"One fish, Two fish, Red fish, Blue fish"

Mathematically Modeling a Fresh Fish Detector

Ibrahim Agyemang (ibagyemang[at]hotmail.com)
Erik Andries (andriese[at]tahpcc.unm.edu)
Dhavide Aruliah (dhavide[at]fields.utoronto.ca)
Mélanie Beck (beck[at]math.mcgill.ca)
Qingguo Li (qlib[at]sfu.ca)
Colin Macdonald (cbm[at]math.sfu.ca)
Matthias Mück (mueck[at]math.toronto.edu)
Robin Swain (rswain[at]math.mun.ca)

Abstract. The PIMS Mathematical Modeling spring workshop presented six different environments to be considered for modeling during the program. For this group, Chris Budd proposed that we study data obtained through experiments using a device designed to determine the freshness of fish. Through an electric current applied to a coil, a needle-shaped probe is projected by a force directly on the surface of a test sample. The depth to which the probe pushes the surface is recorded by the coil as a function of time. The goal of this project is to use the data to indicate what mechanisms govern the dynamics of the probe over time, namely models of ordinary differential equations from which parameters can be extracted to determine fresh fish from those which are not.

Key words. Mathematical modelling, differential equations, noise reduction, nonlinear, visco–elastic medium, impact oscillator.

Contents

1 Introduction

Have you ever been to the supermarket to buy fresh fish and wonder: Just how fresh is this fish? Is the specified freshness of the fish accurate? If not, how could you tell?

In this report we describe the mathematical analysis of a system which is proposed to measure the freshness of fish. The device used to test freshness is a thin needle–like probe with a coil which provides an electromagnetic force to the probe and also measures its motion. We attempt to capture the dynamics of the motion of the probe in a mathematical model which can fit the given data. In doing so we hope to extract information on fish freshness from some fish–dependent parameters in the model(s).

The first section deals with cleaning up the data. Our group has been provided with some raw data collected by the fish–probe apparatus for several different materials:

1. A plaice fish
2. A foam or sponge
3. A clear cling wrap medium
4. A human hand

The data contains noise which was filtered for subsequent analysis. Fast Fourier Transforms (FFT) and wavelet de–noising techniques were applied to the data to remove the noise. The re–constructed data was useful in obtaining estimates for the velocity and acceleration of the probe during the experiments. The scope of our analysis was only concerned with fitting the data for a plaice fish and foam sponge, and does not include analysis of the cling wrap and human hand. Mathematical modeling techniques including nonlinear optimization and the physics of oscillatory systems were used to describe the dynamics of the probe.

2 Data Filtering and Estimation of Acceleration

In this part of the project, wavelet de-noising techniques are applied to the empirical fish probe data for filtering. Based on the filtered data, the acceleration was computed numerically.

2.1 Wavelet Shrinkage De–noising

Our objective was to suppress the noise and recover the signal. Both FFT and wavelet approaches were implemented to fulfill the task. Because of the multi–resolution property of the wavelet transform, wavelet–based de–noising produced better results than the FFT approach.

To recover a signal the noise must be removed before proceeding with further data analysis. The wavelet de-noising procedure consists of three steps:

1. A linear forward wavelet transform
2. A nonlinear shrinkage de-noising
3. A linear inverse wavelet transform

Let $X(t)$ represent a set of observed data, and assume

$$X(t) = S(t) + N(t) \tag{2.1}$$

contains the true signal $S(t)$ with additive noise $N(t)$ as functions in time t to be sampled. Let $\mathcal{W}(\cdot)$ and $\mathcal{W}^{-1}(\cdot)$ denote the forward and inverse wavelet transform operators. Let $\mathcal{D}(\cdot, \lambda)$ denote the de-noising operator with threshold λ. We intend to de-noise $X(t)$ to recover $\widehat{S}(t)$ as an estimate of $S(t)$. The basic version of the procedure consists of three steps, including decomposition, obtaining thresholding detail coefficients, and reconstruction. It is summarized as

$$\begin{aligned} Y &= \mathcal{W}(X) \\ Z &= \mathcal{D}(Y, \lambda) \\ \widehat{S} &= \mathcal{W}^{-1}(Z) \end{aligned} \tag{2.2}$$

For the de-noising operator $\mathcal{D}$, given threshold λ for the data $\mathcal{U}$,

$$\mathcal{D}(\mathcal{U}, \lambda) = \operatorname{sgn}(\mathcal{U}) \max(0, |\mathcal{U}| - \lambda) \tag{2.3}$$

defines nonlinear soft thresholding. Many different schemes have been developed on selection of de-noising operator [1, 2].

In this application, the data was first processed by using 5–level Haar wavelet transform, then we removed the finer wavelet coefficients b_5 and b_4. After that, we performed the inverse transform and recovered the clean data. The processes were carried out by using the wavelet toolbox in Matlab [3]. The results for the fish data are shown in Figures 1 to 2. Figure 2 contains the wavelet de–noising results and the original fish data (offset for comparison). The clean signals for the foam and cling film are included in Figures 3 and 4.

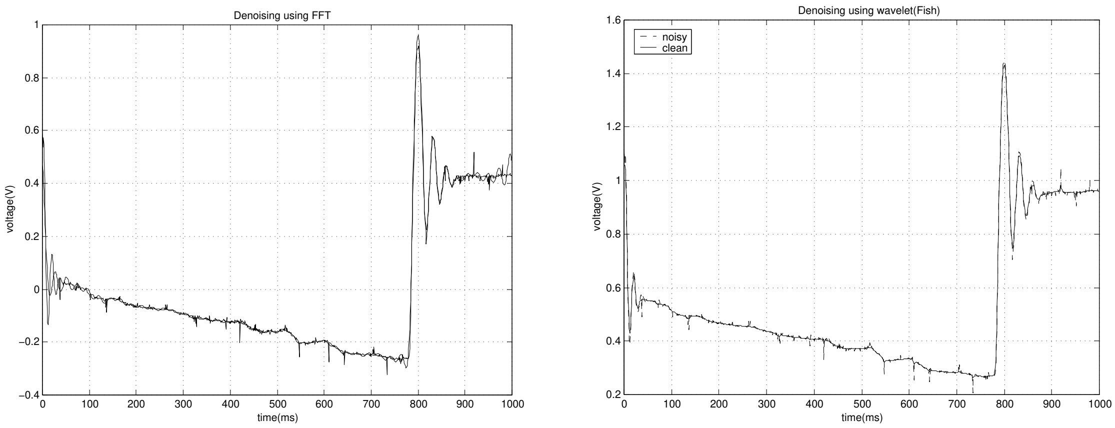

Figure 1: FFT (left) and wavelet (right) de–noising results for the original fish data.

