FFIE/711/116

Approved
Kjeller 25 June 1998

Paul Narum

Paul Narum
Director of Research

HEALTH MONITORING OF SHIP HULLS USING WAVELETS, WAVELET PACKETS AND SYSTEM IDENTIFICATION TECHNIQUES

BREMER Amund Solvi

FFI/RAPPORT-98/02764

FORSVARETS FORSKNINGSINSTITUTT
Norwegian Defence Research Establishment
P O Box 25, N-2007 Kjeller, Norway

FORSVARETS FORSKNINGSINSTITUTT (FFI)
Norwegian Defence Research Establishment

P O BOX 25
N-2007 KJELLER, NORWAY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE
(when data entered)

REPORT DOCUMENTATION PAGE

1) PUBL/REPORT NUMBER	2) SECURITY CLASSIFICATION	3) NUMBER OF PAGES
FFI/RAPPORT-98/02764	UNCLASSIFIED	134
1a) PROJECT REFERENCE	2a) DECLASSIFICATION/DOWNGRADING SCHEDULE	
FFIE/711/116		

4) TITLE

HEALTH MONITORING OF SHIP HULLS USING WAVELETS, WAVELET PACKETS AND SYSTEM IDENTIFICATION TECHNIQUES

5) NAMES OF AUTHOR(S) IN FULL (surname first)

BREMER Amund Solvi

6) DISTRIBUTION STATEMENT

Approved for public release. Distribution unlimited.
(Offentlig tilgjengelig)

7) INDEXING TERMS IN ENGLISH:	IN NORWEGIAN:
a) Wavelets	a) Wavelets
b) Wavelet packets	b) Wavelet-pakker
c) System identification	c) Systemidentifisering
d) Subspace identification	d) Underromsidentifisering
e) Structural monitoring	e) Strukturovervåkning

THESAURUS REFERENCE:

8) ABSTRACT

This report was written as a diploma thesis at Forsvarets forskningsinstitutt during 1997 and 1998. The purpose of the work is to examine methods for detecting and characterizing transient structural vibrations in a ship hull.

The theory of wavelets and wavelet packets is reviewed. An efficient implementation of a wavelet-based matched filter for detecting and characterizing transient vibrations is documented.

As a complementary method, the possibility for using system identification methods for identifying dynamical properties of the hull is also investigated. It is shown how a state-space formulation can be a convenient way of representing a vibrating structure. This state-space formulation can be estimated from measured data, and structural parameters such as eigenfrequencies and damping can then be extracted.

9) DATE	AUTHORIZED BY This page only	POSITION
25 June 1998	Paul Narum P Narum	Director of Research

ISBN 82-464-0273-0

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE
(when data entered)

CONTENTS

HEALTH MONITORING OF SHIP HULLS USING WAVELETS, WAVELET PACKETS AND SYSTEM IDENTIFICATION TECHNIQUES

1 INTRODUCTION

1.1 Background

This thesis is concerned with mathematical methods for extracting and identifying non-stationary signals in a noisy background. It is written at the Norwegian Defence Research Establishment (FFI) as a part of the on-going *Composite Hull Embedded Sensor System* (CHESS), which is a joint project between FFI and Naval Research Laboratory (NRL), Washington. The project has as its objective to develop a real-time signal processing system to monitor and analyse the load on the hull of a prototype Surface Effect Ship (SES) which is currently being built for the Royal Norwegian Navy at Kværner Mandal, Norway.

These ships are meant to operate at high speeds, and will therefore be subjected to a variety of different wave-loads. There are several reasons for wanting to monitor and analyse the response of the hull to these loads. Verifying that the prototype behaves in accordance with the design criteria is important before other ships are built. In critical constructions it is also of great value to to measure the load on the system and to detect impending structural flaws. A structural health monitoring system can prevent extensive damage by issuing early warnings, thus avoiding a total breakdown of the structure.

