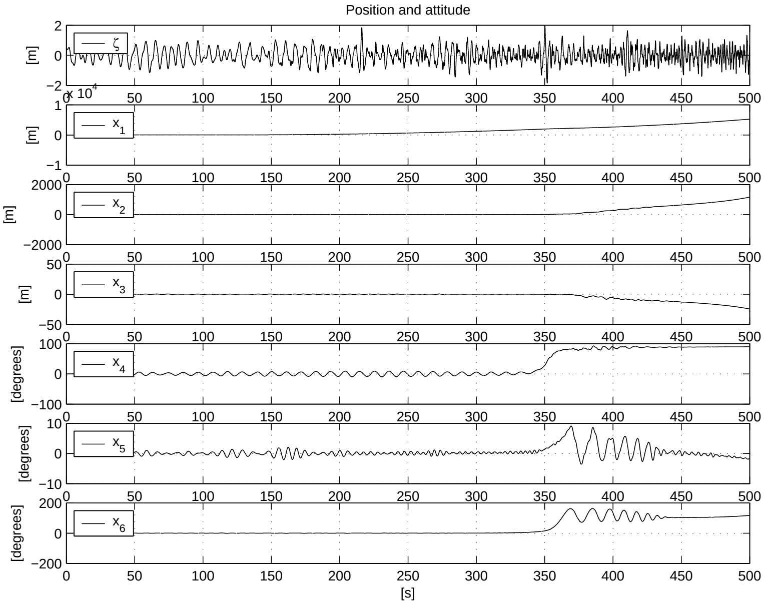

Figure 13.4: Vessel motion, unstable at high speed

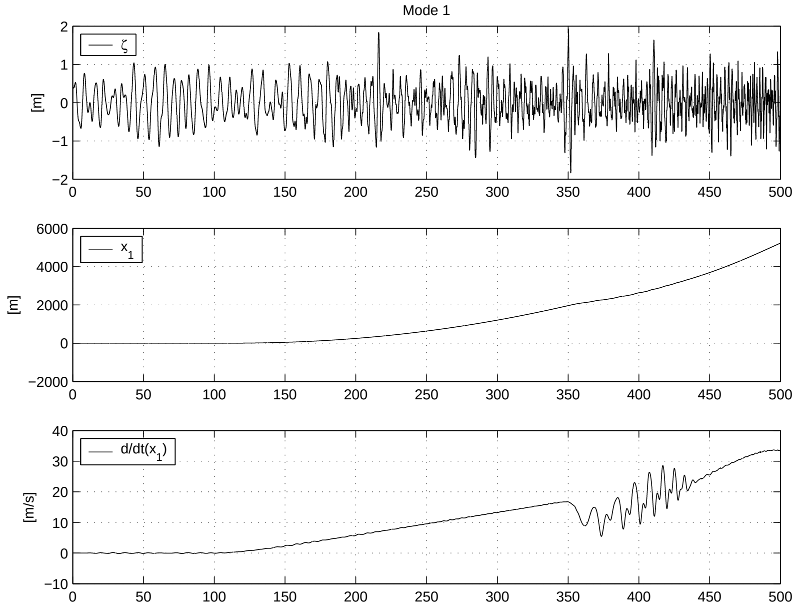

Figure 13.5: Surge motion, unstable

Chapter 14

Conclusions

A low order state-space model can be used to represent the forces from the free surface potential damping. The model order can be as low as between four and eight per mode coupling. These, and previous, results indicate the applicability of the method to data originating from both 3D and strip-method calculations. Here, further verifications between f.d. and t.d. simulations have been carried out. These verifications show the importance of accurate calculations in all steps. In the end, two percent accuracy was achieved within the frequency range of interest.

The state-space model approach has been extended to varying forward speeds, using strip-method like extensions of 3D data. This method is equally applicable to strip-method coefficients. The result is a model, with a relatively small set of parameters, that can represent the forces from the free surface potential damping for varying forward speeds.

However, it may not be this simple. This work appears to prove that the STF strip-method coefficients necessarily cannot satisfy the Kramers-Kronig relations. This relations should be satifisied for all causal, linear systems. The reason why the STF coefficients appears to violate causality is unclear, but some comments are made in section 5.4.

The model can be extended to six degrees-of-freedom, here done using analytical kinematics. This makes it possible to perform simulations with varying forward speeds in the north-east-down frame. The applicability to high turning-rate maneuvering is of course limited, among other reasons because of the assumption of flow symmetry. The model should be suited for other applications such as simulations of cable and pipe laying operations.

This model has been implemented in a simulator in Matlab/Simulink. The output appears reasonable although systematic verification of the produced output has not been carried out. Simulations are computionally efficient, easily runnable on most standard hardware.

The model has been demonstrated to posess several key properties for control — passive and contracting. The added mass matrix is symmetric and constant, even with forward speed. The convolution term is dissipative. The restoring forces can be considered as resulting from motion in a force potential.

14.1 Future work

The derivations of the hydrostatic coefficients should be verified for correctness. There is an interesting symmetry that perhaps can be developed further.

The simulator does not take the speed-dependence of the restoring coefficient into account. This simplification has no particular justification, and a complete simulator should include this.

Adding nonlinear effects is necessary to give fully accurate predictions of motion. In particular, viscous damping in roll has a significant impact on roll motion amplitude. For forward speeds, drag can also have a destabilizing effect that may be important.

The apparent non-causality of the STF coefficients presents a significant problem. Although the usability of the coefficients to predict motion is widely accepted, is would be of great interest to resolve this issue, or at least arrive at a plausible explanation. One likely candidate would be to consider the radiation problem exclusively in the body frame. The input velocities to the system would then be relative instantaneous velocity — including current. The radiation model itself would then be independent of forward speed.

14.2 Final comments

Towards the end, some final comments are in place.

The results presented here does not by any means represent a significant breakthrough or a radically different way to approach the problem of vessel motion in waves. It is much more an attempt to bring to attention some possibly forgotten facts. Similarily, there are prevailing misconceptions that obscure a much clearer picture. The reason appears more of a cultural than technical nature. The hydrodynamic community focuses its attention on motion prediction, and solves this problem with good accuracy. For the hydrodynamicisit, verification is done in the water, in towing tanks or full-scale experiments. Mathematical shortcuts are allowed, and used pragmatically. Thus, strange creatures such as a differential equation with frequency-varying coefficients are allowed to exist. The ultimate criterium is how it eventually performs when compared with experiments.

