EXPERIMENTAL STUDY OF THE EFFECT OF FORWARD SPEED AND FOLLOWING WAVES ON ROLL DAMPING OF FISHING VESSELS

SUIMIN ZHANG

Experimental Study of the Effect of Forward Speed and Following Waves on Roll Damping of Fishing Vessels

by

©Suimin Zhang, B.Sc.

A thesis submitted to the School of Graduate
Studies in partial fulfillment of the
requirements for the degree of
Master of Engineering

Faculty of Engineering and Applied Science
Memorial University of Newfoundland

July 1993

St. John's Newfoundland Canada

ISBN 0-315-86643-8

This Thesis is dedicated to

my parents

Abstract

An extensive experimental program has been carried out to estimate roll damping parameters for three models of fishing vessels having different hull shapes and moving with forward speed. Roll damping parameters are determined using a novel method. This method combines the Energy method and the Modulating Function technique. The results show that this method gets better estimates compared with the original Energy method.

A data processing system was designed to process the experimental data. A parameter C_{error} was introduced to measure the error in roll damping identification. A database system was developed using VAX-Pascal to store the analytical results and perform various kinds of analyses. The data management and processing system in this research work has proved to be very efficient.

The effect of forward speed, initial angle and natural frequency on roll damping is discussed. The effect of forward speed on roll damping was found to be nonlinear. The effect of initial angle is strong at zero and low forward speeds and decreases as the forward speed is increased. The effect of natural frequency was found to be weak.

Ikeda's method was used to predict the roll damping coefficient. The results were compared with the experimental data. It was found that Ikeda's method overestimates the roll damping at higher forward speeds for all three models. This method fails in predicting the eddy damping for ship forms with hard chines. It was noticed that as models move with forward speed, their mean drafts increase. A modification to Ikeda's formula is proposed, making use of this observation. The values predicted by the modified formula fit the experimental data very well.

A preliminary experiment has been done to investigate the effect of following waves on roll damping. It has been found that estimating roll damping parameters,

without allowing for the time variation in the restoring moment, results in over-estimating the values of these parameters. Further work is needed in this area.

ACKNOWLEGEMENTS

In my pursuit of the Master's degree in Engineering, I have received valuable advice and assistance from a number of people and groups, the help of these people and groups is very much appreciated. In particular, I would like to thank:

- My supervisor, Dr. M.R. Haddara for his guidance and support throughout the program, and for his advice and elucidation of the various problems associated with this study.
- Mr. Andrew Kuczora for his assistance in conducting the experiments.
- Dr. Wishahy at C-CORE for providing a Gyroscope for roll measurement.
- IMD for providing the models for experiment.
- Technical service in Engineering for building the experimental setup.
- Staff in Center for Computer Design Engineering for their help in overcoming difficulties related to computer work.
- Mr. Y. Zhang, Mr. Z. Wang and Ms. Xia Wu for helping me with the experiments.

Contents

	Abstract	**i**
	ACKNOWLEGEMENTS	**iii**
	Contents	**iv**
	List of Figures	**vii**
	List of Tables	**x**
	List of Symbols	**xi**
1	**Introduction**	**1**
2	**Experiment**	**5**
	2.1 The Models and Experimental Set Up	5
	2.2 Experimental Parameters	10
	2.2.1 The Measurement of GM Values	11
	2.2.2 The Determination of the Center of Gravity	12
	2.2.3 The Expression for the Restoring Moment	13
3	**Identification of Roll Damping and Data Processing**	**15**
	3.1 Identification of Roll Damping	15
	3.1.1 Energy Method	15

3.1.2 Modified Energy Method 18
3.1.3 Comparison of Energy Method and Modified Energy Method . 19
3.2 Data Acquisition, Processing and Management 21
3.2.1 Data Acquisition 22
3.2.2 Translation of Experimental Data 24
3.2.3 Rearragement of the Data 24
3.2.4 Calculation of Roll Damping Parameters 25

4 Management and Analysis **28**
4.1 Management of the Experimental Results Using Database Techniques 28
4.1.1 Introduction of the Database Management System 28
4.1.2 The Creation of a Database 31
4.1.3 The functions of the data management sub-routine 32
4.1.4 The functions of the data analysis and reporting sub-routine . 33
4.2 Analysis and Discussion 37
4.2.1 Effect of Forward Speed on Roll Damping 37
4.2.2 The Effect of The Initial Angle of Heel 38
4.2.3 The Effect of Natural Frequency 40

5 Prediction of Roll Damping **44**
5.1 Component Analysis 44
5.1.1 Friction Damping 44
5.1.2 Eddy Damping (Naked Hull) 45
5.1.3 Lift Damping 46
5.1.4 Wave Damping 47
5.1.5 Eddy Damping due to the Skeg 48
5.2 Comparison of Predicted Values with Experimental Results 50

5.3 Modification of Ikeda's Formula 54

6 **Effect of Following Waves on Roll Damping** 61

6.1 The Experiment 61

6.2 Analysis and discussion 63

7 **Conclusions** 69

References 71

Appendix A. Content of File S0507 75

Appendix B. Content of File S0507.AGL 76

Appendix C. Content of File S0507.USE 77

Appendix D. Program TRSBAT.PAS 78

Appendix E. Program DAMABAT.FOR 82

Appendix F. Program MODFBAT.FOR 88

Appendix G. Program LABDATA.PAS 99

List of Figures

2.1 Lines' Plan for M363 7
2.2 Lines' Plan for M365 7
2.3 Lines' Plan for M366 8
2.4 Experimental Setup 8
2.5 Key arrangement on the board 9
2.6 Picture of Experimental Setup 9
2.7 GZ curves for three models 14

3.1 Comparison between experiment and predicted response M363 20
3.2 Comparison between experiment and predicted response M365 20
3.3 Comparison between experiment and predicted response M366 21
3.4 The flowgraph of data processing 22
3.5 Roll decay curve before and after the rearrangement 25

4.1 Capabilities of data management system 30
4.2 Effect of the Joint M363 GM=3.12 35
4.3 Effect of the Joint M365 GM=3.39 35
4.4 Effect of the Joint M366 GM=2.02 36
4.5 Effect of F_r and ϕ_A on roll damping M363 39
4.6 Effect of F_r and ϕ_A on roll damping M365 39
4.7 Effect of F_r and ϕ_A on roll damping M366 40

4.8 Effect of natural frequency on roll damping M363 42
4.9 Effect of natural frequency on roll damping M365 42
4.10 Effect of natural frequency on roll damping M366 43

5.1 Pressure distribution due to a skeg 49
5.2 Predicted results using Ikeda's formula M363 (GM=5.33cm 51
5.3 Predicted results using Ikeda's formula M363 (GM=3.12cm 51
5.4 Predicted results using Ikeda's formula M365 (GM=5.22cm 52
5.5 Predicted results using Ikeda's formula M365 (GM=3.96cm 52
5.6 Predicted results using Ikeda's formula M366 (GM=3.82cm 53
5.7 Predicted results using Ikeda's formula M366 (GM=4.35cm 53
5.8 Effect of $\overline{OG}$ values on Wave Damping 55
5.9 $\overline{OG}$ slopes predicted from experimental data 55
5.10 Relationship between slope b and block coefficient 56
5.11 Predicted results using Modified Ikeda's formula M363 (GM=5.33cm . 58
5.12 Predicted results using Modified Ikeda's formula M363 (GM=3.12cm . 58
5.13 Predicted results using Modified Ikeda's formula M365 (GM=5.22cm . 59
5.14 Predicted results using Modified Ikeda's formula M365 (GM=3.96cm . 59
5.15 Predicted results using Modified Ikeda's formula M366 (GM=3.82cm . 60
5.16 Predicted results using Modified Ikeda's formula M366 (GM=4.35cm . 60

6.1 Experimental Setup (following wave) 62
6.2 Roll decay curve with following wave 65
6.3 Heave motion with following wave 65
6.4 Record of incident wave 66
6.5 Effect of following wave on roll damping, GM=5.04 66
6.6 Effect of following wave on roll damping, GM=4.07 67

6.7 Effect of following wave on roll damping, GM=3.13 67

6.8 Effect of following wave on roll damping, GM=2.34 68

List of Tables

2.1 Principal Dimensions For Models 6

2.2 Forward Speed for Test(m/s) 10

2.3 Experimental Parameters for M363 11

2.4 Experimental Parameters for M365 12

2.5 Experimental Parameters for M366 13

4.1 Damping Coefficient V=0.5m/s, id='S' 34

4.2 B_e as a function of forward speed and initial angle, id='S' 34

4.3 B_e as a function of ω and forward speed, initial angle=11° 36

6.1 Wave height in the experiment 62

6.2 Experimental Parameters for M363 (following wave) 63

List of Symbols

L	length of waterline
B	beam of waterline
d	draft
U	forward speed of the model
C_M	midship coefficient
C_B	block coefficient
Δ	model mass(kg)
GM	metacentric height
KG	height of the center of gravity above keel
KB	height of the center of buoyancy above the keel
BM	distance from the center of buoyancy to the metacenter
$\overline{OG}$	distance between the rolling center and the water level
$\overline{OG}_0$	distance between the rolling center and the still water level at zero forward speed.
μ_1, μ_2	parameters of restoring moment
ϕ	roll angle
ϕ_A	initial roll angle
$\dot{\phi}$	roll velocity
$\ddot{\phi}$	roll acceleration
$N(\phi, \dot{\phi})$	damping moment per unit virtual moment of initia
$D(\phi)$	restoring moment per virtual moment of inertia
ω	natural trequency
ζ	nondimensional linear damping coefficient
ϵ	nondimentional nonlinear damping coefficient

B_e equivalent linear damping coefficient

C_{error} error coefficient in roll damping identification

b slope of $\overline{OG}$

B_{F0} friction damping at zero forward speed

B_F friction damping in the presence of forward speed

B_{E0} eddy damping at zero forward speed for one section

B_{E0t} eddy damping at zero forward speed for the whole ship form

B_E eddy damping in the presence of forward speed

B_L lift damping

B_{w0} wave damping at zero forward speed

B_w wave damping in the presence of forward speed

Chapter 1

Introduction

Although roll damping has been extensively studied by many researchers in the past twenty years, very little attention has been paid to the effect of forward motion. Roll damping suffers both quantitative and qualitative variations as a result of forward motion. As the ship speed increases a new roll damping component comes into play: the lift component. The effect of the lift damping becomes predominant at higher speeds.

Barr and Ankudinov[1] considered the roll damping of a ship hull without bilge keels or other damping devices to arise from two sources, wavemaking and viscosity. Viscosity is responsible for damping caused by vortex shedding at areas on the hull which suffer from large slope changes. Schmitke[2] used a similar reasoning to find estimates for damping moment for a warship hull form. He included the contributions from lifting surfaces such as the rudder, skeg and propeller shaft brackets. Both these works ignored the contribution of the bare hull as a lifting surface.

Due to the fact that the ship's hull has poor section shape as a lifting surface and because of its extremely low aspect ratio, it might be expected that the hydrodynamic forces and moments generated by the lift mechanism are much smaller than that generated by the rudder. However, one may quote Crane et al.[4], "... because of its very large profile area, a ship's hull does in fact generate forces and moments far larger than the control forces and moments generated by its rudder", to show that this is not true. This component is very important when the ship is moving with a non zero forward speed. As a matter of fact, as the forward velocity of ship increases, one should expect the lift component of the roll damping moment to constitue the most significant part of the roll damping moment

One of the well known methods of roll damping estimation is the one presented by ikeda et al.[3]. In this method, roll damping for a ship hull is assumed to consist of five components. These are friction damping, wave damping, eddy damping, naked hull lift damping and bilge keel damping. Different empirical formulae are introduced for the calculation of the different components. In calculating the lift component, ikeda et al.[3] assumed the hull to be a lifting surface with a surface area equal to its length multiplied by draft. The angle of attack is equal to the ratio of an effective lateral velocity caused by the rotation of the hull about a center of roll to the forward velocity of the vessel. A semi-empirical expression for the slope of the lift coefficient with respect to the angle of attack, as a function of the ship's length, beam, draft and the midship section coefficient, was presented. It seems that the expression for the slope of the lift coefficient used in ikeda's function is an empirical modification of that provided by Jones formula for a low aspect ratio wing, see Crane et al.[4]. The modification involves using an effective aspect ratio for the hull equal to $(2d/L)$ and adding a function in both the beam length ratio and the midship section coefficient. This function reflects the fact that a thick wing has higher slope for the lift curve. ikeda's formula implies that the lift coefficient is a linear function of the angle of attack. It also assumes that the lift coefficient is independent of the forward speed and of the angle of roll.

An experimental study of the roll damping of a warship hull moving with forward speed by Cumming et al.[5] showed the inviscid damping component to be a nonlinear function of the forward velocity.

Blok and Aalbers[6] investigated the roll damping characteristics for 13 models from MARIN's systematic series of high speed displacement hull forms(FDS series). They reported poor correlation between experimental damping coefficients obtained for these models and estimations obtained using ikeda's empirical method. After modifying the estimation of the lift damping component using the theory of trimmed flat plates by Shuford[7], in addition to other modifications introduced by Schmitke[2] and Graham[8] for the calculations of bilge keel eddy damping, damping coefficients estimates agreed well with those obtained from free roll decay tests.

An experimental investigation of the lift component of roll damping has been done by Haddara and Leung[9]. The models were towed in calm water with different forward speeds at a yaw angle with the hull in the upright condition. The magnitude and the point of action of the lift force are determined by measuring the moment and force acting on the model. It has been found that the equivalent linear damping coefficient due to lift is a nonlinear function of the forward speed of the model. It was also found that ikeda's formula under estimates the lift component in higher forward speed. This experiment was done under a static condition in which the models were not allowed to heave. It may yield different results when the model is allowed more degrees of freedom.

It thus seems, that a further study of the roll damping moment of a ship moving with forward speed is warranted. The accuracy of the assumptions underlying ikeda's method and its limitations should be investigated. It is the main objective of this work to investigate experimentally the roll damping moment of the ship models moving with forward speed.

A few roll decay tests were also obtained for the model in following waves. The main objective of this preliminary investigation is to see what effect following waves have on roll damping.

In roll damping experiments, a large number of roll decay curves are usually obtained. These curves are usually processed one by one. In this work, a new data processing and management technique is used so that the experimental data can be processed and analysed quickly, correctly and completely.

Chapter 2

Experiment

The experiments were performed in the wave tow tank of Memorial University of Newfoundland. The wave tank has inside dimensions of 58.27 m in length, 4.57 m in width, and 3.04 m in depth. Regular and irregular waves can be generated by a piston type wave generator at one end of the tank. At the other end of the tank a parabolic beach, consisting of an aluminum frame covered by wooden slabs, is intented to absorb and dissipate the energy contained in the incident wave and maintain a minimum reflection coefficient. A towing carriage is available for towing tests, resistance tests, current probe calibration, and self propulsion experiments. The carriage has a net weight of 3.9 tonnes and attains a maximum speed of 5 m/s.

2.1 The Models and Experimental Set Up

Models for three small fishing vessels were used in this investigation. They all represent fishing vessels of the less than 25 meters length class. They are all of similar dimensions but have quite different hull forms. Model M363 has a hard chine while M366 has a round bilge. Model M365 has a round bilge with a small rise of floor. The principal dimensions of these models are shown in Table 2.1 and the line plans are shown in Figure 2.1, Figure 2.2 and Figure 2.3.

Table 2.1: Principal Dimensions For Models

Model	M363	M365	M366
Scale	1:12	1:9.1	1:6.8
L(m)	1.551	1.336	1.590
B(m)	0.507	0.506	0.506
d(m)	0.221	0.215	0.205
LCB(m)	-0.109	-0.052	-0.1375
Δ (kg)	79.5	54.5	69.5
C_M	0.746	0.705	0.612
C_B	0.4575	0.3750	0.4214

In the experiment, the models were only allowed three degrees of freedom: roll, heave and pitch. The experimental set up is shown in Figure 2.4. Part A is composed of two rollers which guide a rod fixed to the carriage, this keeps the model moving along the tank and allows it to pitch, heave and roll. Part B is a universal joint, which allows the model to move in roll and pitch. The universal joint is connected to a rod which is supported by two linear bearings. The linear bearings allow the model to heave. The model is moved forward by the action of a force transmitted from the carriage to the universal joint.

Part A and B were mounted on a board as shown in Figure 2.4. The rolling centers of Part A and B are at the same horizontal level. The vertical position of the board can be adjusted. As a result, the roll center can be changed. In addition, a

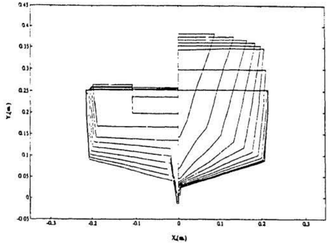

Figure 2.1: Lines' Plan for M363

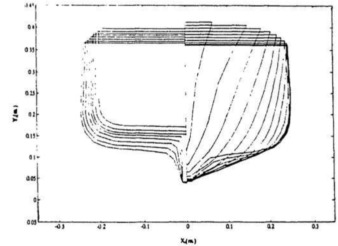

Figure 2.2: Lines' Plan for M365

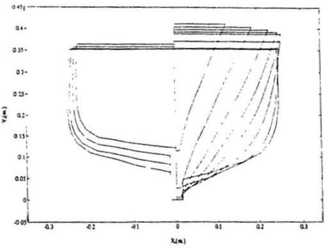

Figure 2.3: Lines' Plan for M365

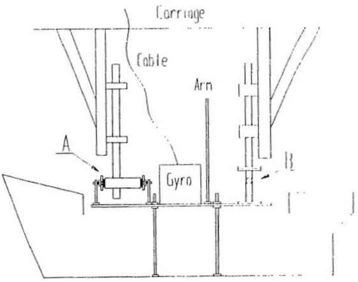

Figure 2.4: Experimental Setup

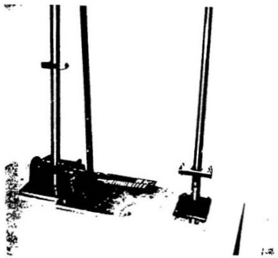

Figure 2.5: Key arrangement on the board

Figure 2.6: Picture of Experimental Setup

gyro is mounted on the board to measure the roll angle. To give the model an initial heel angle at the start of the test, an arm connected to the model at its center of floatation is pushed to the side and then let go. The key arrangement on the board is shown in Figure 2.5 and a picture showing the experimental setup is in Figure 2.6.

2.2 Experimental Parameters

In this experiment, the model was constrained against sway. The metacentric height GM can be changed by changing the position of the center of gravity. This changes the natural frequency of the model. The models were tested under different GM values as shown in Table 2.3, Table 2.4 and Table 2.5 where the natural frequencies were directly measured from the decay curves and I.D. is a character used to identify each GM value and make up file name for each decay curve.

For every GM value, the models were tested at 8 forward speeds varying from 0.0 to 1.5 m/s as shown in Table 2.2. At each forward speed, the roll decay curves were measured for 7 initial angles varied from 7^o to 25^o. At zero forward speed, free roll decay tests were also performed without the joint(Part A) so that the influence of the joint could be found. Therefore, for each GM value, 63 decay curves(8 forward speeds $\times$ 7 initial angles + 7 initial angles without joint) were obtained. More than 1300 decay curves were obtained in total.

The determinations of GM, $\overline{OG_0}$ and μ_1, μ_2 are stated in the following subsections.

