

PARAMETRIC ROLL INSTABILITY of SHIPS

by

IRFAN AHMAD SHEIKH

THESIS

for the degree of

MASTER OF SCIENCE

(Master i Anvendt matematikk og mekanikk)



Faculty of Mathematics and Natural Sciences
University of Oslo

June 2008

Det matematisk- naturvitenskapelige fakultet
Universitetet i Oslo

Preface

Dynamic stability of ships is a research topic which is in the focus of modern researchers. When Brian Hayman presented this idea it was quite interesting for me. I discussed this idea with Brian Hayman in detail and decided to write my Master's Thesis on this topic. I worked on project [3] in the Autumn of 2007, which was the pre- project to this Master's Thesis.

Irfan Ahmad Sheikh

Acknowledgements

First of all I am thankful to Almighty God who gave me the strength and ability to complete this Master's Thesis. Then I am thankful to Brian Hayman for all his support and assistance. He has helped me a great deal in accomplishing this Thesis. I am also thankful to Dr. Olav Rognebakke and Dr. Gaute Storhaug who gave me useful advices during my work. I am also thankful to Anne-Marie Kristensen who conducted the NAPA runs to calculate the GZ values which has been used in this Thesis.

Summary

Parametric roll may be defined as the spontaneous rolling motion of the ship moving in head or following seas that comes about as a result of the dynamic instability of motion. The development of the parametric roll occurs under the conditions that the encounter angular frequency is approximately twice the roll angular frequency, the wavelength is equal to the ship length and the roll damping is insufficient to dissipate the parametric roll energy.

In this Thesis the dynamic roll behaviour for 5 different combinations of wave heights and wavelengths for different initial conditions and different initial wave positions, without and with a realistic amount of damping have been studied. The Methods used are the three Equations of roll motion based on three variants of the stiffness term in the basic Mathieu equation with damping.

Contents

Preface	iii
Acknowledgements	v
Summary	vii
List of figures	xv
List of tables	xvii
1 Introduction	1
1.1 Definitions	1
1.2 Background	2
1.3 Objectives	3
2 Ship Stability and Parametric Rolling	5
2.1 Definitions	5
2.2 Roll Motions of Ship in Calm Water	10
2.3 Stability of Ship in Longitudinal Waves	10
2.4 Parametric Roll Resonance Phenomenon	11
2.4.1 Conditions for the Development of the Phenomenon	12

2.4.2	Development of the Phenomenon During One Roll Period	12
2.5	Effect of Roll Damping	14
3	A Simple Dynamic Model of Parametric Roll	17
3.1	An Analogy to the Rolling Motion	17
3.2	The Mathieu Equation with Damping	18
3.3	Solution of the Mathieu Equation	20
3.4	Solutions without Roll Damping	20
3.4.1	Bounded Solutions	21
3.4.2	Unbounded Solutions	21
3.5	Effect of Roll damping on the Solutions	22
3.5.1	Effect on the Bounded Solutions	22
3.5.2	Effect on the Unbounded Solutions	23
3.6	Ince-Strutt Diagram	24
3.6.1	Original Ince-Strutt Diagram	24
3.6.2	Enlarged Ince-Strutt Diagram	25
4	Ship Data and GZ Data	27
4.1	Ship Data	27
4.2	GZ Data	27
4.3	Some GZ Curves	28
5	Calculation Methods	31
5.1	Three Variants of the Stiffness Term	31
5.2	1st Equation of Roll Motion	32
5.3	2nd Equation of Roll Motion	33

5.4	3rd Equation of Roll Motion	33
5.5	Solutions of the Equations of Roll Motion	34
5.5.1	Solution of the 1st Equation	34
5.5.2	Solution of the 2nd Equation	35
5.5.3	Solution of the 3rd Equation	35
5.6	Estimation of the Coefficients in the Differential Equations	35
5.6.1	Estimation of the Coefficients in Equations (5.16)	36
5.6.2	Estimation of the Coefficients in Equations (5.18)	41
5.6.3	Estimation of the Coefficients in Equations (5.20)	42
5.7	Initial Wave Position	43
5.8	Stability Analysis	43
6	Results	45
6.1	Results for the 1st Method	45
6.1.1	Case 1	46
6.1.2	Case 2	47
6.1.3	Case 3	48
6.1.4	Case 4 and case 5	49
6.2	Results for the 2nd Method	50
6.3	Results for the 3rd Method	51
7	Conclusions	53
A	MATLAB Runs	55
A	MATLAB Input Files	61

Bibliography

95

List of Figures

1.1	Degrees of freedom for a ship	1
1.2	APL CHINA container damage	3
2.1	Centre of gravity and the centre of buoyancy	6
2.2	The metacentre	7
2.3	The righting lever GZ	8
2.4	The three stability conditions	9
2.5	GZ curve	9
2.6	Undamped small roll motions in calm water	10
2.7	Waterlines in wave trough and on wave crest	11
2.8	Parametric roll resonance	12
2.9	Development of parametric roll resonance	13
2.10	Effect of roll damping in calm water	15
3.1	Spring-mass system with damping	17
3.2	Bounded solution of the Mathieu equation	21
3.3	Unbounded solution of the Mathieu equation	22
3.4	Effect of roll damping on the bounded solutions	22
3.5	Effect of roll damping on the unbounded solutions	23

3.6	Original Ince-Strutt diagram	24
3.7	Enlarged Ince-Strutt diagram	25
4.1	GZ curve for the still water case	28
4.2	GZ curves for case 1 and case 2	29
4.3	GZ curves for case 3 and case 4	29
4.4	GZ curves for case 5	30
5.1	GM curves for case 1 and case 2	38
5.2	GM curves used for interpolation	42
5.3	GZ mesh used for interpolation	43
6.1	Stability for case 1	46
6.2	Stability for case 2	48
6.3	Stability for case 3	49
6.4	Stability for case 4	50
6.5	Stability for case 5	50
A.1	Matlab runs	55
A.2	Matlab runs	56
A.3	Matlab runs	56
A.4	Matlab runs	56
A.5	Matlab runs	57
A.6	Matlab runs	57
A.7	Matlab runs	57
A.8	Matlab runs	58

A.9 Matlab runs	58
A.10 Matlab runs	58
A.11 Matlab runs	59
A.12 Matlab runs	59

List of Tables

- 4.1 The combinations of wave heights and wavelengths 28

- 6.1 *GM* values and roll angular frequencies 45
- 6.2 Data in the instability regions for case 1 46
- 6.3 Detail of the runs for case 1 in MATLAB 47
- 6.4 Data in the instability regions for case 2 48
- 6.5 Data in the instability regions for the case 3 49

Chapter 1

Introduction

1.1 Definitions

The motion of a ship can be described with the help of six degrees of freedom shown in Figure 1.1.

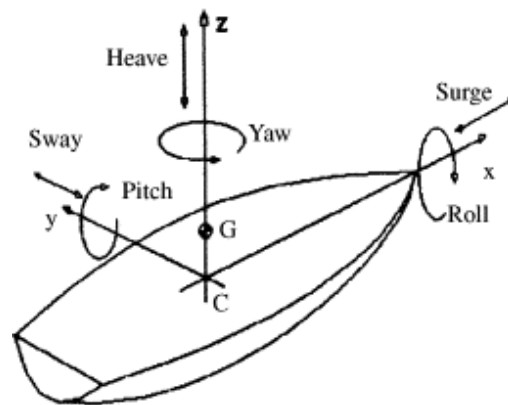


Figure 1.1: Degrees of freedom for a ship.

These degrees of freedom are defined below.

Surge: Surge is the translational motion of a ship along its longitudinal axis (x-axis).

Sway: Sway is the translational motion of a ship along its transverse axis (y-axis).

Heave: Heave is the translational motion of a ship along its vertical axis (z -axis).

Roll: Roll is the rotational motion of a ship about its longitudinal axis (x -axis).

Pitch: Pitch is the rotational motion of a ship about its transverse axis (y -axis).

Yaw: Yaw is the rotational motion of a ship about its vertical axis (z -axis).

Following are some definitions which will help us in understanding the problem.

Head seas: Head seas means that the waves are coming from in front of the ship.

Following seas: Following seas means that the waves are coming from behind the ship.

Longitudinal waves: The waves which are coming either from in front of or from behind the ship are known as longitudinal waves.

Parametric roll: Parametric roll may be defined as the spontaneous rolling motion of the ship moving in head or following seas that comes about as a result of the dynamic instability of motion.

1.2 Background

Modern research on parametric rolling of ships was first conducted in Germany in the late 1930s. The main objective of the research was to explain the capsizing of some small ships such as coasters and fishing vessels in severe following seas. The phenomenon of parametric rolling was observed as a byproduct. This phenomenon was thought to be related to small ships moving in following seas. In the 1990s there were some incidents in which container ships and even some cruise ships experienced heavy rolling in head seas. The APL CHINA casualty in October 1998, described in Paulling [1], is shown in Figure 1.2. This incident focused the attention of researchers on parametric rolling in head seas.

Parametric roll resonance is a phenomenon in which a small initial disturbance in roll can cause an oscillatory rolling motion that grows to a large amplitude after only a few cycles. The development of parametric roll is related to the periodic change of stability as the ship moves in longitudinal waves at a speed when the ship's wave encounter frequency is approximately twice the natural roll



Figure 1.2: APL CHINA container damage.

frequency and the damping of the ship is insufficient to dissipate the parametric roll energy as described in ABS Guide [2]. This phenomenon seems to occur with certain hull forms, and especially those having flared (non-vertical) sides. Large container ships normally have a large flare in the forward region.

A considerable research work has been done on this phenomenon at both Det Norske Veritas (DNV) and the Technical University of Denmark (DTU) as well as other institutions in Europe and the USA. Attention has been largely focused on identifying the conditions that trigger the instability so that the crew can take action to prevent the development of the phenomenon, e.g. by changing speed or course. Some research has studied ways of stabilising the roll motion by means of active fins and other devices.

1.3 Objectives

The main aim of this MSc Project is to study the parameters that lead to parametric roll instability, and to provide easy-to-use tools for modelling dynamic roll motion. Specific objectives are as follows.

- To further develop the MATLAB model developed in the pre-project [3], for simulating the dynamic roll behaviour of a ship travelling in longitudinal waves.

- To develop MATLAB routines to calculate the nonlinear relationship between the hydrostatic righting moment arm (GZ) and roll angle (ϕ) for a ship hull from given hull lines.
- To study the dynamic roll behaviour of a specific container ship with the help of different MATLAB models and then compare the model predictions with those in published literature, and with existing full-scale measurements.
- To perform parametric studies for different values of wave height, wavelength and ship speed to explore stability boundaries for roll response in head or following seas, both in the absence of roll damping and in the presence of realistic amounts of roll damping.

Chapter 2

Ship Stability and Parametric Rolling

2.1 Definitions

First of all definitions of some important terms are presented. These definitions are taken from Rawson et al. [4].

Load waterplane or design waterplane (LWP): That waterplane to which the ship is being designed is called the load waterplane or the design waterplane. This plane may or may not be parallel to the keel.

Load waterline (LWL): The waterline at the level of the load waterplane is called the load waterline.

Fore perpendicular (FP): Fore perpendicular is defined as the vertical line passing through a point at which the load waterline crosses the front face of the bow.

After perpendicular (AP): After perpendicular is defined as the vertical line passing through a point at which the load waterline crosses the rear face of a rudder post of the ship.

Length between perpendiculars (LBP or L_{pp}): The distance between the fore perpendicular FP and the after perpendicular AP is called the length between perpendiculars.

Amidships or midships: The transverse plane midway between the perpendiculars FP and AP is called amidships or midships. The section of the ship at this plane is called the midship section.

Volume of displacement (∇): The total volume of fluid displaced by the ship is called the volume of displacement.

Weight displacement (Δ): The weight of the water displaced by the ship is called the weight displacement. This is also known as the ship displacement since the weight of the water displaced by the ship is equal to the weight of the ship.

Buoyancy: The buoyancy of a body immersed in a fluid is the vertical upthrust it experiences due to displacement of the fluid. The magnitude of the net upward buoyancy force is equal to the weight of the fluid displaced by the body.

Centre of gravity (G): The centre of gravity of a body is that point through which the whole weight of the body may be assumed to act. The centre of gravity of the ship is denoted by G .

Centre of buoyancy (B): The centre of volume of fluid displaced by a ship is known as the centre of buoyancy. This point is denoted by B . When the ship is upright, the centre of buoyancy B is directly below the centre of gravity G of the ship as shown in the Figure 2.1. The buoyancy force passes through the centre

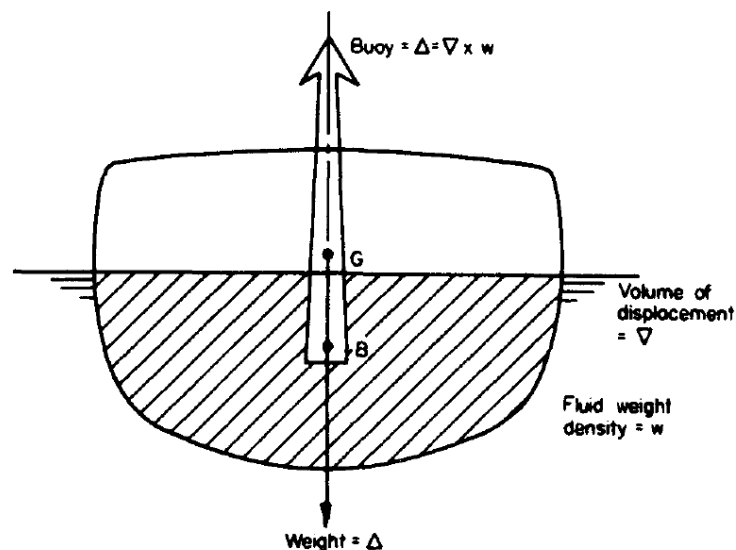


Figure 2.1: Centre of gravity and the centre of buoyancy.

of buoyancy and is given by

$$Buoy = \Delta = \nabla \times w, \quad (2.1)$$

where w is the fluid weight density.

Metacentre (M): Consider the body shown in Figure 2.2 floating upright and freely at waterline WL , whose centre of buoyancy is at B . Let the body be rotated through a small angle in the plane of paper without altering the volume of displacement. For convenience the body is assumed to be fixed and the waterline rotated to W_1L_1 . The centre of buoyancy for this new immersed shape is at B_1 . Lines through B and B_1 normal to their respective waterlines intersect at M which is known as the metacentre.

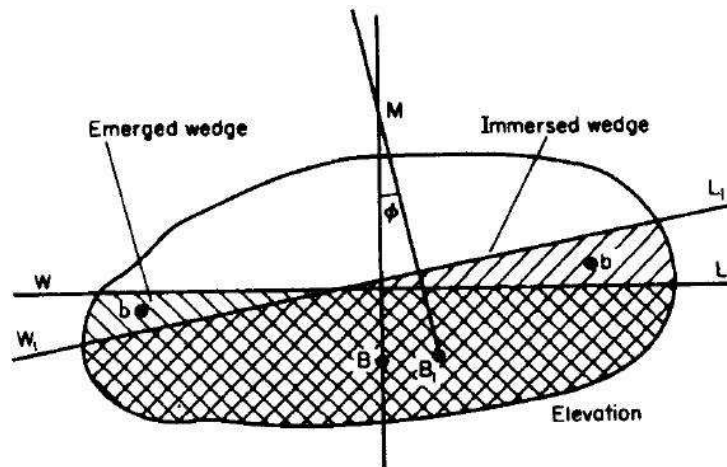


Figure 2.2: The metacentre.

Righting lever (GZ): Consider the body shown in Figure 2.3 floating in a state of equilibrium. In this state the centre of buoyancy B_0 and the centre of gravity G lie on the same vertical line. If the body is now rotated through a small angle β , the centre of buoyancy will move to some new position, B . For convenience, in Figure 2.3 the rotation is shown as being caused by a pure moment about G . The weight of the body acts vertically downwards through G and the buoyancy force acts vertically upwards through B . Thus the body is subject to a moment ΔGZ , where Z is the foot of the normal from G on to the line of action of the buoyancy force. This moment tends to restore the body to the original position. The couple is known as the righting moment and the distance GZ is known as the righting lever.

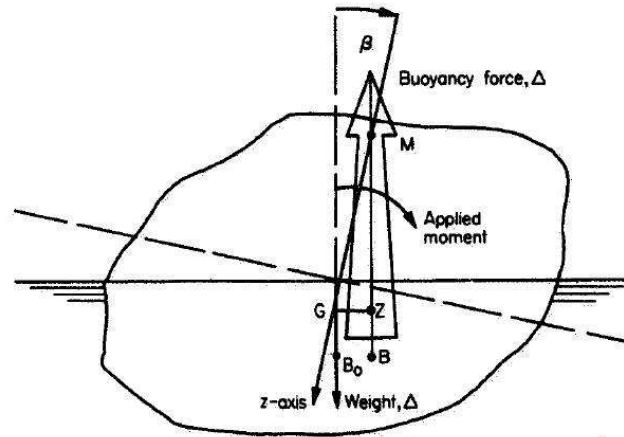


Figure 2.3: The righting lever GZ .

In Figure 2.3 the roll angle is denoted by β , but we shall denote it by ϕ in this report.

Metacentric height (GM): The distance between the Centre of gravity G and the metacenter M is called the metacentric height. The metacentric height GM is a characteristic of the ship which helps determine its stability in the water. There are three stability conditions as explained below.