Researchers in the control community have a different approach to the mathematical tools. An often desired end result is a proof stating that the proposed control strategy works. Furthermore, various optimal control strategies depend on mathematical formulations with a sound basis. Looking at the models typically used presented by hydrodynamicists, this is not the case. The models include mathematical bastards such as frequency-varying coefficients.

However, and as this thesis has presented, retracing certain steps with a control approach in mind can lead to simpler models. Through applying a well-founded mathematical toolbox it is possible to present a model with the same input-output behaviour, but with a much more desirable structure. Other results, such as the isolation of forward speed as a separate parameter is nothing but a simple observation, also leading to simpler calculation along the way.

No marine control engineer should have to spend much worry on frequency-dependent coefficients and non-symmetric added mass matrices. There is no need to complicate matters further.

Bibliography

[1] P. A. Bailey, W. G. Price, and P. Temarel. A unified mathematichal model describing the maneouvering of a ship travelling in a seaway. In *RINA Transactions*, volume 140, pages 131–149, 1998.

[2] J. Balchen, N. A. Jenssen, and S. Sælid. Dynamic positioning of floating vessels based on kalman filtering and optimal control theory. In *Proc. of the IFAC/IPIP Symp. On Automation in Offshore Oil Field Operation*, pages 183–186, Bergen, Norway, 1976.

[3] V. Bertram. *Practical Ship Hydrodynamics.* Butterworth Heinemann, 2000.

[4] G. Birkhoff. *Hydrodynamics – A study in logic, fact and similitude.* Princeton University Press, Princeton, New Jersey, 1960.

[5] W. E. Cummins. The impulse response function and ship motions. *Schiffstechnik*, 1962.

[6] O. Egeland and J. Gravdahl. *Modeling and Simulation for Control.* Marine Cybernetics, Trondheim, Norway, 2002.

[7] J. Falnes. On non-causal impulse respons functions related to propagating water waves. *Applied Ocean Research*, 17:379–389, 1995.

[8] N. Fonseca and C. G. Soares. Non-linear wave induced responses of ships in irregular seas. In *Proc. of 17th International Conference on Offshore Mechanics and Arctic Engineering*, 1998.

[9] N. Fonseca and C. G. Soares. Time-domain analysis of large-amplitude vertical ship motions and wave loads. *Journal of Ship Research*, 42(2):138–153, June 1998.

[10] T. I. Fossen. *Guidance and Control of Ocean Vehicles.* John Wiley and Sons Ltd., 1994.

[11] T. I. Fossen. Matlab GNC Toolbox. *Marine Cybernetics*, 2001. URL: `http://www.marinecybernetics.com/`.

[12] T. I. Fossen. *Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles.* Marine Cybernetics, Trondheim, Norway, 2002.

[13] T. I. Fossen and A. Grøvlen. Nonlinear output feedback control of dynamically positioned ships using vectorial observer backstepping. *IEEE Transactions on Control Systems Technology*, 6(1):121–128, 1998.

[14] T. I. Fossen, K. P. Lindegaard, and R. Skjetne. Inertia shaping techniques for marine vessels using acceleration feedback. In *Proceedings of the IFAC World Congress*, Barcelona, 2002. Elsevier Science.

[15] W. Frank. Oscillation of cylinders in or below the free-surface of deep fluids. Technical Report Report 2375, Naval Ship Research and Development Center, Washington D.C., 1967.

[16] M. Greenhow. High- and low-frequency asymptotic consequences of the Kramers-Kronig relations. *Journal of Engineering Mathematics*, 20:293–306, 1986.

[17] M. J. Grimble, R. J. Patton, and D. A. Wise. The design of dynamic postitioning control systems using stochastic optimal control theory. *Optimal Control Applications and Methods*, 1:167–202, 1980.

[18] Å. Hjulstad. Tilstandsromformulering av frekvensavhengig medreven masse og potensialdemping for overflatefartyer. Semester project, Norwegian University of Science and Technology, 2003. In Norwegian.

[19] Å. Hjulstad, E. Kristiansen, and O. Egeland. State-space representation of frequency-dependent hydrodynamic coefficients. In *Proc. of IFAC Conference on Control Application in Marine Systems*, 2004.

[20] J. Jouffroy and J. Lottin. On the use of contraction theory for the design of nonlinear observers for ocean vehicles. In *American Control Conference (ACC'02)*, 2002.

[21] K. E. Kaasen and K. Mo. Efficient time-domain model for frequency-dependent added mass and damping. In *Proceedings of 23rd International Conference on Offshore Mechanics and Arctic Engineering.* ASME, June 2004.

[22] B.-K. Kim and M. M. Bernitsas. Effect of memory on the stability of spread mooring systems. *Journal of Ship Research*, 43(3):157–169, 1999.

[23] B. K. King. Time-domain analysis of wave exciting forces on ships and bodies. Technical Report Report No. 306, Department of Naval Architecture and Marine Engineering, The University of Michigan, 1987.

[24] E. Kristiansen and O. Egeland. Frequency-dependent added mass in models for controller design for wave motion damping. In *6th IFAC Conference on Manoeuvering and Control of Marine Craft*, pages 90–95, 2003.

[25] W. Lohmiller and J.-J. E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

[26] L. R. Marshall, A. H. Nayfeh, and D. T. Mook. Forward speed effects in the equations of ship motion. *Journal of Sound and Vibration*, 85(3):303–313, 1982.

[27] J. N. Newman. *Marine Hydrodynamics.* MIT Press, Cambridge, Massachusetts, 1977.

[28] T. F. Ogilvie, editor. *Recent Progress Toward the Understanding and Prediction of Ship Motions*, 1964.

[29] O. F. Rognebakke and O. M. Faltinsen. Coupling of sloshing and ship motions. *Journal of Ship Research*, 47(3):208–221, 2003.

[30] S. I. Sagatun and T. I. Fossen. Lagrangian formulation of underwater vehicles' dynamics. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 1029–1034, 1991.

[31] N. Salvesen, E. O. Tuck, and O. Faltinsen. Ship motions and sea loads. In *The Society of Naval Architects and Marine Engineers*, 1970.

[32] M. Schmiechen. Estimation of spectra of truncated transient functions. *Ship Technology Research*, 46:111–127, 1999.

[33] R. Timman and J. Newman. The coupled damping coefficients of a symmetric ship. *Journal of Ship Research*, 5(4), 1962.

[34] G. Torsetnes. Nonlinear control and observer design for dynamic positioning using contraction theory. Master's thesis, NTNU, 2004.