2.2 Numerical Computation of Acceleration

After we obtained the clean data by using wavelet de–noising, the accelerations were computed using finite difference approximation. The position data $x(t)$ can be calculated from the de–noised voltage data $u(t)$ as follows:

$$x(t) = 10^{-3} \times 0.8u(t)\mathrm{mm/V} \tag{2.4}$$

First the velocity $v(t)$ can be computed as

$$v(t) = \frac{x(t+\Delta t) - x(t)}{\Delta t} \tag{2.5}$$

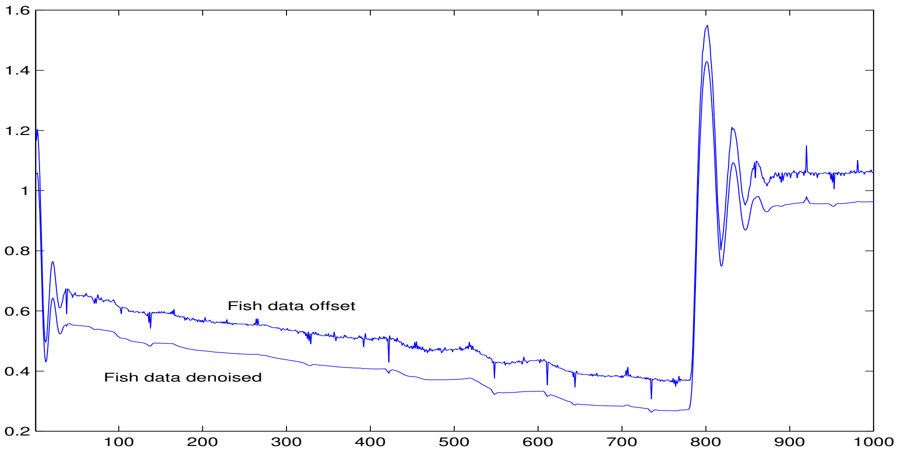

Figure 2: Comparison of the wavelet de–noised data and the original data (offset) for a plaice fish.

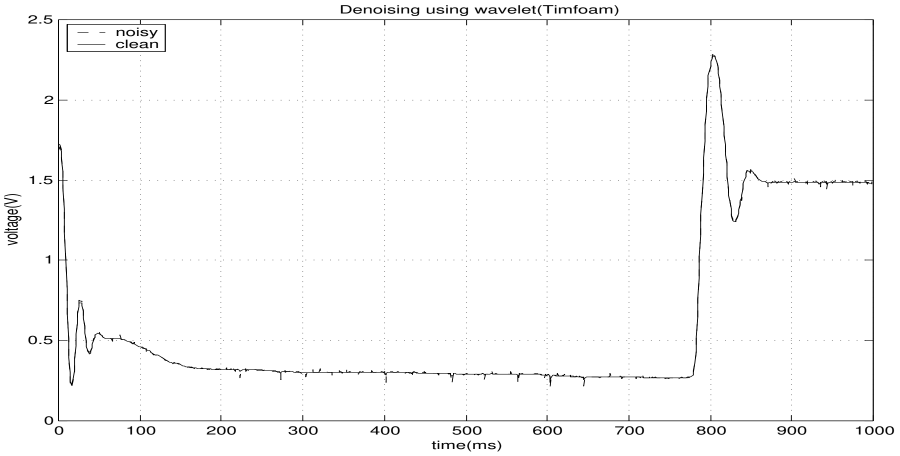

Figure 3: De–noising results of timfoam data.

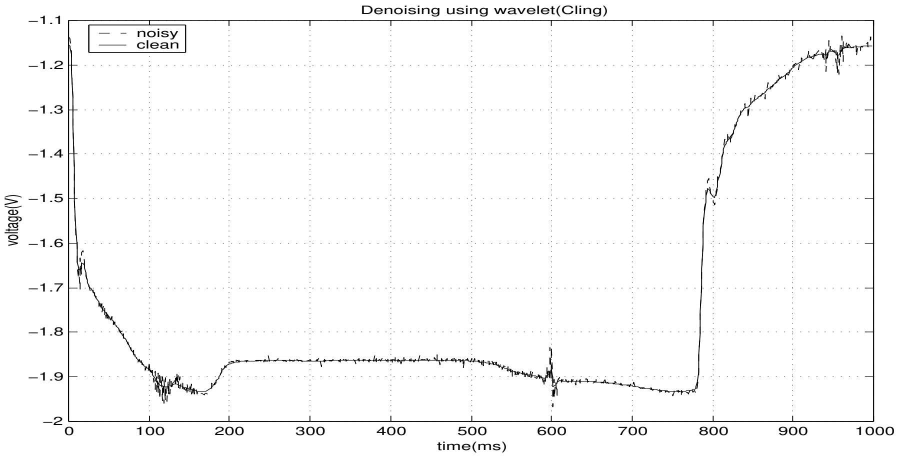

Figure 4: De–noising results of cling film data.

where $\Delta t = 10^{-3}$ is the sample rate of the sensor.

Because of the differentiation process, the noise in the position signal was amplified, and in order to increase the signal to noise ratio (SNR), the de–noising routine was also applied to the velocity data $v(t)$ to generate the clean signal $v_1(t)$. The acceleration $a(t)$ was computed as

$$a(t) \approx \frac{a(t+\Delta t) - a(t)}{\Delta t} \tag{2.6}$$

With the de–noised data we obtained continuous acceleration data for further analysis.

3 Analysis of the Signal

The graphs describing the reponse from the plaice fish and foam material are displyed in Figures 7 and 8. It is obvious from the Figures that the motion of the probe exhibits three distinct behaviors during an experiment (for both the fish and foam cases). Therefore we model these phases as independent processes:

1. Phase A: The initial phase resembles an oscillating decaying exponential function which might be modelled as a mass-spring system with

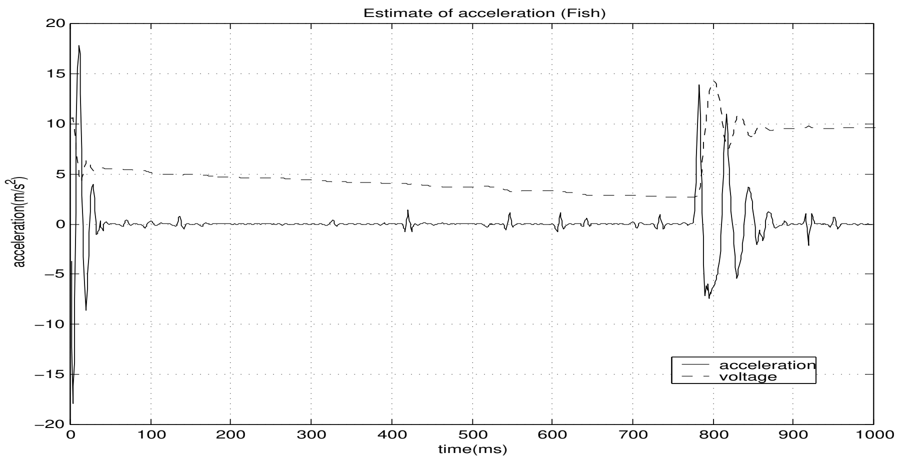

Figure 5: Estimation of acceleration from clean data (fish).