Table 2.2: Forward Speed for Test(m/s)

0.0	0.3	0.5	0.7	0.9	1.1	1.3	1.5

Table 2.3: Experimental Parameters for M363

I.D.	GM(cm)	ω	$\overline{OG}_0(cm)$	μ_1	μ_2
A	5.33	3.796	3.98	0.8052	-1.1035
B	4.54	3.572	4.77	0.9666	-1.2682
R	3.82	3.310	5.49	1.1924	-1.5388
C	3.64	3.180	5.67	1.2616	-1.6221
D	3.24	2.952	6.06	1.4448	-1.8362
S	3.12	2.951	6.19	1.5131	-1.9140
E	2.15	2.455	7.16	2.2844	-2.7878

2.2.1 The Measurement of GM Values

GM is the metacentric height which denotes the distance from the center of gravity to the metacenter, positive upward. GM values can be measured by inclining experiments. In these experiments a small weight is moved a known transverse distance and the heel angle is measured. GM value is calculated by the following equation:

$$GM = \frac{md}{\Delta \cdot \tan\theta} \tag{2.1}$$

where m is the mass of the small weight, Δ is the mass of the model, d is the distance of the small weight from the center and θ is the heel angle. The experiment should be repeated serval times and an average value obtained for GM.

In the experiment, GM values were obtained by moving the small weight in several known distances and the the average value was calculated.

Table 2.4: Experimental Parameters for M365

I.D.	GM(cm)	ω	$\overline{OG}_0(cm)$	μ_1	μ_2
H	5.22	4.457	6.51	-0.4957	-1.0220
I	4.00	3.990	7.73	-0.5909	-1.3431
I	3.96	3.838	7.77	-0.5887	-1.3737
2	3.39	3.605	8.34	-0.6635	-1.5888
J	3.28	3.484	8.45	-0.6868	-1.6202
3	2.57	3.217	9.16	-0.8261	-2.0661
K	2.40	2.986	9.33	-0.8597	-2.2430

2.2.2 The Determination of the Center of Gravity

The height of the center of gravity above the keel can be determined by the following equation:

$$KG = KB + BM - GM \tag{2.2}$$

where KB is the height of the center of buoyancy above the keel, BM is the distance from the center of buoyancy to the metacenter. The values of KB and BM for the test models were specified in the hydrostatic particulars list provided by IMD. If the roll center coincides with the center of gravity, the distance between the roll center and the still water level at zero forward speed can be determined by:

$$\overline{OG}_0 = KG - KD \tag{2.3}$$

where KD is the distance between the still water line to the keel.

Table 2.5: Experimental Parameters for M366

I.D.	GM(cm)	ω	$\overline{OG}_0(cm)$	μ_1	μ_2
Y	4.35	3.423	2.57	-0.1466	-1.8781
X	3.82	3.229	5.80	-0.1514	-1.0165
O	3.57	2.985	6.05	-0.1594	-1.0606
W	3.01	2.831	6.61	-0.1574	-1.3476
N	2.94	2.730	6.68	-0.1557	-1.2919
V	2.02	2.288	7.60	-0.1466	-1.8781
M	1.74	2.087	7.88	-0.1457	-2.1670

2.2.3 The Expression for the Restoring Moment

The expression for the restoring moment is needed for the identification of the roll damping parameters. The restoring moment $D(\phi)$ can be expressed in the following form:

$$D(\phi) = GM \cdot \Delta \cdot g(\phi + \mu_1\phi^3 + \mu_2\phi^5) \tag{2.4}$$

where g is the acceleration due to gravity (m/sec^2), Δ is the model mass(kg) and ϕ is the inclining angle.

The parameters μ_1, μ_2 can be regressed from the GZ curve of a ship with the relationship:

$$D(\phi) = GZ(\phi)\Delta g \tag{2.5}$$

where GZ represents the lever arm of the buoyancy force. The GZ curves for the three models are shown in figure 2.7. Each GZ curve was obtained for a specific GM

value. Assume the GZ and GM values for the GZ curves in Figure 2.7 are $GZ_0(\phi)$ and GM_0. The GZ curve for other GM value can be obtained by the following expression:

$$GZ(\phi) = GZ_0(\phi) + (GM - GM_0)\sin(\phi) \tag{2.6}$$

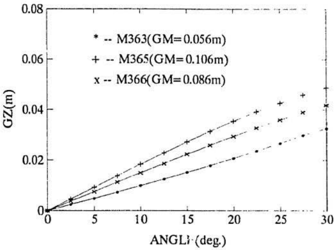

Figure 2.7: GZ curves for three models

Chapter 3

Identification of Roll Damping and Data Processing

3.1 Identification of Roll Damping

The damping parameters for the model were estimated from the free roll decay curves using a novel method. The method combines the Energy method, Haddara and Bennett[10] and a modified version of the Function Modulation Technique introduced by Shinbrot[11], see Haddara and Wu[12]. This method has been called "Modified Energy Method". Both methods are described and compared in the following sections.

3.1.1 Energy Method

The free rolling of a model can be described by the following differential equation:

$$\ddot{\phi} + N(\phi, \dot{\phi}) + D(\phi) = 0 \tag{3.1}$$

where ϕ is the angle of roll, $N(\phi, \dot{\phi})$ and $D(\phi)$ are the damping and restoring moments per unit virtual mass moment of inertia of the model.

The damping model can be expressed in the following nonlineear forms:

$$N(\phi, \dot{\phi}) = 2\zeta\omega(\dot{\phi} + \epsilon|\phi|\dot{\phi}) \quad \text{Linear angle dependence} \tag{3.2}$$

$$N(\phi, \dot{\phi}) = 2\zeta\omega(\dot{\phi} + \epsilon|\dot{\phi}|\dot{\phi}) \quad \text{Quadratic} \tag{3.3}$$

$$N(\phi,\dot{\phi}) = 2\zeta\omega(\dot{\phi} + \epsilon\dot{\phi}^3) \quad \text{Cubic} \tag{3.4}$$

$$N(\phi,\dot{\phi}) = 2\zeta\omega(\dot{\phi} + \epsilon\phi^2\dot{\phi}) \quad \text{Quadratic angle dependence} \tag{3.5}$$

where ζ and ϵ are the nondimensional linear and nonlinear damping coefficients, ω is the natural frequency of the linear roll equation.

In many cases, it is very useful to replace the nonlinear damping moment in equation 3.1 by an equivalent linear damping moment, especially in investigating the effect of different factors on roll damping, such as the effect of forward speed, natural frequency, initial angle, etc. In this case the roll damping moment can be expressed as:

$$N(\phi,\dot{\phi}) = B_e\dot{\phi} \tag{3.6}$$

where $B_e = 2\zeta\omega$ denotes the equivalent linear damping coefficient.

The restoring moment is a function of the form of the underwater part of the ship hull which has beed discussed in Chapter 2.

Rewriting the ship roll decay equation (eq. 3.1) in the following form:

$$\ddot{\phi} + D(\phi) = -N(\phi,\dot{\phi}) \tag{3.7}$$

and multiplying both sides by $\dot{\phi}$ gives:

$$\dot{\phi}\ddot{\phi} + \dot{\phi}D(\phi) = -N(\phi,\dot{\phi})\dot{\phi} \tag{3.8}$$

Writing the left hand terms of above equation in the following form:

$$\begin{aligned} \dot{\phi}\ddot{\phi} &= \frac{1}{2}\frac{d}{dt}(\dot{\phi}^2) \\ D(\phi)\dot{\phi} &= \frac{d}{dt}(G(\phi)) \end{aligned} \tag{3.9}$$

yields:

$$\frac{d}{dt}[\frac{1}{2}\dot{\phi}^2 + G(\phi)] = -N(\phi,\dot{\phi}) \tag{3.10}$$

where

$$G(\phi) = \int_0^{\phi} D(x)dx$$

Integrating equation 3.10 from t_i to t_{i+1} yields

$$V(t_i) - V(t_{i+1}) = \int_{t_i}^{t_{i+1}} N(\phi, \dot{\phi})\dot{\phi}dt \tag{3.11}$$

where t_i and t_{i+1} are two successive instants of time. $V(t)$ is the total energy of the model per unit virtual moment of inertia at time t

$$V(t) = \frac{1}{2}\dot{\phi}^2 + G(\phi) \tag{3.12}$$

Equation 3.11 shows that the energy loss during a small interval of time dt is equal to the energy dissipated in damping in the same interval. Assume the damping model is the Cubic form. Then substituing 3.4 into 3.11 yields

$$V(t_i) - V(t_{i+1}) = \int_{t_i}^{t_{i+1}} 2\zeta\omega(\dot{\phi}^2 + \epsilon\dot{\phi}^4)dt \tag{3.13}$$

or

$$Q_i(t) = b_1 n_{i1} + b_2 n_{i2} \tag{3.14}$$

where

$$\begin{aligned}
Q_i &= V(t_i) - V(t_{i+1}) \\
b_1 &= 2\zeta\omega \\
b_2 &= 2\zeta\omega\epsilon \\
n_{i1} &= \int_{t_i}^{t_{i+1}} \dot{\phi}^2(t)dt \\
n_{i2} &= \int_{t_i}^{t_{i+1}} \dot{\phi}^4(t)dt
\end{aligned} \tag{3.15}$$

Q_i and n_{i1}, n_{i2} can be determined numerically from the roll decay curve. A least square method can then be used to find the coefficients b_1, b_2 which makes the sum of the squares of the difference between the two sides of equation 3.14 a minimum. The parameters of other roll damping models can be obtained in the same way.

3.1.2 Modified Energy Method

A modulating function operator is defined as:

$$\Psi[f(t)] = \int_0^T f(t)A^k(\tau)dt \quad (k = 0, 1, \cdots, n) \tag{3.16}$$

where

$$A^k(\tau) = \exp(-\tau^2/2)H_k(\tau) = (-1)^k \frac{d^k}{d\tau^k}[\exp(-\tau^2/2)] \tag{3.17}$$

and $H_k(\tau)$ is Hermite polynomial of order k and

$$\tau = \frac{t}{T}(T_e + T_s) - T_s = \beta t - T_s \tag{3.18}$$

where

$$\beta = \frac{(T_e + T_s)}{T}$$

The function $A^k(\tau)$ satisfy the following orthogonal relationship

$$\int_{-\infty}^{\infty} \exp(\tau^2/2)A^m(\tau)A^n(\tau)d\tau = n!\sqrt{2\pi}\delta_{mn}$$

where δ_{mn} is Kronecker delta. They also satisfy the following recursion relationships:

$$\tau A^n(\tau) = A^{n+1}(\tau) + nA^{n-1}(\tau)$$

$$\frac{dA^n(\tau)}{d\tau} = -A^{n+1}(\tau)$$

Substitu ng the expression for $N(\phi, \dot{\phi})$ in equation 3.4 and operating on equation 3.1 using Ψ_k, one gets

$$\Psi_k[\dot{V}] = -2\zeta\omega\{\Psi_k[\dot{\phi}^2] + \epsilon\Psi_k[\dot{\phi}^4]\} \tag{3.19}$$

In equation 3.19,

$$\begin{aligned}
\Psi_k[\dot{V}] &= \int_0^T \dot{V}(t)A^k(\tau)dt \\
&= V(T)A^k(T_e) - V(0)A^k(-T_s) - \beta\int_0^T V(t)\frac{dA^k(\tau)}{d\tau}dt \\
&= V(T)A^k(T_e) - V(0)A^k(-T_s) + \beta\Psi_{k+1}[V(t)]
\end{aligned} \tag{3.20}$$

Then equation 3.19 can be expressed as

$$2\zeta\omega_0\{\Psi_k[\dot{\phi}^2] + \epsilon\Psi_k[\dot{\phi}^4]\} = -V(T)A^k(T_e) + V(0)A^k(-T_s) - \beta\Psi_{k+1}[V(t)]$$
$$(k = 0, 1, \cdots, n) \tag{3.21}$$

Using different values of k, one can generate a number of equations similar to equation 3.21 equal to the number of the unknown parameters in equation 3.4. In this case, we need only two equations to solve for ζ and ϵ. One can also generate a larger number of equations and use a least square technique to find the unknown parameters.

When the equivalent linear damping form is used, let $\epsilon = 0$ and ζ can be determined by

$$\zeta = \frac{-V(T)A^k(T_e) + V(0)A^k(-T_s) - \beta\Psi_{k+1}[V(t)]}{2\omega\Psi_k[\dot{\phi}^2]} \tag{3.22}$$

or

$$B_e = \frac{-V(T)A^k(T_e) + V(0)A^k(-T_s) - \beta\Psi_{k+1}[V(t)]}{\Psi_k[\dot{\phi}^2]} \tag{3.23}$$

3.1.3 Comparison of Energy Method and Modified Energy Method

The energy and modified energy methods were used to estimate the damping parameters from the decay curves. The damping parameters obtained by both methods were used to generate free decay curves for the three models. These curves are compared with the decay curves obtained from experiment. The results are shown in Figure 3.1 to Figure 3.3. One can see that the modified energy method provides better predictions than the original energy method and that it is consistent in predicting the damping parameters.

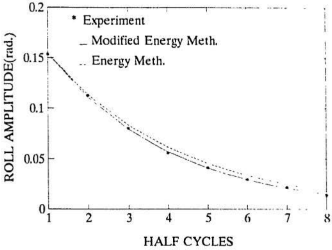

Figure 3.1: Comparison between experiment and predicted response M363

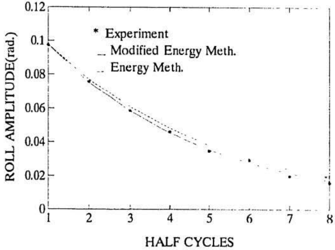

Figure 3.2: Comparison between experiment and predicted response M365

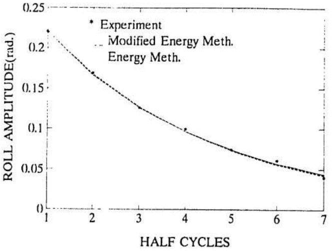

Figure 3.3: Comparison between experiment and predicted response M366

3.2 Data Acquisition, Processing and Management

As stated in Chapter 2, more than 1300 decay curves were measured in the experiment. Usually, the decay curves will be processed one by one. It may take a few weeks of hard work to finish the whole process. In the present work, a special scheme has been designed to process the data in batches. This scheme had to be designed before the experiment, because the file names have key effect on the batch processing. The file names must be composed using certain regulations so that the processing programs can compose the file names automatically and process them one by one. In order to perform batch processing, it takes more time in program design and testing so that the programs work properly. Batch processing gives the benefit that it may only take a few hours in data processing instead of a few weeks of tedious work on the single file processing. It also gives a tidy arrangement of the output files in each processing stage and produces a standard format of results

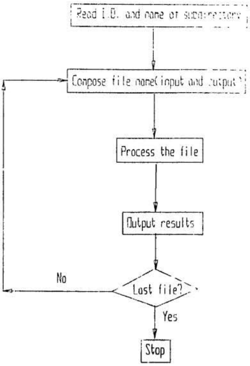

Figure 3.4: The flowgraph of data processing

which provides the possibility of using database techniques in data management and analysis. The flowgraph of data processing is shown in Figure 3.4.

3.2.1 Data Acquisition

In the experiment, a gyroscope was used to measure the roll decay curves. A program named 'S575' was used in data acquisition and plotting using the Keithley system and IBM PC interrupts. This program was developed in the Wave Tank Laboratory of M.U.N. using Microsoft C. The information of each decay curve was stored in a file. For the batch processing requirement, the file names were defined in the following way:

File Name = I.D. + Forward Speed No. + Initial Angle No.

Forward Speed No. = [00, 03, 05, ⋯, 15] for corresponding forward speeds listed in Table 2.2. For the roll decay test without joint(Part A) at zero forward speed, the Forward Speed No. is [63] for M363, [65] for M365 and [66] for M366. Initial Angle No. = [01, 02, 03, ⋯, 07] for initial angles varying from $7^\circ \sim 25^\circ$ in increments of about 3°. I.D. is the identification mentioned in Chapter 2. A few examples of the file names are listed as follows:

S0001 (forward speed = 0.0m/s, initial angle No. $1 \approx 7^\circ$)

S0502 (forward speed = 0.5m/s, initial angle No. $2 \approx 10^\circ$)

S1503 (forward speed = 1.5m/s, initial angle No. $3 \approx 13^\circ$)

S6304 (forward speed = 0.0m/s, initial angle No. $4 \approx 16^\circ$, Without joint)

H6505 (forward speed = 0.0m/s, initial angle No. $5 \approx 19^\circ$, Without joint)

Y6606 (forward speed = 0.0m/s, initial angle No. $6 \approx 22^\circ$, Without joint)

where 'S' is the I.D. for M363(GM=3.12cm) as shown in Table 2.3, 'H' is the I.D. for M365(GM=5.22cm) as shown in Table 2.4 and 'Y' is the I.D. for M366(GM=4.35cm) as shown in Table 2.5. In different stages of analysis, the file name will be the same but with different extension, as will be explained in detail in the following. The files in the stage of data acquisition do not have extensions.

A file named 'S0507' is shown in Appendix A as an example. The data from three channels were collected in the file in three columns. The first column gives the roll angle, the second column gives the pitch angle and the third column gives the forward speed. The integers in the columns indicate the amount of voltages measured by the gyroscope. Offsets and slopes in the file are used to translate the integers into degrees or m/s. The translation can be done by the following formula:

$$A = I * S + O \tag{3.24}$$

where I is the integer in the file, S is the slope and O is the offset.

3.2.2 Translation of Experimental Data

Equation 3.24 was used to obtaine calibrated data. In this stage and the following stages, the data were processed in batches. Input a I.D. such as 'S' will process all the 63 files(8 forward speeds × 7 initial angles + 7 initial angle without joint) at the same time. The program composes the file names automatically and processes the files one by one. A program named "TRSBAT" developed in PASCAL was used to do the translation. The source program is listed in Appendix D. The output files in this stage have the extension '.AGL'. An example of the translated values named 'S0507.AGL' is shown in Appendix B, where the first column is roll angles, the second is pitch angles and the third is forward speed.

3.2.3 Rearragement of the Data

The translated files still need some rearrangment before they can be used in roll damping parameter identification. A program named 'DAMABAT' written in FORTRAN was used to do this work. The source program is listed in Appendix E. The function of the program is listed as follows:

1. Cut off the first half cycle of the data. As stated in Chapter 2, the initial angle was generated by hand through an arm attached to the model. Some heave and pitch coupling are inevitable in the begining of the rolling. Therefore the first half cycle of the data was not used in the analysis.

2. Adjust the x-axis. In the experiment, the gyroscope may not be parallel to the water level and the roll decay curves may have some bias. The x-axis was

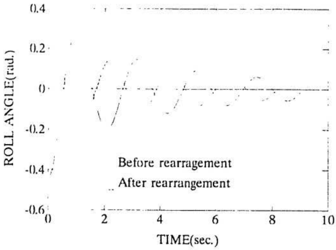

Figure 3.5: Roll decay curve before and after the rearrangement

adjusted to minimize the bias.

3. Measure the natural frequency from the decay curve.

4. Create a file for the identification of roll damping parameters.

The output files in this stage have the extension '.USE'. An example of the output file named 'S0507.USE' is shown in Appendix C. The first five values are sampling frequency($1/s$), natural frequency and coefficients of restoring moment($1, \mu_1, \mu_2$). From sixth to end are the rolling angles. The pitch angle and forward speed were not included in the file in order to save space. Figure 3.5 shows the roll decay curve before and after the rearrangement.

3.2.4 Calculation of Roll Damping Parameters

A program named 'MODFBAT' developed in FORTRAN is used to calculate the roll damping parameters by using Modified Energy Method. The source program is listed in Appendix F. The files obtained by the previous process such as 'S0507.USE' were used as input. A coefficient was used to measure the error. The coefficient is defined as :

$$C_{error} = \Sigma_{i=1}^{n}(A_i - \overline{A}_i)^2 \tag{3.25}$$

where A_i is the amplitude of each half cycle of the decay curve obtained from experiment. $\overline{A}_i$ is the amplitude of each half cycle of the decay curve generated by the roll damping parameters obtained by Modified Energy Method and n is the number of the amplitude of half cycle. n was assigned 5 in the calculation.