[35] M. S. Triantafyllou, M. Bodson, and M. Athans. Real time estimation of ship motions using kalman filtering techniques. *IEEE Journal of Ocean Engineering*, 8(1):9–20, 1983.

[36] J. V. Wehausen. Causality and the radiation condition. *J. Engrg. Math.*, 26(1):153–158, 1992.

[37] E. W. Weisstein. Mathworld. http://mathworld.wolfram.com/.

[38] WAMIT user manual. Technical report, WAMIT Inc, 2002.

[39] S. Årdal. Automatic landing of helicopter on wave-excited ship deck. Master's thesis, NTNU, 2002.

Appendix A

Additional formulae

This appendix includes formulas and tables not included in full in the text.

A.1 Salvesen strip theory

The complex force coefficients from the radiation potential are in the strip theory of Salvesen, Tuck and Faltinsen [31] given by

$$T = T^{\mathrm{c}} + \frac{U}{\mathrm{i}\omega}\begin{pmatrix} 0 & 0 & 0 & 0 & T^{\mathrm{c}}_{13} & -T^{\mathrm{c}}_{12} \\ 0 & 0 & 0 & 0 & T^{\mathrm{c}}_{23} & -T^{\mathrm{c}}_{22} \\ 0 & 0 & 0 & 0 & T^{\mathrm{c}}_{33} & -T^{\mathrm{c}}_{32} \\ 0 & 0 & 0 & 0 & T^{\mathrm{c}}_{43} & -T^{\mathrm{c}}_{42} \\ -T^{\mathrm{c}}_{31} & -T^{\mathrm{c}}_{32} & -T^{\mathrm{c}}_{33} & -T^{\mathrm{c}}_{34} & 0 & 0 \\ T^{\mathrm{c}}_{21} & T^{\mathrm{c}}_{22} & T^{\mathrm{c}}_{23} & T^{\mathrm{c}}_{24} & 0 & 0 \end{pmatrix} \tag{A.1}$$

$$+ \frac{U^2}{\omega^2}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & T^{\mathrm{c}}_{33} & 0 \\ 0 & 0 & 0 & 0 & 0 & T^{\mathrm{c}}_{22} \end{pmatrix} + \text{aft section terms} \tag{A.2}$$

where T^{c} is the zero forward speed complex force coefficient matrix. Assuming $T^{\mathrm{c}}_{jk} = T^{\mathrm{c}}_{kj}$ the first matrix is skew-symmetrical. Also note that for high frequencies the complex force coefficients equal their zero speed value.

Appendix B

Errata, previous work

This work was done in the continuation of project work by the author the previous autumn. As often is inevitable, certain errors where present in that work. This appendix lists some important ones.

Hydrostatic restoring forces The restoring matrix from WAMIT contained an element (in row/column 4,6 and 5,6) that at the time was difficult to explain. These elements should be zero for a ship with body-axis in the same plane as the water plane in equilibrium, whereas in the output data from WAMIT they were non-zero. The explanation is very simple. Centre of gravity is input to WAMIT, whereas centre of boyancy is calculated from the geometry data. As centre of gravity only was given with three decimals, there was a round-off error in the input data to WAMIT, and as a result, the WAMIT data contained a non-zero element at position 4,6.

Appendix C

Matlab code

Some code fragments are included here to illustrate the most important aspects of the developed program. The complete source code and Simulink model is included on the CD, or the authors web site.