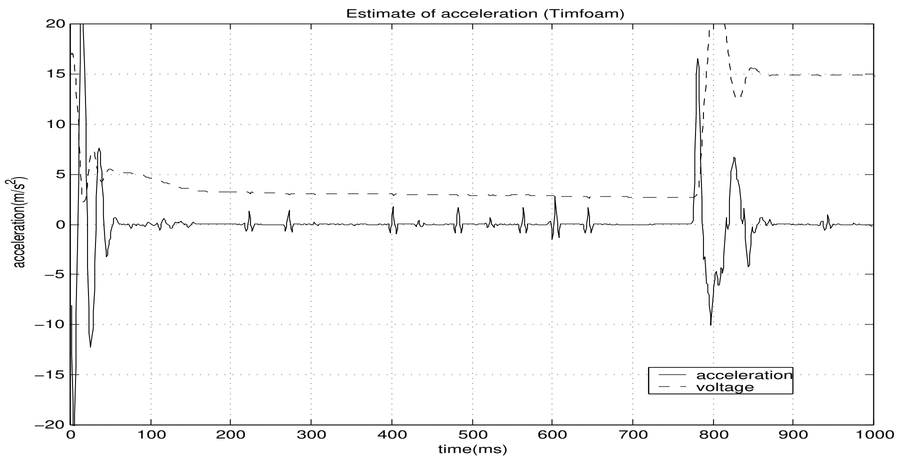

Figure 6: Estimation of acceleration from clean data (foam).

damping. This phase coincides with the initial force applied by the probe on the fish or foam.

2. Phase B: The probe continues to exert a constant force on the fish (foam), and a linearly (exponentially) decaying function is observed which indicates motion with constant or slowly changing velocity.

3. Phase C: In the final phase damped oscillations are again observed as the force is removed.

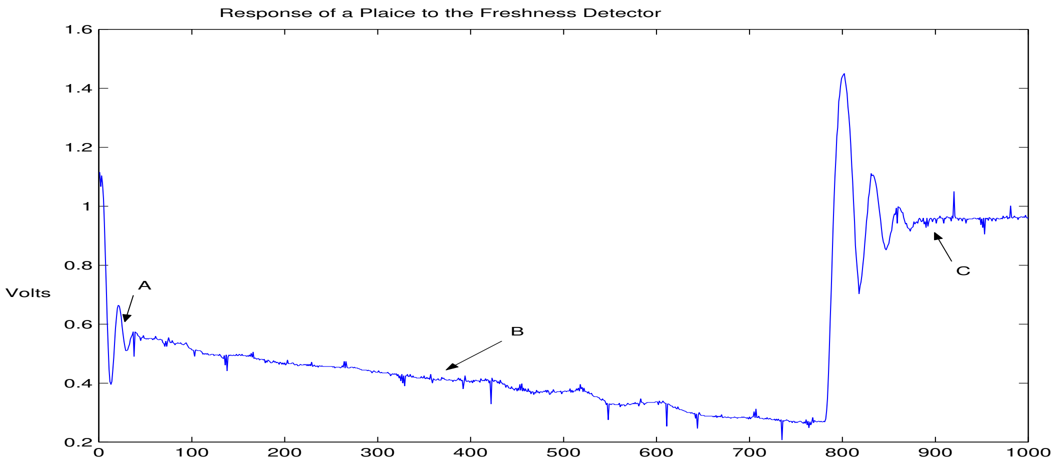

Figure 7: The response of the probe to application of a plaice fish.

The following sections describe the motion of the fish–probe system in the the three phases as isolated processes.

3.1 Phase A

Due to the oscillatory nature of the voltage (position) over time, we first attempted to model the phenomena by a damped linear oscillator

$$M\ddot{x} + \beta\dot{x} + \alpha x + Mg = 0$$

where $M = 10$g is the mass of the probe, g is the gravity constant and $x = x(t)$ is the position of the probe.

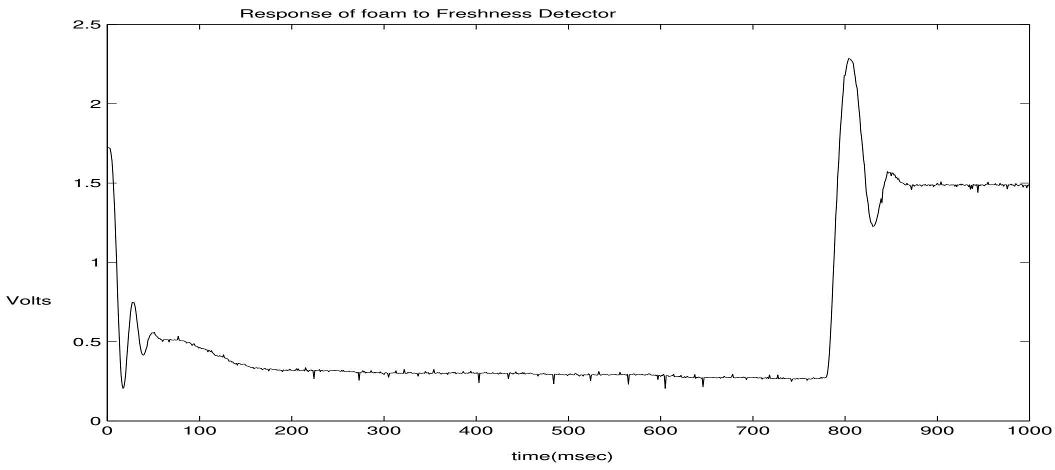

Figure 8: The response of the probe to application to foam block.

With constant coefficients, the solution to the above equation is

$$x(t) = e^{-\delta t}\left[A\cos(\omega t) + B\sin(\omega t)\right] + D,$$

where δ is the decay rate, D is the vertical shift and ω is the frequency. These intermediate parameters ω and δ can be expressed by the ODE parameters:

$$\omega = \sqrt{\frac{\alpha}{M} - \delta^2} \quad \text{and} \quad \delta = \frac{\beta}{2M}.$$

We now have to estimate the unknown parameters of the linear ordinary differential equation from the time series data and then inspect the compatibility of the reconstructed model with the data. The MATLAB nonlinear least squares fitting algorithm, NLINFIT, was used to perform the parameter estimation. The results in Figure 9 indicate that Phase A can indeed be modeled by the damped linear oscillator.