An example of the output of the program is shown in Table 3.1. In the table, line 01) is the input file name. Line 02) is the natural frequency. Lines 03) to 12) are the amplitudes of half cycles. Lines 13) to 16) are the nonlinear damping coefficients $(2\omega\zeta, 2\omega\zeta\epsilon)$ defined in equation 3.2 to 3.5 and the C_{error} defined in equation 3.25. In line 17), the first value is linear equivalent damping coefficient (B_e) and the third value is C_{error}. As we can see in Table 3.1, the errors of nonlinear models are smaller than the error of linear equivalent model, which indicates that the nonlinear models fit the experimental data better than the linear equivalent model.

Table 3.1 A Output of Program MODFBAT

01)	S0507.USE		
02)	2.922418		
03)	0.2869402		
04)	-0.2049789		
05)	0.1474881		
06)	-0.1196999		
07)	0.0929921		
08)	-0.0797468		
09)	0.0616441		
10)	0.0000000		
11)	0.0000000		
12)	0.0000000		
13)	0.4045932	2.036703	2.2077649E-04
14)	0.2950082	0.580191	8.1270322E-05
15)	0.4951415	7.767272	3.0916181E-04
16)	0.4267024	0.581354	1.3132309E-04
17)	0.6101135	0.000000	1.0751081E-03

Chapter 4

Management and Analysis

4.1 Management of the Experimental Results Using Database Techniques

4.1.1 Introduction of the Database Management System

As we saw in the previous Chapter, the output of the calculation is in simple file style(text file). Usually, investigators will analyse the data according to these files, which can be refered to as "simple file approach"[13]. This may work well if the amount of data is small and the relationships between the different components of the data are simple. In the present work, there are more than 1300 results as the one shown in Table 3.1. The usage of the data is quite diverse. Usage includes:

. Output data for different initial angles at a specific forward speed

. Output data for different forward speed at a specific initial angle

. Output data for different natural frequencies at a specific forward speed

As we can see, the data are used in different applications.

When simple file approach is used, it has the following problems:

1. Data redundancy. It is unavoidable that some data elements are used in number of applications as the situation stated above. Since data is required by multiple applications, it often is recorded in multiple data files. In most cases,

the data is stored repeatedly, which may jeopardize the integrity of the data, as well as putting pressure on storage.

2. Data availability constraints. When data are scattered in a number of files, it takes a lot of time and effort to search for the proper data to be used (usually done manually), which may lead to incomplete analysis of the data.

3. Data loss. In simple file approach, the various utilities of the operating system, such as copying, sorting, merging and editing, have to be used to handle the files and prepare data for further calculation or plotting. A small mistake can cause data loss and it will be unrecoverable. As there are many files in the storage, it is easy to forget the name and directory of the file, which may also lead to data loss.

The solution to such problems lies in database management systems(DBMS). DBMS is widely used in business and has spread to science and technology[13]. A database can be defined as[14]: "a common pool of shared data in which the data is interrelated, where each item of the data is stored only once and which represents a service to a wide range of applications."

The most popular database model is the relational model. The relational database can be simply considered as a two-dimensional table. The column is called data item or field and the row is called record. All records are distinct(no duplicate records are allowed). To ensure that all records are distinct, each record has a key. A key can be one field or a combination of a number of fields in the record. The records in the database can be indexed(or sorted) by the key in ascending or descending order.

A data management system is a computer software that builds and uses the database[13]. The capabilities of data management systems are shown in Figure 4.1. The advantages of using a database management system are:

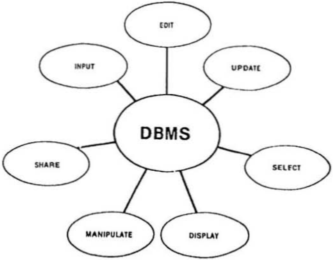

Figure 4.1: Capabilities of data management system(after Rumble and Hampel [13])

1. Redundancy is minimized. Because the data is only stored once. This saves storage space and guarantees the integrity of the data.

2. In a data management system, the data is isolated from the application programs. Changes in data file format, such as increasing a field length or adding a new field, and access methods, do not force modification in the application programs which use the files. This feature is referred as "data independence"[15].

3. With the help of a database, application programs can be developed , maintained and enchanced easily and quickly.

4. With the help of the key field, one can retrieve the required data easily.

5. The data in the database can be shared by different users, which is an important feature in business DBMS and is important in science and technology

applications when a group of people participate in the same project and analyse different aspects of the data.

6. In data management systems, many functions are similar, such as appending, updating, deleting, listing, etc.. It is possible to develop a set of common subroutines which can be used by different databases with a little change, which will save time and effort in programing, especially in large research projects which involve the processing of a great amount of data with different structures.

4.1.2 The Creation of a Database

A database created by PASCAL is used to store the results such as the one shown in table 3.1. PASCAL is better in the field of data management than many other languages such as FORTRAN. The reason is that PASCAL offers a richer repertoier of structured data types[16]. Here record data type is used. The record is a structure with named components which can be of different types. The result of analysis of each decay curve, as shown in Table 3.1, can be considered as a record. The definition of the database can be written as follows:

```
type
  key_type = packed array[1..5] of char;
  f_rec = record
    id          : [key(0, ascending, nochanges, noduplicates)] key_type;
    omega       : real;
    speed       : real;
    amplitude   : array[1..10] of real;
    b44         : array[1..5, 1..3] of real;
    end;
var
  f         : file of f_rec;
```

where 'id' is the field to store the file names of the decay curves such as 'S0507'. This is the key field in the record. The file is accessed in an index mode offered by

VAX-PASCAL[17][18]. To create a database, the file can be opened by:

```
open(f, 'lab0801.dat', history:=unknown,
        organization:=index, access_method:=keyed);
rewrite(f);
```

To read or update the database, the file can be opened by:

```
open(f, 'lab0801.dat', history:=old,
        organization:=index, access_method:=keyed);
resetk(f, 0);
```

where 'lab0801.dat' is the name of the database. A record can be located by:

```
findk(f, 0, 'S0507')
```

where 'S0507' is the key of the record to be found.

A program named 'LABDATA' has been developed to create and manage the database. The source program is listed in Appendix G. The program has two subroutines. One is for data management, the other is for data analysis and reporting.

4.1.3 The functions of the data management sub-routine

The functions of data management sub-routine are listed as follows:

1. Create (or rewrite) database.

2. Append analytical results obtained by Modified Energy Method. The analytical results such as the one shown in Table 3.1 will be added into the database.

3. Data examination. In this function, C_{error} of each record will be compared with a specified value. The program will list all the records in which C_{error} is greater than the specified value. In the analysis, it has been found that

when $C_{error} > 0.01$, the predicted damping coefficient is unacceptable, which means the decay curve generated by the predicted damping coefficients do not fit the decay curve obtained in the experiment well. In this case, the damping coefficients have to be re-estimated according to other results with better conditions. There are more than 1000 records in the database, only 2~3 % of the results needed to be re-estimated, which indicates that the Modified Energy Method has given a very good estimation of the roll damping coefficient.

4. Update analytical result. In this function, the data in a record can be modified.

5. List records in the database.

4.1.4 The functions of the data analysis and reporting sub-routine

As soon as all the analytical results are stored in the database, we can output the results in various combinations. In the analysis, equivalent damping coefficient is used in most of the cases. So without specification, the output damping coefficient is the equivalent damping coefficient.

The main functions of the data analysis and reporting sub-routine are listed as follows:

1. Find error caused by the joint by comparing two sets of data. As stated in Chapter 2, at zero forward speed, both experiments with and without the joint were tested. Two sets of data were compared to find the error caused by the joint. In this function, the average error caused by the joint is calculated and the data were output for plotting. The results are shown in Figure 4.2 ~ Figure 4.4. We can see that the error caused by the joint is almost constant

Table 4.1: Damping Coefficient V=0.5m/s, id='S'

Initial Angle(rad.)	$2\zeta\omega$	$2\zeta\omega_t$	C_{error}
0.09734	0.34164	0.0	3.705E-05
0.14203	0.48094	0.0	4.912E-04
0.17323	0.48768	0.0	3.494E-04
0.20445	0.51416	0.0	2.903E-04
0.24431	0.56082	0.0	6.508E-04
0.25176	0.58284	0.0	7.910E-04
0.28694	0.61011	0.0	1.075E-03

Table 4.2: B_e as a function of forward speed and initial angle, id='S'

Fr	Initial Angle			
	7^o	9^o	11^o	13^o
0.0000	0.3644	0.4512	0.5420	0.6092
0.0769	0.3200	0.3778	0.4504	0.5045
0.1282	0.3471	0.4123	0.4317	0.4685
0.1795	0.3831	0.4191	0.4382	0.4592
0.2308	0.4961	0.5093	0.5324	0.5428
0.2821	0.6102	0.6409	0.6653	0.6487
0.3334	0.7943	0.7572	0.7406	0.7608
0.3847	0.6518	0.7441	0.7647	0.7380

at different initial angles. In the analysis, the error will be deducted from the damping coefficients obtained from the experiments with joint.

2. Output damping coefficients of one forward speed. An example of the output is shown in Table 4.1.

3. Output damping coefficient as a function of speed and initial angle. An example of the output are shown in Table 4.2. where the initial angles are input as many as the user wants.

4. Output damping coefficients a function of ω(natural frequency) and speed. An example of the output is shown in Table 4.3. The user has to specify an initial

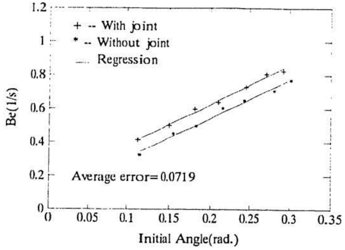

Figure 4.2: Effect of the Joint M363 GM=3.12

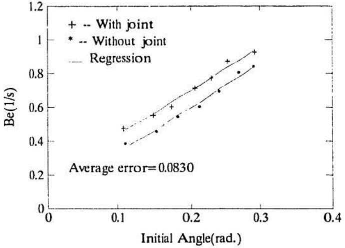

Figure 4.3: Effect of the Joint M365 GM=3.39

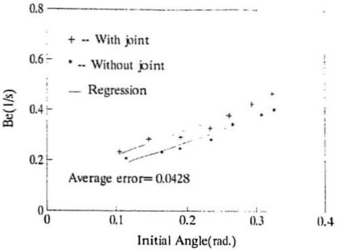

Figure 4.4: Effect of the Joint M366 GM=2.02

angle at first and then select the values of ω by selecting the I.D. listed in Table 2.3, Table 2.4 or Table 2.5.

5. Least square regression. In this function, a straight line is created to fit the data for different initial angles with the same forward speed. Example of using this function can be shown in Figure 4.2 to Figure 4.4.

Table 4.3: B_e as a function of ω and forward speed, initial angle=11°

I.D.	ω	Forward Speed(m/s)							
		0.0	0.3	0.5	0.7	0.9	1.1	1.3	1.5
A	3.796	0.443	0.382	0.372	0.426	0.476	0.554	0.623	0.540
B	3.572	0.445	0.367	0.343	0.375	0.465	0.530	0.654	0.562
R	3.310	0.554	0.435	0.442	0.460	0.531	0.593	0.661	0.706
C	3.180	0.492	0.401	0.386	0.406	0.526	0.605	0.773	0.772
D	2.952	0.488	0.378	0.364	0.382	0.477	0.643	0.734	0.816
S	2.951	0.542	0.450	0.432	0.438	0.532	0.665	0.741	0.765
E	2.455	0.499	0.398	0.396	0.463	0.590	0.756	0.809	0.862

4.2 Analysis and Discussion

The effects of different factors on roll damping have been investigated in detail by the help of the functions provided by the database system introduced in previous section.

4.2.1 Effect of Forward Speed on Roll Damping

The effect of forward speed on the equivalent linear damping coefficient of the models M363, M365 and M366 is shown in Figure 4.5 to 4.7, respectively. It can be seen from Figure 4.5 that there is a minimum point in the damping coefficient of M363 at a Froude number around 0.1 to 0.2. This phenomenon has been observed by several investigators, see Cox and Lloyd[19] and Cumming et al.[5]. The decrease in damping is attributed to a vortex cancellation mechanism caused by the bilge keels. However, model M363 does not have bilge keels but has a hard chine which could be causing the vortex cancellation mechanism in this case. The velocity at which the minimum roll damping occurs can be estimated by the "reduced frequency" relationship[19]:

$$\frac{\omega L_{bk}}{U} = 2\pi \tag{4.1}$$

In the case of the data in Figure 4.5, $\omega = 3.796$ and L_{bk} can be taken as the length of the hard chine which is about 1.0 meter long. Then we get $U = 0.6042$ and $F_r = 0.155$ which is approximately the value observed in Figure 4.5.

In addition, it has been noticed that as the forward speed increases there is a rapid decrease of the eddy damping accompanied by a slow increase in the lift damping until a certain speed is reached. As the forward speed increases beyond this value, lift damping increases rapidly and this causes a steady increase in the total damping of the model. For all three models the damping coefficient increases in a nonlinear manner. Actually, for M363 the damping coefficient reaches a peak at a

Froude number of about 0.33 then decreases again as the velocity is increased. This may be attributed to the deterioration in the lift generating mechanism at higher speeds. The reason that damping for the other models does not behave similarly can be attributed to the fact that model M363 has the highest midship section coefficient which may cause separation of the flow and thus a deterioration in the lift generation.

It should also be pointed out that model M366 has superior damping qualities over model M363 when they are moving with forward speed, in spite of the fact that the reverse is true when they are rolling at zero forward speed. This shows that estimating the damping qualities of ship models at zero forward speed can yield misleading results.

4.2.2 The Effect of The Initial Angle of Heel

The effect of the initial angle of heel, at which the free roll decay starts, on the damping coefficient has been studied. The results can also be seen in Figure 4.5 ~ Figure 4.7 for the models M363, M365 and M366, respectively. The initial angle of heel has the greatest effect near zero speed. At zero and near zero speed, roll damping consists of friction, wave and eddy making components. These three components are functions of the roll amplitude. At speeds near, but greater than, zero the contribution of lift to damping moment is small. As the forward velocity increases, lift effects become predominant and roll damping becomes almost independent of the initial heel angle, as seen from the experimental results shown in Figure 4.5 to 4.7. Roll amplitude has the greatest effect on model M363 damping at zero speed. The effect is less in the case of model M365 and still less in the case of model M366. The fact that M363 has a hard chine while M366 has a round bilge explains this behaviour. At zero forward speed, most of the damping of model M363 is vicous

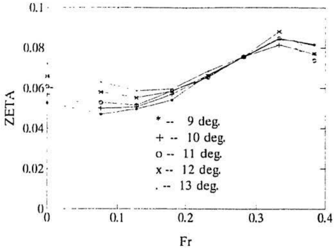

Figure 4.5: Effect of F_r and ϕ_A on roll damping M363

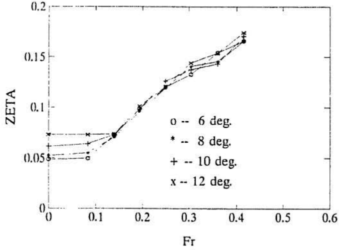

Figure 4.6: Effect of F_r and ϕ_A on roll damping M365

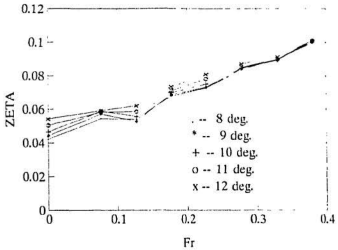

Figure 4.7: Effect of F_r and ϕ_A on roll damping M366

while most of the damping of model M366 is caused by wave generation. Model M365 represents a case in between these two models.

4.2.3 The Effect of Natural Frequency

In the experiments, we changed natural frequency of the model by changing the GM value. As shown in Table 2.3, Table 2.4 and Table 2.5. The square of the natural frequency ω^2 is proportional to GM value. The effect of changing the natural frequency on the equivalent linear damping coefficient is shown in Figure 4.8 to 4.10 for models M363, M365 and M366, respectively. It is seen that the damping is a nonlinear function of the natural frequency. This has been observed in the case of a war ship hull at zero forward speed, see Cumming et al.[5].

It also can been seen in the Figures that the effect of natural frequency on roll damping is not very significant. Generally speaking, the increase of frequency will increase damping in vibration. But for ship rolling, the increase of natural frequency is obtained by the increase of GM value, which will decrease the vertical height of the center of gravity. As a result the magnitude of the damping force arm will be decreased. The combined effect may be the cause that the effect of natural frequency on roll damping is small.

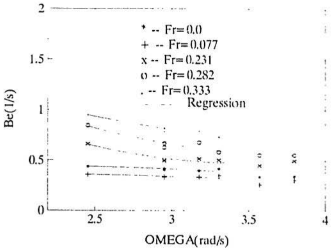

Figure 4.8: Effect of natural frequency on roll damping M363

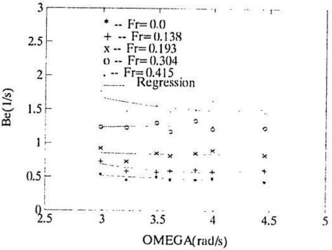

Figure 4.9: Effect of natural frequency on roll damping M365

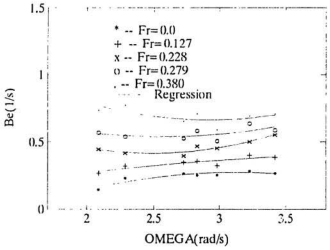

Figure 4.10: Effect of natural frequency on roll damping M366

Chapter 5

Prediction of Roll Damping

The total damping of a small fishing vessel moving with forward velocity can be divided into four components: friction, wave, lift and eddy damping, where eddy damping can be separated into two parts, one caused by the hull and the other caused by the skeg. Each component can be predicted separately[3][20][21]. In this Chapter, the prediction method for each component as proposed by Ikeda and other investigators will be described. Estimated values will be compared with experimental results and suggestion for modification of the present estimating method will be proposed.

5.1 Component Analysis

5.1.1 Friction Damping

Friction Damping is caused by the skin-friction stress on the hull surface. In predicting the value of friction damping, we ignore the effect of waves and regard the ship hull form as an equivalent axisymmetric body. Then the skin friction laws for a flat plate in steady flow are applied to roll motion of the body.

Cited here is Kato's formula modified by Himeno[21](no forward speed):

$$B_{F0} = 0.787 \rho S_f r_f^2 \sqrt{\omega \nu}[1 + 0.00814(\frac{r_s^2 \phi_A^2 \omega}{\nu})^{0.386}] \tag{5.1}$$

the first term in the brackets gives the result for laminar flow, which is used for the naked model hull, while the second term gives the modification for the turbulent flow by Hugh's formula, applicable to both the model hull with bilge keels and the actual ship hull. S_f represents the wetted surface area of the ship hull and r_f the average radius of roll. They can be expressed approximately by the formula:

$$S_f = L(1.7d + C_B B)$$

and

$$r_f = \frac{1}{\pi}\{(0.887 + 0.145C_B)\frac{S}{L} + 2 \cdot \overline{OG}\}$$

In the presence of the forward speed, the friction damping can be expressed as(Tamiya et al.[22]):

$$B_F = B_{F0}(1 + 4.1\frac{U}{\omega L}) \tag{5.2}$$

where B_{F0} represents the friction damping at zero forward speed, which can be predicted by Kato's formula stated above.

5.1.2 Eddy Damping (Naked Hull)

In the absence of ship speed, this component is caused by the flow separation at the bottom of the ship hull near the stem and stern or at the bilge circle near the midship portion. The pressure drop in the separation region gives rise to this damping.