C.1 Main script

The main script that produces all illustrations in this thesis is found in the following listing. The helper scripts storing the intermediate calculation results follow thereafter.

```
do_all = 0;

do_main_block = 1+ do_all;
do_compare_impulsetimeresolution = 0 + do_all;
do_compare_interpmethod = 0 + do_all;
do_explore_restoring_oscillations = 0 + do_all;
do_compare_fineimpulsetimeresolution = 0;
do_make_systems = 0 + do_all;
do_restoring_from_ss = 0 + do_all;
do_compare_restoring = 1 + do_all;
do_make_systems_subtracted_impulse = 1 + do_all;

% clear;
close all;

% process_data
% Wrapper script for restructured processing
%

coarseopt = struct(...
    'Modes',[3 3; 3 5; 5 3; 5 5], ...
    'ImpulseTimesResolution',0.1, ...
    'ImpulseIntegralResolution',0.1, ...
    'CoeffInterpolationResolution',0.1, ...
    'Speeds',[10.3642], ...
    'ForceTestFrequencies',0.5:0.5:1.5, ...
    'FineGrainedInterpolationInterval',0.01);

fineopt = struct(...
    'Modes',[3 3; 3 5; 5 3; 5 5], ...
    'ImpulseTimesMax',120, ...
    'ImpulseTimesResolution',0.05, ...
    'ForceTestFrequencies',0.1:0.05:4, ...
    'ReduceModel', 'no', ...
    'Speeds',[10.3642]);

%opt = coarseopt;
opt = fineopt;

% Do force calculation only at frequencies from the original data.
[not_used, not_used, data_w] = read_fonseca_coeff_set('inndata\fonseca\set2\addda25.dat', [], opt);
opt.ForceTestFrequencies = data_w(1:1:end);

%clear node;
clear v;
symbol = 'y-';
modes = [3 3; 3 5; 5 3; 5 5];
printopts = {'-depsc2','-dpng'};
printprefix = '../figurer/';

warning off MATLAB:MKDIR:DirectoryExists;

if do_main_block,
    nodecount = 0;
    parent = 0;
    clear node;
    % List of points where we have ready impulse responses that can be tested
    % in various ways
    final_list = [];

    % Read input data
    name = 'Original_dataset';
    symbol = 'b--';
    v.T = read_fonseca_T('inndata\fonseca\set2\addda00.dat',[],opt);
    v.Korig = read_fonseca_impulse('inndata\fonseca\set1\shfour.dat');
    [v.coeffs, not_used, data_w] = read_fonseca_coeff_set('inndata\fonseca\set2\addda25.dat', [], opt);
    savepoint;
    % Tag the startingpoint so we can return to it later
    start = last;

    % Generate impulse response the "good" way
    next = start; loadpoint;
    name = 'Variable_speed_method';
    symbol = 'b-';
    v.Kvar = speedvarying_impulse(v.T,opt);
    v.K = v.Kvar.fixed(1);
    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    savepoint; final_list = [final_list last]; good = last;

    % Generate impulse response directly from coefficients
    next = start; loadpoint;
    name = 'Fixed_speed_method';
    symbol = 'k-';
    v.K = impulse_response_from_coeffs(v.coeffs,opt);
    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    savepoint; final_list = [final_list last];

    % Generate impulse response directly from coefficients, don't subtract binf
    next = start; loadpoint;
    name = 'Method_of_original_data';
    symbol = 'k-.';
    opt.ImpulseSubtractBinf = 'no';
    v.K = impulse_response_from_coeffs(v.coeffs,opt);
    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    savepoint; final_list = [final_list last];
    % Use original data
    next = start; loadpoint;
    name = 'Original_memory_function';
    symbol = 'k--';
    v.K = v.Korig;
    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    savepoint; final_list = [final_list last];

    ver_list = [];
    for i = 1:length(final_list),
        next = final_list(i); loadpoint;
        name = [parentname];
        v.testresult = force_test(v.K,v.coeffs,v.Crad,opt);
        v.coeffs = coeff_from_impulse(v.K, v.coeffs.wi, v.coeffs.Ainf, v.K.Binf);
        savepoint; ver_list = [ver_list last];
    end;

    opt = [];
    printdir = [printprefix 'default/']; mkdir(printdir);
    make_plot(node(final_list),modes,@draw_impulseresponse,...
        printopts,[printdir 'impulse_%i_%i'],opt);
    make_plot(node([start 6 9]),modes,@draw_coefficient,...
        printopts,[printdir 'rad_koeff_%i_%i'],opt);
    make_plot(node(ver_list),modes,@draw_force_simulation_results,...
        printopts,[printdir 'forcecompare_%i_%i'],opt);
    make_plot(node(final_list),modes,@draw_restoring_coefficient,...
        printopts,[printdir 'restoring_%i_%i'],opt);

end

if do_compare_impulsetimeresolution,
    next = start; loadpoint;
    name = 'Coarse_impulse_resolution'; symbol = 'g-';
    opt.ImpulseTimesResolution=0.1;
    v.Kvar = speedvarying_impulse(v.T,opt);
    v.K = v.Kvar.fixed(1);
    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    v.testresult = force_test(v.K,v.coeffs,v.Crad,opt);
    savepoint; coarse= last;

    opt = [];
    printdir = [printprefix 'impulse_resolution_coarse/']; mkdir(printdir);
    make_plot(node([6 last]),modes,@draw_impulseresponse,...
        printopts,[printdir 'impulse_%i_%i'],opt);
    make_plot(node([6 last]),modes,@draw_force_simulation_results,...
        printopts,[printdir 'forcecompare_%i_%i'],opt);
    make_plot(node([6 last]),modes,@draw_restoring_coefficient,...
        printopts,[printdir 'restoring_%i_%i'],opt);
end

if do_compare_interpmethod,
    next = start; loadpoint;
    name = 'Linear_interpolation'; symbol = 'g-';
    opt.CoeffInterpolationMode='linear';
    v.T = read_fonseca_T('inndata\fonseca\set2\addda00.dat',[],opt);
    v.Kvar = speedvarying_impulse(v.T,opt);
    v.K = v.Kvar.fixed(1);
    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    v.testresult = force_test(v.K,v.coeffs,v.Crad,opt);
    savepoint;

    opt = [];
    printdir = [printprefix 'interpmethod/']; mkdir(printdir);
    make_plot(node([6 last]),modes,@draw_force_simulation_results,...
        printopts,[printdir 'forcecompare_%i_%i'],opt);
    make_plot(node([6 last]),modes,@draw_restoring_coefficient,...
        printopts,[printdir 'restoring_%i_%i'],opt);
end

if do_explore_restoring_oscillations,
    next = start; loadpoint;
    name = 'Spline_interpolation_everywhere'; symbol = 'g-';
    opt.CoeffInterpolationMode='spline';
    v.T = read_fonseca_T('inndata\fonseca\set2\addda00.dat',[],opt);
    v.coeffs = read_fonseca_coeff_set('inndata\fonseca\set2\addda25.dat', [], opt);
    v.Kvar = speedvarying_impulse(v.T,opt);
    v.K = v.Kvar.fixed(1);
    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    v.testresult = force_test(v.K,v.coeffs,v.Crad,opt);
    savepoint;

    opt = [];
    printdir = [printprefix 'restoring_oscillation/']; mkdir(printdir);
    make_plot(node([6 last]),modes,@draw_force_simulation_results,...
        printopts,[printdir 'forcecompare_%i_%i'],opt);
    make_plot(node([6 last]),modes,@draw_restoring_coefficient,...
        printopts,[printdir 'restoring_%i_%i'],opt);
end


if do_compare_fineimpulsetimeresolution,