The initial and final phases of the motion of the probe were similar in nature - both were oscillating decaying exponential functions.

3.2 Phase B

The second phase of the time series is observed from 50 to 800ms after the initial contact. It is characterised by a steady decay in the magnitude of

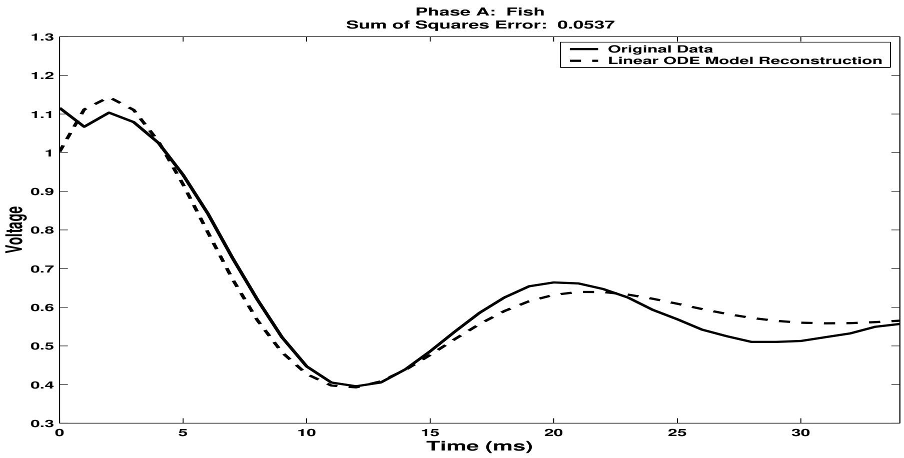

Figure 9: The fit obtained for Phase A by modeling the original fish data with a damped linear oscillator.

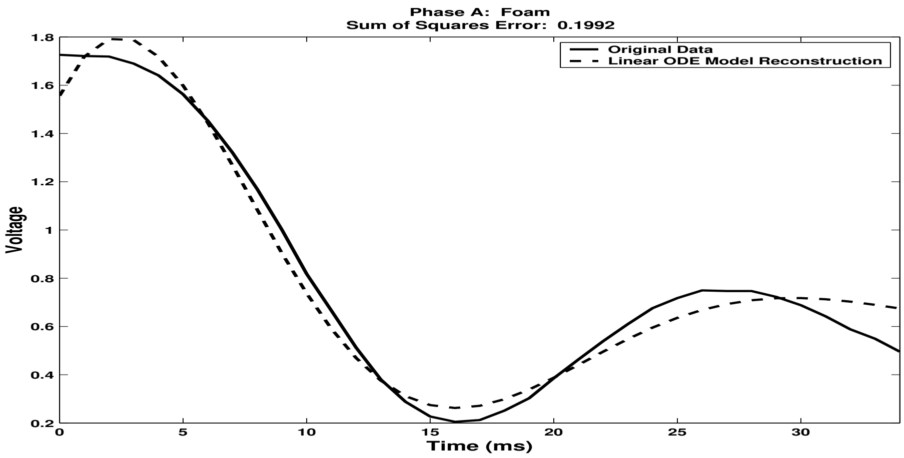

Figure 10: The fit obtained for Phase A by modeling the original foam data with a damped linear oscillator.

displacement following the transient oscillatory behaviour observed in Phase A. We hypothesized that phase B consists of an inelastic deformation because the final equilibrium state at $t = 1000$ms is somewhat depressed below the initial condition at $t = 0$.

To model the decrease in the magnitude of the displacement, we attempt to fit the data first with a decreasing exponential function of the form

$$x(t) = Ae^{-\gamma t} + B, \tag{3.7}$$

with paramters A, B and γ. As an alternative, we used a linear model,

$$x(t) = Ct + D, \tag{3.8}$$

with parameters C and D.

The residuals are minimized in the least squares sense for the above data sets giving the functions

$$\begin{aligned} x(t) &= 28.45e^{-0.01368t} - 27.91 \text{ or} && (3.9)\\ x(t) &= -0.3874t + 0.5357 && (3.10) \end{aligned}$$

that fit the fish data and

$$\begin{aligned} x(t) &= 0.2306e^{-19.56t} + 0.2994 \text{ or} && (3.11)\\ x(t) &= -0.3104t + 0.3928 && (3.12) \end{aligned}$$

for the foam data. These results are plotted in Figure 11 and we note the linear and exponential functions are essentially identical for the fish data, both having a least squares residual of about 0.05. For the foam data, the exponential function provides a much better fit with a least squares residual of 0.016 as opposed to 0.47 for the linear fit; this is to be expected because, as seen in Figure 11, the linear function cannot model the curvature of the data effectively.

As a possible model of this behaviour, consider a particle subjected to a constant force F moving in a visco-elastic medium with elastic coefficient α and viscosity β. With this model, the kinematics are determined by the equation

$$\alpha x(t) + \beta \frac{dx}{dt} = F. \tag{3.13}$$

Integrating this equation gives the exact solution

$$x(t) = \left(X_0 + \frac{F}{\alpha}\right) e^{-\frac{\alpha}{\beta}t} + \frac{F}{\alpha}, \tag{3.14}$$

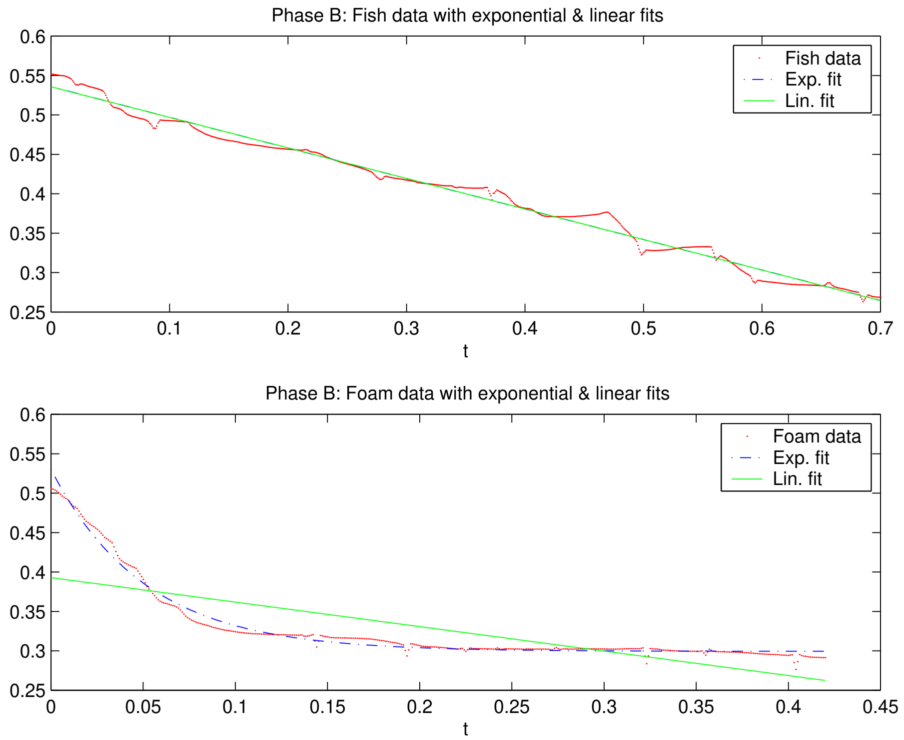

Figure 11: Exponential and linear functions fit to the phase B portion of the data from the fish and the foam experiments.