1. **Positive:** The metacentric height is said to be positive when M lies above G . In this case the righting moment tends to restore the body to the original position as shown in Figure 2.4(a). This is the condition of stable equilibrium.
2. **Neutral:** If M and G coincide then no righting moment acts on the body as shown in Figure 2.4(b). The equilibrium is neutral in this case.
3. **Negative:** The metacentric height is said to be negative when M lies below G . In this case the righting moment tends to increase the rotation angle as shown in Figure 2.4(c). The equilibrium is unstable in this case.

For small values of roll angle ϕ the following relation applies.

$$GZ = GM \sin \phi \simeq GM\phi \quad (2.2)$$

This means that the righting lever and so the righting moment increases as GM increases and vice versa. The greater the GM value or GZ value the greater is the stability of the ship. The stability of a ship is measured by GM for small angles of heel and by GZ for large angles of heel.

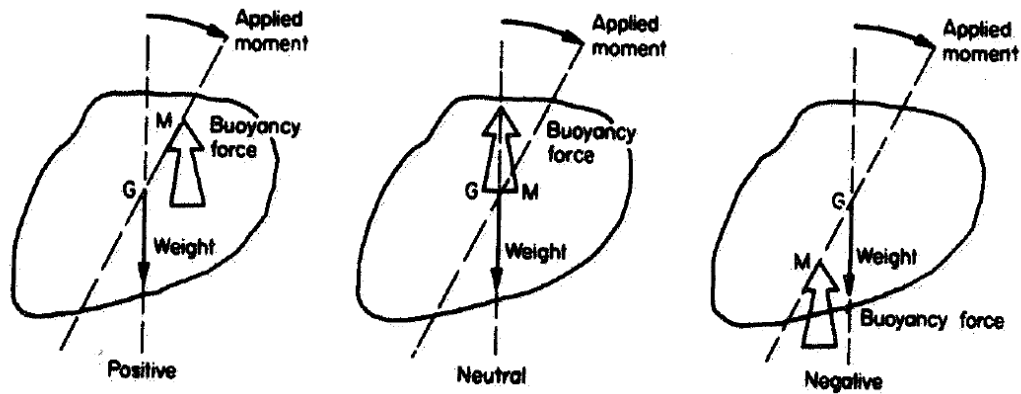
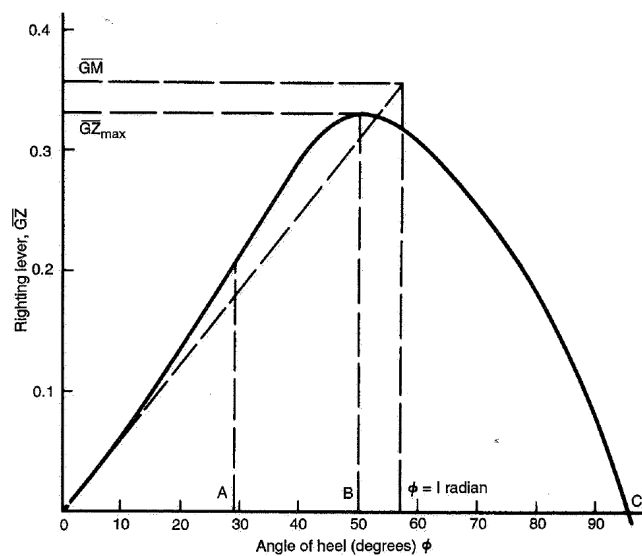


Figure 2.4: The three stability conditions.

GZ curve: For practical applications it is necessary to present ship's stability in the form of righting moments or lever arms about the centre of gravity, as the ship is heeled at constant displacement. A curve which shows variation in the righting lever GZ as the ship is heeled at constant displacement, is known as statical stability curve or GZ curve. Such a curve is shown in Figure 2.5.

Figure 2.5: GZ curve.

The initial slope of the GZ curve represents the metacentric height of ship in still water.

2.2 Roll Motions of Ship in Calm Water

When a ship is in calm water, any disturbance in transverse direction can cause roll motions. When the equilibrium is disturbed, the righting moment or the hydrostatic restoring moment tends to return the ship back to the upright position. When the equilibrium position is attained, the ship does not stop there but due to inertia continues to roll at a decreasing velocity. When a maximum roll angle is attained, the roll restoring moment tends to push the ship back to the upright position. Again when the equilibrium position is attained, the ship does not stop there but due to inertia continues to roll. Thus the ship begins to oscillate about the equilibrium point with a constant roll period and amplitude if the damping is neglected. These free roll oscillations, described in ABS Guide [2], are shown in Figure 2.6.

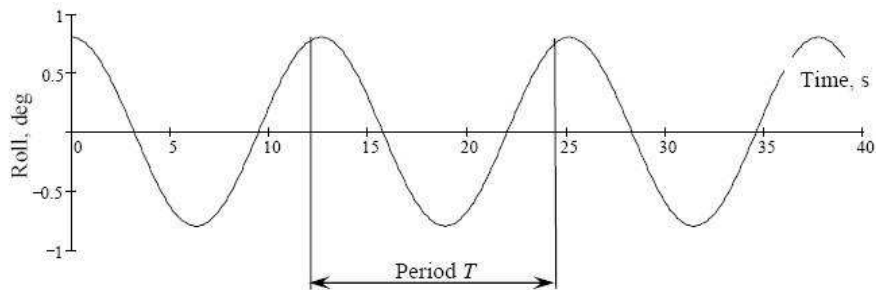


Figure 2.6: Undamped small roll motions in calm water.

The period of such roll oscillations in calm water is known as the *natural roll period*. The roll frequency corresponding to this roll period is called the *natural roll frequency*.

2.3 Stability of Ship in Longitudinal Waves

Now we study the effect of a ship's location on its stability in the longitudinal waves. If the ship is located in a wave trough, the average waterplane width is very much greater than in calm water. The bow and the stern of the ship are more deeply immersed than in calm water and the midship region is less deep as shown in Figure 2.7(a). This makes the mean instantaneous waterplane wider than in calm water with the result that the metacentric height GM is increased

over that in calm water. Thus according to Equation (2.2) the GZ value becomes greater and thereby the righting moment of the ship becomes greater than in calm water. This means that the stability of the ship is increased in a wave trough.

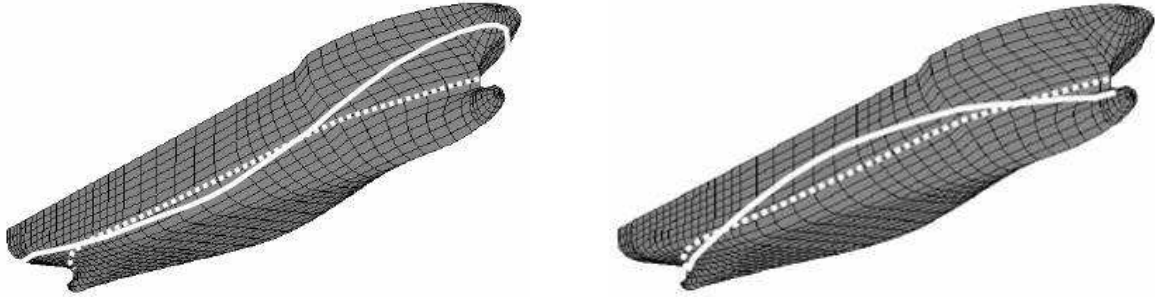


Figure 2.7: (a) Waterline in wave trough.

(b) Waterline on wave crest.

On the other hand when the wave crest is located amidships, the bow and the stern of the ship are less deeply immersed than in calm water as shown in Figure 2.7(b). As a result the average waterplane is narrower than in calm water and the metacentric height GM is decreased as compared to that in calm water. Thus according to Equation (2.2) the GZ value becomes smaller and thereby the righting moment of the ship becomes smaller than in calm water. Hence the stability of the ship is decreased on the wave crest. This phenomenon is described in ABS Guide [2].

2.4 Parametric Roll Resonance Phenomenon

We have come to the result that when a ship is sailing in longitudinal seas, its stability increases in the wave trough and decreases on the wave crest. This change in stability is oscillatory. If this change occurs at approximately twice the natural roll period, roll motions may attain a very high roll angle as a result of parametric roll resonance. A typical development of parametric roll resonance is shown in Figure 2.8.

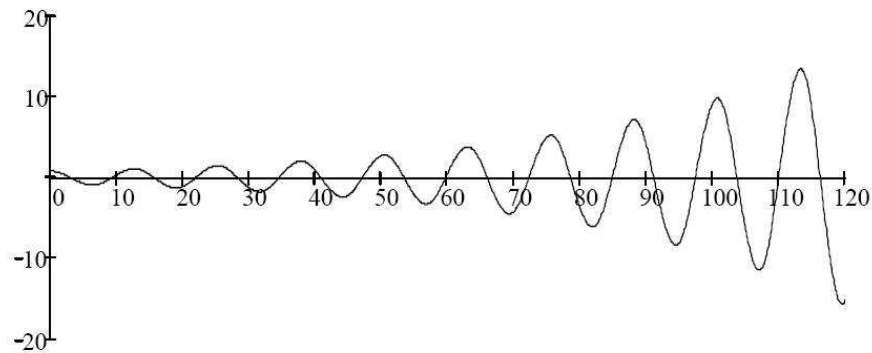


Figure 2.8: Parametric roll resonance.

2.4.1 Conditions for the Development of the Phenomenon

Following are the conditions for the development of the parametric roll resonance phenomenon.

1. The wave encounter frequency is approximately twice the natural roll frequency.
2. The wavelength of the encountering wave is approximately equal to the ship length.
3. The roll damping of the ship is insufficient to dissipate the parametric roll energy.

The increase in the parametric roll motion and roll angle is most rapid when the wave crest is moving away from amidships.

2.4.2 Development of the Phenomenon During One Roll Period

The development of parametric roll resonance is shown in Figure 2.9. The graph in the figure shows the variation in the metacentric height GM and the roll angle with time. The graph also shows the free undamped rolling oscillations in calm water and the mean GM value in waves. We see that at time $t = 0$ the ship

has a roll angle equal to the amplitude of the free roll oscillations, and a GM value equal to the mean GM value. The roll angle begins to decrease and the GM value begins to increase from this time. This is the time when the wave crest is moving away from amidships, and the ship encounters the waves with an encounter frequency approximately twice the natural roll frequency.

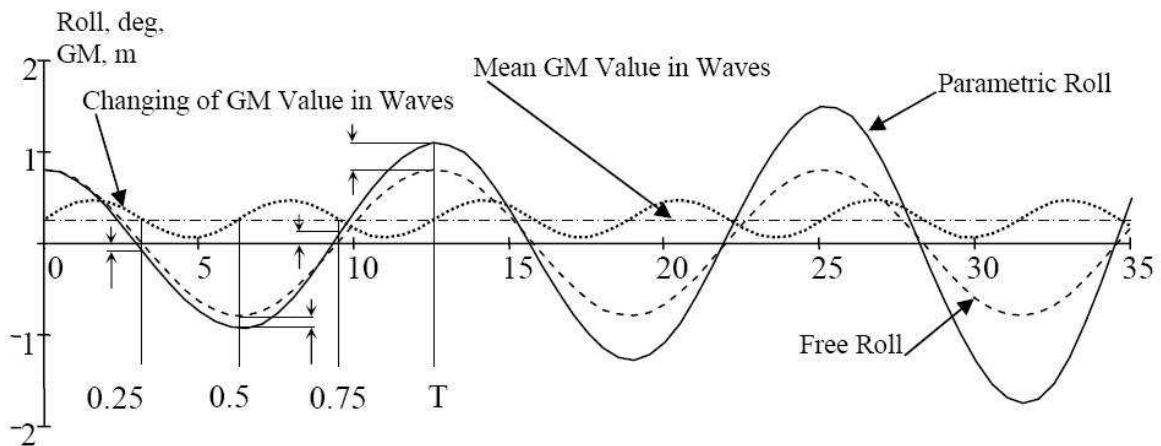


Figure 2.9: Development of parametric roll resonance.

We divide the period of roll oscillation T in four quarters and study the change in the roll angle in every quarter.

First Quarter

In the first quarter the stability of the ship is improving since it is entering a wave trough. As a result the restoring moment tends to push the ship back to the equilibrium position with a larger-than-calm-water moment since the GM value in this quarter is greater than the mean GM value as shown in Figure 2.9. This makes the change in roll angle slightly larger than in calm water as shown in Figure 2.9.

Second Quarter

In the second quarter the ship encounters a wave crest and as a result its stability decreases. The restoring moment now resists the roll motion with a less-than-

calm-water moment since the GM value in this quarter is below the mean GM value as shown in Figure 2.9. As a result the ship attains a roll angle greater than in calm water as shown in Figure 2.9.

Third Quarter

In the third quarter the ship undergoes the same situation as in the first quarter and experiences a restoring moment which is larger-than-calm-water moment since the GM value in this quarter is greater than the mean GM value. The change in roll angle becomes greater than in calm water as shown in Figure 2.9.

Fourth Quarter

In the fourth quarter the situation is similar to that in the second quarter and the ship experiences a restoring moment resisting the roll motion with a less-than-calm-water moment since the GM value in this quarter is below the mean GM value. Hence, at the end of the period T of roll oscillation, the roll angle becomes even greater than in calm water as shown in Figure 2.9.

Overall Effect

If there is no change in the ship's wave encounter frequency, the combination of restoring moment (with a larger-than-calm-water value) and restoring moment resisting the roll (with less-than-calm-water value) can cause the roll angle to grow to a very high level after only a few cycles. This is called the parametric roll resonance phenomenon. This phenomenon is described in Shin et al. [5].

2.5 Effect of Roll Damping

When a ship rolls in calm water, the roll amplitudes decrease successively due to roll damping as shown in Figure 2.10, described in ABS Guide [2].

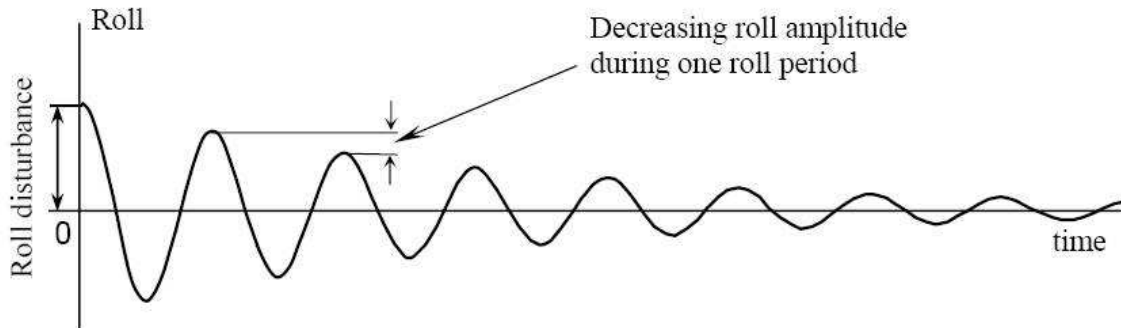


Figure 2.10: Effect of roll damping in calm water.

Roll damping plays an important role in the development of parametric roll resonance. If the loss of energy per cycle caused by roll damping is more than the gain in energy caused by the changing stability in longitudinal seas, the roll angles will not increase and the parametric roll resonance will not develop. When the gain in energy per cycle is more than the loss of energy due to roll damping, the amplitude of the parametric roll starts to grow.

Chapter 3

A Simple Dynamic Model of Parametric Roll

3.1 An Analogy to the Rolling Motion

The dynamic rolling motion of a ship experiencing stability variations in head or following seas is analogous to that of a spring-mass system in which the spring constant k varies sinusoidally with time as described in Paulling [1]. This spring-mass system with damping included, described in Chopra [6, Chap 1], is shown in Figure 3.1.

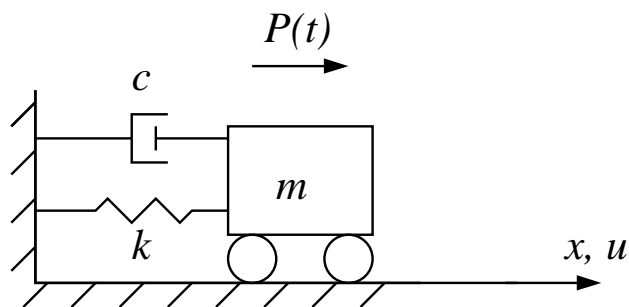


Figure 3.1: Spring-mass system with damping.

The equation of motion for this system is given by

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) = p(t), \quad (3.1)$$

where $u(t)$, $\dot{u}(t)$ and $\ddot{u}(t)$ represent displacement, velocity and acceleration of the mass respectively. The term $m\ddot{u}(t)$ represents the force of inertia, the term $c\dot{u}(t)$ represents the damping force and the term $ku(t)$ represents the force of the spring while $p(t)$ is the external force.