In recent times, it has been found that the drag coefficient of a body in an oscillatory motion varies with the amplitude of the oscillation. The same situation may occur in the case of roll damping. Ikeda et al.[23] investigated this point experimentally for a number of two-dimensional cylinders with ship-like sections. They confirmed through the analysis of the experimental data that the eddy damping coefficient can safely be considered as a constant in case of ship rolling. They further proposed a formula for the eddy damping for ordinary ship hull forms. This can be written in terms of the two dimensional cross-sectional coefficient:

$$B_{E0} = \frac{4}{3\pi}\rho d^4 \omega \phi_A [\frac{r_{max}}{d}]^2 \cdot F[\frac{R}{d}, H_0, \sigma, \frac{\overline{OG}}{d}] \cdot C_p \tag{5.3}$$

where r_{max}, R, σ denote the maximum distance from the center of gravity to the hull surface, bilge radius, area coefficient of the section, respectively. The function F can be determined only by the hull shape and the pressure coefficient C_p by the the ratio of the maximum relative velocity to the mean velocity on the hull surface, $\gamma = v_{max}/v_{mean}$. This can be calculated approximately by a formula given by Ikeda et al., details can be found in reference[3]. The eddy damping for the whole s' g form can be obtained by integrating the sectional values over the ship length.

In the presence of ship forward speed, on the other hand, the separated eddies flow away downstream, with the result that the eddy damping ..ecreases rapidly. In this case, the eddy damping can be corrected by the following emprical formula given by Ikeda et al.[3]

$$B_E = B_{E0t} \cdot \frac{(0.04K)^2}{1 + (0.04K)^2} \tag{5.4}$$

where B_{E0t} is the eddy damping for the whole ship form at zero forward speed and K is the reduced frequency ($K = \omega L/U$).

5.1.3 Lift Damping

As mentioned before, as the forward speed increases the eddy damping decreases rapidly and lift damping prevails. Therefore the lift component becomes the most important part in investigating ship roll damping with forward speed. Yumuro derived a simple formula by applying the lateral force formula used in ship maneuvering research field to the problem of roll damping. This formula was modified by Ikeda et al.[3]. The formula is given in the form of an equivalent linear damping as below

$$B_L = \frac{1}{2}\rho U L d k_N l_o l_R \left(1 + 1.4\frac{\overline{OG}}{l_R} + 0.7\frac{\overline{OG}^2}{l_o l_R}\right) \tag{5.5}$$

where

$$k_N = 2\pi\frac{d}{L} + k(4.1B/L - 0.045) \tag{5.6}$$

$$k = \begin{cases} 0 & C_M \le 0.92 \\ 0.1 \quad \text{for} & 0.92 < C_M \le 0.97 \\ 0.3 & 0.97 < C_M \le 0.99 \end{cases} \tag{5.7}$$

k_N represents the derivative of the lift coefficient of the hull towed obliquely. l_o is the lever defined in such a way that the quantity $l_o\dot{\phi}/U$ correspondents to the incidence angle of the lifting body. l_R denotes the distance from the roll center to the center of lift force. l_o and l_R were given by Ikeda et al. as

$$l_o = 0.3d \quad , \quad l_R = 0.5d \tag{5.8}$$

According to Ikeda's formula, the lift damping is linear, proportional to ship speed and independent of roll amplitude.

5.1.4 Wave Damping

In the case of zero Froude number, the wave damping can be obtained by using the strip method. In this paper,a subroutine of the program SHIPMO[8][24] developed by National Defence Department based on the Close-fit theory was used to calculate the wave damping at zero forward speed.

In the presence of ship speed, it is quite difficult to calculate the wave roll damping theoretically. Ikeda et al calculated the energy loss in the far field due to a pair of horizontal doublets and compared the results with experiments for models of combined flat plates. Through these elementary analyses they proposed an empirical formula for roll damping of ordinary ship forms:

$$\frac{B_w}{B_{w0}} = 0.5[\{(A_2+1)+(A_2-1)\tanh 20(\tau-0.3)\}+(2A_1-A_2-1)\exp\{-150(\tau-0.25)^2\}] \tag{5.9}$$

where $A_1 = 1 + \xi_d^{-1.2} \cdot e^{-2\xi_d}$

$$A_2 = 0.5 + \xi_d^{-1} e^{-2\xi_d}$$

$$\xi_d = \omega^2 d/g \quad , \quad \tau = v\omega/g$$

The terms $A1$ and $A2$ represent the maximum at the point $\tau = 1/4$ and the constant value of B_w/B_{w0} where the value of τ is large. The term B_{w0} stands for the value at zero forward speed.

5.1.5 Eddy Damping due to the Skeg

Most small fishing vessels have skegs to improve their manoeuvrability performance and for the convenience when they are docked on the slipway. Ikeda et al. found that the skeg decreases wave damping and increases eddy damping[20]. They attributed the decrease of wave damping to the fact that the phase of the wave created by the skeg is much different from that created by the main hull.

The creation of eddies at the edge of the skeg leads to an increase in eddy damping. Eddy damping due to a skeg can be divided into two components. One is the normal force component which is created by the pressure variation on a skeg. The other is the hull surface pressure component which is created by the pressure variation on the main hull surface due to the skeg. The normal force component is always positive, while the surface pressure component may be negative.

Ikeda et al.[20] proposed a simple prediction method of eddy component of roll damping due to skeg. A simple pressure distribution on the skeg and on the bottom of a vessel is assumed as shown in Figure 5.1. The pressure coefficient C_{PF} and C_{PR} on the front and the back faces the skeg and the length of the negative pressure

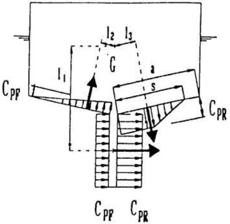

Figure 5.1: Pressure distribution due to a skeg(after Ikeda et al.[20])

region S are assumed on the basis of the experimental results as follows

$$\begin{aligned} C_{PF} &= 1.2 \\ C_{PR} &= -3.8 \\ S &= 1.65 l_s K_c^{2/3} \end{aligned} \tag{5.10}$$

where l_s denotes the length of a skeg, K_c is Keulegan-Carpenter number defined as $U_{max} \cdot T/l_s$, where U_{max} denotes the maximum speed of the edge of the skeg, T the period of the roll motion. Strictly, the value of C_{PR} depends on K_c number, but it is assumed to be constant for simplicity. Integrating the assumed pressure on the skeg and on the hull surface, the roll damping moment M_r for unit length of the hull section can be obtained as follows

$$M_r = \frac{1}{2}\rho U_{sk}^2\{(C_{PF} - C_{PR})l_s l_1 - \frac{1}{2}C_{PF} a l_2 + \frac{3}{4}C_{PR} S l_3\} \tag{5.11}$$

where U_{sk} denotes the velocity of the edge of the skeg, and l_1, l_2 and l_3 the moment levers as shown in Figure 5.1. The equivalent linear damping of the skeg can then be expressed as

$$B_{sk} = \frac{4}{3\pi}\rho\omega\phi_A l_4^2[(C_{PF} - C_{PR})l_s l_1 - \frac{1}{2}C_{PF} a l_2 + \frac{3}{4}C_{PR} S l_3] \tag{5.12}$$

where l_4 is the distance from the center of gravity to the edge of the skeg.

The experimental study by Ikeda et al. indicates that the above formula predicts the eddy damping due to the skeg very well.

5.2 Comparison of Predicted Values with Experimental Results

As stated above, the roll damping of a fishing vessel is composed of four components, namely, friction damping, wave damping, lift damping and eddy damping for naked hull and skeg. These components are calculated using the formulae described in the last section. As many as possible estimated values have been calculated and compared with the experimental data. It was found that the results are quite similar for different conditions (GM values and roll amplitudes) for a specific model. Two figures are selected for each model as shown in Figure 5.2 and Figure 5.3 for M363 , Figure 5.4 and Figure 5.5 for M365 and Figure 5.6 and Figure 5.7 for M366.

At zero forward speed, the predicting method gave quite accurate results for M365 and M366, but lower estimates for M363. As stated previously, M363 has a hard chine which will increase the eddy damping. Ikeda et al.[20] investigated the effect of the hard chine and draw the conclusion that it has twice the value as that calculated for a round bilge vessel, but the results of the present work show that the difference is larger than that. The main reason is that the estimated value obtained by Ikeda's formula gave a very small value for the eddy damping, which suggests that an accurate estimated method of eddy damping for a vessel with hard chine is still lacking. On the other hand, the estimated formula of the skeg proposed by Ikeda gave quite reasonable results based on the fact that the estimating method predicted good results for M365 and M366 at zero forward speed.

In the presence of forward speed, the predicted method over estimates the roll damping for all three models and the difference becomes larger with the increasing

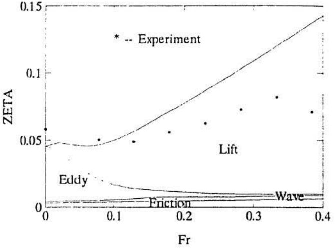

Figure 5.2: Predicted results using Ikeda's formula M363 GM=5.33cm

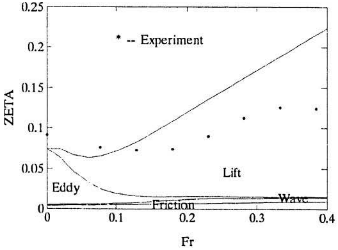

Figure 5.3: Predicted results using Ikeda's formula M363 GM=3.12cm

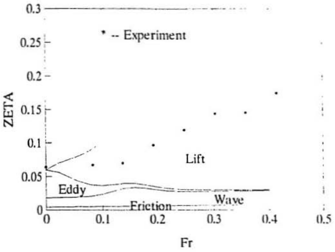

Figure 5.4: Predicted results using Ikeda's formula M365 GM=5.22cm

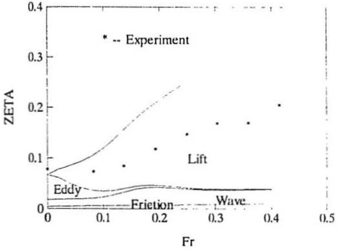

Figure 5.5: Predicted results using Ikeda's formula M365 GM=3.96cm

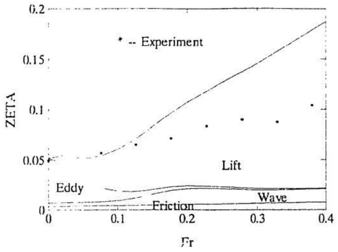

Figure 5.6: Predicted results using Ikeda's formula M366 GM=3.82cm

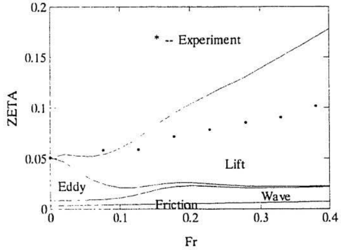

Figure 5.7: Predicted results using Ikeda's formula M366 GM=4.35cm

forward speed as shown in the Figures. This leads us to conclude that Ikeda's formula over estimates lift damping. A modification of Ikeda's formula is therefore suggested based on a phenomenon observed during the experiments.

5.3 Modification of Ikeda's Formula

During the experiments, it was observed that the model sinkage increases with the increase in forward speed. This can be explained easily by basic theory of Fluid Mechanics. When the ship moves in the water, the fluid speed on the ship hull will increase, which creates a low pressure area under the ship hull. The ship's draft increases with forward speed to balance the decreasing pressure under the ship hull. As a result, the distance between the rolling center and the water level $\overline{OG}$ will decrease with the increase of forward speed.

As stated above, all damping components have relationship with $\overline{OG}$ value, but at high forward speed, only lift and wave component need to be considered. Calculations have been done to investigate the effect of $\overline{OG}$ value on the estimation of wave and lift damping. As shown in Figure 5.8, wave damping at zero forward speed shows a minimum around OG/d of 0.15 to 0.25. . Therefore, the effect of a change in $\overline{OG}$ value on wave damping is uncertain, depending on the value of OG/d. The calculation of the lift component shows that lift damping is proportional to $\overline{OG}$ value, which can be explained by the fact that the lever arm of the lift moment is proportional to $\overline{OG}$ value.

The relationship between $\overline{OG}$ and forward speed may be found using the laws of hydrodynamics. Because of the lack of research work in this field, we simply assume that the relationship is linear, i.e.

$$\overline{OG} = \overline{OG}_0 - b \cdot F_r \tag{5.13}$$

where $\overline{OG}_0$ is the $\overline{OG}$ value at zero forward speed, b is the slope and F_r is Froude

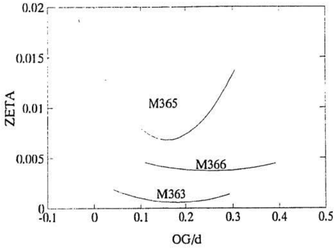

Figure 5.8: Effect of $\overline{OG}$ values on Wave Damping

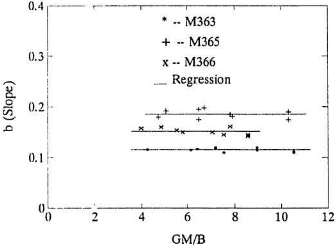

Figure 5.9: $\overline{OG}$ slopes predicted from experimental data

number. For the lift damping component, the modification can be made by replacing $\overline{OG}$ in equation 5.5 by equation 5.13 i.e.

$$B_L = \frac{1}{2}\rho U L d k_N l_o l_R (1 + 1.4\frac{\overline{OG}_0 - b \cdot F_r}{l_R} + 0.7\frac{(\overline{OG}_0 - b \cdot F_r)^2}{l_o l_R}) \qquad (5.14)$$

For the wave damping component of a model moving with forward velocity, one can calculate $\overline{OG}$ values using equation 5.13, then use this $\overline{OG}$ value to calculate the wave damping at zero forward speed B_{w0} using strip theory and, finally calculate B_w using eq. 5.9.

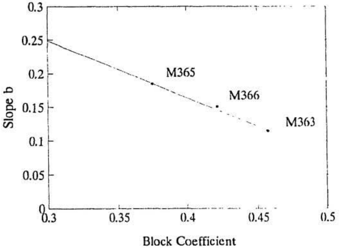

Figure 5.10: Relationship between slope b and block coefficient

The slope b is determined by comparing the experimental data with the estimated values. The search is done automatically by a program which finds a value for b that makes the predicted curve closest to the experimental data. The calculation has been carried out for different GM values. For each GM value, only one or two initial angles have been selected, considering the fact that the effect of initial angle is small at high forward speed as stated in chapter 4. The results are shown in Figure 5.9. It was found that b is almost constant for each model which gives a conclusion that b is only a function of ship form. For M363 b is about 0.115, for M365 0.185 and for M366 0.151. It has been found that b has a linear relationship with the block coefficient C_B as shown in Figure 5.10. The relationship can be expressed as:

$$b = -0.8485C_B + 0.5032 \tag{5.15}$$

The final modification of Ikeda's formula can be expressed as:

$$\begin{aligned} B_L = \frac{1}{2}\rho U L d k_N l_o l_R \{1 &+ 1.4\frac{\overline{OG}_0 - (-0.8485C_B + 0.5032) \cdot F_r}{l_R} \\ &+ 0.7\frac{[\overline{OG}_0 - (-0.8485C_B + 0.5032) \cdot F_r]^2}{l_o l_R}\} \end{aligned} \tag{5.16}$$

The modified results are shown in Figure 5.11 ~ Figure 5.16. Three curves are given in each Figure. One is the curve predicted by Ikeda's formulae. Second one is the curve in which the lift damping has been modified. The third one is the curve in which both wave and lift damping have been modified. For M363 and M366 the second and third curves are quite close, but for M365 the third curve fits the experimental data better, which suggests that both the lift and the wave components need to be modified, especially for ship forms with large value of wave damping component, such as M365. As shown in the figures, the modified curves fit the experimental data much better than the curve predicted by the original Ikeda's formulae. For M366, the modified curves give a perfect fit on the experimental data. For M363 the modified curves fit the experimental data very well except at zero and low forward speed values where eddy damping is under estimated as stated before. For M365 the modified curves still have some difference with experimental data in the range of $0.05 < F_r < 0.25$, which suggests that the relationship between $\overline{OG}$ value and forward speed may not be linear for some ship forms. Actually it was noticed in the experiment that M365 has larger sinkage than the other two model, which may be attributed to the flat bottom of M365.

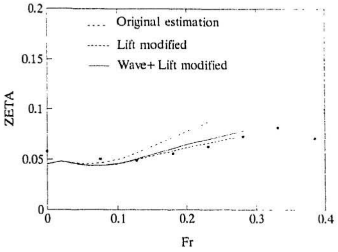

Figure 5.11: Predicted results using Modified Ikeda's formula M363 GM=5.33cm

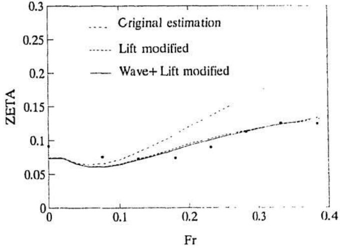

Figure 5.12: Predicted results using Modified Ikeda's formula M363 GM=3.12cm

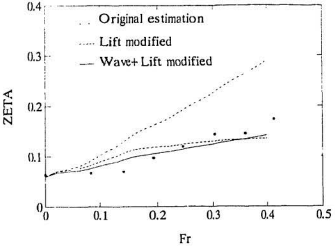

Figure 5.13: Predicted results using Modified Ikeda's formula M365 GM=5.22cm

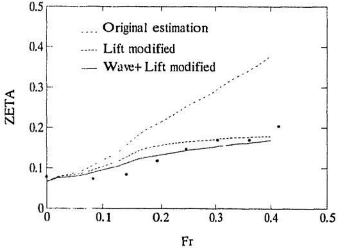

Figure 5.14: Predicted results using Modified Ikeda's formula M365 GM=3.96cm

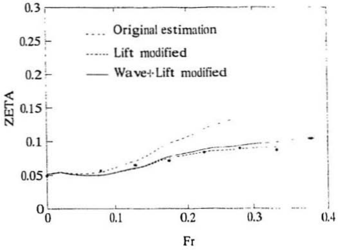

Figure 5.15: Predicted results using Modified Ikeda's formula M366 GM=3.82cm

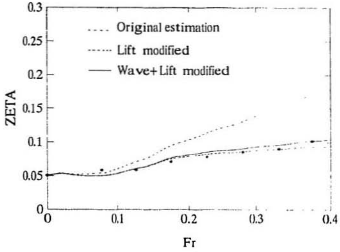

Figure 5.16: Predicted results using Modified Ikeda's formula M366 GM=4.35cm

Chapter 6

Effect of Following Waves on Roll Damping

An experiment has been done to investigate the effect of following waves on roll damping. When a ship is moving in a following wave, the water surface around the ship will change with the transmission of the wave, which will generate heave and pitch and affect the behaviour of roll motion. An experiment was designed to investigate this effect. The details will be stated in the following sections.

6.1 The Experiment

The experiment setup is shown in Figure 6.1. The model was only allowed two degrees of freedom — roll and heave. A dynamometer was used to measure the motion in roll and heave. The rolling center was adjusted to the same level of the center of gravity. An arm was fitted on the model to generate initial angles.

Model M363 was used in this experiment. The principal dimension of the model is listed in Table 2.1. A regular wave was transmited along the model from the stern to the bow. The wave length was taken to be equal to the length of the model, i.e., $\lambda_w = 1.551m$ and the wave period T_w is 1.0 second. Four different wave heights were used in the experiment. These have nominal values of 0.0cm, 5.0cm, 7.0cm and 9.0cm. A probe was set 4.22m away from the midship to measure the incident

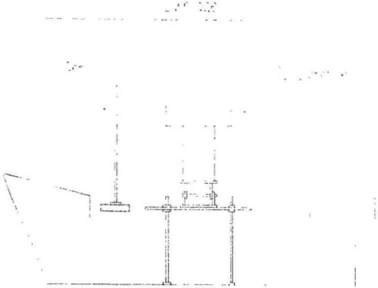

Figure 6.1: Experimental Setup upon (following wave)

wave. The measured wave height is not usually the same as the nominal value. The comparison of the nominal values and measured values is in Table 6.1. The GM values, natural frequencies, periods of rolling and parameters of restoring moment are listed in Table 6.2. The determination of these values have been explained in Chapter 2.