The vibrations induced by the wave-loads have different characteristics. They range from high-frequency local transient vibrations in the panels that constitute the hull, to nearly periodic low-frequency oscillations arising from the approximately periodic loads from the waves. In short, we need to investigate phenomena which are associated with different durations and frequencies.

This behaviour calls for some sort of *local analysis* of the measured time series. As has been noted many times during the last few years, wavelet-based techniques seem especially well suited for such tasks. We also need to understand how the measured data relate to the physical structure of the ship — this can be done using techniques from *system identification.* These two areas are the main topics of this thesis.

1.2 Previous and related work

The wavelet transform is a tool which can be used to investigate signals and functions at different levels of detail. The wavelet transform of a function is commonly said to be a *time–scale* representation. Constructions similar to wavelets have surfaced several times during the present century, often as a tool to characterise singularities and local properties of functions. However, it was not until the late 1970s that time-scale analysis made its final breakthrough, when the geophysicist Jean Morlet found it useful to analyse seismic signals by means of translated and stretched-out versions of a single oscillatory function — what we today call a wavelet. The connection to earlier results was recognised, and the development of the *multiresolution analysis* at the end

2 STRUCTURAL MONITORING AND IDENTIFICATION

This chapter introduces *structural monitoring* and explains its role in structural engineering. We give an overview of the different tasks a *structural health monitoring system* could perform, and motivate its use in the CHESS-project. The proposed sensor system for the CHESS-project is presented, and some of the tasks this system will deal with are illustrated by performing a simple analysis of experimental data. This analysis also illustrates the need for techniques different from traditional Fourier methods, thus providing motivation for much of the rest of the thesis.

2.1 Dynamic behaviour of structures

Assessing the dynamical properties of a proposed structural system is an important part of the design process. The response of the finished structure to the loads it will be subjected to should ideally be modelled and fully known before the construction starts. For example, if one is interested in avoiding resonance between exciting loads and the structure, the structure must be designed in such a way that the *structural eigenfrequencies* are located far away from the frequencies of the dominant loads.

However, the dynamic response of complex structures is in many cases only partially predictable from models and computer simulations. One of the principal reasons for this is that the modelling is frequently based on too strict assumptions. This is typically the case concerning internal and external damping, and possibly the nature of the excitations as well.

Consequently, it is important to be able to measure and characterise the dynamical properties of a system empirically. A sensor system which performs such a *structural health monitoring* can serve a wide range of other tasks as well, as illustrated by the following list.

- *Load Monitoring*
  Monitoring the loads on a structure could reveal whether it is being subjected to unsustainable loads. If the monitoring is performed in real-time, it could be coupled with an alarm system. Depending on the type of structure, an operator may be alerted and take appropriate action.

- *Verification*
  Investigating whether an actual system behaves in accordance with the design criteria is important. It could reveal whether the structure possesses significant unmodelled dynamics. That could help identify processes that need to be accommodated for in the model, and further basic understanding as to what the most important dynamics are. Weaknesses in prototype design could be detected and then corrected in later models.

- *Damage detection*
  Closely related to verification is damage detection. If incipient structural flaws are detected at an early stage, substantial damage arising from a total breakdown might be avoided. Knowledge of when and where structural damage arises