3.2 The Mathieu Equation with Damping

When the ship moves through head or following seas, its time-varying GM which is analogous to the spring constant k , can be approximated by a sinusoidal function of time given by Equation (3.2), described in Paulling [1].

$$GM(t) = GM_0(1 + C \cos \omega t) \quad (3.2)$$

Here,

$$\begin{aligned} GM_0 &= \text{still water GM,} \\ C &= \text{fractional variation of GM,} \\ \omega &= \text{encounter angular frequency.} \end{aligned}$$

The equation of roll motion with damping included and without any external force is given by

$$I_x \frac{d^2\phi}{dt^2} + B \frac{d\phi}{dt} + \Delta\phi(GM_0 + CGM_0 \cos \omega t) = 0, \quad (3.3)$$

where

$$\begin{aligned} I_x &= \text{mass moment of inertia in roll, including added mass,} \\ \phi &= \text{angle of roll,} \\ B &= \text{viscous roll damping coefficient,} \\ \Delta &= \text{ship displacement.} \end{aligned}$$

Equation (3.3) applies only for small values of ϕ . We divide both sides of Equation (3.3) by I_x

$$\frac{d^2\phi}{dt^2} + \frac{B}{I_x} \frac{d\phi}{dt} + \left(\frac{\Delta GM_0}{I_x} + \frac{C\Delta GM_0}{I_x} \cos \omega t \right) \phi = 0 \quad (3.4)$$

We make a substitution,

$$\omega_n^2 = \frac{\Delta GM_0}{I_x}, \quad (3.5)$$

where ω_n is the natural roll angular frequency for the undamped system. Equation (3.4) becomes,

$$\frac{d^2\phi}{dt^2} + \frac{B}{I_x} \frac{d\phi}{dt} + (\omega_n^2 + C\omega_n^2 \cos \omega t)\phi = 0 \quad (3.6)$$

Basic (standard) form of equation of motion without $\cos \omega t$ term is

$$\frac{d^2\phi}{dt^2} + 2\zeta\omega_n \frac{d\phi}{dt} + \omega_n^2\phi = 0, \quad (3.7)$$

where ζ is the damping ratio which is defined as

$$\zeta = \frac{B}{B_{cr}}, \quad (3.8)$$

where B_{cr} is the critical roll damping coefficient.

By comparing the coefficients of $\frac{d\phi}{dt}$ in Equations (3.6) and (3.7) we get

$$\frac{B}{I_x} = 2\zeta\omega_n \quad (3.9)$$

We put Equation (3.8) in Equation (3.9)

$$\begin{aligned} \frac{B}{I_x} &= 2\omega_n \frac{B}{B_{cr}} \\ \Rightarrow B_{cr} &= 2\omega_n I_x \end{aligned} \quad (3.10)$$

We have found the expression for B_{cr} for later use. Now we put Equation (3.9) in Equation (3.6)

$$\frac{d^2\phi}{dt^2} + 2\zeta\omega_n \frac{d\phi}{dt} + (\omega_n^2 + C\omega_n^2 \cos \omega t)\phi = 0 \quad (3.11)$$

We make the change of variable,

$$\tau = \omega t \quad \Rightarrow \quad d\tau = \omega dt \quad \Rightarrow \quad \frac{1}{dt} = \frac{\omega}{d\tau} \quad \Rightarrow \quad \frac{1}{dt^2} = \frac{\omega^2}{d\tau^2}$$

Equation (3.11) becomes,

$$\omega^2 \frac{d^2\phi}{d\tau^2} + 2\zeta\omega\omega_n \frac{d\phi}{d\tau} + (\omega_n^2 + C\omega_n^2 \cos \tau)\phi = 0 \quad (3.12)$$

We divide both sides of Equation (3.12) by ω^2

$$\frac{d^2\phi}{d\tau^2} + 2\zeta \frac{\omega_n}{\omega} \frac{d\phi}{d\tau} + \left(\frac{\omega_n^2}{\omega^2} + C \frac{\omega_n^2}{\omega^2} \cos \tau \right) \phi = 0 \quad (3.13)$$

We define

$$\delta = \frac{\omega_n^2}{\omega^2}, \quad \epsilon = C\delta \quad (3.14)$$

The equation of roll motion now becomes,

$$\frac{d^2\phi}{d\tau^2} + 2\zeta\sqrt{\delta}\frac{d\phi}{d\tau} + (\delta + \epsilon\cos\tau)\phi = 0 \quad (3.15)$$

or

$$\ddot{\phi}(\tau) + 2\zeta\sqrt{\delta}\dot{\phi}(\tau) + (\delta + \epsilon\cos\tau)\phi(\tau) = 0 \quad (3.16)$$

This ordinary 2nd order differential equation where the spring constant varies sinusoidally, is known as the Mathieu equation with damping. The term $(\epsilon\cos\tau)\phi(\tau)$, is called the *parametric excitation term* because the development of parametric roll resonance is accounted for this term.

3.3 Solution of the Mathieu Equation

The Mathieu equation is an ordinary 2nd order differential equation. We shall solve this equation numerically by using Runge-Kutta-method (ode45) in MATLAB. This method solves a system of equations of 1st order. We can show that a system of n ordinary differential equations of order m can be written as a system of $n \cdot m$ equations of 1st order. Thus the Mathieu equation can be transformed to a set of two 1st order equations. We introduce the following to unknowns,

$$\begin{aligned} y_1(\tau) &= \phi(\tau) \\ y_2(\tau) &= \dot{\phi}(\tau). \end{aligned} \quad (3.17)$$

By using these to unknowns Equation (3.16) can be written as

$$\begin{aligned} \dot{y}_1(\tau) &= y_2(\tau) \\ \dot{y}_2(\tau) &= -2\zeta\sqrt{\delta}y_2(\tau) - (\delta + \epsilon\cos\tau)y_1(\tau). \end{aligned} \quad (3.18)$$

The coefficients in these differential equations must be known and the initial conditions for the unknown functions $y_1(\tau)$ and $y_2(\tau)$ in Equations (3.17) must be specified.

3.4 Solutions without Roll Damping

The Mathieu equation without damping may have two types of solutions, bounded or stable solutions and unbounded or unstable solutions as described in [5]. The type of solution depends on the combination of the parameters δ and ϵ . These two types of solutions are explained below.

3.4.1 Bounded Solutions

For some pairs of δ and ϵ the amplitude of the roll motions remains within a certain interval as shown in Figure 3.2(a). This figure shows the roll motion for a pair ($\delta = 0.1$, $\epsilon = 0.2$) with damping excluded and initial conditions ($\phi(0) = 5$ deg, $\dot{\phi}(0) = 0$). We see that the roll angle remains between -5 and 5 degrees. This solution is bounded or stable solution.

Figure 3.2(b) shows the graph between ϕ and $\dot{\phi}$. This is called a phase plane plot. The phase plane plot of the solution shows that $\dot{\phi}$ does not diverge, because the solution is bounded.

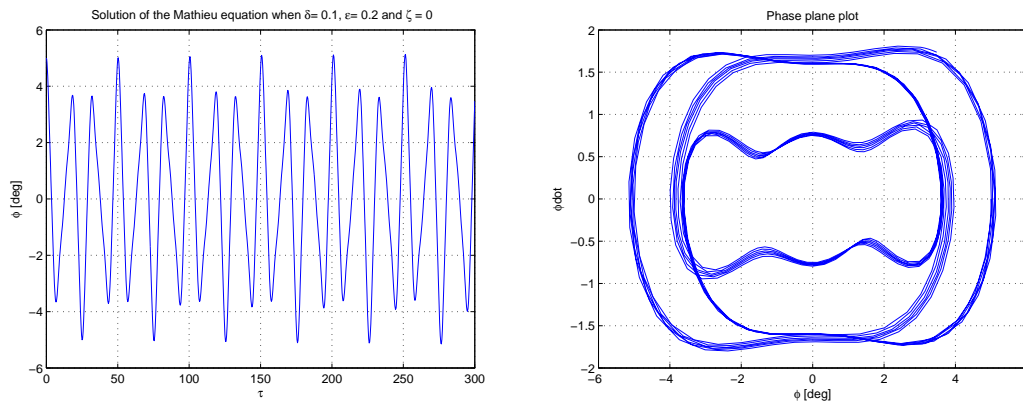


Figure 3.2: (a) Bounded solution of the Mathieu equation (b) Phase plane plot.

3.4.2 Unbounded Solutions

For some pairs of δ and ϵ the amplitude of the roll motion does not remain within a certain interval but continues to grow as shown in Figure 3.3(a). This figure shows the roll motion for a pair ($\delta = 0.15$, $\epsilon = 0.2$) with damping excluded and initial conditions ($\phi(0) = 5$ deg, $\dot{\phi}(0) = 0$). We see that the amplitude of the roll angle increases in every period and becomes more than 30 degrees after only a few cycles. This solution is unbounded or unstable solution. The phase plane plot in Figure 3.3(b) shows that $\dot{\phi}$ diverges, because the solution is unbounded.

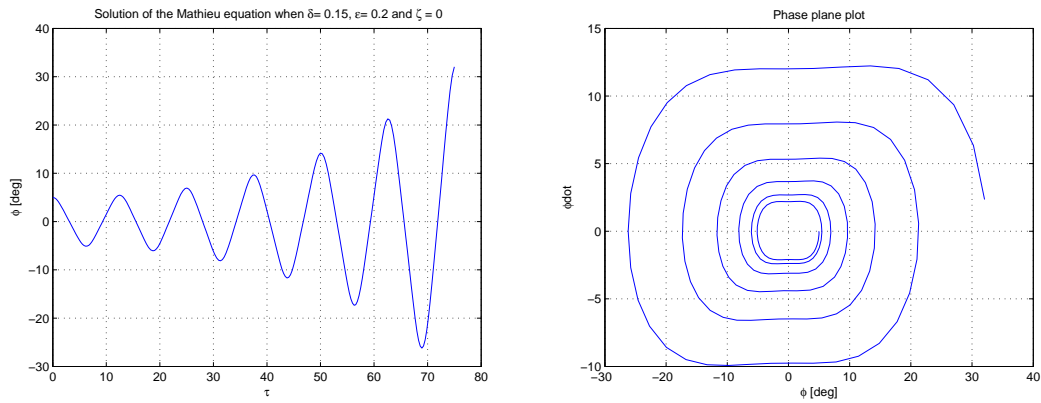


Figure 3.3: (a) Unbounded solution of the Mathieu equation (b) Phase plane plot.

3.5 Effect of Roll damping on the Solutions

The effect of roll damping on the two types of solutions is studied separately.

3.5.1 Effect on the Bounded Solutions

When the roll damping is included, the amplitude of the roll motions for bounded solutions decreases with time as shown in Figure 3.4. Figure 3.4(a) shows the roll motion for the pair $(\delta = 0.1, \epsilon = 0.2)$, with the initial conditions $(\phi(0) = 5 \text{ deg}, \dot{\phi}(0) = 0)$ and a damping ratio $\zeta = 0.05$. Figure 3.11(b) shows the roll motions with different initial conditions $(\phi(0) =, \dot{\phi}(0) = 1.5)$ but the same damping ratio. We see that the roll motion dies after some time.

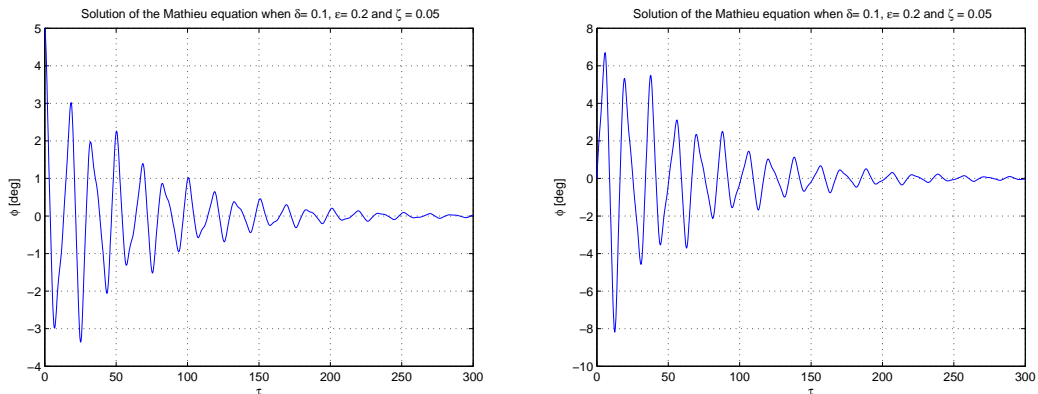


Figure 3.4: Effect of roll damping on the bounded solutions.

3.5.2 Effect on the Unbounded Solutions

There exists a transition value for roll damping for every pair of parameters δ and ϵ for unbounded solution. ABS Guide [2] calls this a threshold value but we will call it a transition value. If roll damping is less than the transition value, the roll motion will be unbounded. If the roll damping is larger than the transition value, the roll motion will be bounded. In Figure 3.5 the effect of roll damping on the roll motions for the pair $(\delta = 0.15, \epsilon = 0.2)$ with initial conditions $(\phi(0) = 5 \text{ deg}, \dot{\phi}(0) = 0)$, is shown.

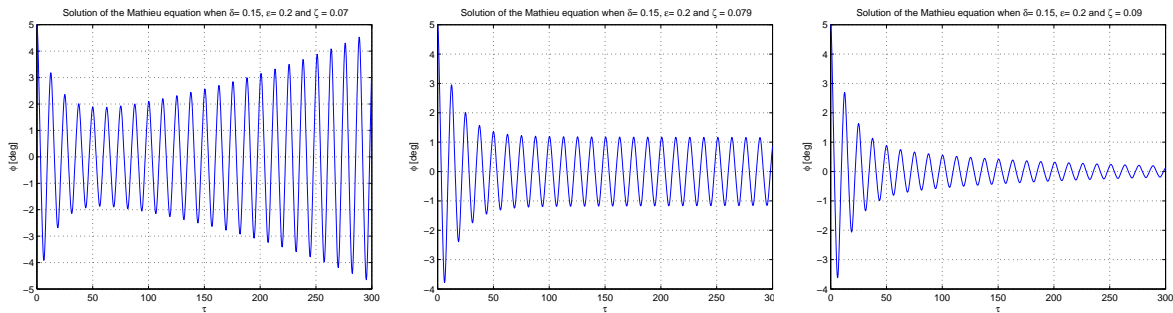


Figure 3.5: (a) Roll damping is less than the transition value
 (b) Roll damping almost equal to the transition value
 (c) Roll damping is greater than the transition value.

We see in Figure 3.5(a) that the amplitude of the roll motions increases with time. This means that the roll damping is less than the transition value and therefore unable to prevent the development of the parametric roll resonance.

In Figure 3.5(b) we see that the amplitude of the roll motions decreases initially but then remains constant at a certain roll angle. In this case the roll damping is equal to the transition value.

In Figure 3.5(c) we see that the amplitude of the roll motions decreases in every cycle due to roll damping. In this case the roll damping is large enough to prevent the development of the parametric roll resonance.

3.6 Ince-Strutt Diagram

3.6.1 Original Ince-Strutt Diagram

The combinations of the two types of solutions can be shown in Figure 3.6. This figure is known as Ince-Strutt diagram as described in Paulling [1]. The shaded areas in the figure correspond to the bounded solutions and the unshaded areas correspond to the unbounded solutions. The unshaded areas are commonly referred to as instability regions.

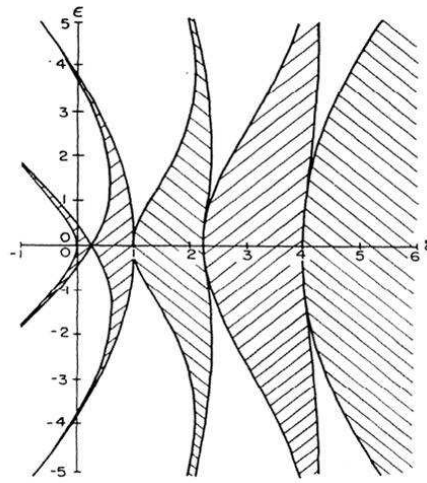


Figure 3.6: Original Ince-Strutt diagram.

We see in this figure that the first instability region is centred on a value $\delta = 1/4$. In Equation (3.14) we have defined δ as the ratio of the square of the natural roll frequency to the square of the encounter frequency. Thus we get the relation,

$$\begin{aligned} \delta &= \frac{\omega_n^2}{\omega^2} = \frac{1}{4} \\ \Rightarrow \frac{\omega_n}{\omega} &= \frac{1}{2} \\ \Rightarrow \omega &= 2\omega_n. \end{aligned} \tag{3.19}$$

This means that any small disturbance in the ship's roll angle will grow to a high amplitude when the wave encounter frequency is approximately twice the natural roll frequency. This is the condition for the development of parametric roll resonance which we have mentioned in Chapter 2.4.1. The unbounded motion

in the first instability region is commonly referred to as the principal parametric resonance as described in Shin et al. [5].

The second instability region is centred on a value $\delta = 1$. For this value the ship's wave encounter frequency is equal to the natural roll frequency. Development of parametric roll resonance is still possible if the other conditions mentioned in Chapter 2.4.1 are fulfilled and the amplitude of the variation in GM is sufficiently large. The unbounded motion in this region is defined as the fundamental parametric resonance as described in Shin et al. [5].

3.6.2 Enlarged Ince-Strutt Diagram

In the original Ince-Strutt diagram the scaling of δ and ϵ is not proportional and hence it is difficult to read the correct values of δ and ϵ . Therefore we introduce an enlarged Ince-Strutt diagram which is properly scaled as shown in Figure 3.7, taken from [8]. We shall use this diagram in the stability analysis. We note that both the axis in this diagram are scaled by a factor of 4. Thus the first instability region is centred on $\delta = 1$ and the second instability region is centred on $\delta = 4$ as shown in Figure 3.7.