where $x(0) = X_0$ is the initial condition for the ODE. In the absence of an elastic term, the kinematics are determined by

$$\beta \frac{dx}{dt} = F \tag{3.15}$$

with corresponding exact solution

$$x(t) = X_0 + \frac{F}{\beta} t. \tag{3.16}$$

It also seems plausible that we could derive Equation (3.16) by considering Darcy's Law for the flow of fluid through a porous medium (the fish flesh). However, we unfortunately did not have sufficient time to develop this idea.

The parameters in the solutions Equation (3.14) and Equation (3.16) can be determined by matching with Equation (3.7) and Equation (3.8).

Further, in the limit of vanishing elasticity α, the solution Equation (3.14) reduces to Equation (3.16) since

$$\begin{aligned}
x(t) &= \left(X_0 - \frac{F}{\alpha}\right) e^{-\frac{\alpha}{\beta} t} + \frac{F}{\alpha} \\
&= \left(X_0 - \frac{F}{\alpha}\right) \left(1 - \frac{t}{\beta}\alpha + \frac{1}{2}\left(\frac{t}{\beta}\right) \alpha^2 + \cdots\right) + \frac{F}{\alpha} \\
&= X_0 + (F - \alpha X_0)\left(\frac{t}{\beta}\right) - \frac{1}{2}(F - \alpha X_0)\left(\frac{t}{\beta}\right)^2 \alpha + \cdots \\
&\to X_0 + \frac{F}{\beta} t \quad \text{as} \quad \alpha \to 0.
\end{aligned}$$

Thus, this model suggests that the foam acts as a visco-elastic medium during phase B while the fish acts as a viscous medium with vanishing elasticity. Indeed, this is supported by the small time constant γ in the exponential fit for the fish data.

3.3 Phase C

We define phase C as beginning when the force on the probe is released at around 780ms. We consider both the fish and foam data during this phase.

3.3.1 The Fish Data

We first consider the fish data. After the force is released, the probe is pushed upwards by the skin of the fish and enters an apparent damped oscillation.

Our initial model was to describe the motion by the ODE

$$\mu\ddot{x} + \widetilde{\beta}\dot{x} + \widetilde{\alpha}x + \mu g = 0, \tag{3.17}$$

where $\mu = 1 + \frac{M}{m}$ is the effective mass of the oscillator (M is the mass of the probe, m is a mass unit of the skin) and $\widetilde{\alpha}$ and $\widetilde{\beta}$ are respectively the restoring and friction parameters. The μg term describes the force due to gravity. We fitted the general solution

$$x(t) = Ae^{-\delta t}\cos(wt - \phi) + D, \tag{3.18}$$

to the data by matching the parameters A, δ, ω, ϕ and D where δ and ω are related to ODE cofficients by

$$\delta = \frac{\widetilde{\beta}}{2\mu}, \tag{3.19}$$

$$\omega^2 = \frac{\widetilde{\alpha}}{\mu} - \delta^2. \tag{3.20}$$

Results of a least squares fit are shown in Figure 12.

We note that the fit apparently matches the first period of the oscillations quite well but the solution quickly drifts out of phase. Therefore, the model described by Equation (3.18) should be modified. One strategy is to replace the constant coefficients $\widetilde{\alpha}$ and $\widetilde{\beta}$ with functions of x and $\dot{x}$ respectively. We considered these corrections up to second order, i.e.,

$$\widetilde{\alpha}(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 \tag{3.21}$$

$$\widetilde{\beta}(\dot{x}) = \beta_0 + \beta_1\dot{x} + \beta_2\dot{x}^2 \tag{3.22}$$

$$\tag{3.23}$$

However, no significant change was noticed between this non-linear model and the linear approach shown in Figure 12.

After the failure of the non-linear model, we attempted to gain additional insight into the physics of the problem. Therefore, we considered the velocity and acceleration of the probe as computed from the de-noised data (see Figure 13).

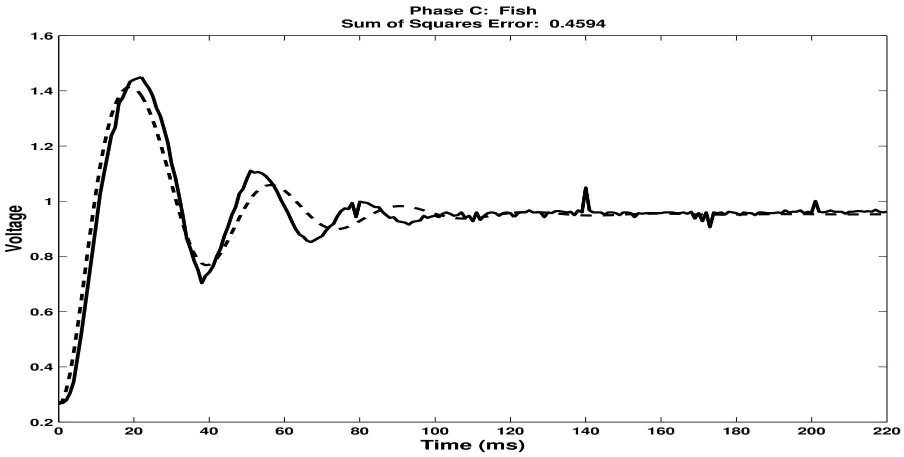

Figure 12: Phase C — The fit obtained by modeling the original fish data with a damped nonlinear oscillator.

We noticed that between times $t_1 = 790$ms and $t_2 = 808$ms, the acceleration is almost constant, indicating that the probe is a constant acceleration over 18ms which does not coincide with motion due to a damped oscillator. Note the agreement with the velocity which appears to have a linear behaviour during the same time. However, before and after this acceleration plateau, the motion seems plausibly described by damped oscillations. For convenience we call the time interval $[790\text{ms}, 808\text{ms}]$ Section II and refer to the intervals $[780\text{ms}, 789\text{ms}]$ and $[809\text{ms}, 1000\text{ms}]$ as Section I and Section III, respectively.