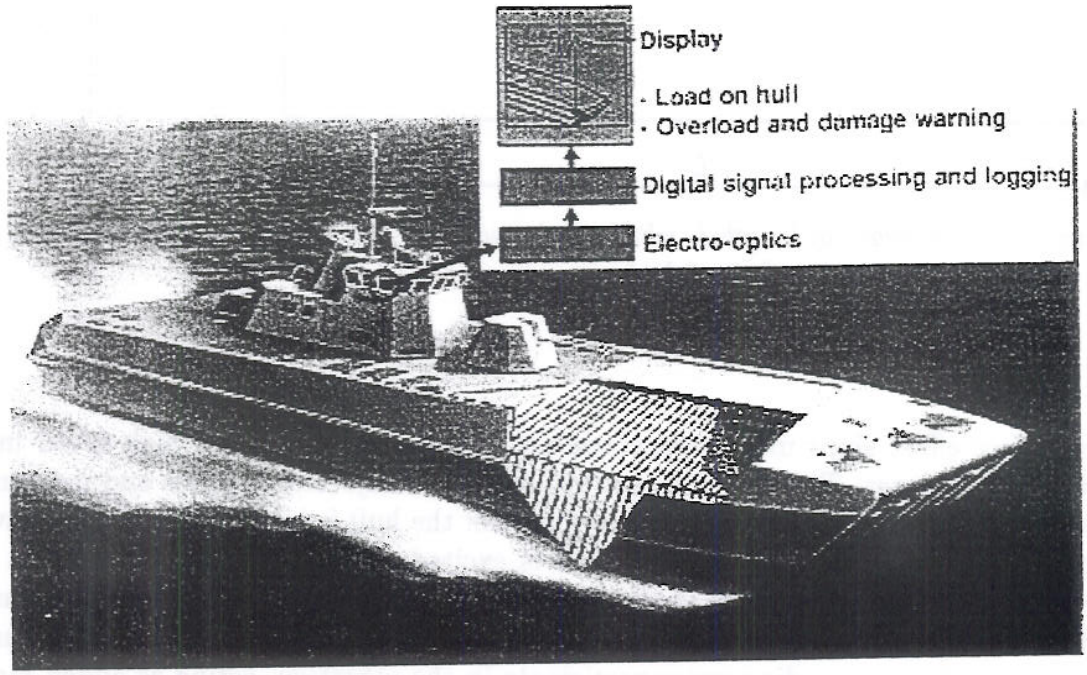


Figure 2.1 *A surface effect ship with its fibre-optic structure monitoring sensor system partially exposed. The inset pictorially represents the flow of signals from detection to presentation.*

is also important when performing preventive maintenance. Several methods for detecting damage exist. They are in general based on detecting changes in certain characteristic structural properties, e.g. in vibrational modes or electrical conductivity, [5].

- *Logging*
  Creating a database which stores the loads and structural behaviour is useful for the purposes mentioned above. The database should be *temporal*, reflecting possible time-development of the loads and structural characteristics.

All these areas are relevant to the CHESS-project.

2.2 The CHESS-project

The CHESS-project is a collaboration between Forsvarets Forskninginstitutt (FFI) and Naval Research Laboratory (NRL) in Washington. The aim is to develop a real-time sensor system which is going to be installed on the hull of a prototype of a new breed of surface effect ships commissioned by The Royal Norwegian Navy, pictured in Figure 2.1. The actual implementation of the proposed sensor system is not truly relevant to this report as the methods we will discuss are of a general nature. However, it will be easier to get an intuitive appreciation of the topics covered if they are related directly to applications. This warrants a brief description of some key areas in the CHESS-project and the physical processes behind the signals which are analysed.

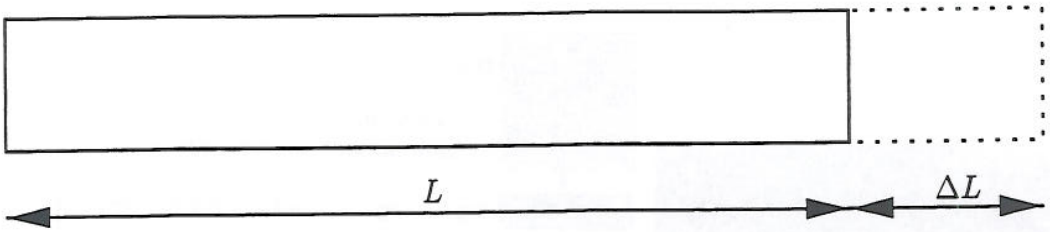


Figure 2.2 *A beam of length L which is elongated to a length $L + \Delta L$ is subjected to a strain of $\epsilon = \Delta L/L$.*

2.2.1 Wave-hull interaction

The surface effect ships mentioned above are designed to operate at high speed in a variety of sea-states. The ship will then regularly be subjected to forces and moments induced by impacts of the waves. When a part of the hull is hit by an incoming wave, a local vibration in the plates of the hull is excited, as well as global vibrations in the whole ship. The *modes* of these vibrations are determined mainly by *structural parameters* such as stiffness, density, geometry and internal damping of the structure. In addition, the surrounding water participate in the vibration, acting as added mass and contributing to external damping.