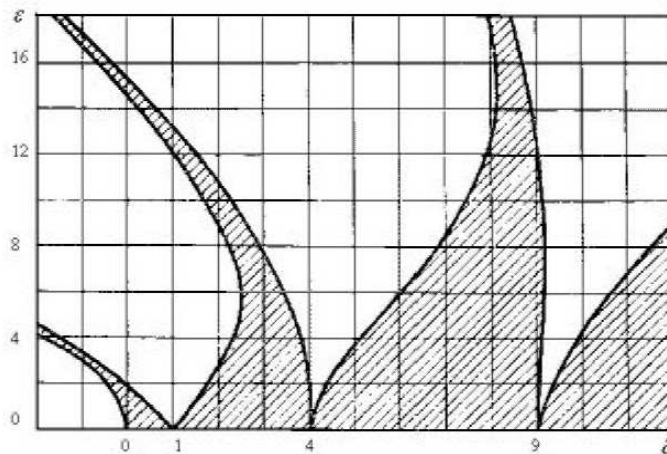


Figure 3.7: Enlarged Ince-Strutt diagram.

Chapter 4

Ship Data and GZ Data

4.1 Ship Data

We shall study the dynamic roll behaviour of a container ship whose data is given below.

Ship Length, $L_{pp} = 232\text{ m}$

Ship Displacement, $\Delta = 54616\text{ tons force}$
 $= 54616 \times 1000\text{ kg} \times 9.81\text{ m s}^{-2}$
 $= 5.358 \times 10^8\text{ N}$

Ship Speed, $V = 21.3\text{ knots}$
 $= 21.3 \times 0.514\text{ m s}^{-1}$
 $= 10.95\text{ m s}^{-1}$

4.2 GZ Data

We are provided the GZ data for a container ship by DNV. These GZ data are computed by a computer program NAPA which has been run for the still water

case and for 21 wave positions at 15 different heel angles. NAPA has been run for 5 different combinations of wave heights and wavelengths. The detail of these combinations of wave heights and wavelengths and corresponding values of wave steepness are given in Table 4.1.

Table 4.1: The combinations of wave heights and wavelengths.

Case	Wave height H [m]	Wavelength L [m]	Wave steepness H/L
1	11.6	232	$0.05 = 1/20$
2	8.4	232	$0.036 \approx 1/28$
3	5.6	232	$0.024 \approx 1/42$
4	5.6	155	$0.036 \approx 1/28$
5	5.6	116	$0.048 \approx 1/21$

We see in Table 4.1 that in the first three cases the wavelength is equal to the ship length, in the case 4 it is about $2/3$ of the ship length and in the case 5 it is $1/2$ of the ship length. We also note that the wave steepness has a minimum value in the case 3.

4.3 Some GZ Curves

The GZ curve for the still water case is shown in Figure 4.1.

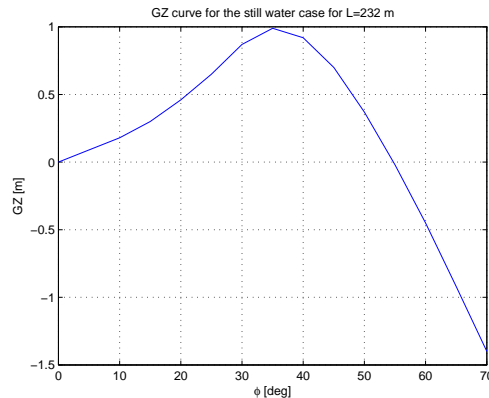


Figure 4.1: GZ curve for the still water case.

The GZ curves for the crest amidships position and the trough amidships position for all the the five combinations of wave heights and wavelengths are shown in Figures 4.2, 4.3 and 4.4

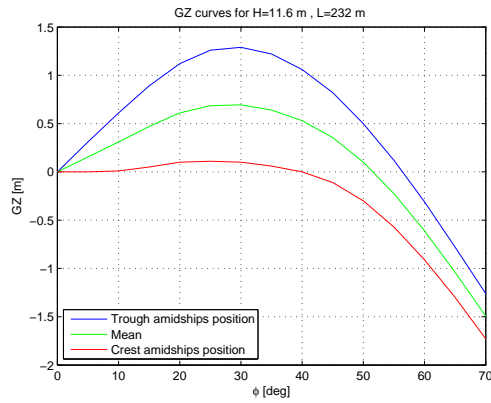
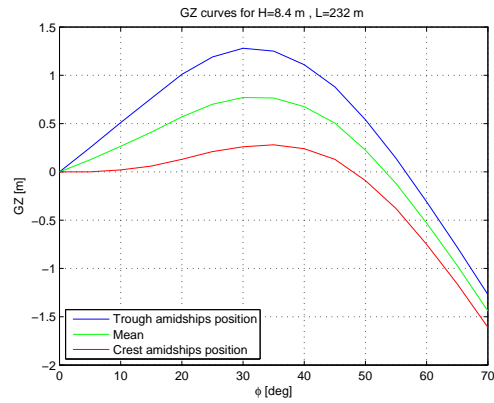


Figure 4.2: (a) GZ curves for the case 1



(b) GZ curves for the case 2.

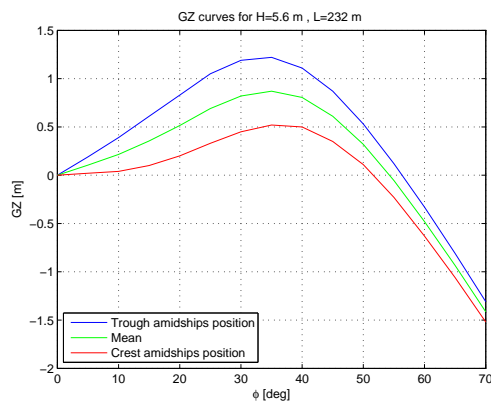
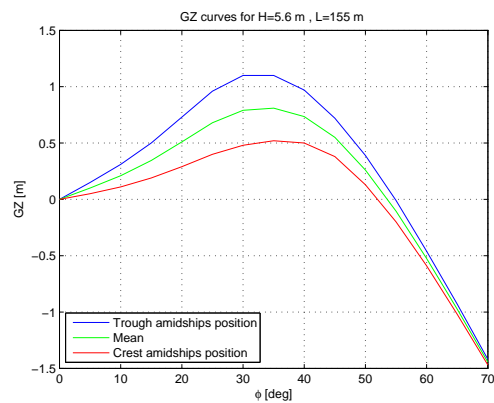


Figure 4.3: (a) GZ curves for the case 3



(b) GZ curves for the case 4.

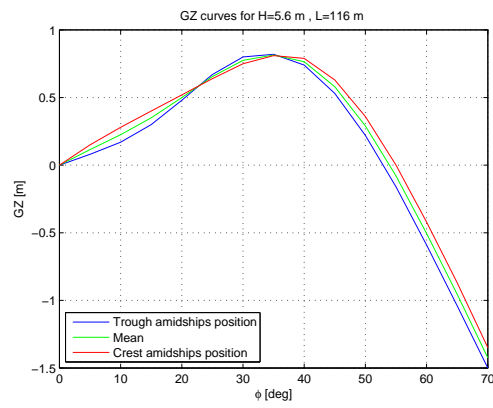


Figure 4.4: GZ curve for the case 5.

Chapter 5

Calculation Methods

We shall use three calculation methods. These methods are actually equations of roll motion corresponding to three variants of the stiffness term which are given below.

5.1 Three Variants of the Stiffness Term

We define the following three variants of the stiffness term.

1. Linearised (small roll angle) formulation using sinusoidally interpolated GM values between values for crest amidships and trough amidships positions.
2. Linearised (small roll angle) formulation using GM values obtained by interpolating between the real GM values for wave positions at $L/20$ intervals.
3. Nonlinear formulation using GZ values obtained by interpolating between GZ values calculated for wave positions at $L/20$ intervals and roll angles between -70 and 70 degrees.

The GM values used in the first two variants are based on GZ values calculated for 10 degrees roll angle. Now we shall find the equations of roll motion corresponding to these three variants of the stiffness term.

5.2 1st Equation of Roll Motion

We shall replace the still water GM values in the stiffness term in Equation (3.4) with the sinusoidally interpolated GM values. For this purpose we define mean metacentric height, GM_m , and variation in mean metacentric height, CGM_m , in Equations (5.1) and (5.2) respectively. All the GM values used are based on GZ values calculated for 10 degrees roll angle.

$$GM_m = \frac{1}{2}(GM_{cr} + GM_{tr}) \quad (5.1)$$

$$CGM_m = \frac{1}{2}(GM_{cr} - GM_{tr}) \quad (5.2)$$

Here,

GM_{cr} = GM value for crest amidships position,

GM_{tr} = GM value for trough amidships position.

We replace GM_0 and CGM_0 in Equation (3.4) with GM_m and CGM_m respectively

$$I_x \frac{d^2\phi}{dt^2} + B \frac{d\phi}{dt} + \Delta\phi(GM_m + CGM_m \cos \omega t) = 0 \quad (5.3)$$

We note that when $\omega t = 0$ the stiffness term becomes,

$$\begin{aligned} \Delta\phi(GM_m + CGM_m) &= \Delta\phi \frac{1}{2}(GM_{cr} + GM_{tr}) + \frac{1}{2}(GM_{cr} - GM_{tr}) \\ &= \Delta\phi GM_{cr}, \end{aligned}$$

while for $\omega t = \pi$ it becomes

$$\begin{aligned} \Delta\phi(GM_m - CGM_m) &= \Delta\phi \frac{1}{2}(GM_{cr} + GM_{tr}) - \frac{1}{2}(GM_{cr} - GM_{tr}) \\ &= \Delta\phi GM_{tr}. \end{aligned}$$

We divide both sides of Equation (5.3) by I_x

$$\frac{d^2\phi}{dt^2} + \frac{B}{I_x} \frac{d\phi}{dt} + \left(\frac{\Delta GM_m}{I_x} + \frac{\Delta CGM_m}{I_x} \cos \omega t \right) \phi = 0 \quad (5.4)$$

We make a substitution,

$$\omega_r^2 = \frac{\Delta GM_m}{I_x}, \quad (5.5)$$

where ω_r is the roll angular frequency for the undamped system. This frequency is different from ω_n used previously because it is based on GM_m rather than GM_0 , and these quantities are not the same. Equation (5.4) becomes,

$$\frac{d^2\phi}{dt^2} + \frac{B}{I_x} \frac{d\phi}{dt} + (\omega_r^2 + C\omega_r^2 \cos \omega t)\phi = 0 \quad (5.6)$$

or

$$\frac{d^2\phi}{dt^2} + \frac{B}{I_x} \frac{d\phi}{dt} + \omega_r^2(1 + C \cos \omega t)\phi = 0 \quad (5.7)$$

or

$$\ddot{\phi}(t) + \frac{B}{I_x} \dot{\phi}(t) + \omega_r^2(1 + C \cos \omega t)\phi(t) = 0 \quad (5.8)$$

This is the 1st equation of roll motion.

5.3 2nd Equation of Roll Motion

We shall now replace the sinusoidally interpolated GM values in the stiffness term in Equation (5.3) with the GM values obtained by interpolating between the real GM values for wave positions at $L/20$ intervals. This means that we shall replace $(GM_m + CGM_m \cos \omega t)$ with GM_i . By making this replacement we get

$$I_x \frac{d^2\phi}{dt^2} + B \frac{d\phi}{dt} + \Delta \phi GM_i = 0 \quad (5.9)$$

We divide both sides of Equation (5.9) by I_x

$$\frac{d^2\phi}{dt^2} + \frac{B}{I_x} \frac{d\phi}{dt} + \frac{\Delta}{I_x} GM_i \phi = 0 \quad (5.10)$$

or

$$\ddot{\phi}(t) + \frac{B}{I_x} \dot{\phi}(t) + \frac{\Delta}{I_x} GM_i(t)\phi(t) = 0 \quad (5.11)$$

This is the 2nd equation of roll motion.

5.4 3rd Equation of Roll Motion

We shall now replace the stiffness term in Equation (5.9) with the restoring moment function in which the GZ values are obtained by interpolating between

GZ values calculated for wave positions at $L/20$ intervals and roll angles between -70 and 70 degrees. This means that we shall replace $\Delta \phi GM_i$ with ΔGZ_i . By making this replacement we get

$$I_x \frac{d^2 \phi}{dt^2} + B \frac{d\phi}{dt} + \Delta GZ_i = 0 \quad (5.12)$$

We divide both sides of Equation (5.12) by I_x . We get

$$\frac{d^2 \phi}{dt^2} + \frac{B}{I_x} \frac{d\phi}{dt} + \frac{\Delta}{I_x} GZ_i = 0 \quad (5.13)$$

or

$$\ddot{\phi}(t) + \frac{B}{I_x} \dot{\phi}(t) + \frac{\Delta}{I_x} GZ_i(\phi, t) = 0 \quad (5.14)$$

This is the 3rd equation of roll motion.

5.5 Solutions of the Equations of Roll Motion

All the three equations of roll are ordinary 2nd order differential equations. We shall solve these equations numerically by using Runge-Kutta-method (ode45) in MATLAB. As we have explained in Chapter 3.3, we need to transform every equation to a set of two 1st order equations. We do this transformation for every equation separately.

5.5.1 Solution of the 1st Equation

To solve the 1st equation of roll motion, i.e. Equation (5.8), we introduce the following to unknowns,

$$\begin{aligned} y_3(t) &= \phi(t) \\ y_4(t) &= \dot{\phi}(t). \end{aligned} \quad (5.15)$$

By using these to unknowns Equation (5.8) can be written as

$$\begin{aligned} \dot{y}_3(t) &= y_4(t) \\ \dot{y}_4(t) &= -\frac{B}{I_x} y_4(t) - \omega_r^2 (1 + C \cos \omega t) y_3(t). \end{aligned} \quad (5.16)$$

The coefficients in these differential equations must be known and the initial conditions for the unknown functions $y_3(t)$ and $y_4(t)$ in Equations (5.15) must be specified.

5.5.2 Solution of the 2nd Equation

To solve the 2nd equation of roll motion, i.e. Equation (5.11), we introduce the following to unknowns,

$$\begin{aligned} y_5(t) &= \phi(t) \\ y_6(t) &= \dot{\phi}(t). \end{aligned} \quad (5.17)$$

By using these to unknowns Equation (5.11) can be written as

$$\begin{aligned} \dot{y}_5(t) &= y_6(t) \\ \dot{y}_6(t) &= -\frac{B}{I_x} y_6(t) - \frac{\Delta}{I_x} GM_i(t) y_5(t). \end{aligned} \quad (5.18)$$

The coefficients in these differential equations must be known and the initial conditions for the unknown functions $y_5(t)$ and $y_6(t)$ in Equations (5.17) must be specified.

5.5.3 Solution of the 3rd Equation

To solve the 3rd equation of roll motion, i.e. Equation (5.14), we introduce the following to unknowns,

$$\begin{aligned} y_7(t) &= \phi(t) \\ y_8(t) &= \dot{\phi}(t). \end{aligned} \quad (5.19)$$

By using these to unknowns Equation (5.14) can be written as

$$\begin{aligned} \dot{y}_7(t) &= y_8(t) \\ \dot{y}_8(t) &= -\frac{B}{I_x} y_8(t) - \frac{\Delta}{I_x} GZ_i[y_7(t), t]. \end{aligned} \quad (5.20)$$

The coefficients in these differential equations must be known and the initial conditions for the unknown functions $y_7(t)$ and $y_8(t)$ in Equations (5.19) must be specified.

5.6 Estimation of the Coefficients in the Differential Equations

We shall make an estimation of the coefficients in the differential equations given by Equations (5.16), (5.18) and (5.20).

5.6.1 Estimation of the Coefficients in Equations (5.16)

Following is the list of coefficients which we need to estimate for solving the differential equations given by Equations (5.16).

- Mass moment of inertia (I_x)
- Roll angular frequency (ω_r)
- Fractional variation of GM (C)
- Encounter angular frequency (ω)
- Roll damping coefficient (B)

We note that ω_r and C shall be used only in the first equation of roll motion. Now we find these coefficients one by one.

Mass Moment of Inertia (I_x)

We shall find the mass moment of inertia by using the relation

$$\omega_1 = \sqrt{\frac{\Delta GM_0}{I_x}}, \quad (5.21)$$

where,

- ω_1 = roll angular frequency based on the roll period as estimated by Eriksen [7, Chap 5],
- GM_0 = still water GM value based on the GZ value calculated for the still water case,
- Δ = ship displacement.

From Equation (5.21) we get

$$I_x = \frac{\Delta GM_0}{\omega_1^2} \quad (5.22)$$

We are given the value of Δ . We need to calculate GM_0 and ω_1 .