    next = start; loadpoint;
    name = 'Fine_impulse_resolution'; symbol = 'g-.';
    opt.ImpulseTimesResolution=0.01;
    v.Kvar = speedvarying_impulse(v.T,opt);
    v.K = v.Kvar.fixed(1);
    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    v.testresult = force_test(v.K,v.coeffs,v.Crad,opt);
    savepoint;

    opt = [];
    printdir = [printprefix 'impulse_resolution_fine/']; mkdir(printdir);
    make_plot(node([6 last coarse]),modes,@draw_impulseresponse,...
        printopts,[printdir 'impulse_%i_%i'],opt);
    make_plot(node([6 last coarse]),modes,@draw_force_simulation_results,...
        printopts,[printdir 'forcecompare_%i_%i'],opt);
    make_plot(node([6 last coarse]),modes,@draw_restoring_coefficient,...
        printopts,[printdir 'restoring_%i_%i'],opt);
end

if do_make_systems,
    next = good; loadpoint;
    name = 'State-space_system'; symbol = 'r-';
    opt.ForceImpulseToZero = 'no';
    opt.ImpulseTimesMax = 60;
    opt.ImpulseTimesResolution = 0.3;
    opt.TimeDecimation = 1;

    v.Kvar = speedvarying_impulse(v.T,opt);
% [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    [v.ss_model, v.system] = impulse_to_reduced_statespace(v.Kvar, v.T, opt);
    v.testresult = force_test_ss(v.ss_model.fixed,0:0.05:120, v.coeffs,v.Crad,opt);
    opt.ImpulseTimesMax = 300;
    opt.ImpulseTimesResolution = 0.1;
    v.K = impulse_from_statespace(v.ss_model.fixed, opt);
    savepoint; statespace = last;

    opt = [];
    opt.ImpulsePlotMaxTime = 300;
    printdir = [printprefix 'state_space/']; mkdir(printdir);
    make_plot(node([6 last]),modes,@draw_force_simulation_results,...
        printopts,[printdir 'forcecompare_%i_%i'],opt);
    make_plot(node([6 last]),modes,@draw_impulseresponse,...
        printopts,[printdir 'impulse_%i_%i'],opt);
end

if do_restoring_from_ss,
    % I am not sure if this code works properly
    next = statespace; loadpoint;
    name = 'Restoring_from_state-space'; symbol = 'r-.';
    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    v.testresult = force_test_ss(v.ss_model.fixed,0:0.05:120, v.coeffs,v.Crad,opt);

    savepoint;

    opt = [];
    printdir = [printprefix 'state_space_newrestoring/']; mkdir(printdir);
    make_plot(node([6 statespace last]),modes,@draw_force_simulation_results,...
        printopts,[printdir 'forcecompare_%i_%i'],opt);
    make_plot(node([6 last]),modes,@draw_restoring_coefficient,...
        printopts,[printdir 'restoring_%i_%i'],opt);
end

if do_compare_restoring,
    next = start; loadpoint;
    name = 'Truncated_impulse';
    opt.ImpulseTimesMax = 30;
    opt.ForceImpulseToZero = 'no';
    symbol = 'k:';
    v.Kvar = speedvarying_impulse(v.T,opt);
    v.K = v.Kvar.fixed(1);
    v.K.subtracted = v.K.Kijt(:,:,end);
    v.K.Kijt = v.K.Kijt - repmat(v.K.Kijt(:,:,end),[1 1 length(v.K.tpts)]);
    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    v.testresult = force_test(v.K,v.coeffs,v.Crad,opt);
    savepoint; trunc1 = last;

    next = start; loadpoint;
    name = 'Long_impulse';
    opt.ImpulseTimesMax = 300;
    opt.ForceImpulseToZero = 'no';
    symbol = 'k-.';
    v.Kvar = speedvarying_impulse(v.T,opt);
    v.K = v.Kvar.fixed(1);
    v.K.subtracted = v.K.Kijt(:,:,end);
    v.K.Kijt = v.K.Kijt - repmat(v.K.Kijt(:,:,end),[1 1 length(v.K.tpts)]);
    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    v.testresult = force_test(v.K,v.coeffs,v.Crad,opt);
    savepoint; trunc2 = last;

    opt = [];
    printdir = [printprefix 'compare_restoring/']; mkdir(printdir);
    make_plot(node([6 trunc1 trunc2]),modes,@draw_restoring_coefficient,...
        printopts,[printdir 'restoring_%i_%i'],opt);
    make_plot(node([6 trunc1 trunc2]),modes,@draw_impulseresponse,...
        printopts,[printdir 'impulse_%i_%i'],opt);
    make_plot(node([6 trunc1 trunc2]),modes,@draw_force_simulation_results,...
        printopts,[printdir 'forcecompare_%i_%i'],opt);
end

if do_make_systems_subtracted_impulse,
    next = start; loadpoint;
    name = 'State-space_system'; symbol = 'r-.';
    opt.ForceImpulseToZero = 'no';
    opt.TimeDecimation = 1;
    opt.ReduceModel = 'yes';
    opt.ImpulseTimesMax = 80;
    opt.ImpulseTimesResolution = 0.2;
    opt.ShiftHalfStep = 'yes';

    v.Kvar = speedvarying_impulse(v.T,opt);
    v.K = v.Kvar.fixed(1);
    v.K.Kijt = v.K.Kijt - repmat(v.K.Kijt(:,:,end),[1 1 length(v.K.tpts)]);
    v.Kvar.constant = v.Kvar.constant - repmat(v.Kvar.constant(:,:,end),[1 1 length(v.K.tpts)]);
    v.Kvar.linear = v.Kvar.linear - repmat(v.Kvar.linear(:,:,end),[1 1 length(v.K.tpts)]);
    v.Kvar.quadratic = v.Kvar.quadratic - repmat(v.Kvar.quadratic(:,:,end),[1 1 length(v.K.tpts)]);

    [v.ss_model, v.system] = impulse_to_reduced_statespace(v.Kvar, v.T, opt);
% opt.ImpulseTimesMax = 300;
% v.regeneratedK = impulse_from_statespace(v.ss_model.fixed, opt);
% [v.Crad,v.Cradijw] = calc_radiation_restoring(v.regeneratedK,v.coeffs,opt);

    [v.Crad,v.Cradijw] = calc_radiation_restoring(v.K,v.coeffs,opt);
    v.testresult = force_test_ss(v.ss_model.fixed(1),0:opt.ImpulseTimesResolution:120, v.coeffs,v.Crad,opt);
    savepoint; statespace2 = last;

    next = statespace2; loadpoint;
    name = 'Ver_impulse_response'; symbol = 'k--';
    v.K = impulse_from_statespace(v.ss_model.fixed, opt);
    v.testresult = force_test(v.K, v.coeffs, v.Crad, opt);
    savepoint; verimpulse = last;

    opt = [];
    opt.ImpulsePlotMaxTime = 300;
    printdir = [printprefix 'state_space2/']; mkdir(printdir);
    make_plot(node([6 statespace2 verimpulse ]),modes,@draw_force_simulation_results,...
        printopts,[printdir 'forcecompare_%i_%i'],opt);
    make_plot(node([6 statespace2 verimpulse ]),modes,@draw_impulseresponse,...
        printopts,[printdir 'impulse_%i_%i'],opt);
    make_plot(node([6 statespace2 verimpulse ]),modes,@draw_restoring_coefficient,...
        printopts,[printdir 'restoring_%i_%i'],opt);
end


if (0),
    next = statespace2; loadpoint;
    name = 'Continuous_model'; symbol = 'k-.';
    for m = 1:size(modes,1),
        i = modes(m,1);
        j = modes(m,2);
        v.ss_model.fixed.single(i,j).ss = d2c(v.ss_model.fixed.single(i,j).ss);
    end
    v.ss_model.fixed.ss = d2c(v.ss_model.fixed.ss);
    v.testresult = force_test_ss(v.ss_model.fixed(1),0:opt.ImpulseTimesResolution:120, v.coeffs,v.Crad,opt);
    savepoint; continous = last;
end
```

```
v = node(next).data;
opt = node(next).opt;
symbol = node(next).symbol;
parent = next;
parentname = node(next).name;
```

```
% Make checkpoint
nodecount = nodecount + 1;

node(nodecount).data = v;
node(nodecount).name = name;
node(nodecount).parent = parent;
node(nodecount).opt = opt;
node(nodecount).symbol = symbol;
disp(['Finished␣with␣"' name '"']);
name = '';
clear v opt;

last = nodecount;
```