Table 6.1: Wave height in the experiment

Nominal value(cm)	5.0	7.0	9.0
Measured value(cm)	3.68	6.12	7.31

Three channels were used to collect the data, one for rolling, the second for heaving and the third for the incident wave. An example of the collected data is

Table 6.2: Experimental Parameters for M363 (following wave)

I.D.	GM(cm)	ω	$T_r(sec.)$	μ_1	μ_2
4	5.04	3.75	1.68	0.8556	-1.1500
5	4.07	3.35	1.88	1.1080	-1.4402
6	3.13	2.94	2.14	1.4966	-1.8810
7	2.02	2.34	2.69	2.4529	-2.9807

shown in Figure 6.2 to Figure 6.4. The phase of the wave was measured with respect to a coordinate whose origin is located at the midship. The phase of the incident wave was found to be close to the phase of the heave motion and they have same period(1.0 second).

The same data processing and management technique presented in Chapter 3 and Chapter 4 were used to process the data. For each GM value, there are 7 initial angles and 4 wave heights. The wave height can be treated as the same as forward speed. The programs presented in the Chapter 3 can be directly used for processing the data here. The analytical results are stored in the same database discussed in Chapter 4 and analysis was done by the help of the same data management system. A few new procedures were introduced in the system to meet the specific need of this experiment.

6.2 Analysis and discussion

The effect of following wave on the equivalent linear damping coefficient is shown in Figure 6.5 to 6.8, respectively. These figures show that damping coefficient in waves are larger than those measured in calm water. There may be three reasons for this. First, damping parameters were estimated neglecting time variations in the restoring moment. This may have caused a false increase in the values of the damping parameter. Second, the increase in the surface area of the model as a result of the passing wave. Third, the effect of coupled heave motion. As can also be seen from the figures, there is much scatter in the results. The scatter for $GM = 5.04cm$ is more pronounced than the other two cases. For the $GM = 5.04cm$ the roll natural period is almost twice the wave period which may indicate a parametric resonance effect.

This is a very preliminary investigation. Further research work needs to be done in the following aspects:

- The mathematical modeling for identification of roll damping coefficients needs to be improved. The method of roll damping identification stated in Chapter 3 is used in this analysis. As we can seen in equation 3.1, the restoring moment is considered to be independent of time. This is correct in still water condition. When ship moves in waves, the restoring moment is changing with time. Therefore a new function $D(\phi, t)$ has to be found to express the restoring moment of a ship moving in regular waves.

- The phase difference of the incident wave and roll motion may have an effect on roll damping. This should be investigated in experiment, which can be done by collecting the roll decay curves in the same conditions(GM value and initial angle) for many times and comparing the results.

- Effect of parametric resonance should be investigated.

- Effect of the coupling of heave and pitch into roll should be investigated.

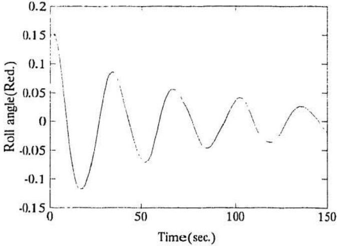

Figure 6.2: Roll decay curve with following wave

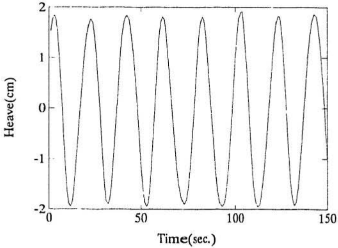

Figure 6.3: Record of heave motion with following wave

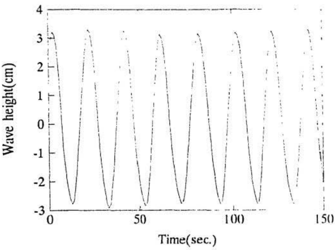

Figure 6.4: Incident wave

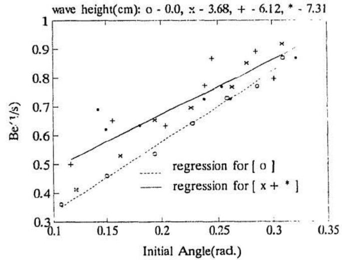

Figure 6.5: Effect of following wave on roll damping, GM=5.04

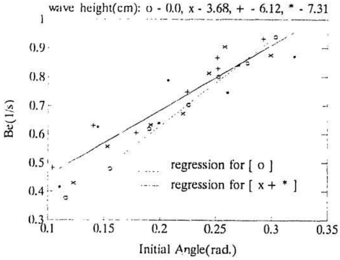

Figure 6.6: Effect of following wave on roll damping, GM=4.07

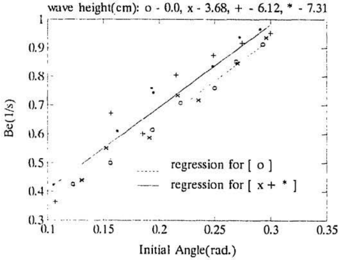

Figure 6.7: Effect of following wave on roll damping, GM=3.13

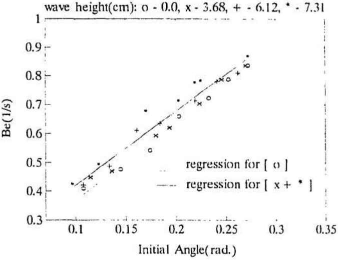

Figure 6.8: Effect of following wave on roll damping, GM=2.34

Chapter 7

Conclusions

A method which combines the Energy and the Modulation Function methods was used to analyse the decay curves obtained from the free roll decay experiments using models of three small fishing vessels. This method has proved to give better estimates for the roll damping parameters than Energy method.

A database system created using VAX-Pascal is used to store the analytical results and perform various kinds of analyses. It has been shown that this system is very useful in the analysis of the experimental results.

Ikeda's method was used to predict the roll damping coefficient. The estimated values predicted by Ikeda's method do not fit the experiment data well. The main reason is that Ikeda's method over estimates roll damping at higher forward speed. In addition, Ikeda's method is not suitable for estimating the eddy damping of a ship form with hard chine. It was observed that the model's sinkage increases with the increase of forward speed. A modification of Ikeda's formula has been proposed based on the modification of the distance between the center of gravity and the water line. The values predicted by the modified Ikeda's formula give better fit to the experimental data. The modified Ikeda's formula was proposed based on the experiment of the three models of fishing vessels. Further research is needed for other ship forms.

The following conclusions can be obtained from the results of the analysis of the experimental data:

1. At higher forward speed, roll damping is nonlinear for all three models.
2. For M363 it has been found that the damping coefficient reaches a peak at a Froude number of about 0.33 then decreases again as the velocity is increased. This may be attributed to the deterioration in the lift generating mechanism at higher spc ds and the high values for the midship section coefficient of M363.
3. The effect of initial heel angle is strong at zero and low forward speed but becomes weak when the forward speed increases, which indicates that the lift damping can be considered to be independent of the roll amplitude.
4. The effect of the natural frequency on roll damping is not very strong, which may be caused by the fact that the increase of the natural frequency decreases the magnitude of the damping force arm.
5. Roll damping in following wave is an area where much work is still needed.

References

[1] Barr, R.A. and Ankudinov, V., "Ship Rolling, its Prediction and Reduction Using Roll Stabilization", Marine Technology, Vol. 14, No. 1, pp. 19-41, 1977.

[2] schmitke, R.T., "Ship Sway, Roll and Yaw Motions in Oblique Seas", Transactions of the Society of Naval Architects and Marine Engineers, Vol. 86, pp. 26-46, 1978.

[3] ikeda, Y., Himeno, Y. and Tanaka, N., "A Prediction Method for Ship Roll Damping", Report No. 00405 of Department of Naval Architecture, University of Osaka Prefecture, 1978.

[4] Crane, C.L., Eda, H. and Landsburg, A.C., CONTROLLABILITY, Chapter 9 of PRINCIPLES OF NAVAL ARCHITECTURE, Vol. III, Lewis, E.V. editor, Published by the Society of Navval Architects and Marine Engineers, New Jersey, 198?

[5] Cumming, D., Haddara, M.R. and Graham, R., "Experimental Investigation of Roll Damping Characteristics of Destroyer Model", Proceedings of the Fouth International Conference on Stability of Ships and Ocean Vehicles, Napoli, Italy, pp. 159-166, 1990.

[6] Blok, J.J. and Aalberts, A.B., "Roll Damping Due to Lift Effects on High Speed Monohulls", Proceedings of the First International Conference on Fast Sea Transportation, FAST'91, Trondheim, Norway, pp 1331-1348, 1991.

[7] Shuford Jr., C.L., "A Theoretical and Experimental Study of Planning Surfaces Including Effects of Cross-Section and Plan Form", Report 1355, National Advisory Committee for Aeronautics, Langley Aeronautical Lab, 1958.

[8] Graham, R., "SHIPMO3: Improved Viscous Roll Damping Prediction for the SHIPMO computer program", Technical Report of Defence Research Establishment Atlantic, 86/211, May 1986.

[9] Haddara, M.R. and Leung, S.K., "Experimental investigation of the lift component of roll damping", to be published in Ocean Engineering.

[10] Haddara, M.R. and Bennett, P., "A Study of the Angle Dependence of Roll Damping Moment", Ocean Engineering, Vol. 16, No. 4, pp. 411-427, 1989.

[11] Shinbrot, M., "On the Analysis of Linear and Nonlinear Dynamical Systems from Transient Response Data", NACA TN 3288, 1954.

[12] Haddara, M.R. and Wu, X., "Parameter Identification of Nonlinear Rolling Motion in Random Seas", to be published in the International Shipbuilding Progress.

[13] Rumble Jr., J.R. and Hampel,V.E., "Database Management in Science and Technology", Published by Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1984.

[14] Davis, B., "Database in Perspective", NCC Puplications, Manchester, 1980

[15] Atre, S., "Data Base: Structured Techniques for Design, Performance, and Management", Published by A Wiley-Interscience Publication, New York, 1980.

[16] Cherry, G. W., "Pascal Programming Structures - an introduction to systemetic programming", Published by Reston Publishing Company, Virginia, 1980.

[17] Digital Equipment Corporation, "VAX Pascal User Manual", Order Number:AA-H485F-TE, Maynard, Massachusetts, Dec. 1989.

[18] Digital Equipment Corporation, "VAX Pascal Reference Manual", Order Number:AA-L369D-TE, Maynard, Massachusetts, Dec. 1989.

[19] Cox, G.G. and Lloyd, A.R., "Hydrodynamic Design Basis for Navy Ship Roll Stabilization", Transactions of Society of Naval Architecture and Marine Engineers, vol. 85, pp. 51-93, 1977.

[20] ikeda, Y., Tanaka, N. and Himeno, Y., "Effect of hull form and appendage on roll motion of small fishing vessel", Second Int. Conf. on Stability of Ships and Ocean Vechicles, Tokyo, Oct. 1982.

[21] Himeno, Y., "Prediction of ship roll damping — state of art", Dept. of Naval Architecture and Marine Engineering, University of Michigan, Report No. 239, 1981.

[22] Tamiya, M., and Komura, T., "Topics on Ship Rolling Characteristics with Advanced Speed", JSNA Japan, Vol 132, 1972.

[23] ikeda, Y., Himeno, Y., and Tanaka, N., "On Eddy Making Component of Roll Damping Force on Naked Hull", JSNA Japan, Vol. 142 , 1977.

[24] Graham, R. and Trudelle, C., "SHIPMO4: An Update User's Manual for the SHIPMO Computer Program Incorporating an Extended Hydrostatics Capabil-

ity and An Improved Viscous Roll Damping Model", Technical Report of Defence Research Establishment Atlantic, 87/304, March 1987.

[25] Gerald, C.F. and Whertley, P.O., "Applied Numerical Analysis", Published by Addison-Wesley Publishing Company, California, 1984.

Appendix A. Content of File S0507

```
* Roll Damping tests
DATE: Tue Nov 03 14:47:10 1992
SCALE_HI: 10.00000
SCALE_LO: -10.00000
SAMPLE: 20.00
NUMCHAN: 3
TOTIME: 9.000000
Chan Gain    Offset         Slope
OFF_SLOPE:
1    5    -7.58437619E+001 2.31460325E-003 Roll (deg)
2    5    -7.58437619E+001 2.31460325E-003 Pitch (deg)
3    2    -4.92759072E+000 1.49645144E-004 Speed(m/s)
NUMPOINTS 181
END
22041 31612 36292
22397 31623 36279
22952 31631 36290
23840 31613 36268
24872 31607 36279
26068 31573 36295
27408 31539 36271
28794 31491 36277
30178 31457 36268
31563 31408 36278
32890 31348 36271
34170 31306 36280
35393 31279 36271
36476 31248 36274
37427 31236 36255
38268 31219 36264
38929 31224 36240
39427 31222 36244
39745 31237 36249
39864 31252 36273
39824 31264 36261
39600 31266 36254
39234 31278 36272
38737 ...
```

Appendix B. Content of File S0507.AGL

```
-0.43332 -0.04668 0.50333
-0.41894 -0.04623 0.50139
-0.39652 -0.04591 0.50303
-0.36065 -0.04664 0.49974
-0.31896 -0.04688 0.50139
-0.27064 -0.04825 0.50378
-0.21651 -0.04963 0.50019
-0.16052 -0.05157 0.50109
-0.10461 -0.05294 0.49974
-0.04866 -0.05492 0.50124
0.00495 -0.05734 0.50019
0.05666 -0.05904 0.50153
0.10606 -0.06013 0.50019
0.14981 -0.06138 0.50064
0.18823 -0.06187 0.49779
0.22221 -0.06256 0.49914
0.24891 -0.06235 0.49555
0.26903 -0.06243 0.49615
0.28187 -0.06183 0.49690
0.28668 -0.06122 0.50049
0.28506 -0.06074 0.49869
0.27602 -0.06066 0.49764
0.26123 -0.06017 0.50034
0.24115 -0.06021 0.49914
0.21695 -0.06078 0.49989
0.18965 -0.06062 0.50034
0.15810 -0.06005 0.50109
0.12590 -0.06138 0.49989
0.09112 -0.06138 0.50064
0.05565 -0.06130 0.50034
0.02074 -0.06126 0.50079
-0.01388 -0.06082 0.50288
-0.04781 -0.05981 0.49974
-0.07839 -0.05944 0.49854
-0.10691 ···
```

Appendix C. Content of File S0507.USE

0.05
2.922418
1.0
-1.5131
-1.9140
0.2869402
0.2853242
0.2762752
0.2614901
0.2414121
0.2172141
0.1899062
0.1583551
0.1261581
9.1376141E-02
5.5907138E-02
2.1004137E-02
-1.3616863E-02
-4.7549862E-02
-7.8130864E-02
-0.1066519
-0.1337589
-0.1558959
-0.1741159
-0.1881329
-0.1977479
-0.2035649
-0.2049789
-0.2007369
-0.1920119
-0.1819929
-0.1670459
-0.1482609
-0.1277389
-0.1048739
-8.1362866E-02
...

Appendix D. Program TRSBAT.PAS

```
{ TRSBAT.PAS
  Developed by Suixin Zhang, Sept. 1992, MUN.

  Translation of Experimental Data. Original experimental
  data is interges which indicate the amount of  valteges
  measured by the equipments, such as gyroscope. The Translation
  is done by the formula:
              A = I*S + O
  where I is the integer in the file, S is the slope and O is the
  offset
}

program trsbat(input, output, newf, oldf, nf);
const
  n_angle=7;
  n_speed=9;
type
  tname = packed array[1..35] of char;
var
  newf, oldf, nf: text;
  angle: array[1..n_angle] of string(2);
  speed: array[1..n_speed] of string(2);
  i, j: integer;
  in_f, out_f: tname;
  X: char;
  X1: string(4);

{subroutine for translation of experimental data}
procedure trsl(in_f: tname; out_f: tname);
var
  i, mm : integer;
  a1, a2, b1, b2, c1, c2, l1, l2, l3: real;
begin
 {Open input and output file}
  open(oldf, in_f, history:=old);
  open(newf, out_f,  history:= unknown);
  reset(oldf);
  rewrite(newf);
 {Skip 9 lines of message}
  for i:=1 to 9 do
    readln(oldf);
 {Read offset and slope}
  read(oldf, mm);
  read(oldf, mm);
  read(oldf, a1);
  read(oldf, a2);
  readln(oldf);

  read(oldf, mm);
  read(oldf, mm);
  read(oldf, b1);
  read(oldf, b2);
  readln(oldf);
```

```
  read(oldf, mm);
  read(oldf, mm);
  read(oldf, c1);
  read(oldf, c2);
  readln(oldf);

  readln(oldf);
  readln(oldf);

  while not eof(oldf) do
  begin
   {Read data of roll, picth and forward speed}
    read(oldf, l1);
    read(oldf, l2);
    readln(oldf, l3);
   {Do translation}
    l1:=l1*a2 + a1;
    l2:=l2*b2 + b1;
    l3:=l3*c2 + c1;
    l1:=l1*3.14159/180.0;
    {Output results}
    writeln(newf, l1:12:6, ' ', l2:9:5, ' ', l3:9:5,' ');
  end;
  close(oldf);
  close(newf);
end;

{Main program}
begin
{ 7 initial angle numbers }
  angle[1]:='01';
  angle[2]:='02';
  angle[3]:='03';
  angle[4]:='04';
  angle[5]:='05';
  angle[6]:='06';
  angle[7]:='07';
{ 8 forward speed}
  speed[1]:='00';
  speed[2]:='03';
  speed[3]:='05';
  speed[4]:='07';
  speed[5]:='09';
  speed[6]:='11';
  speed[7]:='13';
  speed[8]:='15';

  open(nf, 'trsbat_n.dat', history:=old);
{ trsbat_n.dat contains I.D. and name of subdirectory
  of the files. Example of contect:

  A
```

```
  T363
  B
  T363
  H
  T365
  Y
  T366
  ...
}
  reset(nf);

  while not eof(nf) do
  begin
    readln(nf, X); { read I.D. }
    readln(nf, X1); { read name of sub-directory}
    speed[9]:=substr(X1, 3,2);
   {forward seed number for test without joint,
    63 for M363, 65 for M365 and 66 for M366 }

    writeln(speed[9]);
    for i:=1 to n_speed do
    begin
      for j:=1 to n_angle do
      begin
       {Compose input file name with sub-directory}
        in_f:='[grad.szhang.roll.'+X1+']'+ X +
               speed[i] + angle[j] + '.';
        writeln(X+speed[i]+angle[j]);

       {Compose output file name with sub-directory}
        out_f:='[grad.szhang.roll.'+X1+']'+ X +
                speed[i] + angle[j] + '.agl';
        trsl(in_f, out_f);
      end;
    end;
  end;
end.
```