Calculation of GM_0

Equation (2.2) for the still water case becomes,

$$GZ_0 = GM_0 \phi, \quad (5.23)$$

where GZ_0 is the still water GZ value. We shall use GZ_0 value for 10 degrees roll angle calculated by NAPA.

$$\phi = 10 \text{ deg} \times \frac{\pi}{180} = 0.174 \text{ rad}$$

$$GZ_0 \text{ for } 10 \text{ deg} = 0.18 \text{ m}$$

By putting the values of ϕ and GZ_0 in Equation (5.23) we get

$$\begin{aligned} 0.18 &= GM_0 \times 0.174 \\ \Rightarrow GM_0 &= \frac{0.18}{0.174} = 1.032 \text{ m} \end{aligned}$$

Calculation of ω_1

We shall calculate ω_1 with the help of the roll period, T_1 as estimated by Eriksen [7, chap 5]. The reason of using this roll period is that the container ship studied by Eriksen [7] is the same ship we are studying.

$$\text{Roll period, } T_1 = 21.4 \text{ s}$$

Corresponding roll angular frequency is given by

$$\omega_1 = \frac{2\pi}{T_1} = \frac{2\pi}{21.4} = 0.294 \text{ rad s}^{-1}$$

Calculation of I_x

By putting the values of Δ , GM_0 and ω_1 in Equation (5.22) we get

$$\begin{aligned} I_x &= \frac{5.358 \times 10^8 \text{ kg m s}^{-2} \times 1.032 \text{ m}}{(0.294)^2 \text{ s}^{-2}} \\ &= 6.397 \times 10^9 \text{ kg m}^2 \end{aligned}$$

Roll Angular Frequency (ω_r)

We shall find the roll angular frequency with the help of Equation (5.5). By taking square root on both the sides of Equation (5.5) we get,

$$\omega_r = \sqrt{\frac{\Delta GM_m}{I_x}}. \tag{5.24}$$

We have the values of Δ and I_x . We need to calculate GM_m .

Calculation of GM_m

We shall use Equation (5.1) to find GM_m . We need to calculate GM_{cr} and GM_{tr} first. To find these values we shall use the GZ values for crest amidships and trough amidships positions calculated by NAPA. We use Equation (2.2) for 10 degrees roll angle.

$$\phi = 10 \text{ deg} = 0.174 \text{ rad}$$

$$GM_{cr} = \frac{GZ_{cr}}{\phi} = \frac{GZ_{cr}}{0.174} \quad (5.25)$$

$$GM_{tr} = \frac{GZ_{tr}}{\phi} = \frac{GZ_{tr}}{0.174} \quad (5.26)$$

By putting the values of GM_{cr} and GM_{tr} in Equation (5.1) we can find the value of GM_m . In Figure 5.1 GM_{cr} values, GM_{tr} values and GM_m values for case 1 and case 2 are plotted.

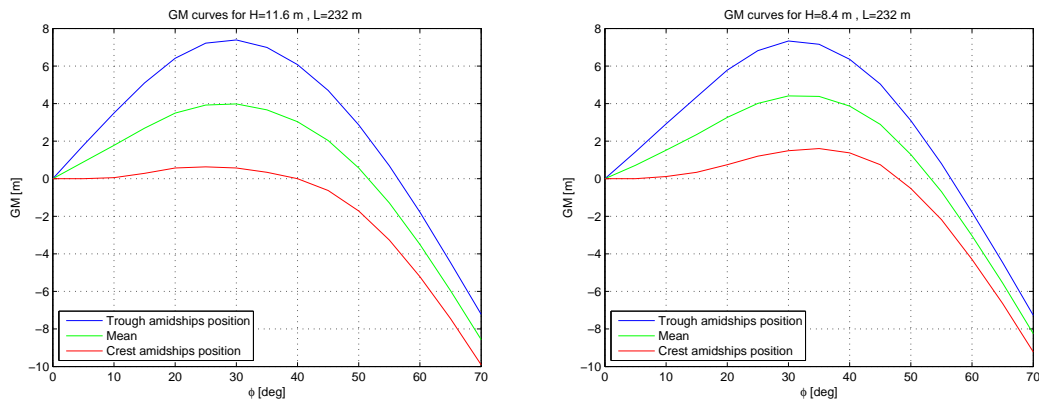


Figure 5.1: (a) GM curves for the case 1

(b) GM curves for the case 2.

We can calculate ω_r by putting the values of Δ , I_x and GM_m in Equation (5.24).

Fractional Variation of GM_m (C)

We shall calculate the value of C with the help of Equations (5.1) and (5.2). By dividing both sides of Equation (5.2) by GM_m we get

$$C = \frac{GM_{cr} - GM_{tr}}{2GM_m} \quad (5.27)$$

By putting Equation (5.1) in Equation (5.27) we get

$$C = \frac{GM_{cr} - GM_{tr}}{GM_{cr} + GM_{tr}} \quad (5.28)$$

The values of GM_{cr} and GM_{tr} can be calculated by using Equations (5.25) and (5.26) respectively.

Encounter Angular Frequency (ω)

The encounter angular frequency, ω , is given by

$$\omega = 2\pi f, \quad (5.29)$$

where f is encounter frequency which is given by

$$f = \frac{V_{rel}}{L}, \quad (5.30)$$

where

- V_{rel} = relative velocity of the ship
- L = wavelength

We are given the value of L , but we need to calculate V_{rel} .

Calculation of V_{rel}

The relative velocity of the ship is the sum of the ship speed, V and the wave velocity, v . We are given the ship speed, V , but we have to calculate the wave velocity, v .

Wave velocity, v , is given by

$$v = \frac{g}{\omega_w}, \quad (5.31)$$

where

- g = acceleration of gravity
- ω_w = wave angular frequency

Wave angular frequency, ω_w , is given by

$$\omega_w = 2\pi f_w, \quad (5.32)$$

where f_w is wave frequency which is given by

$$f_w = \frac{v}{L}, \quad (5.33)$$

where L is the wavelength.

We put Equation (5.33) in Equation (5.32)

$$\omega_w = \frac{2\pi v}{L} \quad (5.34)$$

Now we put Equation (5.34) in Equation (5.31)

$$\begin{aligned} v &= \frac{gL}{2\pi v} \\ \Rightarrow v^2 &= \frac{gL}{2\pi} \\ \Rightarrow v &= \sqrt{\frac{gL}{2\pi}} \end{aligned} \quad (5.35)$$

By putting the values of g and L in Equation (5.35) we can calculate v and then we can find V_{rel} by adding V and v . We can calculate encounter frequency, f , by putting the values of V_{rel} and L in Equation (5.30) and then we can calculate encounter angular frequency, ω by putting the value of f in Equation (5.29).

Roll Damping Coefficient (B)

We can find the roll damping coefficient, B , with the help of Equation (3.8) which gives

$$B = \zeta B_{cr} \quad (5.36)$$

where

$$\begin{aligned} \zeta &= \text{damping ratio,} \\ B_{cr} &= \text{critical damping coefficient.} \end{aligned}$$

According to information received from DNV the damping ratio, ζ , for a container ship may vary from 2 to 10%. At low ship speeds it may be from 2 to 3%, but at

full speed it could be from 5 to 6% for small roll amplitudes, possibly increasing to about 10% for large roll amplitudes

For the critical damping coefficient, B_{cr} , we have found an expression given by Equation (3.10) which is

$$B_{cr} = 2\omega_n I_x, \tag{5.37}$$

where

- I_x = mass moment of inertia,
- ω_n = natural roll angular frequency for the undamped system.

As we have mentioned in Chapter 5.2 that ω_n is based on GM_0 , i.e. still water GM. To make a realistic estimation of B_{cr} we replace ω_n in Equation (5.37) with ω_1 which is the roll angular frequency based on the roll period as estimated by Eriksen [7, chap 5]. The roll angular frequency, ω_1 , is more logical to use because it has been calculated in the actual sea state rather than in the calm water. Equation (5.37) becomes,

$$B_{cr} = 2\omega_1 I_x, \tag{5.38}$$

By putting the values of ω_1 and I_x in Equation (5.38) we get

$$\begin{aligned} B_{cr} &= 2 \times 6.397 \times 10^9 \text{ kg m}^2 \times 0.294 \text{ rad s}^{-1} \\ &= 3.761 \times 10^9 \text{ kg m}^2 \text{ s}^{-1} \end{aligned}$$

5.6.2 Estimation of the Coefficients in Equations (5.18)

We have made the estimation of all the coefficients except $GM_i(t)$ in the differential equations given by Equations (5.18).

Estimation of $GM_i(t)$

$GM_i(t)$ is the time varying function in which the GM values are obtained by interpolating between the real GM values for wave positions at $L/20$ intervals. The GM values used are based on GZ values calculated for 10 degrees roll angle. The GM curves used for interpolation for the case 1 and case 3 are shown in Figure 5.2. The interpolation is done in MATLAB.

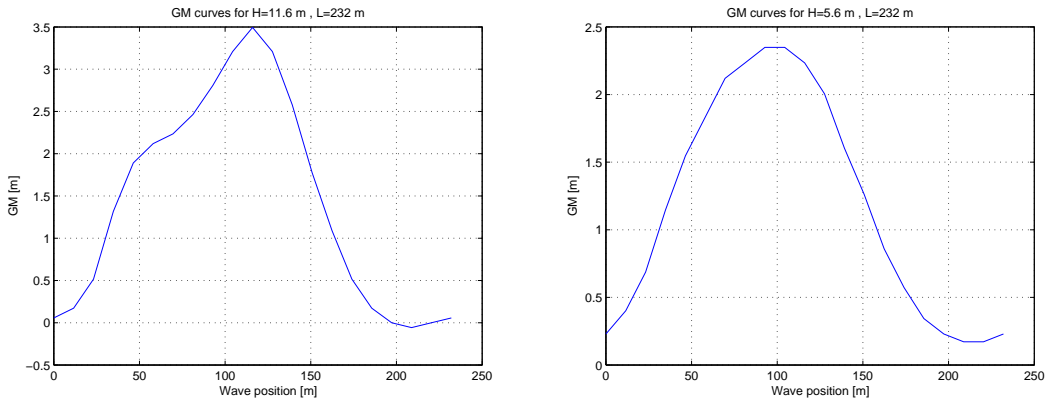


Figure 5.2: (a) GM curves for the case 1

(b) GM curves for the case 3.

5.6.3 Estimation of the Coefficients in Equations (5.20)

We have made the estimation of all the coefficients except $GZ_i[\phi(t), t]$ in the differential equations given by Equations (5.20).

Estimation of $GZ_i[\phi(t), t]$

$GZ_i[\phi(t), t]$ is the restoring moment function in which the GZ values are obtained by interpolating between GZ values calculated for wave positions at $L/20$ intervals and roll angles between -70 and 70 degrees. The GZ mesh used for interpolation for the case 1 is shown in Figure 5.3. The interpolation is done in MATLAB.

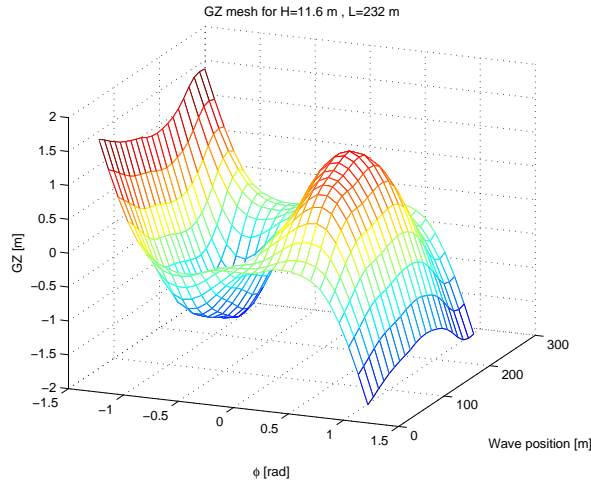


Figure 5.3: GZ mesh for the case 1.

5.7 Initial Wave Position

We shall analyse the dynamic roll behaviour of the container ship for three different initial wave positions. These are

1. Crest amidships position
2. Trough amidships position
3. Zero wave elevation at the FP

5.8 Stability Analysis

We shall find the stability limits of the container ship for all the five combinations of wave heights and wavelengths. For this purpose we need to calculate the stability parameters in the Mathieu equation, i.e. δ and ϵ . Then by using the Ince-Strutt diagram we can find the stability limits. We have defined the parameters δ and ϵ in Equation (3.14). We replace ω_n in Equation (3.14) with ω_r which is based on GM_m rather than GM_0 . We get

$$\delta = \frac{\omega_r^2}{\omega^2}, \quad \epsilon = C\delta \quad (5.39)$$

The encounter angular frequency, ω , depends on the relative velocity of the ship, V_{rel} . Thus for different values of V_{rel} we get different encounter angular frequencies, and corresponding to these frequencies we get different pairs of δ and ϵ . Each pair of δ and ϵ represents a point on the Ince-Strutt diagram. If this point lies in the stable region then the roll motion will be stable and if it lies in the unstable region then the roll motion will be unstable.

Chapter 6

Results

We present the results obtained from the three calculation methods separately.

6.1 Results for the 1st Method

The values of GM_{cr} , GM_{tr} , GM_m and ω_r calculated for all the five cases are given in Table 6.1. We note that we can calculate ω_r only for the 1st method.

Table 6.1: GM values and roll angular frequency.

Case	GM_{cr} [m]	GM_{tr} [m]	GM_m [m]	ω_r [rad s ⁻¹]
1	0.057	3.495	1.776	0.385
2	0.115	2.922	1.518	0.356
3	0.229	2.234	1.232	0.321
4	0.630	1.776	1.203	0.317
5	1.604	0.974	1.289	0.328

We have found the values of δ and ϵ corresponding to different relative velocities of the ship. These pairs of δ and ϵ give us different points on the Ince-Strutt diagram. We present the Ince-Strutt diagrams showing these points for each case separately. We also present the results obtained from the stability analysis for these cases.

6.1.1 Case 1

For case 1 we obtain the points on the line in the Ince-Strutt diagram shown in Figure 6.1.

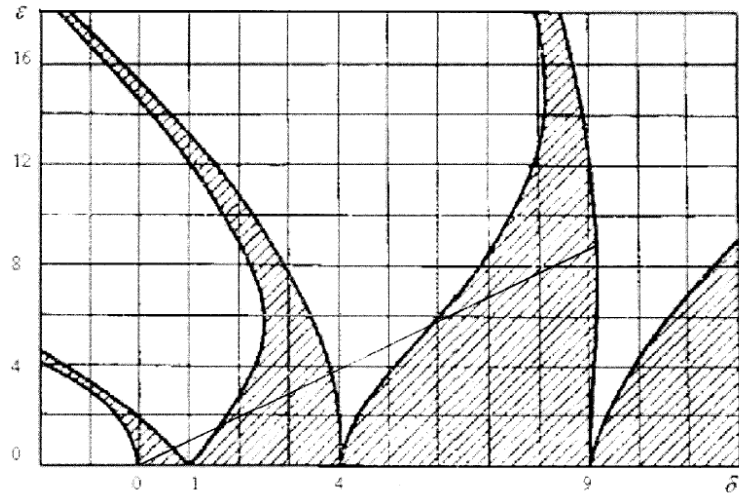


Figure 6.1: Stability for case 1.

We have done the stability analysis for the relative velocities between 5 m/s and 50 m/s . All the velocities more than 50 m/s have a stable response because these represent the points which lie in the bounded region between $\delta = 0$ and $\delta = 1$ in the Ince-Strutt diagram. We present the data for case 1 in the first two instability regions in Table 6.2.

Table 6.2: Data in the instability regions for case 1.

Instability region	$V_{rel} [m/s]$		$\omega [rad/s]$		$\frac{\omega_r}{\omega}$	
	min	max	min	max	min	max
1st	21.54	35.0	0.583	0.948	0.660	0.407
2nd	11.69	14.75	0.317	0.40	1.217	0.965

We note that at $V_{rel} = 18 [m/s]$ lies in the stable region and $V_{rel} = 28 [m/s]$ lies in the unstable region. We have checked the stability on these points and some other points in MATLAB. The MATLAB figures are attached in appendix . We present the detail of these runs in Table.

Table 6.3: Detail of the runs for case 1 in MATLAB.

Run	V_{rel} [m/s]	ϕ_0 [deg]	$\dot{\phi}_0$ [deg/s]	ζ
1	18	2	0	0
2	18	0	0.85	0
3	18	4	0	0
4	18	0	1.54	0
5	28	2	0	0
6	28	0	0.85	0
7	28	4	0	0
8	28	0	1.54	0
9	18	4	0	0.1
10	28	4	0	0.1
11	18	2	0	0
12	18	4	0	0
13	18	2	0	0
14	18	4	0	0
15	28	2	0	0
16	28	4	0	0
17	28	2	0	0
18	28	4	0	0
19	34.5	4	0	0
20	34.5	4	0	0.1
21	21.53	4	0	0
22	21.54	4	0	0
23	35.2	4	0	0
24	35.2	8	0	0

6.1.2 Case 2

For case 2 we obtain the points on the line in the Ince-Strutt diagram shown in Figure 6.2.

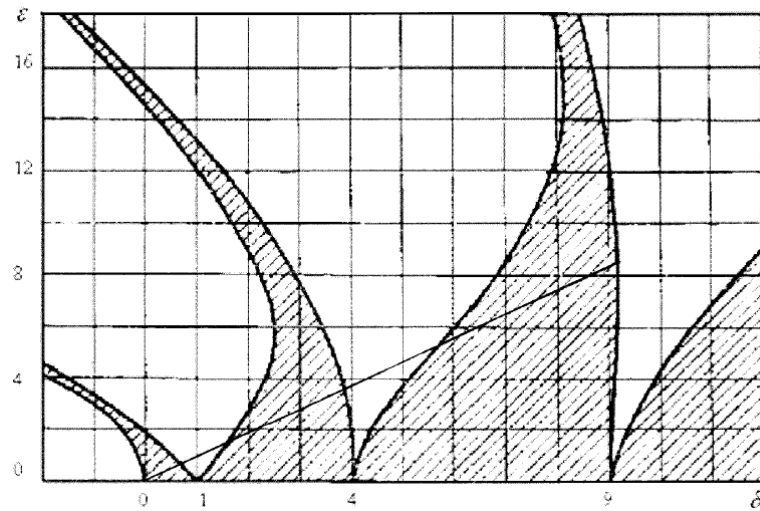


Figure 6.2: Stability for case 2.