C.2 Impulse response calculation

The set of three impulse responses (constant, linear and quadratic in speed) is found in the following listing.

```
function Kvar = speedvarying_impulse(T, options)
% Calculates speedvarying impulses

defaultopt = struct ( ...
          'ImpulseIntegralResolution',0.001, ...
          'ImpulseIntegralMax',17, ...
          'Modes',[3 3; 3 5; 5 3; 5 5], ...
          'Speeds',[0 5.1821 10.3642], ...
          'ForceImpulseToZero','yes', ...
          'ImpulseTimesResolution',0.05, ...
          'ImpulseTimesMax',30);

% If just 'defaults' passed in, return the default options in the first
% return variable.
if nargin==1 & nargout <= 1 & isequal(T,'defaults')
   Kc = defaultopt;
   return
end
if nargin < 1, error('More␣arguments␣required'); end
if nargin < 2, options=[]; end
if (isempty(options)),
    options = defaultopt;
end

t = 0:optionget(options,'ImpulseTimesResolution',defaultopt):optionget(options,'ImpulseTimesMax',defaultopt);

K0ijt = zeros(6,6,length(t));
K1ijt = zeros(6,6,length(t));
K2ijt = zeros(6,6,length(t));

modi = optionget(options,'Modes',defaultopt);
for m = 1:size(modi,1),
    i = modi(m,1);
    j = modi(m,2);
    disp(['K␣' int2str(i) int2str(j)]);
    K0ijt(i,j,:) = impulse_response_ij(T, i, j, t, options);
end

dw = optionget(options,'ImpulseIntegralResolution',defaultopt);
wi = 0:dw:optionget(options,'ImpulseIntegralMax',defaultopt);

ad = T_da_for_w(T, 3, 3, wi)';
bd = T_b_for_w(T, 3, 3, wi)';
% Calculate speed-dependent parts
for s=1:length(t),
    K1ijt(3,5,s) = 2/pi * dw * trapz ( ad .* cos(wi'.*t(s)) );
```

```
end
K1ijt(5,3,:) = -K1ijt(3,5,:);

K_org = zeros(1,length(t));
for s=1:length(t),
    K_org(s) = 2/pi*dw*trapz ( bd(2:end) ./ wi(2:end)'.^2 .* cos(wi(2:end)'.*t(s)) );
end

K2ijt(5,5,:) = K_org;

if (strcmp(optionget(options,'ForceImpulseToZero',defaultopt),'yes')),
    % Force K to zero in the last 5% of the data interval
    forcelength = fix(length(t)/20);
    forcevector = ones(1,size(K0ijt,3));
    forcevector(end-forcelength:end) = 1 - (0:1/forcelength:1);
    for m = 1:size(modi,1),
        i = modi(m,1);
        j = modi(m,2);
        K0ijt(i,j,:) = squeeze(K0ijt(i,j,:)).*forcevector';
        K1ijt(i,j,:) = squeeze(K1ijt(i,j,:)).*forcevector';
        K2ijt(i,j,:) = squeeze(K2ijt(i,j,:)).*forcevector';
    end
end

Kvar.constant = K0ijt;
Kvar.linear = K1ijt;
Kvar.quadratic = K2ijt;
Kvar.tpts = t;


speeds = optionget(options,'Speeds',defaultopt);
for sp = 1:length(speeds),
    speed = speeds(sp);
    Kvar.fixed(sp).speed = speed;
    Kvar.fixed(sp).Kijt = zeros(size(K0ijt));
    for m = 1:size(modi,1),
        i = modi(m,1);
        j = modi(m,2);
        Kvar.fixed(sp).Kijt(i,j,:) = K0ijt(i,j,:) + speed .* K1ijt(i,j,:) + speed^2 .* K2ijt(i,j,:) ;
    end
    Kvar.fixed(sp).tpts = t;

    Binf5 = zeros(6,6);
    Binf5(3,5) = Binf5(3,5) + speed*T.mode(3,3).ainf;
    Binf5(5,3) = Binf5(5,3) - speed*T.mode(3,3).ainf;
    Binf5(5,5) = Binf5(5,5);
    Kvar.fixed(sp).Binf = Binf5;
end
```