Appendix E. Program DAMABAT.FOR

```
*       DAMABAT.FOR
*       Rearragement of data
*       VAX-FORTRAN
*
*       Suimin Zhang
*       Oct. 1992, Engineering, M.U.N.

        real Y(600), NH(50), NO(50), XH(50), XX(5), YY(5), MM0, MM1, MM2
        character filen*5, in_f*35, out_f*35

*       Input coefficient of restoring moment
        print*, 'input MM1, MM2'
        read*, MM1, MM2

*       HH is the interval of measurement
        HH = 0.05

        open(7, file='N_FILE1.DAT', status='old')
*       N_FILE1.DAT contain name of the input file
*       produced by NAMEMK.PAS

        open(17, file='N_FILE2.DAT', status='old')
*       N_FILE2.DAT contain name of the output file
*       produced by NAMEMK.PAS

        do while (.true.)
          read(7, '(A)', end=111) in_f
          read(17, '(A)', end=111) out_f
          open(9, file=in_f, status='old')
          open(8, file=out_f, status='unknown')
          READ(9, *) Y(1)

          IF (Y(1) .GT. 0) THEN
            TU=-1.0
          ELSE
            TU= 1.0
          END IF
          Y(1) = TU*Y(1)
          I=1
          do while (.true.)
            I=I+1
            read(9,*, END=11) Y(I)
            Y(I) = TU*Y(I)
          end do
 11       close(9)
          N=I-1
          PRINT*, 'N=', N

*         Find amplitude of half cycle
          CALL DASEL(Y,N,XH,NH,NO,3)

          XX(1)=NH(1)
          XX(2)=NH(3)
```

```
        XX(3)=NH(5)
        YY(1)=XH(1)
        YY(2)=XH(3)
        YY(3)=XH(5)
        XINT=NH(2)

        CALL LAGINT(XX, YY, 3, XINT, YOUT)
        E1 = XH(2) + YOUT
        DO I=1,N
          Y(I)=Y(I)-E1/2.0
        END DO

        TT=HH *( (NO(3) -NO(1)) + (NO(4) - NO(2)) )/ 2.0
        WW=6.2832/TT

        WRITE(8,*) HH
        WRITE(8,*) WW
        WRITE(8,*) MM0
        WRITE(8,*) MM1
        WRITE(8,*) MM2
        DO I = NH(1), N
          WRITE(8,*) Y(I)
        END DO

        close(8)
        close(9)
      end do
 111  close(7)
      end


*     Find the amplitude of half cycle of
*     the decay curve

      SUBROUTINE DASEL(Y,N,XH,NN,NO,NUB)
      REAL Y(*), XH(*), NN(*), NO(*)

      IF (Y(1).LT. 0.0) THEN
         KID=1
      ELSE
         KID=2
      END IF

      IF (KID.EQ. 1) THEN
        SIGN=1.0
      ELSE
        SIGN=-1.0
      END IF

      X0=SIGN*Y(1)

      NSTEP=1
```

```
      DO I=1,N

          IF ( SIGN*Y(I) .GT. X0) THEN
            X0=SIGN*Y(I)
            N0=I
          END IF

          IF ( (SIGN*Y(I) .GT. 0.0).AND. (NSTEP .EQ.1) ) THEN
              NSTEP=2
              J=1
              NO(J)=I
          END IF

          IF ((SIGN*Y(I) .LE. 0.0) .AND. (NSTEP .EQ. 2).AND.
     &     ( (N0-NN(J-1)) .GT. NUB) ) THEN
             XH(J)=Y(N0)
             NN(J)=N0
             J=J+1
             NO(J)=I
             SIGN=-1.0*SIGN
             X0=SIGN*Y(I)
          END IF
          IF ( (I.EQ.N) .AND. (SIGN*Y(I) .GT. 0.0).AND. (SIGN*Y(I)
     &       .LT. X0)) THEN
             XH(J)=X0/SIGN
             NN(J)=N0
             J=J+1
             SIGN=-1.0*SIGN
             X0=SIGN*Y(I)
          END IF
        END DO
        NH=J-1
        XH(J)=0.0
        XH(J+1)=0.0
        XH(J+2)=0.0
      END


*     Program  cited from [25]
      SUBROUTINE LAGINT(X,Y,N,XINT,YOUT)
C       THIS SUBROUTINE PERFORMS LAGRANGIAN
C     INTERPOLATION WITHIN A SET OF (X,Y) PAIRS TO GIVE THE Y
C     VALUE CORRESPONDING TO XINT. THE DEGREE OF THE INTERPO-
C     LATING POLYNNOMIAL IS ONE LESS THAN THE NUMBER OF
C     POINTS SUPPLIED
C
C     X  -- ARRAY OF VALUEES OF THE INDEPENDENT VARIABLE
C     Y  -- ARRY OF FUNCTION VALUES CORRESPONDING TO X
C     N  -- NUMBER OF POINTS
C     XINT -- THE X VALUE FOR WHICH ESTIMATE OF Y IS DESIRED
C     YOUT -- THE Y VALUE RETURNED TO CALLER
C
```

```
      REAL X(N), Y(N), XINT, YOUT, TERM
      YOUT = 0.0
      DO I = 1,N
        TERM = Y(I)
        DO J = 1,N
          IF (I .NE. J) THEN
            TERM = TERM * ( XINT - X(J) )/( X(I) - X(J) )
          END IF
        END DO
        YOUT = YOUT + TERM
      END DO
      END
```

```
{A program developed in Pascal to
 compose the name of files
 I.D. and name of subdirectory are needed

 This program needs to be compiled separately from the
 Fortran codes list above}

program name(input, output);
const
  n_angle=7;
  n_speed=9;
type
  tname = packed array[1..35] of char;
var
  oldf, outf, nf: text;
  angle: array[1..n_angle] of string(2);
  speed: array[1..n_speed] of string(2);
  i, j: integer;
  in_f, out_f: tname;
  X: char;
  X1: string(4);

begin
  angle[1]:='01';
  angle[2]:='02';
  angle[3]:='03';
  angle[4]:='04';
  angle[5]:='05';
  angle[6]:='06';
  angle[7]:='07';
  speed[1]:='00';
  speed[2]:='03';
  speed[3]:='05';
  speed[4]:='07';
  speed[5]:='09';
  speed[6]:='11';
  speed[7]:='13';
```

```
speed[8]:='15';
speed[9]:='63';
write('input EXP_ID:');
readln(X);
write('input name of subdirectory:');
read(X1);

{open file for storing the file name with *AGL extension}
open(nf, 'n_file1.dat', history:=old);
rewrite(nf);

{open file for storing the file name with *USE extension}
open(outf, 'n_file2.dat', history:=unknown);
rewrite(outf);

speed[9]:=substr(X1, 3,2);
writeln(speed[9]);
for i:=1 to n_speed do
begin
     for j:=1 to n_angle do
     begin
       in_f:='[grad.szhang.roll.'+X1+']'+ X + speed[i] + angle[j] +
                 '.AGL';
       out_f:='[grad.szhang.roll.'+X1+']'+ X + speed[i] + angle[j] +
                 '.USE';
       writeln(X+speed[i]+angle[j]);
       writeln(nf, in_f);
       writeln(outf, out_f);
     end;
  end;
end.
```

Appendix F. Program MODFBAT.FOR

```
*       PROGRAM MODFBAT.FOR
*       VAX-FORTRAN
*       Roll damping parameter idetification using
*       Modified Energy Method
*
*       Suimin Zhang
*       Oct. 1992, Engineering, M.U.N.

        REAL PH(250), Y(250),H(250, 10), PHD(250), A(250, 10),
      ! E(4, 5),
      !   PSN1(10), PSN2(10), PSN3(10, 10), EN(5, 250), VK(250),
      !   VP(250),
      !   V(250), OM, MM0, MM1, MM2, MUSA,
      !   B44(4,2), CQ2(4), SPHDQ(4), ENG(5),
      !   NH(10), XH(10), B_L(2), NH_A(10), XH_A(10)

        CHARACTER XXX*1, XXX1*4, IN_F*35, ANGLE(7)*2,
      ! SPEED(9)*2

        PI = 3.1415926
        TS = 0.0
        TE = 5
        KC = 4
        ITER = 5
        OPEN(15, FILE='ID_DIR.DAT', STATUS='OLD')
*       ID_DIR.DAT contain I.D. and name of subdirectory of the files.
*       of the files. It is the same file used in TRSBAT.PAS and
*       DAMABAT.FOR. Example of contect
*
*       A
*       T363
*       H
*       T365
*       Y
*       T366

        OPEN(20, FILE='MODF_RST.DAT', STATUS='UNKNOWN')
*       MODF_RST.DAT is a file for outputing analytical result

        DO WHILE (.TRUE.)
          READ(15, '(A)', END=113) XXX
*         Read I.D
          READ(15, '(A)', END=113) XXX1
*         Read name of subdirectory

*         7 initial angle numbers
          ANGLE(1)='01'
          ANGLE(2)='02'
          ANGLE(3)='03'
          ANGLE(4)='04'
          ANGLE(5)='05'
          ANGLE(6)='06'
          ANGLE(7)='07'
```

```
*       8 forward speed numbers
        SPEED(1)='00'
        SPEED(2)='03'
        SPEED(3)='05'
        SPEED(4)='07'
        SPEED(5)='09'
        SPEED(6)='11'
        SPEED(7)='13'
        SPEED(8)='15'
*       Forward speed number for test with joint
*       '63' for M363, '65' for M365, '66' for M366
*       Obtain the value for name of subdirectory(T363, T365 and T366)
        SPEED(9)=XXX1(3:4)

        DO L = 1,9
          DO M = 1, 7
*           Compose the file name including subdirectory
            IN_F='[GRAD.SZHANG.ROLL.'//XXX1//']'//XXX//SPEED(L)
     !        //ANGLE(M)//'.USE'
            WRITE(20, *) IN_F(24:)

            OPEN(77, FILE=IN_F, STATUS='OLD')

*           Read data for the file
            READ(77,*) DT
            READ(77,*) OM
            READ(77,*) MM0
            READ(77,*) MM1
            READ(77,*) MM2
            MM0=1.0
            TPER = 2.0*PI/OM

            SOM = OM**2
            I=0
            DO WHILE (.TRUE.)
              I=I+1
              READ(77, *, END=11) PH(I)
            END DO
  11        N=I - 1
*           end of readingg

*           Begin roll damping idetification by
*           Modified Energy Method

            PERIOD= N*DT

            ALFA = (TE + TS)/PERIOD

*           A - Function calculations
            DO I = 1, N
              TSM = (I - 1) * DT
              TAU = TSM * ALFA - TS
              H(I, 1) = 1.0
```

```
      H(I, 2) = TAU
      DO K = 3, KC+1
        H(I,K) = TAU * H(I,K-1) - (K-1-1) * H(I,K-2)
      END DO
    END DO

    DO I = 1, N
      TSM = (I - 1)*DT
      TAU = TSM * ALFA - TS
      DO K = 1, KC+1
        A(I,K) = H(I,K) * EXP(-TAU**2/2.0)
        IF (I.LE.6) THEN
*            WRITE(88,*) I, K, A(I,K)
        END IF
      END DO
    END DO

*   Calculation of the roll velocity
    PHD(1) = -(PH(3) - 4.0*PH(2) + 3.0*PH(1) )/(2.0*DT)
    PHD(N) = (3.0*PH(N) - 4.0*PH(N-1) + PH(N-2))/(2.0*DT)
    DO I = 2, N-1
      PHD(I) = (PH(I+1) - PH(I-1))/(2.0*DT)
    END DO

*   ------------
    DO I = 1, N
      PHS = PH(I)**2
      PHDS = PHD(I)**2
      PHC = PH(I)**3
      PHQ = PH(I)**4
      MUSA = 0.5*MM1*PHS + MM2*PHQ/3.0
      VK(I) = 0.5 * PHDS
      VP(I) = 0.5 * SOM * PHS * (1.0 + MUSA)
      V(I) = VK(I) + VP(I)
      EN(1,I) = PHDS
      EN(2,I) = ABS(PH(I))* PHDS
      EN(3,I) = ABS(PHD(I)) * PHDS
      EN(4,I) = PHS * PHDS
      EN(5,I) = PHDS**2
    END DO

*   Calculation of the integrals
    DO K = 1, 2
      SUMN1 = 0.0
      SPHDS = 0.0
      SPHDQ(1) = 0.0
      SPHDQ(2) = 0.0
      SPHDQ(3) = 0.0
      SPHDQ(4) = 0.0
      DO I = 2, N-1
        SUMN1 = SUMN1 + V(I) * A(I, K+1)
        SPHDS = SPHDS + EN(1,I)*A(I,K)
        SPHDQ(1) = SPHDQ(1) + EN(2,I)*A(I,K)
```

```
          SPHDQ(2) = SPHDQ(2) + EN(3,I)*A(I,K)
          SPHDQ(3) = SPHDQ(3) + EN(4,I)*A(I,K)
          SPHDQ(4) = SPHDQ(4) + EN(5,I)*A(I,K)
        END DO

        CC1 = V(1) * A(1, K+1) + V(N) * A(N, K+1)
        PSN1(K) = DT * (CC1 + 2.0*SUMN1)/2.0

        CQ1 = EN(1,1)*A(1,K) + EN(1,N) * A(N,K)
        PSN2(K) = DT*(CQ1 + 2.0 * SPHDS)/2.0

        CQ2(1) = EN(2,1)*A(1,K) + EN(2, N) * A(N, K)
        CQ2(2) = EN(3,1)*A(1,K) + EN(3, N) * A(N, K)
        CQ2(3) = EN(4,1)*A(1,K) + EN(4, N) * A(N, K)
        CQ2(4) = EN(5,1)*A(1,K) + EN(5, N) * A(N, K)

        PSN3(K,1) = DT* (CQ2(1) + 2.0 * SPHDQ(1)) / 2.0
        PSN3(K,2) = DT* (CQ2(2) + 2.0 * SPHDQ(2)) / 2.0
        PSN3(K,3) = DT* (CQ2(3) + 2.0 * SPHDQ(3)) / 2.0
        PSN3(K,4) = DT* (CQ2(4) + 2.0 * SPHDQ(4)) / 2.0
      END DO

*     Calculation of the matrix elements
      DO II=1,4
        DO K = 1,2
          J = K
          E(J,1) = PSN2(K)
          E(J,2) = PSN3(K,II)
          E52 = V(N) * A(N,K) - V(1) * A(1,K)
          E(J,3) = -(E52 + ALFA*PSN1(K))
          IF (II.EQ.1) THEN
            B_L(K)=E(J,3)/E(J,1)
*            PRINT*, 'B_L', K, B_L(K)
          END IF
        END DO

        DO 300 I = 1,2
          IF (E(I,I).EQ.0) THEN
            GOTO 300
          END IF
          PIVOT = E(I,I)
          DO J=I,3
            E(I,J) = E(I,J)/PIVOT
          END DO
          DO 200 K = 1,2
            IF (K .EQ. I) THEN
              GOTO 200
            END IF
            SOB = E(K,I)
            DO J = I, 3
              E(K,J) = E(K,J) - SOB * E(I,J)
            END DO
200       CONTINUE
```

```
 300        CONTINUE
            B44(II,1)=E(1,3)
            B44(II,2)=E(2,3)
          END DO


          WRITE(20, *) OM

          N_OLD=N
          N1=0
          N2=0
          X01=PH(1)

*         Find the amplitude of half cycle of the
*         experimental decay curve
          CALL DASEL(PH, N1, N2, N, NH, XH, NUB)
          WRITE(20, *) X01
          DO I=1, 9
            IF (I.LE. NUB) THEN
              WRITE(20, *) XH(I)
            ELSE
              WRITE(20, *) 0.0
            END IF
          END DO

          DO I =1, 5
            IF (I.NE.5) THEN
              B1=B44(I,1)
              B2=B44(I,2)
              KK=I
            ELSE
              B1=B_L(1)
              B2=0.0
              KK=1
            END IF

*           Produce a decay curve from the analytical
            roll damping coefficients using Louger-Kuto
            CALL DEMKBAT(Y, DT, OM, MM0, MM1, MM2, B1, B2, KK,
     !                   N_OLD , X01)

*           Find the amplitude of half cycle of
*           analytical decay curve
            CALL DASEL(Y, N1, N2, N_OLD, NH_A, XH_A, NUB_A)

            ENG(I)=0.0

*           Calculate the error coefficient
            DO K=1, ITER
              ENG(I)=ENG(I) + (XH(K)-XH_A(K))**2
            END DO

*             PRINT*, I, ENG(I)
```

```
          END DO

          DO I = 1,5
            IF (I.NE.5) THEN
              WRITE(20, *) B44(I,1), B44(I,2), ENG(I)
            ELSE
              WRITE(20, *) B_L(1), 0.0 , ENG(I)
            END IF
          END DO
          CLOSE(77)
        end do
      end do
    END DO
113 print*, '---------END--------'
    END

*     Subroutine for finding the amplitude of half cycle
*     of the decay curve
      SUBROUTINE DASEL(Y,N1,N2,N, NH, XH, NUB)
      REAL Y(*), NH(*), XH(*)
      IF (Y(1).LT. 0.0) THEN
          KID=1
      ELSE
          KID=2
      END IF

      IF (KID.EQ. 1) THEN
        SIGN=1.0
      ELSE
        SIGN=-1.0
      END IF
      X0=SIGN*Y(1)
      NSTEP=1
      I=1
      DO I=1,N

          IF ( SIGN*Y(I) .GT. X0) THEN
            X0=SIGN*Y(I)
            N0=I
          END IF

          IF ( (SIGN*Y(I) .GT. 0.0).AND. (NSTEP .EQ.1) ) THEN
              NSTEP=2
              J=1
          END IF

          IF ((SIGN*Y(I) .LE. 0.0) .AND. (NSTEP .EQ. 2)) THEN
             NH(J)=N0
             XH(J)=X0/SIGN
             J=J+1
             SIGN=-1.0*SIGN
             X0=SIGN*Y(I)
```

```
          END IF
        END DO
        NUB=J-1
        DO K=J,J+10
          XH(J)=0.0
        END DO
      END


*     Subroutine for producing analytical decay curve
*     using RUNGE_KUTTA method

      SUBROUTINE DEMKBAT(Y, HH, WW, MM0, MM1, MM2, B1, B2,
     ! KK, N, X01)

      INTEGER I, N
      REAL Y(*), X0(2), X(2, 600), F(2), XEND(2), XWRK(4,4),
     & HH,WW,MM0, MM1,MM2,TT

      X0(1)=X01
      X0(2)=0

      TT = 0.0

*     Solve differential equation by the RUNGE_KUTTA method

      X(1,1)=X0(1)
      X(2,1)=X0(2)
      CK_FLOW=0.0
      DO I = 2, N
        CALL RKSYST(TT, HH, X0, XEND, XWRK, F, 2, WW, MM0,
     &               MM1, MM2, B1, B2, KK, CK_FLOW)
        X(1,I) = XEND(1)
        X(2,I) = XEND(2)
        TT = TT + HH
        X0(1) = XEND(1)
        X0(2) = XEND(2)
      END DO

      DO I = 1, N
        Y(I)=X(1,I)
      END DO
   20 FORMAT(X, I4, 3F10.4)
      END
```

```
*      RUNGE_KUTTA Method, Program from [25]

       SUBROUTINE RKSYST(T0, H, X0, XEND, XWRK, F, N, WW, MM0,
      & MM1, MM2, B1, B2, KK, CK_FLOW)
*
*      This subroutine solve a sysstem of N first order diffrential
*      equations by the RUNGE-KUTTA method. The equations are of the
*      are of the form DX1/DT = F1(X, T), DX2/DT = F2(X,T), etc,
*      where X = (X1, X2, X3, ......, Xn)
*
*      DERIVS - A subroutine that computr values of the N derivatives.
*               It must decleared external by the caller.
*      T0     - The initial value of independent valiable
*      H      - The INCREMENT TO T, THE STEP SIZE
*      X0     - The array that holds the initial valus of the functions
*      XEND   - An array that return the final values of the functions
*      XWRK   - An array to hold the values of the RK fomular, K1, K2,
*               K3, K4 .
*      N      - The number of equations to be solved
*      F      -An array that holds the derivatives
*      ------------------------------------------------------------------
*
       REAL X0(N), XEND(N), XWRK(4,N), F(N), H, T0, WW, MM0
      &     MM1, MM2, B1, B2
       INTEGER I, N, KK
*      Get K1
       CALL DERIVS(X0, T0, F, N, WW, MM0, MM1, MM2, B1, B2, KK, CK_FLOW)

       IF (CK_FLOW.EQ.1.0) GOTO 999
       DO I = 1, N
         XWRK(1,I) = H * F(I)
         XEND(I)   = X0(I) + XWRK(1,I)/2.0
       END DO

*      Get K2
       CALL DERIVS(XEND, T0 + H/2.0, F, N, WW, MM0, MM1, MM2, B1, B2,
      !  KK, CK_FLOW)
       IF (CK_FLOW.EQ.1.0) GOTO 999

       DO I = 1, N
         XWRK(2,I) = H * F(I)
         XEND(I)   = X0(I) + XWRK(2,I)/2.0
       END DO

*      Get K3
       CALL DERIVS(XEND, T0 + H/2.0, F, N,
      &          WW, MM0, MM1, MM2, B1, B2, KK, CK_FLOW)
       IF (CK_FLOW.EQ.1.0) GOTO 999

       DO I = 1, N
         XWRK(3, I) = H * F(I)
```

```
            XEND(I) = X0(I) + XWRK(3,I)
          END DO

*         Get K4
          CALL DERIVS(XEND, T0 + H, F, N,
     &              WW, MM0, MM1, MM2, B1, B2, KK, CK_FLOW)
          IF (CK_FLOW.EQ.1.0) GOTO 999

          DO I = 1, N
            XWRK(4, I) = H * F(I)
          END DO

*         Compute result
          DO I = 1, N
            XEND(I) = X0(I) + ( XWRK(1,I) + 2.0*XWRK(2,I) + 2.0 * XWRK(3,I)
     &              + XWRK(4,I) ) / 6.0
          END DO
999       RETURN
          END


*         Subroutine for calculation of damping moment of
*         different models and resotoring moment

          SUBROUTINE DERIVS(XEND, T, F, N, WW, MM0, MM1, MM2, B1,
     !     B2, KK, CK_FLOW)

          REAL XEND(N), T, F(N),B1,B2,WW,MM0, MM1,MM2
          INTEGER N, KK

          F(1) = XEND(2)

*         Linear angle dependence
          IF (KK .EQ. 1) THEN
             IF ((ABS(XEND(1)).GT.9999) .OR. (ABS(XEND(2)).GT.9999)) THEN
               CK_FLOW=1.0
             END IF
             F(2) = -B1 * XEND(2) - B2 * ABS(XEND(1)) * XEND(2) - WW**2
     &         * (MM0*XEND(1) +MM1*XEND(1)**3 + MM2*XEND(1)**5)
          END IF

*         Quadratic
          IF (KK .EQ. 2) THEN
             IF ((ABS(XEND(1)).GT.9999) .OR. (ABS(XEND(2)).GT.9999)) THEN
               CK_FLOW=1.0
             END IF

             F(2) = -B1 * XEND(2) - B2 * ABS(XEND(2)) * XEND(2) - WW**2
     &         * (MM0*XEND(1) +MM1*XEND(1)**3 + MM2*XEND(1)**5)
          END IF

*         Quadratic angle dependence
```

```
      IF (KK .EQ. 3) THEN
         IF ((ABS(XEND(1)).GT.9999) .OR. (ABS(XEND(2)).GT.9999)) THEN
           CK_FLOW=1.0
         END IF
         F(2) = -B1 * XEND(2) - B2 * XEND(1)**2 * XEND(2) - WW**2
     &    * (MM0*XEND(1) +MM1*XEND(1)**3 + MM2*XEND(1)**5)
      END IF

*     Cubic
      IF (KK .EQ. 4) THEN
         IF ((ABS(XEND(1)).GT.9999) .OR. (ABS(XEND(2)).GT.9999)) THEN
           CK_FLOW=1.0
         END IF
         F(2) = -B1 * XEND(2) - B2 * XEND(2)**3 - WW**2 *
     &   (MM0*XEND(1) +MM1*XEND(1)**3 + MM2*XEND(1)**5)

      END IF
      END
```