We present the data for case 2 in the first two instability regions in Table 6.4.

Table 6.4: Data in the instability regions for case 2.

Instability region	V_{rel} [m/s]		ω [rad/s]		$\frac{\omega_r}{\omega}$	
	min	max	min	max	min	max
1st	20.16	32.3	0.546	0.875	0.407	0.652
2nd	10.96	13.63	0.297	0.369	0.965	1.20

We note that the first instability region has become smaller and the second instability region has moved a little bit to the right.

6.1.3 Case 3

For case 3 we obtain the points on the line in the Ince-Strutt diagram shown in Figure 6.3.

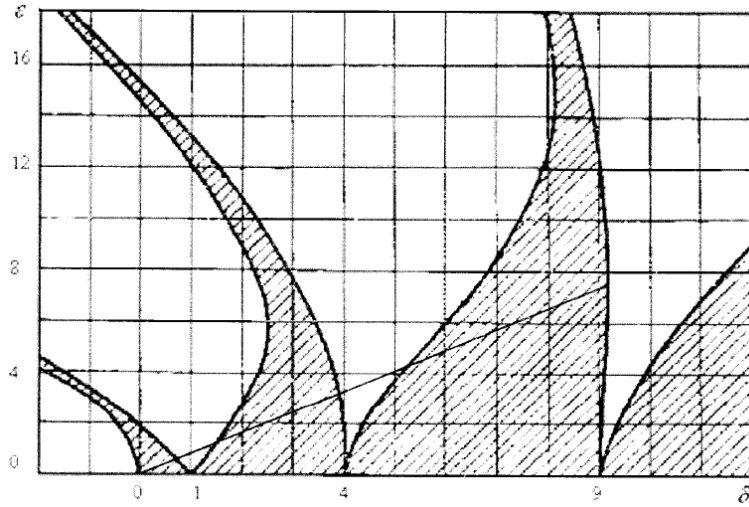


Figure 6.3: Stability for case 3.

We present the data for for case 3 in the first two instability regions in Table 6.5.

Table 6.5: Data in the instability regions for the case 3.

Instability region	V_{rel} [m/s]		ω [rad/s]		$\frac{\omega_r}{\omega}$	
	min	max	min	max	min	max
1st	18.78	28.5	0.509	0.772	0.416	0.631
2nd	10.26	12.19	0.278	0.330	0.972	1.155

We note that the first instability region has become more smaller and the second instability region has moved a little bit to the right.

6.1.4 Case 4 and case 5

The Ince-Strutt diagrams for case 4 and case 5 are shown in Figures 6.4 and 6.5 respectively.

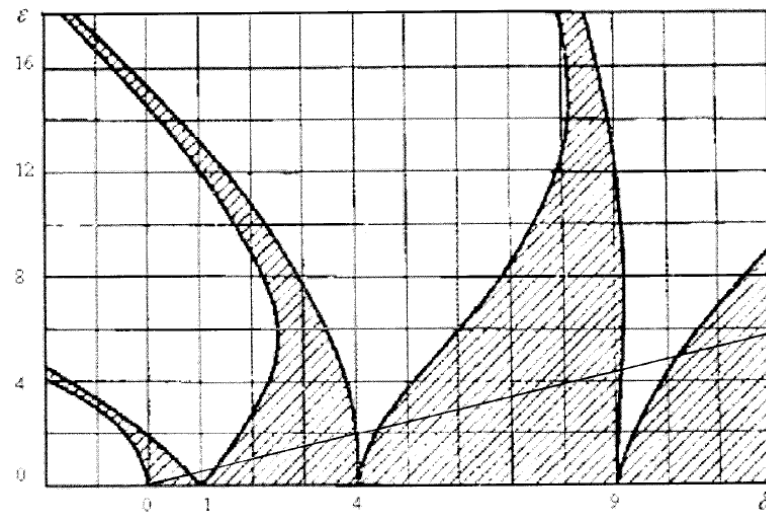


Figure 6.4: Stability for case 4.

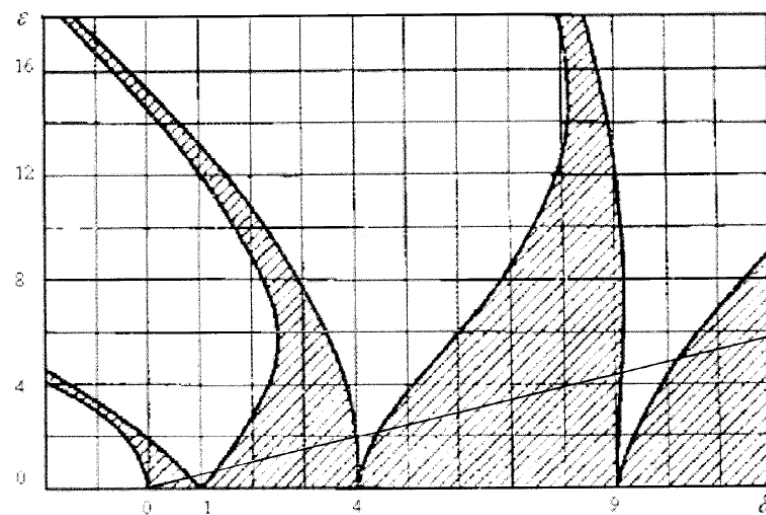


Figure 6.5: Stability for case 5.

We note from the Ince-Strutt diagrams for case 4 and case 5 that the first instability region has become more smaller than the previous case. the second instability region has moved a little bit to the right.

6.2 Results for the 2nd Method

The trend of the results obtained by using the 2nd method is the same as for the 1st method.

6.3 Results for the 3rd Method

The trend of the results obtained by using the 3rd method is the same as for the 1st method. The only difference is that a change in the initial wave position effects the roll response more than that in the first two methods.

Chapter 7

Conclusions

As we have mentioned in the previous chapter that the overall trend of the dynamic roll behaviour is the same for the three methods. We can conclude the following things from the results of the stability analysis.

1. In the MATLAB runs from 1 to 8 we conclude that the roll response is independent of the initial conditions.
2. From the run 9 and 10 it is obvious that roll motion of the stable solution dies after some time and that of an unstable solution becomes stable for just a few seconds.
3. The runs 11, 12, 15, and 16 are for the trough amidships position and the runs 13, 14, 17 and 18 are for the zero wave elevation condition from the FP. From these runs we conclude that the initial wave position does not effect the stability of the roll response.
4. In the run 19 the roll response for a point having unstable response is shown. In the run 20 this response becomes stable with a damping ratio 0.1.
5. The runs 23 and 24 show that a greater initial roll angle can change a stable response to an unstable response.

Appendix A

MATLAB Runs

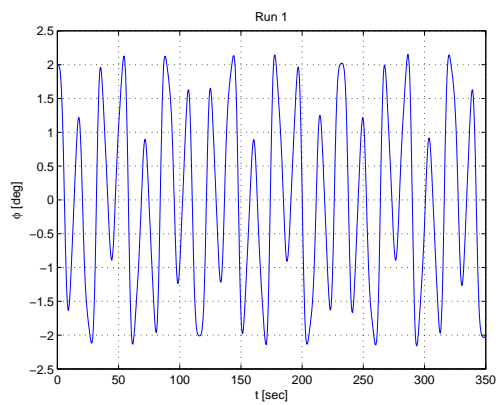
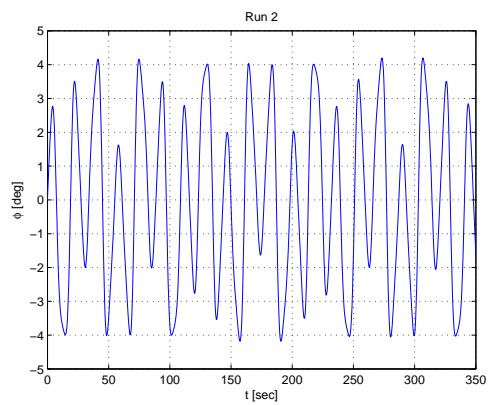


Figure A.1: (a) Run 1



(b) Run 2.

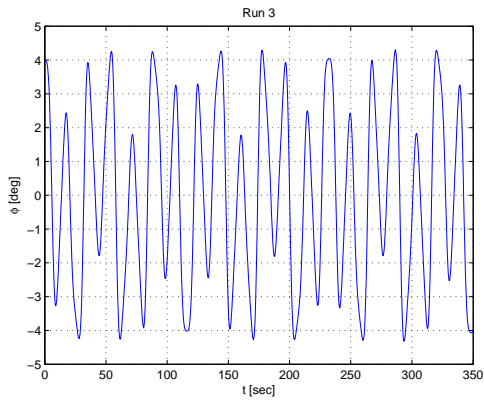
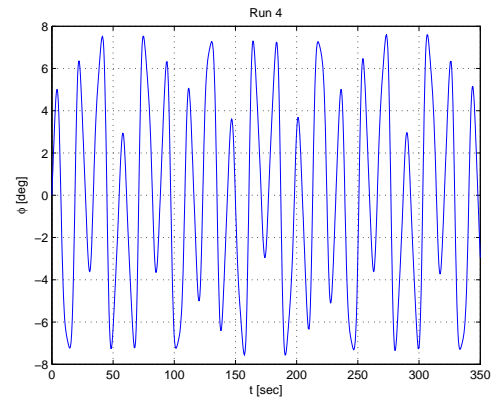


Figure A.2: (a) Run 3



(b) Run 4.

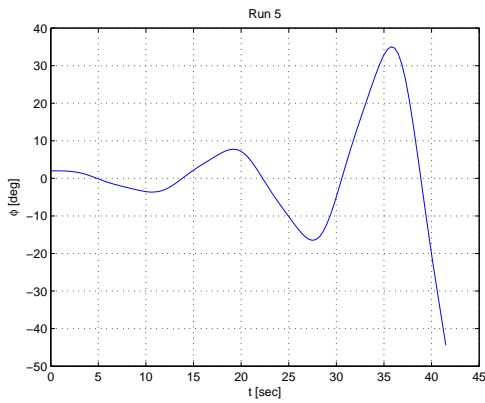
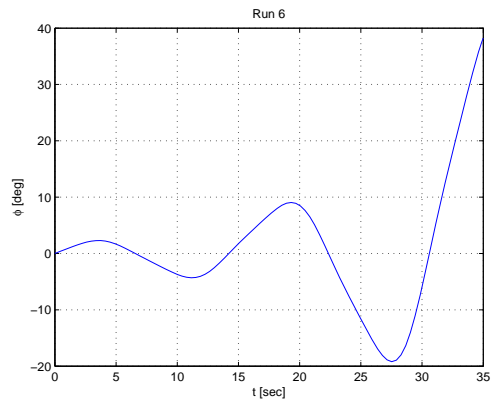


Figure A.3: (a) Run 5



(b) Run 6.

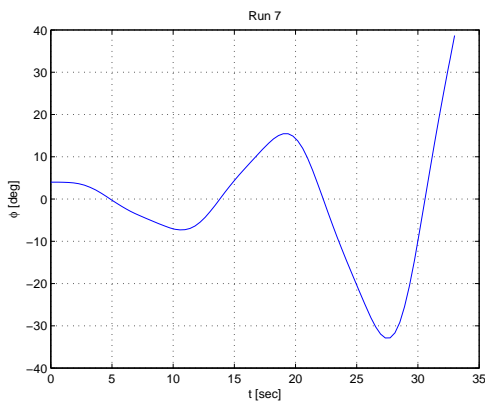
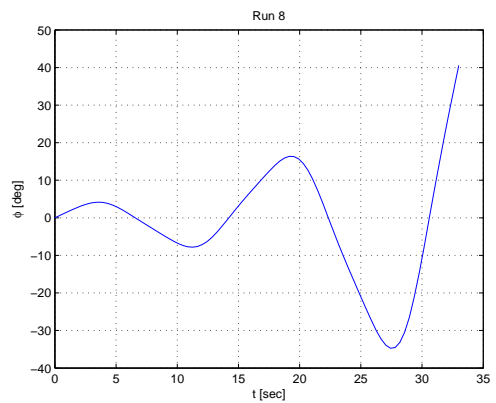


Figure A.4: (a) Run 7



(b) Run 8.

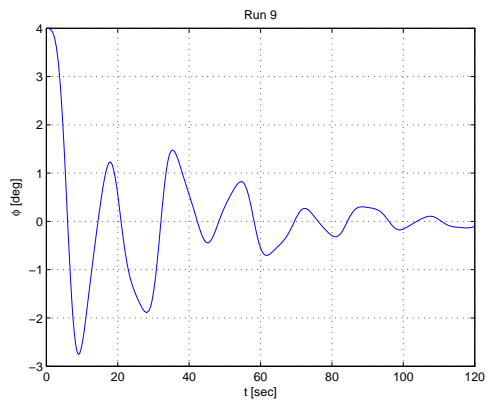
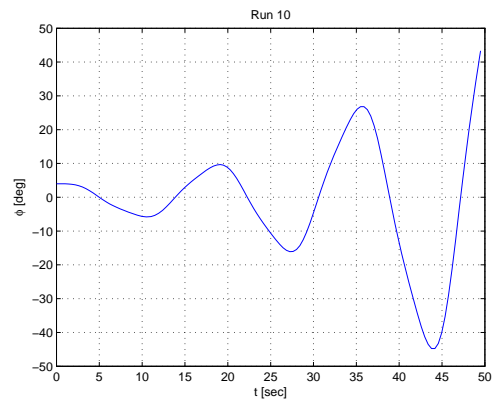


Figure A.5: (a) Run 9



(b) Run 10.

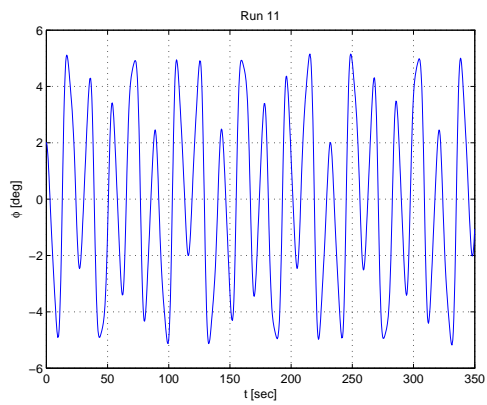
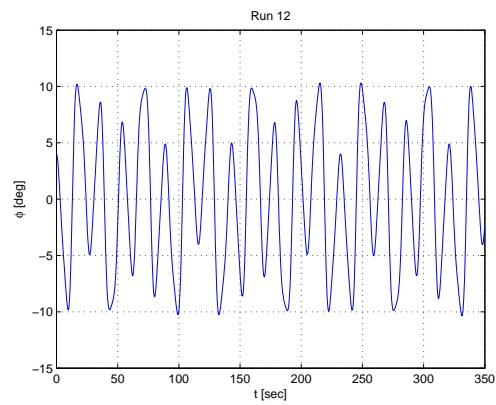


Figure A.6: (a) Run 11



(b) Run 12.

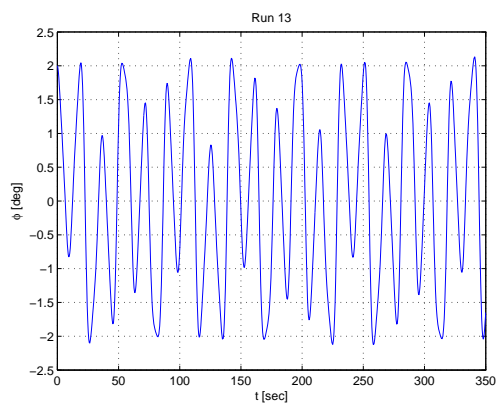
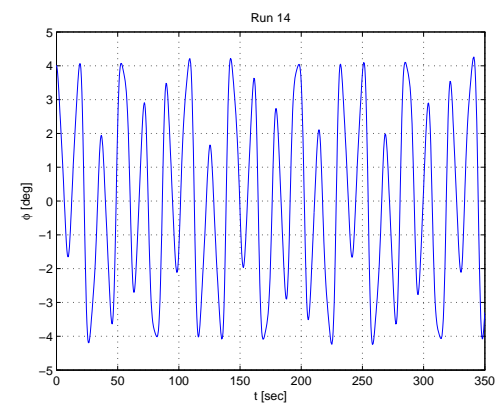


Figure A.7: (a) Run 13



(b) Run 14.

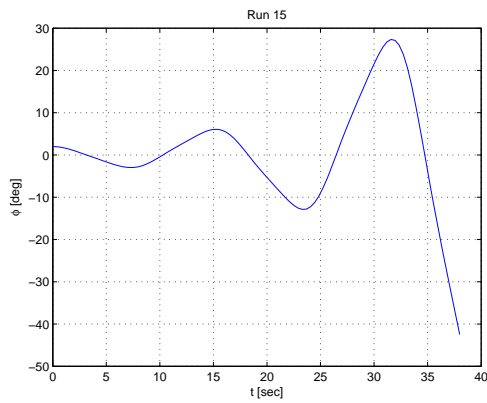
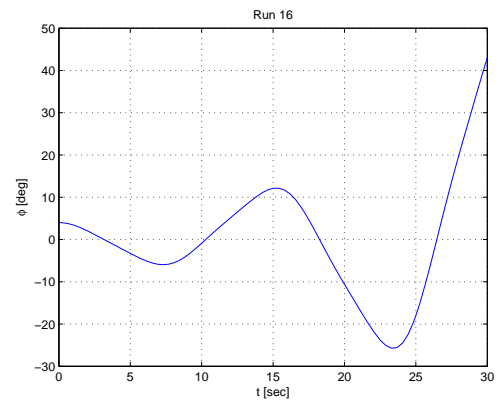


Figure A.8: (a) Run 15



(b) Run 16.