C.3 Model reduction

The model reduction is done using the code in the following listing.

```
function [dempingss, system] = impulse_to_reduced_statespace(Kvar, T , options)

defaultopt = struct ( ...
    'ModelOrders', 4:10, ...
    'UseOrder', 6, ...
    'Modes',[3 3; 3 5; 5 3; 5 5], ...
    'ReduceModel', 'yes', ...
    'ShiftHalfStep', 'no', ...
    'InitialErrorBound', 0.01, ...
    'AugmentWithBinf', 'no', ...
    'TimeDecimation', 1);

if nargin < 3,
    options = [];
end
if (isempty(options)),
    options = defaultopt;
end


K0ijt = Kvar.constant;
K1ijt = Kvar.linear;
K2ijt = Kvar.quadratic;
t = Kvar.tpts;
%dT = t(2)-t(1);

modelorders = optionget(options,'ModelOrders',defaultopt);
useorder = optionget(options,'UseOrder',defaultopt);
decimation = optionget(options,'TimeDecimation',defaultopt);

timepoints = 1:decimation:length(t);
dT = t(timepoints(2))-t(timepoints(1));

modi = optionget(options,'Modes',defaultopt);
dempingss.constant.ss = ss(zeros(6));
dempingss.linear.ss = ss(zeros(6));
dempingss.quadratic.ss = ss(zeros(6));
for m = 1:size(modi,1),
    i = modi(m,1);
    j = modi(m,2);
    disp(sprintf('Creating␣statespace␣for␣mode␣%i%i',i,j));

    if (strcmp(optionget(options,'ShiftHalfStep',defaultopt),'yes')),
        K0ijt(i,j,:) = (K0ijt(i,j,:) + cat(3,K0ijt(i,j,2:end),0))/2;
        K1ijt(i,j,:) = (K1ijt(i,j,:) + cat(3,K1ijt(i,j,2:end),0))/2;
        K2ijt(i,j,:) = (K2ijt(i,j,:) + cat(3,K2ijt(i,j,2:end),0))/2;
    end

    scale0 = K0ijt(i,j,1);
    if (K1ijt(i,j,1)~=0),
        scale1 = K1ijt(i,j,1);
    else
        scale1 = 1;
    end
    if (K2ijt(i,j,1)~=0),
        scale2 = K2ijt(i,j,1);
    else
        scale2 = 1;
    end

    bound = optionget(options,'InitialErrorBound', defaultopt);
    disp('Constant␣component');
    [system(i,j).Ap,system(i,j).Bp,system(i,j).Cp,system(i,j).Dp,TOTBND,system(i,j).svh] = ...
        IMP2SS(squeeze(K0ijt(i,j,timepoints))'./scale0,-1,1,1,bound);
    disp('Linear␣component');
    [system(i,j).Ap1,system(i,j).Bp1,system(i,j).Cp1,system(i,j).Dp1,TOTBND,system(i,j).svh1] = ...
        IMP2SS(squeeze(K1ijt(i,j,timepoints))'./scale1,-1,1,1,bound);
    disp('Quadratic␣component');
    [system(i,j).Ap2,system(i,j).Bp2,system(i,j).Cp2,system(i,j).Dp2,TOTBND,system(i,j).svh2] = ...
        IMP2SS(squeeze(K2ijt(i,j,timepoints))'./scale2,-1,1,1,bound);
    % Scale back up from what we did with the impulse response
    system(i,j).dss = ss(system(i,j).Ap,system(i,j).Bp,system(i,j).Cp,system(i,j).Dp,dT) *scale0;
    system(i,j).dss1 = ss(system(i,j).Ap1,system(i,j).Bp1,system(i,j).Cp1,system(i,j).Dp1,dT)*scale1;
    system(i,j).dss2 = ss(system(i,j).Ap2,system(i,j).Bp2,system(i,j).Cp2,system(i,j).Dp2,dT)*scale2;
    % Change to continous time
    disp('Change␣to␣continuous␣time');
    system(i,j).ss = d2c(system(i,j).dss,'tustin') ∗dT;
    system(i,j).ss1 = d2c(system(i,j).dss1,'tustin') ∗dT;
    system(i,j).ss2 = d2c(system(i,j).dss2,'tustin') ∗dT;

    if (strcmp(optionget(options,'ReduceModel',defaultopt),'yes')),
        disp('Reducing␣model␣order');
        system(i,j).redlitt.ss = schmr(system(i,j).ss,1,50);
        system(i,j).redlitt.ss1 = schmr(system(i,j).ss1,1,50);
        system(i,j).redlitt.ss2 = schmr(system(i,j).ss2,1,50);
        for order = modelorders,
            %[system(i,j).red(order).ss, bnd(i,j,order)] = balmr(system(i,j).redlitt.ss,1,order);
            [system(i,j).red(order).ss] = schmr(system(i,j).redlitt.ss,1,order);
            [system(i,j).red(order).ss1] = schmr(system(i,j).redlitt.ss1,1,order);
            [system(i,j).red(order).ss2] = schmr(system(i,j).redlitt.ss2,1,order);
        end
        dempingss.constant.ss(i,j) = system(i,j).red(useorder).ss;
        dempingss.linear.ss(i,j) = system(i,j).red(useorder).ss1;
        dempingss.quadratic.ss(i,j) = system(i,j).red(useorder).ss2;
    else
        dempingss.constant.ss(i,j) = system(i,j).ss;
        dempingss.linear.ss(i,j) = system(i,j).ss1;
        dempingss.quadratic.ss(i,j) = system(i,j).ss2;
    end
end


speeds = optionget(options,'Speeds',defaultopt);
for sp = 1:length(speeds),
    speed = speeds(sp);
    dempingss.fixed(sp).speed = speed;
    dempingss.fixed(sp).ss = dempingss.constant.ss + speed ∗ dempingss.linear.ss + speedˆ2 ∗ dempingss.quadratic.ss;

    % Assemble SISO−models for fixed speed, to make some tests faster
    if (strcmp(optionget(options,'ReduceModel',defaultopt),'yes')),
        for m = 1:size(modi,1),
            i = modi(m,1);
            j = modi(m,2);
            dempingss.fixed(sp).single(i,j).ss = system(i,j).red(useorder).ss + ...
                                                    system(i,j).red(useorder).ss1 ∗ speed + ...
                                                    system(i,j).red(useorder).ss2 ∗ speedˆ2;
        end
    else
        for m = 1:size(modi,1),
            i = modi(m,1);
            j = modi(m,2);
            dempingss.fixed(sp).single(i,j).ss = system(i,j).ss + ...
                                                    system(i,j).ss1 ∗ speed + ...
                                                    system(i,j).ss2 ∗ speedˆ2;
        end
    end

    Binf5 = zeros(6,6);
    Binf5(3,5) = Binf5(3,5) + speed∗T.mode(3,3).ainf;
    Binf5(5,3) = Binf5(5,3) − speed∗T.mode(3,3).ainf;
    Binf5(5,5) = Binf5(5,5);
    dempingss.fixed(sp).Binf = Binf5;
end


if (strcmp(optionget(options,'AugmentWithBinf',defaultopt),'yes')),
    % Lookup table for the linear part
    lookup = zeros(6,6,3); % Two for indexes and the last for sign
    lookup([1 3],5,1) = [1 3];
    lookup([1 3],5,2) = 3;
    lookup([2 4],6,1) = [2 4];
    lookup([2 4],6,2) = 2;
    lookup(5,[1 3],1) = 3;
    lookup(5,[1 3],2) = [1 3];
    lookup(6,[2 4],1) = 2;
    lookup(6,[2 4],2) = [2 4];

    lookup(1:4,5,3) = 1;
    lookup(1:4,6,3) = -1;
    lookup(5,1:4,3) = -1;
    lookup(6,1:4,3) = 1;

    for m = 1:size(modi,1),
        i = modi(m,1);
        j = modi(m,2);
        if (mod(i,2)==mod(j,2)) & (lookup(i,j,3)~=0),
            depi = lookup(i,j,1);
            depj = lookup(i,j,2);
            sign = lookup(i,j,3);

            dempingss.linear.ss(i,j) = dempingss.linear.ss(i,j) + sign*speed*T.mode(depi,depj).ainf;
        end
    end
end
```