The different types of motion of the probe in these three sections suggest that the probe is starting to perform a harmonic oscillation (Section I) and looses contact near $t_1 = 790$ms. During Section II, the probe (apparenty in "free-fall") is dealt with below. Near time $t_2 = 808$ms the skin and the probe collide and again behave as a single oscillator. The motion in Section I and III must be driven by the ODE given in Equation (3.18). Therefore the frequency ω, the decay rate δ, and the baseline D of the signal must be in agreement. However, we allowed that the intermediate Section II introduces an amplitude and a phase correction to the solution of the ODE in the last

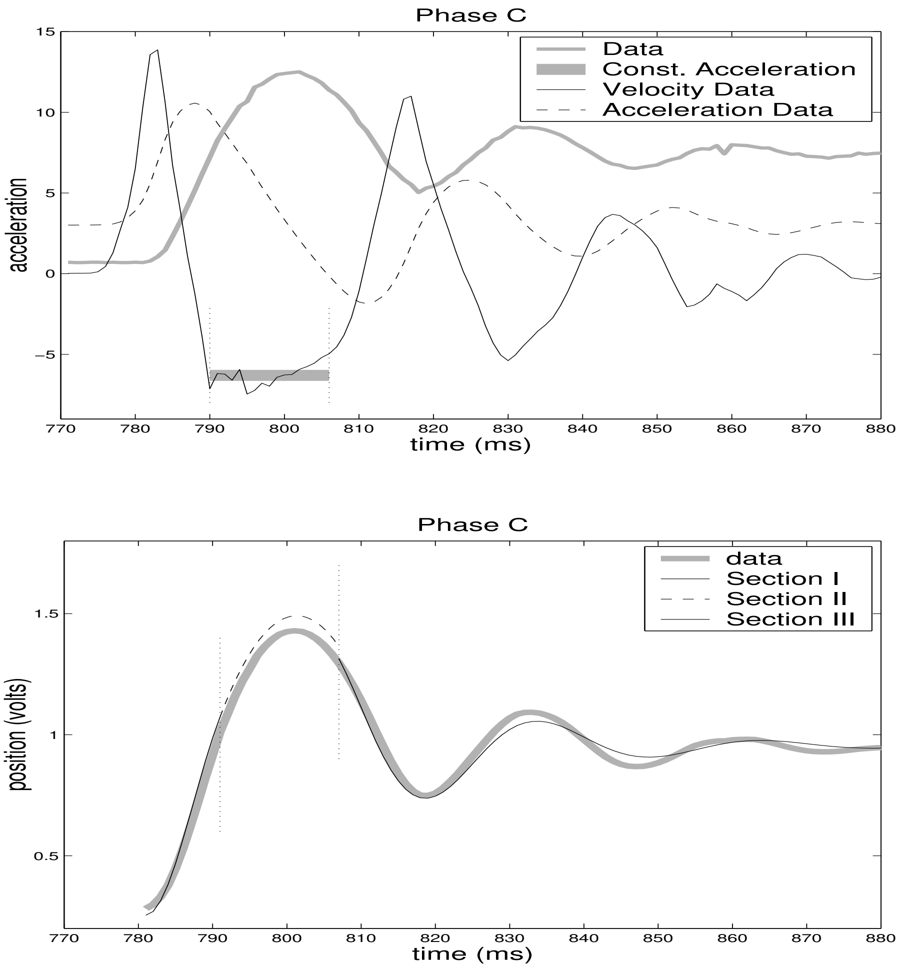

Figure 13: (Above) Original Phase C data with acceleration overlay. The region where the probe is in free fall is highlighted.
(Below) The fit obtained by modeling the original fish data as an impact oscillator (loss of contact model).

section. We then fitted the (de-noised) data to the functions

$$x(t) = Ae^{-\delta t}\cos(\omega t + \phi) + D, \quad t \in \text{Section I}, \tag{3.24}$$

$$x(t) = Be^{-\delta t}\cos(\omega t + \theta) + D, \quad t \in \text{Section III}. \tag{3.25}$$

The results of the least squares fit is displayed in the relevant parts of Figure 13 (i.e., the two solid curves). We emphasize that the fit is reasonably close to the data. This confirms our conjecture that the probe performs a damped oscillation as long as it is in contact with the skin.

We focus our attention on Section II to understand the motion of the probe over the entire time range of Phase C. After losing contact with the skin, the probe is constantly accelerated over the time range of Section II and therefore its position function $x(t)$ obeys the ODE

$$\ddot{x} - \widehat{g} = 0. \tag{3.26}$$

The constant $\widehat{g}$ can be determined from the acceleration data in Figure 13. We expected that $\widehat{g}$ match the gravitational constant $g = -9.81\text{m/s}^2$. However, by taking the average of the acceleration data over the plateau in Section II, $\widehat{g}$ turned out to be $\widehat{g} = -6.3\text{m/s}^2$. A plausible explanation for this reduced gravitational acceleration could be the effects of friction from inside the apparatus.

To solve Equation (3.26) we chose initial conditions such that the position function $x(t)$ and the velocity $\dot{x}(t)$ became contionuous at the point, $t_1 = 790\text{ms}$. The solution is a parabolic path and is plotted in Figure 13 as a dashed curve. Note that the "free fall" not only fits the motion in Section II very well but also connects to the damped oscillatory motion in Section III.

Unfortunately, it is difficult to determine $\widetilde{\alpha}$ and $\widetilde{\beta}$ because of their dependence on μ. This is problematic because we have very little information about μ; mostly because we did not succeed in finding a way to extract significant information about the motion of the fish skin during Section II. A possible approach is that some information could be extracted by assuming that momentum is conserved by the collision at t_2. However, we had insufficient time to properly persue this option.

3.3.2 The Foam Data

We began modeling the foam data in much the same way as we initially modeled the fish data, i.e., as a a damped linear oscillator. The resulting model was not a good fit.

The next model attempted was the generalized damped oscillator with non-linear coefficients (again, with up to quadratic corrections) for viscosity, $\beta = \beta_0 + \beta_1 \dot{x} + \beta_2 \dot{x}^2$, and the restoring force, $\alpha = \alpha_0 + \alpha_1 x + \alpha_2 x^2$. This new model demonstrated considerable improvement over the linear case, i.e., the reconstructed ODE model was an excellent fit (see Figure 14.