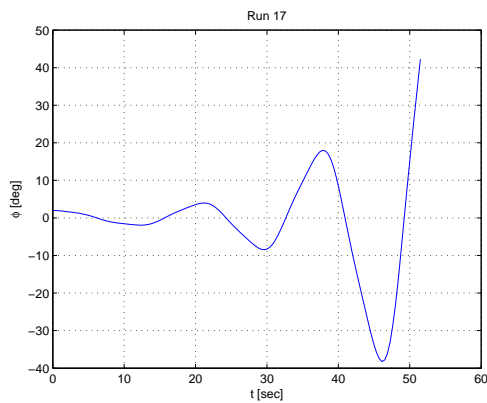
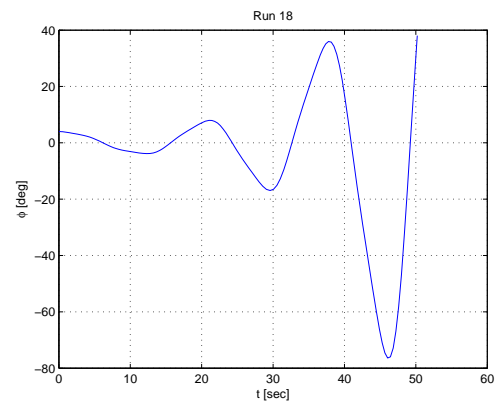


Figure A.9: (a) Run 17



(b) Run 18.

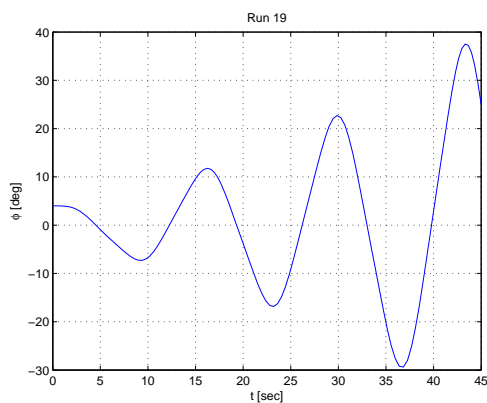
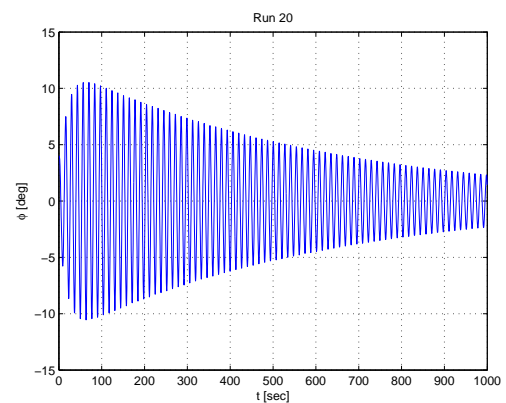


Figure A.10: (a) Run 19



(b) Run 20.

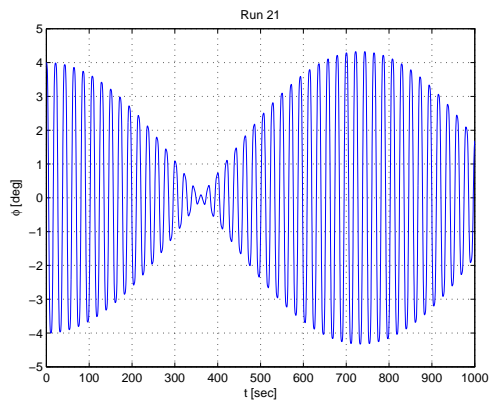
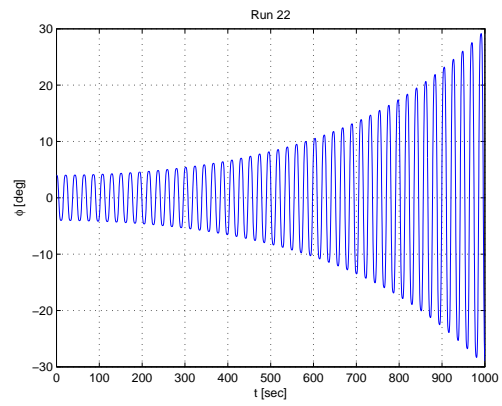


Figure A.11: (a) Run 21



(b) Run 22.

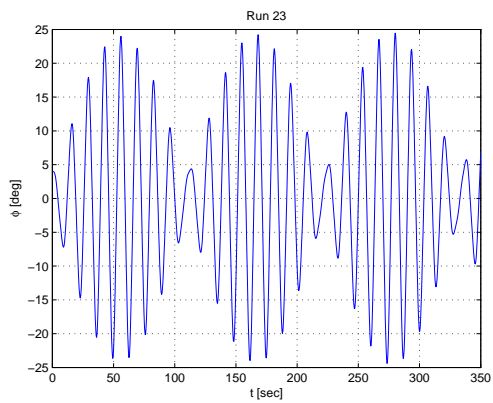
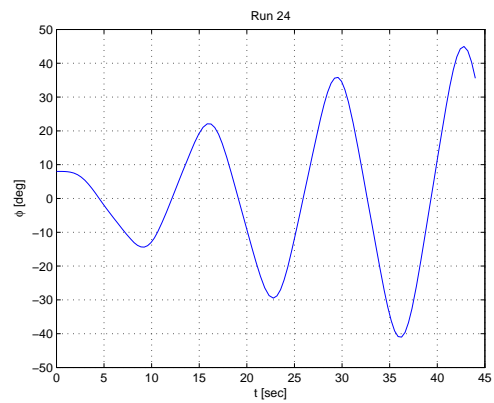


Figure A.12: (a) Run 23



(b) Run 24.

Appendix A

MATLAB Input Files

```
% GM_mean.m %
```

```
clear all
```

```
V_rel = 31;
```

```
L=232;
```

```
GZ_cr = 0.01;
```

```
% 1) H = 11.6 m, L = 232 m
```

```
GZ_tr = 0.61;
```

```
% L=232;
```

```
% GZ_cr = 0.02;
```

```
% 2) H = 8.4 m, L = 232 m
```

```
% GZ_tr = 0.51;
```

```
% L=232;
```

```
% GZ_cr = 0.04;
```

```
% 3) H = 5.6 m, L = 232 m
```

```
% GZ_tr = 0.39;
```

```
% L=155;
```

```
% GZ_cr = 0.11;
```

```
% 4) H = 5.6 m, L = 155 m
```

```
% GZ_tr = 0.31;
```

```
% L=116;
```

```
% GZ_cr = 0.28;
```

```
% 5) H = 5.6 m, L = 116 m
```

```
% GZ_tr = 0.17;
```

```
phi_10 = deg2rad(10);
GM_cr = GZ_cr/phi_10;
GM_tr = GZ_tr/phi_10 ;

GM_m = 1/2*(GM_cr+GM_tr);

%CGM_m= 1/2*(GM_cr-GM_tr);

% weight of ship

M=54616;
g=9.81;
Delta = M*1000*g;

% roll frequency

Tg = 21.4;
wg = 2*pi/Tg;

% moment of inertia
GZ_0 = 0.18;
GM_0 = GZ_0/phi_10;

Ix = Delta*GM_0/wg^2;           % we find Ix with the help of wg & GM_0

wr = sqrt(Delta*GM_m/Ix);     % roll frequency

B_cr = 2*wg*Ix;

C = (GM_cr-GM_tr)/(GM_cr+GM_tr);

v=sqrt((g*L)/(2*pi));         % wave velocity
f = V_rel/L;
w = 2*pi*f;                   % Encounter angular frequency

delta=wr^2./w.^2;
r=sqrt(delta);
```

```
epsilon=C*delta;

delta_s= delta*4;
epsilon_s = C*delta_s;
epsilon_s2=epsilon_s/2;

plot(delta_s, epsilon_s2,'r')
grid on
xlabel('delta')
ylabel('epsilon')
% title ('Delta-epsilon curve for the container ship')

% GM_interp.m %

% interpolation of GM

clear all

t=0:30;
V_rel=30;
L_pp=232;

L=232;
S_input= load ('WavePosition_232.txt');
%GZ = load ('GZ_11_6_232.txt');
%GZ = load ('GZ_8_4_232.txt');
  GZ = load ('GZ_5_6_232.txt');

% L=155;
% S_input= load ('WavePosition_155.txt');
% GZ = load ('GZ_5_6_155.txt');

% L=116;
% S_input= load ('WavePosition_116.txt');
% GZ = load ('GZ_5_6_116.txt');
%a=V_rel.*t;

S= (L_pp-L)/2- V_rel*t;           % crest amidships position at t=0
```

```
%S= L_pp/2- V_rel*t;           % trough amidships position at t=0

%S= L_pp-L/4- V_rel*t;         % zero wave elevation at FP at t=0

e=numel(S);

for k=1:1:e

    while S(k) < 0

        S(k) = S(k) + L;
    end

    while S(k) > L

        S(k) = S(k) - L;

    end

end

phi_10=deg2rad(10);
GM= GZ/phi_10;

GMi = interp1(S_input,GM,S,'spline');

plot (S_input,GM)

%plot (S,GMi,'o',S_input,GM,'r')
grid on

xlabel('Wave position [m]')
ylabel('GM [m]')
title('GM curves for H=5.6 m , L=232 m')

% weight of ship

M=54616;
g=9.81;
Delta = M*1000*g;
```

```
% roll frequency

Tg = 21.4;
wg = 2*pi/Tg;

% moment of inertia
GZ_0 = 0.18;
GM_0 = GZ_0/phi_10;

Ix = Delta*GM_0/wg^2;

% roll frequency

wr = sqrt(Delta*GMi/Ix);
c_cr = 2*Ix*wr;

%c= c_cr*zeta;

% Encounter angular frequency
f = V_rel/L;          % wave frequency
w = 2*pi*f;
Tw = 1./f;
r=wr/w;

%% GZ_interp.m

% interpolation of GZ

clear all

t=0:0.5:15;
V_rel=30;
L_pp=232;

L=232;
S_input= load ('WavePosition_232.txt');
import_GZ_table GZ_table_11_6_232.xls;
%import_GZ_table GZ_table_8_4_232.xls;
```

```
%import_GZ_table GZ_table_5_6_232.xls;

% L=155;
% S_input= load ('WavePosition_155.txt');
% import_GZ_table GZ_table_5_6_155.xls;

% L=116;
% S_input= load ('WavePosition_116.txt');
% import_GZ_table GZ_table_5_6_116.xls;

S= (L_pp-L)/2- V_rel*t;

%S= L_pp/2- V_rel*t;

%S= L_pp-L/4- V_rel*t;

e=numel(S);

for k=1:1:e

    while S(k) < 0

        S(k) = S(k) + L;
    end

    while S(k) > L

        S(k) = S(k) - L;

    end

end

phi = -70:5:70;

X = deg2rad(phi);
Y = S_input';
GZ = Ark1;
```

```
Xi = -1.23:0.05:1.23;
Yi = S';

GZi = interp2(X,Y,GZ,Xi,Yi,'spline');

mesh (X,Y,GZ)
hold on
%mesh (Xi,Yi,GZi+3), hold

% surf (X,Y,GZ)
% hold on
% surf (Xi,Yi,GZi+3)
% shading interp;
% colormap(summer)

xlabel('ϕ [rad]')
ylabel('Wave position [m]')
zlabel ('GZ [m]')
title('GM curves for H=5.6 m , L=232 m')

% mathieu.m %

k = menu('Choose a plot','Plot between roll angle and tau','Plot between phi and ph

if k==1          % plot between roll angle and tau

    tf_m= input('Enter the final time [sec]: ');
    tspan_m= [0, tf_m];
    w_m= 1;
    tauspan_m = w_m*tspan_m;
    ip= input('Enter the initial roll angle [deg]: ');
    iv= input('Enter the initial roll velocity: ');
    ip_r=deg2rad(ip);
    iv_r=deg2rad(iv);
    y0_m= [ip_r, iv_r];
    delta_m= input('Enter the coefficient δ: ');
    epsilon_m= input('Enter the coefficient ε: ');
    zeta_m= input('Enter the damping ratio ζ: ');
```

```

    [tau_m, y_m] = ode45(@sysfil_mathieu,tauspan_m,y0_m,[],delta_m,epsilon_m,zeta_m);

    size(tau_m);
    size(y_m);

    y=rad2deg(y_m);

    plot(tau_m, y(:,1))
    grid on
    xlabel('τ')
    ylabel('φ [deg]')
    title(['Solution of the Mathieu equation when δ = ',num2str(delta_m), ', ε = '])

else % Plot between phi and phidot

    plot(y(:,1), y(:,2))
    grid on
    xlabel('φ [deg]')
    ylabel('')
    title ('Phase plane plot')

end

% sysfil_mathieu.m %

function ydot=sysfil_mathieu(tau,y,delta,epsilon,zeta)

ydot=[y(2); -2*zeta*sqrt(delta)*y(2)-(delta + epsilon*cos(tau))*y(1)];

% test4a.m %

function varargout = test4a(varargin)
% TEST4 M-file for test4.fig
%     TEST4, by itself, creates a new TEST4 or raises the existing
%     singleton*.
%
%     H = TEST4 returns the handle to a new TEST4 or the handle to

```

```
% the existing singleton*.
%
% TEST4('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in TEST4.M with the given input arguments.
%
% TEST4('Property','Value',...) creates a new TEST4 or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before test4_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to test4_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help test4

% Last Modified by GUIDE v2.5 16-Mar-2008 19:42:09

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',      mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @test4_OpeningFcn, ...
                  'gui_OutputFcn',  @test4_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin1)
    gui_State.gui_Callback = str2func(varargin1);
end

if nargout
    [varargout1:nargout] = gui_mainfcn(gui_State, varargin:);
else
    gui_mainfcn(gui_State, varargin:);
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before test4 is made visible.
function test4_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to test4 (see VARARGIN)

% Choose default command line output for test4
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes test4 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = test4_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout1 = handles.output;

function tf4_input_Callback(hObject, eventdata, handles)
% hObject    handle to tf4_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tf4_input as text
%        str2double(get(hObject,'String')) returns contents of tf4_input as a double
```

```
% --- Executes during object creation, after setting all properties.
function tf4_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tf4_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundcolor'))
    set(hObject,'BackgroundColor','white');
end

function iv4_input_Callback(hObject, eventdata, handles)
% hObject    handle to iv4_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of iv4_input as text
%       str2double(get(hObject,'String')) returns contents of iv4_input as a double

% --- Executes during object creation, after setting all properties.
function iv4_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to iv4_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundcolor'))
    set(hObject,'BackgroundColor','white');
end

function zeta4_input_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to zeta4_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of zeta4_input as text
%         str2double(get(hObject,'String')) returns contents of zeta4_input as a c

% --- Executes during object creation, after setting all properties.
function zeta4_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to zeta4_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgro
    set(hObject,'BackgroundColor','white');
end

function ip4_input_Callback(hObject, eventdata, handles)
% hObject    handle to zeta4_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of zeta4_input as text
%         str2double(get(hObject,'String')) returns contents of zeta4_input as a c

% --- Executes during object creation, after setting all properties.
function ip4_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to zeta4_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackground...
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in plot4_pushbutton1.
function plot4_pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to plot4_pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Get user input from GUI
tf4 = str2double(get(handles.tf4_input,'String'));
t4span = [0, tf4];

```

```

ip = str2double(get(handles.ip4_input,'String'));
iv = str2double(get(handles.iv4_input,'String'));

```

```

ip4=deg2rad(ip);
iv4=deg2rad(iv);
y0_4= [ip4 ; iv4];

```

```

B_cr = 3.761e9 ;
zeta= str2double(get(handles.zeta4_input,'String'));
B4=B_cr*zeta;

```

```

[t4, y4] = ode45(@test4_system,t4span,y0_4,[],B4);

```

```

% plot between roll angle and tau

```

```

size(t4);
size(y4);
y=rad2deg(y4);

```

```

axes(handles.t_phi_axes)
plot(t4, y(:,1))
grid on
xlabel('t [sec]')