Appendix G. Program LABDATA.PAS

```
{     LABDATA.PAS

    * Database management system                          *
    * created by VAX Pascal                               *
    * for exprimental data management and analysis *

    By Suimin Zhang
    Sept. 1992, Engineering. M.U.N.        }

program labdata(input, output);
label
  11, 13; {for exit}
type
{ Definition of database for storing the analytica result}
  tkey = packed array[1..5] of char;
  f_rec = record
    id      : [key(0, ascending, nochanges, noduplicates)] tkey;
    speed   : real;
    i_angle: real;
    omega   : real;
    amp     : array[1..10] of real;
    b_eng   : array[1..5, 1..3] of real;
    end;

{ Definition of database for storing the result of list square
  regression}
  lsr_rec = record
    lsr_id  : [key(0, ascending, nochanges, noduplicates)] tkey;
    lsr_a   : real;
    lsr_b   : real;
    lsr_err : real;
    lsr_gm  : real;
    lsr_sp  : real;
    end;
var
  f          : file of f_rec;
  f_lsr      : file of lsr_rec;
  m_select, m_choice : integer;
  m_yn       : char;


{ Clear screen}
procedure clearsc;
var
  i : integer;
begin
  for i:=1 to 3 do
  begin
    writeln;
  end;
end;

{function for geting the friction effect of the joint}
```

```
function  friction(id: char)  : real;
label
  19;
var
  idd    : char;
  rr, omega     : real;
  fcomp : text;
begin
  open(fcomp, 'compl.txt', history := old);
  reset(fcomp);
  friction := 0.0;
  while not eof(fcomp) do
  begin
    readln(fcomp, idd, rr, omega);
    if (idd=id) then
    begin
      friction := rr;
      goto 19;
    end;
  end;
  19:
  close(fcomp);
end;

{function for geting the natural frequency}
function  w(id: char) : real;
label
  19;
var
  idd    : char;
  rr, omega     : real;
  fcomp : text;
begin
  open(fcomp, 'compl.txt', history := old);
  reset(fcomp);
  w := 0.0;
  while not eof(fcomp) do
  begin
    readln(fcomp, idd, rr, omega);
    if (idd=id) then
    begin
      w := omega;
      goto 19;
    end;
  end;
  19:
  close(fcomp);
end;


{Create database(LAB0801.DAT) by file type: f}
procedure cr_db;
var
```

```
  n_f : text;
  id_key: tkey;
begin
  open(f, 'lab0801.dat', history:=unknown,
       organization:=indexed, access_method:=keyed);
  rewrite(f);
  open(n_f, 'n_0801.dat', history:=old);
  reset(n_f);
  while not eof(n_f) do
  begin
    readln(n_f, id_key);
    writeln(id_key);
    f^.id:= id_key;
    put(f);
  end;
  close(f);
  close(n_f);
  writeln;
  writeln('*** database created ***');
  writeln;
end;

{Data Examination}
procedure exam_db;
label 15;
var
  f_check   : text;
  f_name    : packed array[1..10] of char;
  i, kk     : integer;
  id_key    : tkey;
  err       : real;
begin
  open(f, 'lab0801.dat', history:=unknown,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  write('Damping form(1-5):');
  readln(kk);
  write('Error Level:');
  readln(err);
  write('Output file name:');
  readln(f_name);
  write('Begin ID_EXP (Press RETURN if from top):');
  readln(id_key);

  if (f_name<>'') then
  begin
    open(f_check, f_name, history:=unknown);
    rewrite(f_check);
  end;
  if (id_key<>'') then
  begin
    findk(f, 0, id_key);
    if ufb(f) then
```

```
    begin
      writeln(id_key, ' not found! Press any key to return');
      read(id_key);
      goto 15;
    end;
  end;
  while not eof(f) do
  begin
    writeln('ID_KEY  Phi0        B1        B2        Error');
    i:= 1;
    while ((i<22) and (not eof(f))) do
    begin
      if ((f^.b_eng[kk,3] > err) or (f^.b_eng[kk,1] = 0)) then
      begin
         writeln(f^.id,' ', f^.amp[1]:8:5, ' ', f^.b_eng[kk,1]:8:5, '
              f^.b_eng[kk,2]:8:5,  f^.b_eng[kk,3]);
         if (f_name<>'') then
           writeln(f_check, f^.id,' ', f^.amp[1]:8:5, ' ',
                   f^.b_eng[kk,1]:8:5, ' ',
                   f^.b_eng[kk,2]:8:5,  f^.b_eng[kk,3]);
         i := i+1;
      end;
      get(f);
    end;
    writeln('--------------------------------------------------------');
    m_yn:='Y';
    write('Continue? (Y/N)');
    readln(m_yn);
    if ((m_yn='N') or (m_yn='n')) then
      goto 15;
  end;
  15:
  close(f);
end;

{List items in the database}
procedure ls_db;
label 15;
var
  i, kk     : integer;
  id_key    : tkey;
begin
  open(f, 'lab0801.dat', history:=unknown,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  write('Damping form(1-5):');
  readln(kk);
  write('Begin ID_EXP (Press RETURN if from top):');
  readln(id_key);

  if (id_key<>'') then
  begin
    findk(f, 0, id_key);
```

```
    if ufb(f) then
    begin
      writeln(id_key, ' not found! Press any key to return');
      read(id_key);
      goto 15;
    end;
  end;
  while not eof(f) do
  begin
    writeln('ID_KEY  Phi0       B1        B2        Error');
    i:= 1;
    while ((i<22) and (not eof(f))) do
    begin
      writeln(f^.id,' ', f^.amp[1]:8:5, ' ', f^.b_eng[kk,1]:8:5, ' ',
              f^.b_eng[kk,2]:8:5, f^.b_eng[kk,3]);
      get(f);
      i:=i+1;
    end;
    writeln('----------------------------------------------------');
    m_yn:='Y';
    write('Continue? (Y/N)');
    readln(m_yn);
    if ((m_yn='N') or (m_yn='n')) then
      goto 15;
  end;
  15:
  close(f);
end;


{natural frequency average}
procedure rpt_omega;
label 15;
var
  kk    : integer;
  id_key   : tkey;
  i, omega    : real;
begin
  open(f, 'lab0801.dat', history:=unknown,
      organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  readln;
  write('ID:');
  readln(id_key);
  if (id_key<>'') then
  begin
    findk(f, 0, id_key);
    if ufb(f) then
    begin
      writeln(id_key, ' not found! Press any key to return');
      read(id_key);
      goto 15;
    end;
```

```
  end
  else
    goto 15;

  omega :=0.0;
  i:= 0.0;
  while ((substr(id_key,1,1)=substr(f^.id, 1,1)) and (not ufb(f))) do
  begin
    omega := omega + f^.omega;
    i:= i+1.0;
    get(f);
  end;
  writeln('Omega =', i, omega/i);
  readln(kk);
  15:
  close(f);
end;


{ Update forward speed from measured value}
procedure updt_sp;
var
  lab_f  : text;
  lab_fn : packed array[1..9] of char;
  id_key : tkey;
  i, sp, sp_i, qq  : real;
begin
  open(f, 'lab0801.dat', history:=old,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  while not eof(f) do
  begin
    id_key:=f^.id;
    lab_fn:=id_key+'.agl';
    open(lab_f, lab_fn, history:=old);
    reset(lab_f);
    i:=0.0; sp:=0.0;
    while not eof(lab_f) do
    begin
      read(lab_f,qq);
      read(lab_f,qq);
      readln(lab_f,sp_i);
      sp:=sp+sp_i;
      i:=i+1.0;
    end;
    close(lab_f);
    sp:=sp/i;
    f^.speed:=sp;
    writeln(lab_fn,' speed =', f^.speed:5:3);
    update(f);
    get(f);
  end;
  close(f)
```

```
end;

{ Update initial angle}
procedure updt_agl0;
var
  ang_f  : text;
  lab_fn : packed array[1..9] of char;
  id_key : tkey;
  agl1, agl2   : real;
begin
  open(f, 'lab0801.dat', history:=old,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  open(ang_f, 'agl0801.dat', history:=old);
  reset(ang_f);
  while not eof(ang_f) do
  begin
    readln(ang_f,id_key, agl1, agl2);
    writeln(id_key, agl1, agl2);
    findk(f, 0, id_key);
    if not ufb(f) then
    begin
      f^.i_angle := agl1;
      update(f);
    end
    else
    begin
      writeln(id_key, ' not found !');
    end;
  end;
  close(f)
end;

{ Append data from *.rst}
procedure appe_db;
var
  eng_f             : text;
  id_key, disp_k : tkey;
  ttt               : string(10);
  in_file           : packed array[1..20] of char;
  i, j              : integer;
  gitem             : real;

begin
  readln;
  write('Input file name (Such as: MODF1112.RST)');
  readln(in_file);
  open(f, 'lab0801.dat', history:=old,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  open(eng_f, in_file, history:=old);
  reset(eng_f);
  while not eof(eng_f) do
```

```
  begin
    readln(eng_f, ttt);
    id_key:=substr(ttt, 2, 5);
    findk(f, 0, id_key);
    if ufb(f) then
    begin
      writeln(id_key);
      f^.id := id_key;
      readln(eng_f, f^.omega);
      for i:=1 to 10 do
      begin
        readln(eng_f, f^.amp[i]);
      end;
      for i:=1 to 5 do
      begin
          read(eng_f, f^.b_eng[i,1]);
          read(eng_f, f^.b_eng[i,2]);
          readln(eng_f, f^.b_eng[i,3]);
      end;
      put(f);
    end
    else
    begin
      writeln(id_key, 'already exist, record not appended');
    end;
  end;
  close(f);
end;

{ Update data from file *.RST}
procedure updt_db;
var
  eng_f           : text;
  id_key, disp_k  : tkey;
  ttt : string(10);
  i, j            : integer;
  in_file         : packed array[1..20] of char;
  gitem           : real;
begin
  readln;
  write('Input file name (Such as: MODF1112.RST)');
  readln(in_file);
  open(f, 'lab0801.dat', history:=old,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  open(eng_f, in_file, history:=old);
  reset(eng_f);
  while not eof(eng_f) do
  begin
    readln(eng_f, ttt);
    id_key:=substr(ttt, 2, 5);

    findk(f, 0, id_key);
```

```
    if not ufb(f) then
    begin
      writeln(f^.id);
      readln(eng_f, f^.omega);
      for i:=1 to 10 do
      begin
        readln(eng_f, f^.amp[i]);
      end;
      for i:=1 to 5 do
      begin
          read(eng_f, f^.b_eng[i,1]);
          read(eng_f, f^.b_eng[i,2]);
          readln(eng_f, f^.b_eng[i,3]);
      end;
      update(f);
    end
    else
    begin
      writeln(id_key, 'not found');
    end;
  end;
  close(f);
end;

(Output damping coefficients of one forward speed)
procedure rpt_b;
var
  rpt_f           : text;
  rpt_fn          : packed array[1..10] of char;
  id_key          : tkey;
  speed           : packed array[1..2] of char;
  i, j, kk        : integer;
  speed_v         : array[1..8] of real;
  len             : real;
begin
  speed_v[1] := 0.0;
  speed_v[2] := 0.3;
  speed_v[3] := 0.5;
  speed_v[4] := 0.7;
  speed_v[5] := 0.9;
  speed_v[6] := 1.1;
  speed_v[7] := 1.3;
  speed_v[8] := 1.5;
  open(f, 'lab0801.dat', history:=old,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  clearsc;
  readln;
  write('Damping form(1-5):');
  readln(kk);
  write('Output file name:');
  readln(rpt_fn);
  write('KEY NAME:');
```

```
  readln(id_key);
  speed:=substr(id_key, 2,2);
  findk(f, 0, id_key);

  if not ufb(f) then
  begin
    open(rpt_f, rpt_fn, history:=unknown);
    rewrite(rpt_f);
    clearsc;
    writeln('file   amp       b1       b2        error');
    i:=1;
    while (substr(f^.id, 2,2)=speed) and (not ufb(f)) do
    begin
      writeln(f^.id, f^.amp[1], f^.b_eng[kk,1],
              f^.b_eng[kk,2], f^.b_eng[kk,3]);
      writeln(rpt_f, f^.amp[1], f^.b_eng[kk,1],
              f^.b_eng[kk,2], f^.b_eng[kk,3]);
      get(f);
      i:=i+1;
    end;
  end
  else
    writeln(' File not found!');
  close(f);
end;

{Output damping coefficient of forward speed via
 initial angle}
procedure rpt_va;
type
  tspeed = packed array[1..2] of char;
  tangle = packed array[1..2] of char;
var
  rpt_f          : text;
  rpt_fn         : packed array[1..10] of char;
  speed          : packed array[1..2] of char;
  id_key         : tkey;
  speed_a        : array[1..8] of tspeed;
  speed_v        : array[1..8] of real;
  angle          : array[1..7] of tangle;
  i, j, kk, m    : integer;
  len            : real;
  exp_id         : char;

begin
  open(f, 'lab0801.dat', history:=old,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  clearsc;

  kk := 5;
  write('len:');
  readln(len);
```

```
write(' exp_id:');
readln(exp_id);
writeln(exp_id);

write('Output file name:');
readln(rpt_fn);


speed_a[1] := '00';
speed_a[2] := '03';
speed_a[3] := '05';
speed_a[4] := '07';
speed_a[5] := '09';
speed_a[6] := '11';
speed_a[7] := '13';
speed_a[8] := '15';

speed_v[1] := 0.0;
speed_v[2] := 0.3;
speed_v[3] := 0.5;
speed_v[4] := 0.7;
speed_v[5] := 0.9;
speed_v[6] := 1.1;
speed_v[7] := 1.3;
speed_v[8] := 1.5;

angle[1] := '01';
angle[2] := '02';
angle[3] := '03';
angle[4] := '04';
angle[5] := '05';
angle[6] := '06';
angle[7] := '07';
open(rpt_f, rpt_fn, history:=unknown);
rewrite(rpt_f);
clearsc;

for m:=1 to 8 do
begin
  id_key := exp_id + speed_a[m] + '01';
  speed:=substr(id_key, 2,2);
  writeln(id_key);
  findk(f, 0, id_key);

  if not ufb(f) then
  begin
    write(rpt_f, speed_v[m]/sqrt(9.8*len) , ' ');
    while (substr(f^.id, 2,2)=speed) and (not ufb(f)) do
    begin
      write(rpt_f, f^.b_eng[kk,1], ' ');
      get(f);
    end;
    writeln(rpt_f);
```

```
    end
    else
      writeln(' File not found!');
  end;
  close(f);
end;

{Output damping coefficient of a specific
 initial angle}
procedure rpt_ampl;
label 17;
type
  tspeed = packed array[1..2] of char;
  tangle = packed array[1..2] of char;
var
  rpt_f             : text;
  rpt_fn            : packed array[1..10] of char;
  speed             : packed array[1..2] of char;
  id_key            : tkey;
  speed_a           : array[1..8] of tspeed;
  speed_v           : array[1..8] of real;
  angle             : array[1..7] of tangle;
  i, j, kk, m       : integer;
  len, err          : real;
  exp_id            : char;
  ampl, amp3, angl, b441, ang2, b442, ang, b44, ee : real;
  any_key           : packed array[1..1] of char;

begin
  open(f, 'lab0801.dat', history:=old,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  clearsc;

  kk := 5;

  write('len:');
  readln(len);
  write(' exp_id:');
  readln(exp_id);
  err := friction(exp_id);
  writeln(exp_id,' ', err);

  while true do
  begin
{loop added}
  writeln;
  writeln;
  ampl:=0.0;
  write('amplitude (0 exit):');
  readln(ampl);
  if (ampl=0.0) then goto 17;
  write('Output file name:');
```

```
readln(rpt_fn);
ampl:= ampl*3.1416/180.0;

speed_a[1] := '00';
speed_a[2] := '03';
speed_a[3] := '05';
speed_a[4] := '07';
speed_a[5] := '09';
speed_a[6] := '11';
speed_a[7] := '13';
speed_a[8] := '15';

speed_v[1] := 0.0;
speed_v[2] := 0.3;
speed_v[3] := 0.5;
speed_v[4] := 0.7;
speed_v[5] := 0.9;
speed_v[6] := 1.1;
speed_v[7] := 1.3;
speed_v[8] := 1.5;

angle[1] := '01';
angle[2] := '02';
angle[3] := '03';
angle[4] := '04';
angle[5] := '05';
angle[6] := '06';
angle[7] := '07';
open(rpt_f, rpt_fn, history:=unknown);
rewrite(rpt_f);
clearsc;

for m:=1 to 8 do
begin
  id_key := exp_id + speed_a[m] + '01';
  speed:=substr(id_key, 2,2);
  writeln(id_key);
  findk(f, 0, id_key);

  if not ufb(f) then
  begin
    write(rpt_f, speed_v[m]/sqrt(9.8*len) , ' ');
    ee:=0.0;
    j:=1 ;
    while (substr(f^.id, 2,2)=speed) and
          (not ufb(f)) and (ee=0.0) do
    begin
      amp3 := abs(f^.amp[1]);
      if (amp3 <= ampl) then
      begin
        ang1:=amp3;
        b441:=f^.b_eng[kk,1];
        if (j=7) then
```

```
          begin
            b442:=b441;
            ang2:=0.0;
          end;
        end
        else
        begin
          ang2:=amp3;
          b442:=f^.b_ ng[kk,1];
          ee:=1.0;
          if (j=1) then
          begin
            b441:=b442;
            ang1:=0.0;
          end;
        end;
        get(f);
        j:=j+1
      end;
      b44:= b441 + (b442-b441)*(ampl - ang1)/(ang2-ang1);
      writeln(ang1, ampl, ang2, b441, b44, b442);
      writeln(rpt_f, b44-err);
    end
    else
      writeln(' File not found!');
  end;
  close(rpt_f);
{ loop added }
  end;
  17:
  close(f);
end;


{Output damping coefficient of different natural
 frequency OMEGA}
procedure rpt_freq;
type
  tspeed = packed array[1..2] of char;
  tangle = packed array[1..2] of char;
var
  rpt_f           : text;
  rpt_fn          : packed array[1..10] of char;
  speed           : packed array[1..2] of char;
  id_key          : tkey;