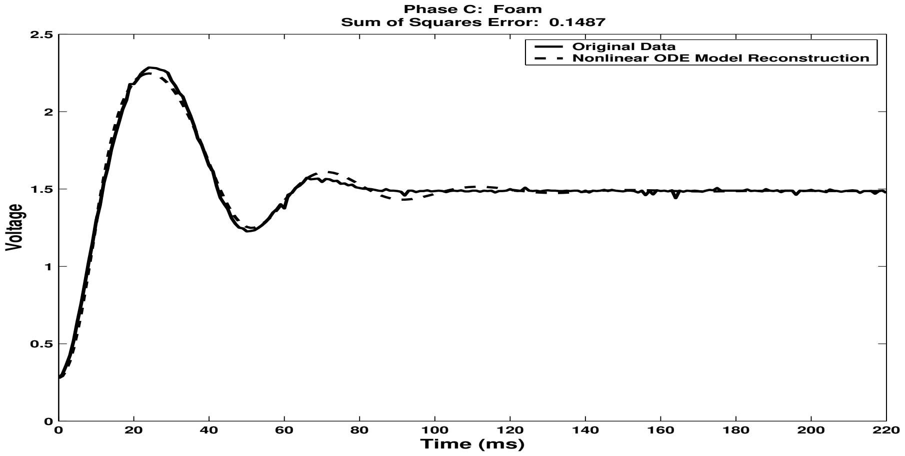

Figure 14: Phase C — The fit obtained by modeling the original foam data with a damped non-linear oscillator.

From here, we can conclude that the probe probably did not leave the surface during Phase C. Overall, the probe response for foam can accuractely be modeled as a visco-elastic system, as opposed to the probe response for the fish — indicating that the fish is not a perfect visco-elastic material.

4 Conclusions

We gained a fundamental understanding of the dynamics of the given problem for both the fish and the foam data. In this sense, the modeling was successful. In particular, the loss-of-contact model as an impact oscillator during Phase C was surprisingly successful. However, although we have identified many parameters of the model, correlating these parameters with freshness requires several (perferably many) samples of fish of various freshness. Thus, when it comes to predicting freshness of the fish, we simply do not have enough data to make any claims.

It seems plausible that the restoring and damping parameters would be indicators of freshness. Recall that, in Phase C, we were unable to extract the values of these parameters from the frequency, ω, and decay rate, δ, since we could not determine μ. However, it is possible that ω and δ are themselves indicators using μ as a fish-dependent constant.

5 Acknowledgements

Many thank goes to Chris Budd for providing the group with an interesting project to model and for his guidance as a mentor. Also, many thanks to Jim Verner for his support, laptop, and car.

A Source Code

A.1 Phase A Codes

A.1.1 Phase_A_Fit.m

```
%===================================================================================
function [P, V, F]= Phase_A_Fit( time, voltage, t1, t2, baseline )
% Estimate the parameters (coefficients) of the nonlinear ODE for Phase A
% time:       time vector of the signal (SCALE IS 'MS', i.e. 0ms, 1ms, 2ms, ...
% voltage:    position data of the probe/needle
% -- time and voltage must be column vectors
% t1:         time of the first crest in the data
% t2:         time of the second crest in the data
% -- from t1 and t2, a rough estimate of the decay rate and period can be computed,
%    whereby all subsequent initial parameters can be computed
% baseline:   guess for the steady-state voltage for Phase A
%===================================================================================

    % Reconstructed Voltage
    global V;

    % Estimate parameters
    Mass    = 10;
    T       = t2-t1;                               % Period
    Omega   = 2*pi/T;                              % Frequency
    Delta   = -1/T * log(voltage(t2)/voltage(t1)); % Decay rate
    Beta    = 2*Mass*Delta;                        % Damping coefficient
    Alpha   = Mass*(Omega^2 + Delta^2);            % Restoring force coeffient
    Shift   = baseline;                            % Steady-state voltage
    A       = voltage(1)-baseline;                 % Initial amplitude for cos(omega t)
    B       = 0.1;                                 % Initial amplitide for sin(omega t)
    P0      = [Delta Omega Shift A B];             % Consolidate initial parameters into one vector

    %Find Optimal Parameters
    P = nlinfit(time,voltage,@Phase_A_NLObj,P0);

    % Unroll the parameters
    Delta = P(1);
    Omega = P(2);
    Shift = P(3);
    A     = P(4);
    B     = P(5);

    % Reconstruct the voltage from the estimated parameters
    V = exp(-Delta*time).*( A *cos(Omega*time) + B*sin(Omega*time) ) + Shift;

    % Compute the sum of squares error term (not normalized to the length of the vector
    F = sum((V-voltage).^2);

return
```

A.1.2 Phase_A_NLObj.m

```
%==========================================================================
function x = Phase_A_NLObj(P,t)
% Objective function to be minimized by NLINFIT
%==========================================================================

    % Unroll parameters
    Delta = P(1);
    Omega = P(2);
    Shift = P(3);
    A     = P(4);
    B     = P(5);

    % Equation to fit
    x = exp(-Delta * t) .* (A*cos(Omega*t) + B*sin(Omega*t)) + Shift;
return
```

A.1.3 Phase_C_Fit.m

```
%==================================================================================
function [P, V, F] = Phase_C_Fit(time,voltage,t1,t2,baseline)
% Estimate the parameters (coefficients) of the nonlinear ODE for Phase C
% time:       time vector of the signal (SCALE IS 'MS', i.e. 0ms, 1ms, 2ms, ...
% voltage:    position data of the probe/needle
% -- time and voltage must be column vectors
% t1:         time of the first crest in the data
% t2:         time of the second crest in the data
% -- from t1 and t2, a rough estimate of the decay rate and period can be computed,
%    whereby all subsequent initial parameters can be computed
% baseline:   guess for the steady-state voltage for Phase C
%==================================================================================

    % Reconstructed Voltage
    global V;

    % Estimate parameters
    Mass    = 10;
    T       = t2-t1;                                % Period
    Omega   = 2*pi/T;                               % Frequency
    Delta   = -1/T * log(voltage(t2)/voltage(t1));  % Decay Rate
    Beta    = 2*Mass*Delta;                         % Damping coefficient
    Alpha   = Mass*(Omega^2 + Delta^2);             % Restoring Force Coefficient
    Shift   = baseline;                             % Steady-state voltage
    P0      = [[Beta 0 0] [Alpha 0 0] Shift];       % Consolidate initial parameters into one vector
    IC      = [voltage(1)-baseline; 0];             % Initial conditions for ODE

    %Find Optimal Parameters
    [P,F] = fminsearch(@Phase_C_Obj, P0, [], time, voltage, IC);
return
```