The parameters which depend on the amount of water present will clearly vary with different impacts — this may result in variations in the vibrational modes. A reliable damage detection system based on changes in vibrational modes must therefore be able to distinguish between this natural variation of the modes and changes induced by material flaws.

One of the reasons for the interest in load monitoring and damage detection in conjunction with these SES is that the hull will mainly be constructed from composite materials such as glass fibre reinforced polyester. These materials have until recently not been much used in ship construction, and consequently, there is a great deal of uncertainty with regard to their performance and durability.

A damage which can result from "wear and tear" is *delamination*, when the hard, protective fibre-reinforced skins detach and separate from each other or the foam core. This change in structure changes the structural parameters that determine the vibrational modes of the plate. There is thus reason to believe that a change in the vibrational frequencies might indicate structural damage. This effect has also been observed experimentally, where it is noted that the changes in frequency are most pronounced for higher order modes. This is the background for our interest in detecting transients and especially higher order harmonics.

2.2.2 Fibre optic strain sensors

The CHESS-project has developed sensors that locally measure relative elongation, or *strain*, in the hull. In one dimension, the strain ϵ of a beam of length L which is stretched to a length $L + \Delta L$ is defined as $\epsilon = \Delta L/L$, as shown in Figure 2.2. The sensors we will use for measuring the strain at various locations on the hull are all

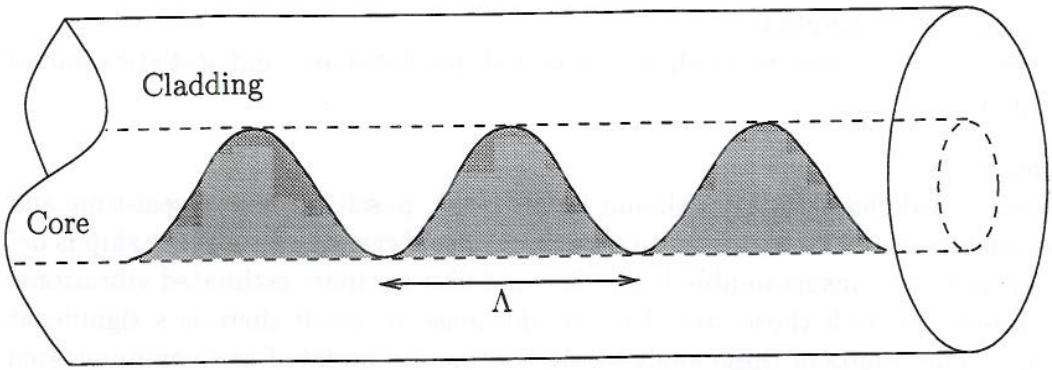


Figure 2.3 Schematic illustration of a fibre optic Bragg grating sensor. A cosine-modulated refractive index is written onto the light-transmitting core.

fibre optic Bragg grating sensors. The fibres used have a *grating* written onto the light transmitting core, as indicated in Figure 2.3. By a grating we here mean a periodic modulation of the index of refraction. When broadband light is incident on this grating, narrow-band light will be reflected by the grating. The *Bragg-condition* states that [29] the reflected light will have a wavelength λ_B given by

$$\lambda_B = 2n\Lambda, \tag{2.1}$$

where Λ is the period of the grating and n is the refractive index of the material. When the fibre is stretched or compressed, the period Λ will vary as a function of time. The relative elongation of the fibre $\epsilon(t)$ at an instant t can then be calculated according to

$$\frac{\Delta\lambda_B}{\lambda_B} = \alpha\epsilon(t) \tag{2.2}$$

where $\Delta\lambda_B$ is the change in the reflected wavelength, and α is a constant of proportionality. It is not not equal to unity as one might think; the reason for this is that some additional optical effects not included in (2.1) must be accommodated for. We will not go into further details concerning the physics of the sensor system.