```

```
ylabel('ϕ [deg]')
title('Solution with sinusoidally interpolated GM')

% Plot between phi and phidot

axes(handles.phi_phidot_axes)
plot(y(:,1), y(:,2))
grid on
xlabel('ϕ [deg]')
ylabel(' [deg/s]')
title ('Phase plane plot')

% test4b.m %

clear all

k = menu('Choose a plot','Plot between roll angle and tau','Plot between phi and

if k==1

    % plot between roll angle and tau

    tf =input('Enter the final time [sec]: ');
    tspan= [0, tf];

    ip= input('Enter the initial roll angle [deg]: ');
    iv= input('Enter the initial roll velocity [deg/s]: ');

    ip_r=deg2rad(ip);
    iv_r=deg2rad(iv);

    y0= [ip_r, iv_r];

    B_cr = 3.761e9 ;
    zeta= input('Enter the damping ratio, zeta: ');

    B=B_cr*zeta;

    [t, y] = ode45(@test4_system,tspan,y0,[],B);
```

```
size(t);
size(y);

y_d=rad2deg(y);

plot(t, y_d(:,1))
grid on

xlabel('t [sec]')
ylabel('ϕ [deg]')
title('Run 1')

else

% Plot between phi and phidot

plot(y_d(:,1), y_d(:,2))
grid on

xlabel('ϕ [deg]')
ylabel(' [deg/s]')
title ('Phase plane plot')

end

% test4_system.m %

function ydot=test4_system(t,y,B)

V_rel = 28;

L=232;
GZ_cr = 0.01; % 1) H = 11.6 m, L = 232 m
GZ_tr = 0.61;

% L=232;
% GZ_cr = 0.02; % 2) H = 8.4 m, L = 232 m
```

```

% GZ_tr = 0.51;

% L=232;
% GZ_cr = 0.04;
% GZ_tr = 0.39;

% L=155;
% GZ_cr = 0.11;
% GZ_tr = 0.31;

% L=116;
% GZ_cr = 0.28;
% GZ_tr = 0.17;

phi_10 = deg2rad(10);
GM_cr = GZ_cr/phi_10;
GM_tr = GZ_tr/phi_10;

% moment of inertia, Ix

M=54616;
g=9.81;
Delta = M*1000*g;

Tg = 21.4;
wg = 2*pi/Tg;

GZ_0 = 0.18;
GM_0 = GZ_0/phi_10;

Ix = Delta*GM_0/wg^2;

% roll frequency, wr

GM_m = 1/2*(GM_cr+GM_tr);
wr = sqrt(Delta*GM_m/Ix);

C = (GM_cr-GM_tr)/(GM_cr+GM_tr);

```

% 3) H = 5.6 m, L = 232 m
% 4) H = 5.6 m, L = 155 m
% 5) H = 5.6 m, L = 116 m
% 10 deg converted to radians
% GM for the crest amidships position
% GM for the trough amidships position
% mass of ship
% ship displacement
% roll period given in Eriksen [7, Ch. 7]
% corresponding roll frequency% given
% GZ for still water
% GM for still water
% Mean GM
% fractional variation in mean GM

```

% Encounter angular frequency

f = V_rel/L;                                % wave frequency
w = 2*pi*f;
ydot=[y(2); -B/Ix*y(2)- wr^2*(1+ C*cos(w*t))*y(1)];      % crest amidships po
%ydot=[y(2); -B/Ix*y(2)- wr^2*(1+ C*cos(w*t+pi))*y(1)];  % trough amidships
%ydot=[y(2); -B/Ix*y(2)- wr^2*(1+ C*cos(w*t+3/2*pi))*y(1)]; % zero wave elevati

% test5a.m %

function varargout = test5a(varargin)
% TEST5 M-file for test5.fig
%     TEST5, by itself, creates a new TEST5 or raises the existing
%     singleton*.
%
%     H = TEST5 returns the handle to a new TEST5 or the handle to
%     the existing singleton*.
%
%     TEST5('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in TEST5.M with the given input arguments.
%
%     TEST5('Property','Value',...) creates a new TEST5 or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before test5_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to test5_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help test5

% Last Modified by GUIDE v2.5 16-Mar-2008 20:00:19

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',      mfilename, ...

```

```

        'gui_Singleton', gui_Singleton, ...
        'gui_OpeningFcn', @test5_OpeningFcn, ...
        'gui_OutputFcn', @test5_OutputFcn, ...
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin1)
    gui_State.gui_Callback = str2func(varargin1);
end

if nargout
    [varargout1:nargout] = gui_mainfcn(gui_State, varargin:);
else
    gui_mainfcn(gui_State, varargin:);
end
% End initialization code - DO NOT EDIT

% --- Executes just before test5 is made visible.
function test5_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to test5 (see VARARGIN)

% Choose default command line output for test5
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes test5 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = test5_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
varargout1 = handles.output;
```

```
function tf5_input_Callback(hObject, eventdata, handles)
```

```
% hObject handle to tf5_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of tf5_input as text
% str2double(get(hObject,'String')) returns contents of tf5_input as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function tf5_input_CreateFcn(hObject, eventdata, handles)
```

```
% hObject handle to tf5_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackground...
    set(hObject,'BackgroundColor','white');
end
```

```
function ip5_input_Callback(hObject, eventdata, handles)
```

```
% hObject handle to ip5_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of ip5_input as text
```

```
% str2double(get(hObject,'String')) returns contents of ip5_input as a double
```

```

% --- Executes during object creation, after setting all properties.
function ip5_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ip5_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgro
    set(hObject,'BackgroundColor','white');
end

```

```

function iv5_input_Callback(hObject, eventdata, handles)
% hObject    handle to iv5_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of iv5_input as text
%       str2double(get(hObject,'String')) returns contents of iv5_input as a double

```

```

% --- Executes during object creation, after setting all properties.
function iv5_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to iv5_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgro
    set(hObject,'BackgroundColor','white');
end

```

```

function c3_input_Callback(hObject, eventdata, handles)

```

```
% hObject    handle to c3_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c3_input as text
%         str2double(get(hObject,'String')) returns contents of c3_input as a double

% --- Executes during object creation, after setting all properties.
function c3_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c3_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in plot5_pushbutton1.
function plot5_pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to plot5_pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get user input from GUI
tf5 = str2double(get(handles.tf5_input,'String'));
t5span = [0, tf5];

ip = str2double(get(handles.ip5_input,'String'));
iv = str2double(get(handles.iv5_input,'String'));

ip5=deg2rad(ip);
iv5=deg2rad(iv);

y0_5= [ip5 ; iv5];
```

```
B_cr = 3.761e9 ;
zeta = str2double(get(handles.c3_input,'String'));
B5=B_cr*zeta;

S_input= (0:1:20)*11.6;           % Wave positions for L=232
%S_input= (0:1:20)*7.75;         % Wave positions for L=155
%S_input = (0:1:20)*5.8;         % Wave positions for L=116

%GZ = load ('GZ_11_6_232.txt');
GZ = load ('GZ_8_4_232.txt');
%GZ = load ('GZ_5_6_232.txt');

%GZ = load ('GZ_5_6_155.txt');
%GZ = load ('GZ_5_6_116.txt');

[t5, y5] = ode45(@test5_system,t5span,y0_5,[],B5,S_input,GZ);

% plot between roll angle and tau

size(t5);
size(y5);
y=rad2deg(y5);

axes(handles.t_phi_axes)
plot(t5, y(:,1))
grid on
xlabel('t [sec]')
ylabel('phi [deg]')
title('Solution with real interpolated GM')

% Plot between phi and phidot

axes(handles.phi_phidot_axes)
plot(y(:,1), y(:,2))
grid on
xlabel('phi')
ylabel('phi dot')
```

```
title ('Phase plane plot')

%% test5b.m

clear all

k = menu('Choose a plot','Plot between roll angle and tau','Plot between phi and ph

if k==1

    % plot between roll angle and tau

    tf =input('Enter the final time [sec]: ');
    tspan= [0, tf];

    ip= input('Enter the initial roll angle [deg]: ');
    iv= input('Enter the initial roll velocity [deg/s]: ');

    ip_r=deg2rad(ip);
    iv_r=deg2rad(iv);

    y0= [ip_r, iv_r];

    B_cr = 3.761e9 ;
    zeta= input('Enter the damping ratio, zeta: ');

    B=B_cr*zeta;

    [t, y] = ode45(@test5_system,tspan,y0,[],B);

    size(t);
    size(y);

    y_d=rad2deg(y);

    plot(t, y_d(:,1))
    grid on

    xlabel('t [sec]')
```

```
    ylabel('φ [deg]')
    title('Run 000')

else
    % Plot between phi and phidot
    plot(y_d(:,1), y_d(:,2))
    grid on
    xlabel('φ [deg]')
    ylabel(' [deg/s]')
    title ('Phase plane plot')

end

% test5_system.m %

function ydot=test5_system(t,y,B,S_input,GZ)

phi_10 = deg2rad(10);

V_rel=28;

L_pp=232;

L=232;
%L=155;
%L=116;

S= (L_pp-L)/2- V_rel*t;           % crest amidships position

%S= L_pp/2- V_rel*t;             % trough amidships position

%S= L_pp-L/4- V_rel*t;           % zero wave elevation at the FP

while S < 0

    S = S + L;

end
```

```
while S > L

    S = S - L;

end

GM = GZ/phi_10;

GMi = spline(S_input,GM,S);

% moment of inertia, Ix

M=54616;           % mass of ship
g=9.81;
Delta = M*1000*g;  % weight of the ship

Tg = 21.4;         % roll period given in Eriksen [7, Sec5
wg = 2*pi/Tg;      % corresponding roll frequency% given r

GZ_0 = 0.18;       % GZ for still water
GM_0 = GZ_0/phi_10; % GM for still water

Ix = Delta*GM_0/wg^2;

ydot=[y(2); -B/Ix*y(2)-Delta/Ix*GMi*y(1)];

% test7a.m %

function varargout = test7a(varargin)
% TEST7A M-file for test7a.fig
%     TEST7A, by itself, creates a new TEST7A or raises the existing
%     singleton*.
%
%     H = TEST7A returns the handle to a new TEST7A or the handle to
%     the existing singleton*.
%
%     TEST7A('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in TEST7A.M with the given input arguments.
```

```

%
% TEST7A('Property','Value',...) creates a new TEST7A or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before test7a_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to test7a_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help test7a

% Last Modified by GUIDE v2.5 01-May-2008 18:55:42

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @test7a_OpeningFcn, ...
                  'gui_OutputFcn',  @test7a_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin1)
    gui_State.gui_Callback = str2func(varargin1);
end

if nargout
    [varargout1:nargout] = gui_mainfcn(gui_State, varargin:);
else
    gui_mainfcn(gui_State, varargin:);
end
% End initialization code - DO NOT EDIT

% --- Executes just before test7a is made visible.
function test7a_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.

```

```
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to test7a (see VARARGIN)

% Choose default command line output for test7a
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes test7a wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = test7a_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout1 = handles.output;

function tf7_input_Callback(hObject, eventdata, handles)
% hObject    handle to tf7_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tf7_input as text
%        str2double(get(hObject,'String')) returns contents of tf7_input as a double

% --- Executes during object creation, after setting all properties.
function tf7_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tf7_input (see GCBO)
```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgro
    set(hObject,'BackgroundColor','white');
end

```

```

function ip7_input_Callback(hObject, eventdata, handles)
% hObject handle to ip7_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ip7_input as text
% str2double(get(hObject,'String')) returns contents of ip7_input as a double

```

```

% --- Executes during object creation, after setting all properties.
function ip7_input_CreateFcn(hObject, eventdata, handles)
% hObject handle to ip7_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgro
    set(hObject,'BackgroundColor','white');
end

```

```

function iv7_input_Callback(hObject, eventdata, handles)
% hObject handle to iv7_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```
% Hints: get(hObject,'String') returns contents of iv7_input as text
%         str2double(get(hObject,'String')) returns contents of iv7_input as a double

% --- Executes during object creation, after setting all properties.
function iv7_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to iv7_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackground...
    set(hObject,'BackgroundColor','white');
end

function c7_input_Callback(hObject, eventdata, handles)
% hObject    handle to c7_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c7_input as text
%         str2double(get(hObject,'String')) returns contents of c7_input as a double

% --- Executes during object creation, after setting all properties.
function c7_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c7_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackground...
    set(hObject,'BackgroundColor','white');
end
```

```

% --- Executes on button press in plot7_pushbutton.
function plot7_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to plot7_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get user input from GUI
tf7 = str2double(get(handles.tf7_input,'String'));
t7span = [0, tf7];

ip = str2double(get(handles.ip7_input,'String'));
iv = str2double(get(handles.iv7_input,'String'));

ip7=deg2rad(ip);
iv7=deg2rad(iv);

y0_7= [ip7 ; iv7];

B_cr = 3.761e9 ;
zeta= str2double(get(handles.c7_input,'String'));
B=B_cr*zeta;

% Read in table of GZ values and set up phi and position values

phi = -70:5:70;           % In put roll angle
X = deg2rad(phi);        % roll angle in radians

Y = (0:1:20)*11.6;       % Wave positions for L=232

%Y = (0:1:20)*7.75;      % Wave positions for L=155
%Y = (0:1:20)*5.8;       % Wave positions for L=116

GZ = xlsread('GZ_table_11_6_232.xls');    % Reads GZ table for H = 11.6 m, L =
%GZ = xlsread('GZ_table_8_4_232.xls');    % Reads GZ table for H = 8.4 m, L =
%GZ = xlsread('GZ_table_5_6_232.xls');    % Reads GZ table for H = 5.6 m, L =

%GZ = xlsread('GZ_table_5_6_155.xls');    % Reads GZ table for H = 5.6 m, L =
%GZ = xlsread('GZ_table_5_6_116.xls');    % Reads GZ table for H = 5.6 m, L =

```

```
[t7, y7] = ode45(@test7_system,t7span,y0_7, [],B,X,Y,GZ);

% plot between roll angle and tau

size(t7);
size(y7);
y=rad2deg(y7);

axes(handles.t_phi_axes)
plot(t7, y(:,1))
grid on
xlabel('t [sec]')
ylabel('ϕ [deg]')
title('Solution with interpolated GZ')

% Plot between phi and phidot

axes(handles.phi_phidot_axes)
plot(y(:,1), y(:,2))
grid on
xlabel('ϕ [deg]')
ylabel(' [deg/s]')
title ('Phase plane plot')

% --- Executes when figure1 is resized.
function figure1_ResizeFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% test6b.m %

k = menu('Choose a plot','Plot between roll angle and tau','Plot between phi and ph

if k==1

    % plot between roll angle and tau
```

```
t_final=input('Enter the final time: ');

tspan= [0,t_final];

ip= input('Enter the initial roll angle [deg]: ');
iv= input('Enter the initial roll velocity [deg/s]: ');

ip_r=deg2rad(ip);
iv_r=deg2rad(iv);

y0= [ip_r, iv_r];

B_cr = 3.761e9 ;
zeta= input('Enter the damping ratio, zeta: ');

B=B_cr*zeta;

phi = -70:5:70;
X = deg2rad(phi);

Y = (0:1:20)*11.6;

import_GZ_data GZ_-70_70.xls;
GZ=data;

[t, y] = ode45(@test7_system,tspan,y0, [],B,X,Y,GZ);

size(t);
size(y);

y_d=rad2deg(y);

plot(t, y_d(:,1))
grid on

xlabel('t [sec]')
ylabel('φ [deg]')
title('Run 000')
```

```
else

    % Plot between phi and phidot
    plot(y6(:,1), y6(:,2))
    grid on
    xlabel('phi')
    ylabel('phi dot')
    title(['Phase plane plot with interpolated values of GM for  $\phi = 10 \text{ deg} =$ 
    0.1745 rad and  $c = ', num2str(c6)])

end

% test7_system.m %

function ydot=test7_system(t,y,B,X,Y,GZ)

t;
y;

V_rel=13.79;

L_pp=232;

L=232;
%L=155;
%L=116;

S= (L_pp-L)/2- V_rel*t;           % crest amidships position

%S = L_pp/2- V_rel*t;           % trough amidships position

%S= L_pp-L/4- V_rel*t;           % zero wave elevation at FP

while S < 0

    S = S + L;
end$ 
```

```
while S > L

    S = S - L;
end

S;

% Interpolate GZ value from table
%GZi = interp2(phitable,postable,GZtable,y7(1),S,'spline');

GZi = interp2(X,Y,GZ,y(1),S,'spline');

% weight of ship

M=54600;
g=9.81;
Delta = M*1000*g;

% roll frequency

Tg = 21.4;
wg = 2*pi/Tg;

% moment of inertia
phi_10 = pi/18;
GZ_0 = 0.18;
GM_0 = GZ_0/phi_10;

Ix = Delta*GM_0/wg^2;

ydot=[y(2); -B/Ix*y(2)- Delta/Ix*GZi];
```


Bibliography

- [1] Paulling, J. R. On Parametric Rolling of Ships. 10th International Symposium on Practical Design of Ships and Other Floating Structures, American Bureau of Shipping, 2007.
- [2] ABS Guide for the assessment of parametric roll resonance in the design of container carriers, American Bureau of Shipping, 2004.
- [3] Sheikh, I. A. Dynamic Stability of Ships, Pre-project in Solid Mechanics for the Degree of Master of Science, Faculty of Mathematics and Natural Sciences, University of Oslo, 2007
- [4] Rawson, K.J. and Tupper, E.C. Basic Ship Theory. Butterworth Heinemann, 2001.
- [5] Shin, Y.S., Belenky V.L., Paulling, J. R., Weems, K.M. and Lin, W.M. Criteria for Parametric Roll of Large Containerships in Longitudinal Seas, Appendix 3 from a paper presented at the SNAME Annual Meeting, 30 September 2004.
- [6] Chopra, A.K. Dynamics of Structures, 3rd ed., Pearson Education, Inc., New Jersey, 2007.
- [7] Eriksen, S. M. R. Parametric Roll Motions of Ships, M.Sc. Thesis, Department of Marine Technology, Norwegian University of Science and Technology (NTNU), Trondheim, 2007.
- [8] http://www.math.fsu.edu/~ssimakhi/Svetlana%20Simakhina_files/Sveta_thesis_pdf.pdf