A.1.4 Phase_C_Obj.m

```
%==========================================================================
```

```
function f = Phase_C_Obj(P,t,x,IC)
% Compute Objective Function where the minimizer, x, for the objective function is
% derived from the numerical solution of the ODE
%==========================================================================

    % Plug in current ODE parameters and compute current minimizer, x, based on the
    % the numerical solution to the second-order of ODEs
    [T,X] = ode15s(@Phase_C_Ode, t, IC, [], P);

    % Extract reconstructed voltage
    global V;
    shift = P(end);
    V = X(:,1) + shift;
    f = sum( (x-V).^2 );
    disp(sprintf('f = %9.7f',f));
return
```

A.1.5 Phase_C_Ode.m

```
%==========================================================================
function f = Phase_C_Ode(t,x,P)
% M-function which define the system 2nd-order ODE as system of 1st-order
% ODEs which hypothetically describe the behavior of Phase C
%==========================================================================

    % Define the coefficients of the ODE
    M = 10;
    [Beta,Alpha] = tweak_parameters(x,P);

    % Compute the RHS of system of 1st-order ODEs
    f  = zeros(2,1);
    f(1) = x(2);
    f(2) = (-Beta/M)*x(2) + (-Alpha/M)*x(1);
return

%==========================================================================
function [Beta,Alpha] = tweak_parameters(x,P)
% Define the coefficients of the ODE
%==========================================================================

    % Unroll the parameter vector into meaningful components
    beta  = P(1:3);
    alpha = P(4:6);

    % Define the nonlinear damping and restoring force
    Beta  = beta(1)  + beta(2)*x(2)  + beta(3)*x(2)^2;
    Alpha = alpha(1) + alpha(2)*x(1) + alpha(3)*x(1)^2;
return
```

A.2 Phase C Codes

The code phasec_minimizer.m matches the parameters for Section I and III of Phase C as described above. It uses phasec_cost.m as the cost function to minimize.

A.2.1 phasec_minimizer.m

```
%PHASEC_MINIMIZER  This script minimizes the parameters in the
%  general solution for Section I and Section III of Phase C
%  simultaneously

close all;
clear all;

% uncomment for the fish data
%rawdata = load('b1.mat');

% uncomment for the clean fish data:
load('b1_clean');
rawdata(:, 1) = [1:1024]';
rawdata(:, 2) = u1';

% the time boundaries
T10 = 780+1;  % these are off by one!
T1F = 790+1;
T20 = 806+1;
T2F = 879+1;

time1offset = rawdata(T10,1);
time1 = rawdata(T10:T1F, 1) - time1offset;
depth1 = rawdata(T10:T1F, 2);

time2offset = rawdata(T20,1);
time2 = rawdata(T20:T2F, 1) - time2offset;
depth2 = rawdata(T20:T2F, 2);

% this is just used for plotting
alldepth = rawdata(T10:T2F, 2);
alltime = rawdata(T10:T2F, 1);

% number of times to restart optimization.  basically we want to
% avoid local minima
N = 100;

% stores parameters values and the associated cost value
Xs = zeros(N,6);
fvals = zeros(N,1);

figure(1);
clf;
plot(alltime, alldepth, 'b-');
hold on;

for n = 1:N
  % these are initial values for the parameters
  X0 = [.04*rand+.02 .4*rand .1*rand+.9 -1*rand -.4*rand pi*rand-pi/2];

  [X,fval] = fminsearch(@pc_sec1_cost_5, X0, [], time1, depth1, time2, depth2);
  Xs(n,:) = X;
```

```
  fvals(n) = fval;

  delta = X(1);
  omega = X(2);
  D = X(3);
  A = X(4);
  B = X(5);
  phi2 = X(6);

  phi = atan(-delta/omega);

  fval
  y1 = A*exp(-delta*time1) .* cos(omega*time1 + phi) + D;
  y2 = B*exp(-delta*time2) .* cos(omega*time2 + phi2) + D;
  plot(time1+time1offset, y1, 'r-');
  plot(time2+time2offset, y2, 'm-');
  pause(0);
  if(mod(n, 10) == 0)
      disp(sprintf('%10d/%d done', n, N));
  end
end

% find the index of one of the global minima
[minfval, js] = min(fvals);
j = js(1);

minfval
delta = Xs(j,1)
omega = Xs(j,2)
D = Xs(j,3)
A = Xs(j,4)
B = Xs(j,5)
phi2 = Xs(j,6)
phi = atan(-delta/omega);

y1 = A*exp(-delta*time1) .* cos(omega*time1+phi) + D;
y2 = B*exp(-delta*time2) .* cos(omega*time2+phi2) + D;

figure(2);
clf;
plot(alltime, alldepth, 'b-');
hold on;
plot(time1+time1offset, y1, 'r-');
plot(time2+time2offset, y2, 'r-');
pause(0);

figure(3);
clf;
plot(sort(fvals));

save pc_plot HACK_y1 HACK_y1p HACK_y2 HACK_y2p T10 T1F T20 T2F delta ...
    omega D A B phi phi2 minfval time1 time2 time1offset time2offset ...
    alltime alldepth
```

A.2.2 phasec_cost.m

```
function cost = phasec_cost(X, time1, depth1, time2, depth2)
%PHASEC_COST  Cost function to minimized for phase B
%  The X is the input to the function and the times and depths
%  never change (they are the parameters to this function)

% see below...
global HACK_y1 HACK_y1p HACK_y2 HACK_y2p;

delta = X(1);
omega = X(2);
D = X(3);
A = X(4);
B = X(5);
phi2 = X(6);

phi = atan(-delta/omega);

y1 = A*exp(-delta*time1) .* cos(omega*time1 + phi) + D;
y2 = B*exp(-delta*time2) .* cos(omega*time2 + phi2) + D;

cost = sum((y1 - depth1).^2) + sum((y2 - depth2).^2);

% don't ask don't tell...
HACK_y1 = y1;
HACK_y1p = -A*delta*exp(-delta*time1) .* cos(omega*time1+phi) - ...
    A*exp(-delta*time1) .* sin(omega*time1+phi)*omega;
HACK_y2 = y2;
HACK_y2p = -A*delta*exp(-delta*time2) .* cos(omega*time2+phi2) - ...
    A*exp(-delta*time2) .* sin(omega*time2+phi2)*omega;
```

References

[1] D.L., Donoho, De-noising by soft-thresholding, IEEE trans. on Information theory, 41(3), 613-627, 1995

[2] XP. Zhang, M.D., Desai, Adaptive de-noising based on SURE risk, IEEE signal processing letters, 5(10),265-267, 1998

[3] Wavelet toolbox for user with Matlab manual, Ver.2, 2000

[4] W.E., Boyce, R.C., DiPrima, Elementary Differential Equations and Boundary Value Problems, Sixth Edition., John Wiley & Sons, Inc., 1997