  speed_a         : array[1..8] of tspeed;
  speed_v         : array[1..8] of real;
  angle           : array[1..7] of tangle;
  i, j, kk, m     : integer;
  exp_id          : char;
  omega, ampl, amp3, ang1, b441, ang2, b442, ang, b44, ee, err: real;

begin
```

```
open(f, 'lab0801.dat', history:=old,
     organization:=indexed, access_method:=keyed);
resetk(f, 0);
clearsc;

kk := 5;
write('amplitude:');
readln(ampl);
write('Output file name:');
readln(rpt_fn);
ampl:= ampl*3.1416/180.0;

speed_a[1] := '00';
speed_a[2] := '03';
speed_a[3] := '05';
speed_a[4] := '07';
speed_a[5] := '09';
speed_a[6] := '11';
speed_a[7] := '13';
speed_a[8] := '15';

speed_v[1] := 0.0;
speed_v[2] := 0.3;
speed_v[3] := 0.5;
speed_v[4] := 0.7;
speed_v[5] := 0.9;
speed_v[6] := 1.1;
speed_v[7] := 1.3;
speed_v[8] := 1.5;

angle[1] := '01';
angle[2] := '02';
angle[3] := '03';
angle[4] := '04';
angle[5] := '05';
angle[6] := '06';
angle[7] := '07';
open(rpt_f, rpt_fn, history:=unknown);
rewrite(rpt_f);
clearsc;
write(' exp_id(0 exit):');
readln(exp_id);
err := friction(exp_id);
writeln(exp_id,' ', err);

while (exp_id <> '0') do
begin
  omega := w(exp_id);
  writeln('Omega = ', omega);
  write(rpt_f, omega);
  for m:=1 to 8 do
  begin
    id_key := exp_id + speed_a[m] + '01';
```

```
      speed:=substr(id_key, 2,2);
      writeln(id_key);
      findk(f, 0, id_key);

      if not ufb(f) then
      begin
        ee:=0.0;
        j:=1;
        while (substr(f^.id, 2,2)=speed) and
              (not ufb(f)) and (ee=0.0) do
        begin
          amp3 := abs(f^.amp[1]);
          if (amp3 <= ampl) then
          begin
            angl:=amp3;
            b441:=f^.b_eng[kk,1];
            if (j=7) then
            begin
              b442:=b441;
              ang2:=0.0;
            end;
          end
          else
          begin
            ang2:=amp3;
            b442:=f^.b_eng[kk,1];
            ee:=1.0;
            if (j=1) then
            begin
              b441:=b442;
              angl:=0.0;
            end;
          end;
          get(f);
          j:=j+1
        end;
        b44:= b441 + (b442-b441)*(ampl - angl)/(ang2-angl);
        writeln(angl, ampl, ang2, b441, b44, b442);
        write(rpt_f, b44 - err, ' ');
      end
      else
        writeln(' File not found!');
    end;
    writeln(rpt_f);
    write(' exp_id(0 exit):');
    readln(exp_id);
    err := friction(exp_id);
    writeln(exp_id,' ', err);
  end;
  close(f);
end;

(Compare two set of file)
```

```
procedure comp;
var
  rpt_f              : text;
  rpt_fn             : packed array[1..10] of char;
  id_key1, id_key2 : tkey;
  speed              : packed array[1..2] of char;
  i, j, kk           : integer;
  sum1, sum2, no    : real;
begin
  open(f, 'lab0801.dat', history:=old,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  clearsc;
  no:=7.0;
  write('Damping form(1-5):');
  readln(kk);
  write('KEY NAME 1:');
  readln(id_key1);
  write('KEY NAME 2:');
  readln(id_key2);

  speed:=substr(id_key1, 2,2);
  findk(f, 0, id_key1);
  sum1:=0.0;
  if not ufb(f) then
  begin
    clearsc;
    writeln('file   amp      b1       b2        error');
    while (substr(f^.id, 2,2)=speed) and (not ufb(f)) do
    begin
      writeln(f^.id,f^.amp[1], f^.b_eng[kk,1],
              f^.b_eng[kk,2], f^.b_eng[kk,3]);
      sum1:=sum1 + f^.b_eng[kk,1];
      get(f);
    end;
  end
  else
    writeln(' File  1 not found!');

  speed:=substr(id_key2, 2,2);
  findk(f, 0, id_key2);
  sum2:=0.0;
  if not ufb(f) then
  begin
    clearsc;
    writeln('file   amp      b1       b2        error');
    while (substr(f^.id, 2,2)=speed) and (not ufb(f)) do
    begin
      writeln(f^.id,f^.amp[1], f^.b_eng[kk,1],
              f^.b_eng[kk,2], f^.b_eng[kk,3]);
      sum2:=sum2 + f^.b_eng[kk,1];
      get(f);
    end;
```

```
  end
  else
    writeln(' File2 not found!');

  writeln(' Average difference: ', (sum1-sum2)/no);
  close(f);
  readln(i);
end;

{Least square regression for a specific
 forward speed number}
procedure lsr;
var
  rpt_f           : text;
  rpt_fn          : packed array[1..10] of char;
  id_key          : tkey;
  speed           : packed array[1..2] of char;
  i, kk, no       : integer;
  x, y            : array[1..7] of real;
  xsum, ysum, xysum, xxsum, a, b: real;
  dd, err         :    real;
begin
  open(f, 'lab0801.dat', history:=old,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);
  clearsc;

  write('Damping form(1-5):');
  readln(kk);
  write('Output file name:');
  readln(rpt_fn);
  write('KEY NAME:');
  readln(id_key);
  write(' dd:');
  readln(dd);
  write(' initial angle:');

  speed:=substr(id_key, 2,2);
  findk(f, 0, id_key);

  if not ufb(f) then
  begin
    open(rpt_f, rpt_fn, history:=unknown);
    rewrite(rpt_f);
    clearsc;
    i:=0;
    while (substr(f^.id, 2,2)=speed) and (not ufb(f)) do
    begin
      i:=i+1;
      x[i]:=(f^.amp[1]);
      y[i]:=f^.b_eng[kk,1] - dd;
      writeln(i, x[i], y[i]);
      get(f);
```

```
    end;
    no:=i;

    xsum:=0;
    ysum:=0;
    xysum:=0;
    xxsum:=0;

    for i:= 1 to no do
    begin
      xsum := xsum + x[i];
      ysum := ysum + y[i];
      xysum := xysum + x[i]*y[i];
      xxsum := xxsum + x[i]*x[i];
    end;
    b := (no*xysum - xsum*ysum)/(no*xxsum - xsum*xsum);
    a := (ysum - b*xsum)/no;
    err := 0;
    for i:=1 to no do
    begin
      err := err + abs(y[i] - (a + b*x[i]));
      writeln(x[i], y[i], a+b*x[i]);
      writeln(rpt_f, x[i], y[i], a+b*x[i]);
    end;
    writeln('-----');
    writeln('a = ', a, 'b =', b, 'error=', err);
  end
  else
    writeln(' File not found!');
  close(f);

end;

{batch least square regression
 results stored in LSR0801.DAT}
procedure lsr_bat;
type
  tspeed = packed array[1..2] of char;
var
  f_rpt           : text;
  f_name          : text;
  id_key          : tkey;
  speed           : tspeed;
  speed_a         : array[1..8] of tspeed;
  speed_v         : array[1..8] of real;
  i, i1, kk, no       : integer;
  x, y            : array[1..7] of real;
  xsum, ysum, xysum, xxsum, a, b: real;
  dd, err, gm :    real;
  exp_id  :    char;
begin
  speed_a[1] := '00';
  speed_a[2] := '03';
```

```
speed_a[3] := '05';
speed_a[4] := '07';
speed_a[5] := '09';
speed_a[6] := '11';
speed_a[7] := '13';
speed_a[8] := '15';

speed_v[1] := 0.0;
speed_v[2] := 0.3;
speed_v[3] := 0.5;
speed_v[4] := 0.7;
speed_v[5] := 0.9;
speed_v[6] := 1.1;
speed_v[7] := 1.3;
speed_v[8] := 1.5;

open(f_lsr, 'lsr0801.dat', history:=unknown,
     organization:=indexed, access_method:=keyed);
rewrite(f_lsr);

open(f_rpt, 'lsr0801.rpt', history:=unknown);
rewrite(f_rpt);

open(f_name, 'lsr_n.dat', history:=old);
reset(f_name);

open(f, 'lab0801.dat', history:=old,
     organization:=indexed, access_method:=keyed);
resetk(f, 0);

clearsc;
kk := 5;

while not eof(f_name) do
begin
  readln(f_name, exp_id);
  readln(f_name, dd);
  readln(f_name, gm);

  for i1 := 1 to 8 do
  begin
    id_key := exp_id + speed_a[i1] + '01';
    writeln(id_key);
    speed:=substr(id_key, 2,2);
    findk(f, 0, id_key);

    if not ufb(f) then
    begin
      clearsc;
      i:=0;
      while (substr(f^.id, 2,2)=speed) and (not ufb(f)) do
      begin
        i:=i+1;
```

```
            x[i]:=(f^.amp[1]+f^.amp[2]+f^.amp[3])/3.0;
            y[i]:=f^.b_eng[kk,1] - dd;
{            writeln(i, x[i], y[i]);}
            get(f);
          end;
          no:=i;

          xsum:=0;
          ysum:=0;
          xysum:=0;
          xxsum:=0;

          for i:= 1 to no do
          begin
            xsum := xsum + x[i];
            ysum := ysum + y[i];
            xysum := xysum + x[i]*y[i];
            xxsum := xxsum + x[i]*x[i];
          end;
          b := (no*xysum - xsum*ysum)/(no*xxsum - xsum*xsum);
          a := (ysum - b*xsum)/no;
          err := 0;
          for i:=1 to no do
          begin
            err := err + abs(y[i] - (a + b*x[i]));
{            writeln(x[i], y[i], a+b*x[i]);}
          end;
          writeln(id_key, a, b, err);
          f_lsr^.lsr_id:= id_key;
          f_lsr^.lsr_a := a;
          f_lsr^.lsr_b := b;
          f_lsr^.lsr_err := err;
          f_lsr^.lsr_gm  := gm;
          f_lsr^.lsr_sp  := speed_v[i1];
          put(f_lsr);
          writeln(f_rpt, id_key, gm, a, b, err);
        end
        else
          writeln(' File not found!');
      end;
    end;
    close(f);
    close(f_lsr);
end;

{Least square regression for one GM}
procedure lsr_gm1;
type
  tspeed = packed array[1..2] of char;
var
  f_rpt_n         : packed array[1..10] of char;
  f_rpt           : text;
  id_key          : tkey;
```

```
  speed          : tspeed;
  speed_a        : array[1..8] of tspeed;
  speed_v        : array[1..8] of real;
  i, il, kk, no  : integer;
  x, y, dd       : array[1..7] of real;
  xsum, ysum, xysum, xxsum, a, b  : real;
  err, gm, angle, len         : real;
  exp_id         :   char;

begin
  speed_a[1] := '00';
  speed_a[2] := '03';
  speed_a[3] := '05';
  speed_a[4] := '07';
  speed_a[5] := '09';
  speed_a[6] := '11';
  speed_a[7] := '13';
  speed_a[8] := '15';

  speed_v[1] := 0.0;
  speed_v[2] := 0.3;
  speed_v[3] := 0.5;
  speed_v[4] := 0.7;
  speed_v[5] := 0.9;
  speed_v[6] := 1.1;
  speed_v[7] := 1.3;
  speed_v[8] := 1.5;

  write('Input len:');
  readln(len);

  write('Initial Angle:');
  readln(angle);
  angle := 3.1416*angle/180.0;

  write('Input exp_id:');
  readln(exp_id);

  write('Input report name:');
  readln(f_rpt_n);

  open(f_rpt, f_rpt_n, history:=unknown);
  rewrite(f_rpt);

  open(f, 'lab0801.dat', history:=old,
       organization:=indexed, access_method:=keyed);
  resetk(f, 0);

  if ((exp_id='A') or (exp_id ='B') or (exp_id='C')
       or (exp_id='D') or (exp_id='E')) then
  begin
    findk(f, 0, 'E6301');
    for il := 1 to 7 do
```

```
      dd[i1] := f^.b_eng[5,1];
      get(f);
   findk(f, 0, exp_id+'0001');
   for i1 := 1 to 7 do
      dd[i1] := f^.b_eng[5,1] - dd[i1];
      get(f);
end;

if ((exp_id='H') or (exp_id ='I') or (exp_id='J') or (exp_id='K'))
then
begin
   findk(f, 0, exp_id+'6501');
   for i1 := 1 to 7 do
      dd[i1] := f^.b_eng[5,1];
      get(f);
   findk(f, 0, exp_id+'0001');
   for i1 := 1 to 7 do
      dd[i1] := f^.b_eng[5,1] - dd[i1];
      get(f);
end;

if ((exp_id='L') or (exp_id ='M') or (exp_id='N') or (exp_id='O')
   or (exp_id='P') ) then
begin
   findk(f, 0, exp_id+'6601');
   for i1 := 1 to 7 do
      dd[i1] := f^.b_eng[5,1];
      get(f);
   findk(f, 0, exp_id+'0001');
   for i1 := 1 to 7 do
      dd[i1] := f^.b_eng[5,1] - dd[i1];
      get(f);
end;

clearsc;
kk := 5;

   for i1 := 1 to 8 do
   begin
     id_key := exp_id + speed_a[i1] + '01';
     writeln(id_key);

     speed:=substr(id_key, 2,2);
     findk(f, 0, id_key);

     if not ufb(f) then
     begin
       clearsc;
       i:=0;
       while (substr(f^.id, 2,2)=speed) and (not ufb(f)) do
       begin
         i:=i+1;
         x[i]:=(abs(f^.amp[1])+abs(f^.amp[2])+abs(f^.amp[3]) )/3.0;
```

```
          y[i]:=f^.b_eng[kk,1] - dd[i];
{          writeln(i, x[i], y[i]);}
          get(f);
        end;
        no:=i;

        xsum:=0;
        ysum:=0;
        xysum:=0;
        xxsum:=0;

        for i:= 1 to no do
        begin
          xsum := xsum + x[i];
          ysum := ysum + y[i];
          xysum := xysum + x[i]*y[i];
          xxsum := xxsum + x[i]*x[i];
        end;
        b := (no*xysum - xsum*ysum)/(no*xxsum - xsum*xsum);
        a := (ysum - b*xsum)/no;
        err := 0;
        for i:=1 to no do
        begin
          err := err + abs(y[i] - (a + b*x[i]));
        end;
        writeln(f_rpt, speed_v[i1]/sqrt(9.8*len),' ', a,' ', b,' ',
                a + b*angle, ' ', err);
      end
      else
        writeln(' File not found!');
    end;

  close(f);
  close(f_rpt);
end;

{begin main program}
begin
  while true do
  begin
    writeln;
    writeln;
    writeln;
    writeln('________________________________________________________________');
    writeln('| VAX PASCAL/VMS 5.4          Suimin Zhang   Sept.1992 |');
    writeln('|                                                       |');
    writeln('|                      <MAIN MENU>                      |');
    writeln('|                                                       |');
    writeln('|   EXPERIMENTAL DATA MANAGEMENT AND PROCESSING         |');
    writeln('|                                                       |');
    writeln('|              1.   DATA MANAGEMENT                     |');
    writeln('|                                                       |');
    writeln('|              2.   DATA ANALYSIS AND REPORT            |');
```

```
writeln('|                                             |');
writeln('|            0.  EXIT                         |');
writeln('|                                             |');
writeln('|            Please Select                    |');
writeln('|                                             |');
writeln('|                                             |');
writeln('|_____________________________________________|');
writeln;
writeln;
writeln;
read(m_select);
if m_select = 0 then goto 11;

if m_select = 1 then
begin
  while true do
  begin
    writeln;
    writeln;
    writeln;
    writeln;
    writeln;
    writeln;
    writeln('          <-- Data Management -->');
    writeln;
    writeln('        1. create(or rewrite) dababase');
    writeln('        2. Update forwardspeed');
    writeln('        3. Update initail angle');
    writeln('        4. Append analytical result by E.M.F. Method');
    writeln('        5. Data examination (error finding) ');
    writeln('        6. Update analytical result by E.M.F. Method');
    writeln('        7. List records in the Databse');
    writeln;
    writeln('        0. Return to Main Menu');
    writeln;
    write('          Please select:');
    writeln;
    writeln;
    writeln;
    writeln;
    writeln;
    read(m_choice);
    case m_choice of
      1: begin
           writeln(' All data in the database may be deleted !');
           write(' Are you sure?(Y/N)');
           m_yn:='N';
           readln(m_yn);
           If ((m_yn='Y') or (m_yn='y')) then
             cr_db
           else
             writeln(' Database not created (or rewrite)');
         end;
```

```
      2: updt_sp;
      3: updt_ag10;
      4: appe_db;
      5: exam_db;
      6: updt_db;
      7: ls_db;
      0: goto 13
    end;
  end;
end;
if m_select = 2 then
begin
  while true do
  begin
    writeln;
    writeln;
    writeln;
    writeln;
    writeln;
    writeln;
    writeln('      <-- Data Analysis and Report -->');
    writeln;
    writeln('      1. output damping coef. of one speed');
    writeln('      2. compare two file');
    writeln('      3. single least square regression');
    writeln('      4. batch least square regression(LSR0801.DAT)');
    writeln('      5. least square regression for one GM');
    writeln('      6. output damping coef. of speed vie angle ');
    writeln('      7. output damping coef. of 1 amplitude amp[1]');
    writeln('      8. output damping coef. of differenr OMEGA');
    writeln('      9. statistics natural frequency');
    writeln('      0. Return to Main Menu');
    writeln;
    write('       Please select:');
    writeln;
    writeln;
    writeln;
    writeln;
    read(m_choice);
    case m_choice of
      1: rpt_b;
      2: comp;
      3: lsr;
      4: lsr_bat;
      5: lsr_gm1;
      6: rpt_va;
      7: rpt_ampl;
      8: rpt_freq;
      9: rpt_omega;
      0: goto 13
    end;
  end;
end;
```

```
  13:
  end;
11:
end.
```