C.4 Plotting code

The plotting code simplifies batch creation of result plots.

```
function make_plot(nodes, modes, plotfunction, printopts, location, options)

% Make plots
for m = 1:size(modes,1),
    modei = modes(m,1);
    modej = modes(m,2);

    fh = [];
    for i = 1:length(nodes),
        d = nodes(i).data;
        fh = feval(plotfunction,fh,d,modei,modej,nodes(i).symbol, options);
    end
    legend({nodes.name},0);
    for popt = printopts,
        print(gcf, popt{1}, sprintf(location, modei, modej));
    end
    saveas(gcf, sprintf(location, modei, modej), 'fig');

end
```

```
function fh = draw_impulseresponse(fh, data_node, i, j, symbol, options);

defaultopt = struct ( ...
    'ImpulsePlotMaxTime',30);

if nargin<6, options = []; end;
maxtime = optionget(options,'ImpulsePlotMaxTime',defaultopt);

Kfixed = data_node.K;

if isempty(fh),
    fh = figure;
    hold on;
    title(['K_{' int2str(i) int2str(j) '}']);
    xlabel('s');
    ylabel(enhetstreng(i,j,0));
    grid on;
end;
figure(fh);

plotinterval = find(Kfixed.tpts<=maxtime);
plot( Kfixed.tpts(plotinterval) , squeeze(Kfixed.Kijt(i,j,plotinterval)) ,symbol);
```

```
function fh = draw_force_simulation_results(fh, data_node, i, j, symbol, options)

if isempty(fh),
    fh = figure;

    xlabel('Excitation␣frequency␣(\omega␣)');
    ylabel('Relative␣error');
    title(['Convolution␣calculation␣compared␣to␣freq.var.␣coefficients,␣K_{' int2str(i) int2str(j) '}']);

    grid on; hold on;
end

figure(fh);

res = data_node.testresult;

td = res.mode(i,j).td_amplitude;
fd = res.mode(i,j).fd_amplitude;
freq_range = res.freq_range;

plot(freq_range,(td−fd)./fd,symbol);

v = axis;
v(3) = −0.1;
v(4) = 0.1;
axis(v);
```

```
function fh = draw_restoring_coefficient(fh, data_node, i, j, symbol, options)

if isempty(fh),
    fh = figure;
    title(['C_{' int2str(i) int2str(j) '}']);
    xlabel('Frequency␣(\omega␣)');
    grid on; hold on;
end

figure(fh);

coeffs = data_node.coeffs;
cijw = data_node.Cradijw;

frequency_points = find(coeffs.wi<3);

plot(coeffs.wi(frequency_points),squeeze(cijw(i,j,frequency_points)),symbol);
```

```
function fh = draw_coefficient(fh, data_node, i, j, symbol, options);

c = data_node.coeffs;

if isempty(fh),
    fh = figure;
    subplot(2,1,1);
    title(sprintf('Hydrodynamic␣coefficients␣as␣function␣of␣frequency,␣from␣mode␣%i␣to␣mode␣%i', j, i));
    xlabel('[rad/s]');
    ylabel(['Added␣mass␣' enhetstreng(i,j,1)]);
    grid on; hold on;

    subplot(2,1,2);
    xlabel('[rad/s]');
    ylabel(['Potential␣damping␣' enhetstreng(i,j,0)]);
    grid on; hold on;

end;


figure(fh);

subplot(2,1,1);
plot(c.wi, squeeze(c.Aijw(i,j,:)),symbol);
subplot(2,1,2);
plot(c.wi, squeeze(c.Bijw(i,j,:)),symbol);
```

Afterword

For printing reasons, as well as other practical concerns, there were a few weeks between finishing the thesis and delivering it. This created the opportunity to provide a small afterword. The comments it contains are some my personal experiences with the work.

Writing a thesis such as this is continous process, starting well before the final results are achieved. The direction may not be fully known in advance, even the scope of the work may be uncertain. This has very much been the case here.

Part of this may simply be because of the choice to do the work abroad, creating practical challenges as well as the time-consuming, though exciting, process of establishing new relations. Writing a thesis at home makes it easier to draw on the availble resources — one knows where to get help for what problems. Of course, extra time is granted (six weeks, for a total of 26), and this time is most certainly necessary and should be used carefully.

However, it would clearly have been advantageous to more firmly decide on problem formulation and scope before leaving. Preferably, the topic should be one of common interest and activity between the participating institutions. My topic was related to a previous work done at IST, and as it was at the moment not a active research topic there, it was sometimes not that easy to get the desired time from local staff. The main reason for this was the fact that one of the main planned contact points at IST left the job just a week after I arrived. The help and interest received from others was very good, and Nuno Fonseca deserves special mention, but as time progressed and problems arose, more focus would have been helpfull.

My advice to other students is to plan carefully any stay abroad, and especially a stay involving a final thesis or large work. Make sure that local students and staff have interest in the work, and the necessary time available. Still, I would thoroughly recommend taking part of the study abroad, it is an experience worth to have.

Then, on to some comments more relating to the content of this thesis. Work would probably have progressed very differently if [1] had been discovered earlier. Professor Thor I. Fossen deserves thanks for finding it and sending me a copy. It is amazing how much harder it is to utilize previous research that is not available electronically in publication databases. The contents of that article is not included in the detail it would deserve in this work. However, good understanding of that article, as well as the strip-method derivations of [31], should open many eyes. While I in this thesis dwelve a great deal on related subjects, it was only after understanding [1] that I really understood what was going on.

Not having the opportunity to incorporate this new understanding into the existing material presented here, I will have to settle on briefly stating my thoughts, and hope somone (or me) at a later time will have the opportunity to present them more thoroughly.

- The added mass matrix is constant, symmetric and positive definite. (And in particular: speed-invariant.)
- The damping from the radiation potential is constant, that is, the kernel in the convolution integral does not change. The local, relative, velocity should be used as input, however. This makes it unnecessary to split the damping in speed-indepedendent and speed-dependent parts.
- Both added mass and damping should be calculated in the body frame.
- The radiation restoring forces come from the changing direction of ambient (current) flow as the vessel moves. Using relative fluid velocity is the proper way of handling this problem.
- Relative fluid velocities should also, probably, be used in the forces expressed by the Lie bracket ("Coriolis and sentripetal" forces), but I do not know for sure.
- The radiation potential gives linear damping at constant velocities, even though the damping is said to be zero at $\omega = 0$.

These are mentioned in the main text, but not given the required attention they deserve. As said, someone can settle this soon, hopefully. There are so many different model formulations that it is very easy to get lost.

The work has given me much greater understanding of the problem. Hopefully, it will be of use to others.

smund Hjulstad, Trondheim, August 2004.

Index