2.2.3 The proposed signal processing system

With regard to the discussion in the previous sections, we may formulate the main tasks of this signal processing system.

- *Detection*
  The system should detect the different transient vibrations in the hull. For the strongest transients, this is relatively easy. The hypothesised *superharmonic vibrations* are expected to be very weak, so care must be taken when designing the detection routines.
- *Characterisation*
  The most important parameter to estimate is the initial amplitude (strain) of the vibration. In order to characterise the transients more fully, their frequencies and damping should also be estimated. We are also interested in characterising the interrelationship between the signals from the different sensors, such as the cross-correlation.

- *Maintaining the database*
  The resultant characteristics should be stored in a database, and statistics should be updated.

- *Analysis*
  The identified characteristics should be analysed, possibly both in real-time and after each cruise. The real-time analysis identifies if certain areas of the ship is being subjected to unsustainable loads. It could also compare estimated vibrational characteristics with those stored in the database to see if there is s significant change. The results of these analyses are continually updated and communicated to the operator of the ship.

  There is no need for a real-time analysis when performing the post-mission analysis, and this could allow for a more careful analysis of the data.

The specifications of the signal processing system are not yet finalised, so aspects different than those presented here may well come into consideration.

2.2.4 Experimental data

A prototype of the sensor system outlined above has already been tested on a full-scale ship, and several hours of data are available for analysis. The dynamic response of the glass-fibre material used in the hull has been tested at MARINTEK, Trondheim. In this report, we concentrate on analysing the data obtained from the ship.

In Figure 2.4 we show the orientation of the strain sensors relative to the ship. The sensor marked L measures *longitudinal* strain in the hull, whereas the sensors marked T1, T2 and T3 measure *transversal* strain in the panel. In this report we will exclusively consider data obtained from L and T2.

A complex structure such as a ship will in general have many different vibrational modes, and these may be spread over a wide frequency range. Some eigenfrequencies may be located close together, others are perhaps rarely excited. All are probably masked by noise. The end result is that robust estimation of structural parameters is generally considered a great challenge. However, some insight into the dynamical behaviour of our ship can be obtained by relatively simple methods. In Figure 2.5 we show estimated spectral density for 55 minutes data obtained from sensors L and T2. The spikes in the plot may indicate structural eigenfrequencies.

Several different phenomena can be identified from these data. The most common *transient component* is a damped vibration of about 2 Hz corresponding to a *long-ship* vibrational mode. This is a global vibration where the whole hull takes part. An example of these characteristic vibrations is shown in Figure 2.6 where several transients of approximately 2 Hz can be seen. The data shown is atypical in that transients normally do not come that frequently.

A somewhat rarer vibration is a vibration around 17 Hz which corresponds to a local vibration of a panel in the hull, probably excited by a direct wave impact.

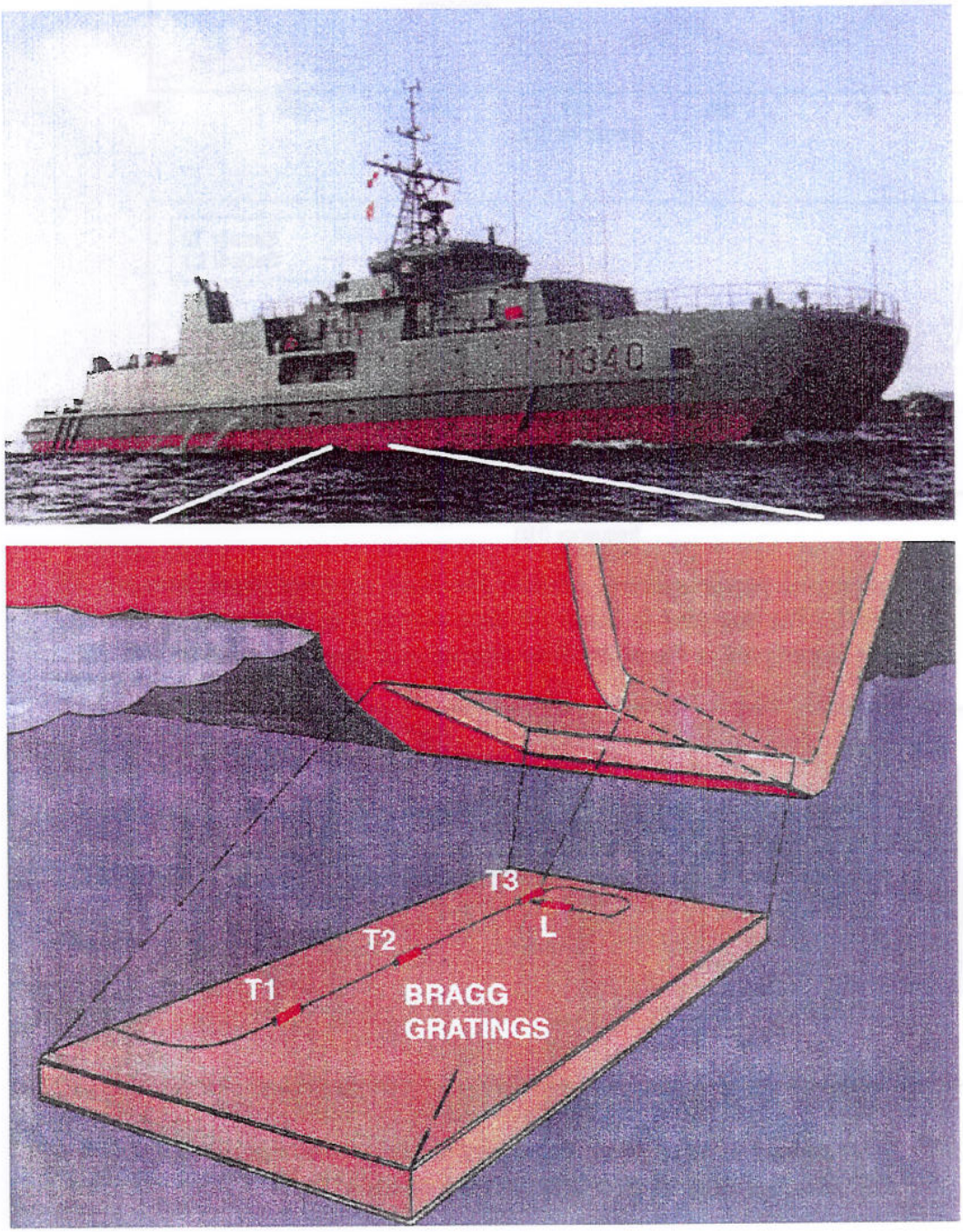


Figure 2.4 The orientation of the strain sensors used in the experiments.

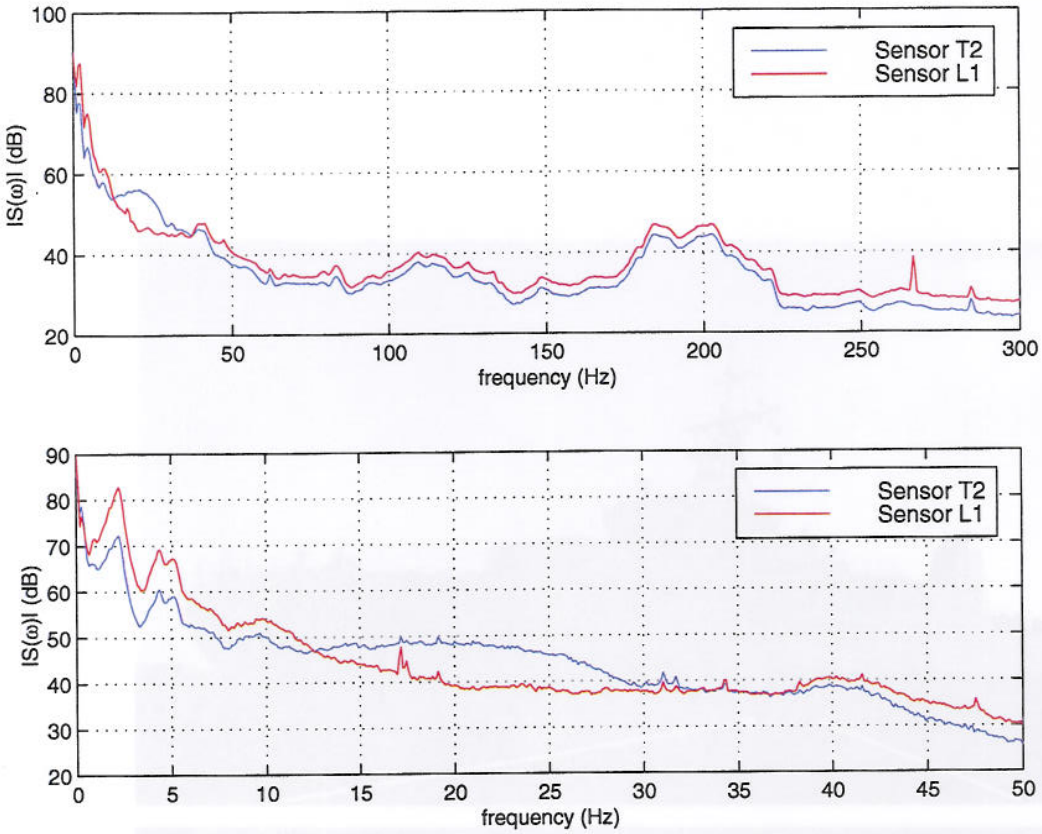


Figure 2.5 *Estimated power spectral density for data obtained from sensors T2 and L. The top plot is based on data sampled at frequency of* 600 *Hz. The lower plot is based on data filtered and downsampled to* 100 *Hz.*

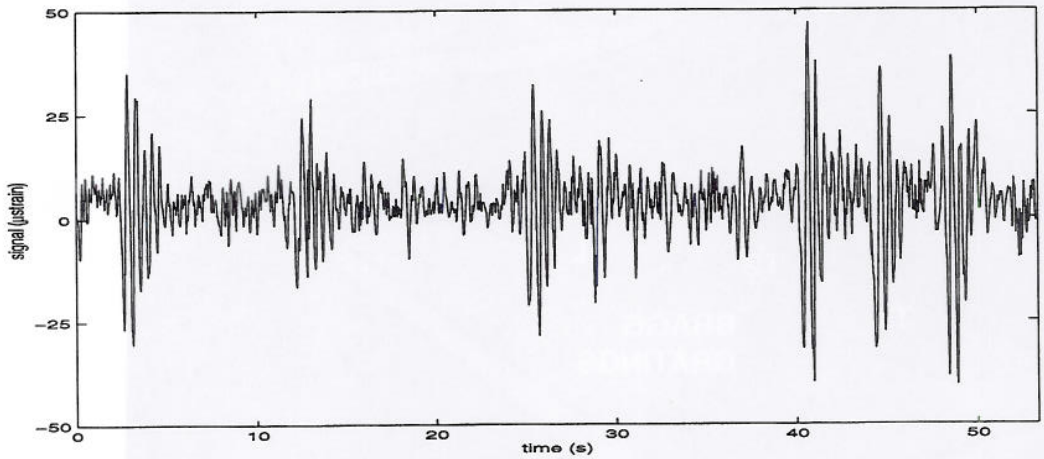


Figure 2.6 *A* 50*-second excerpt from the experimental strain measurements on a ship showing several strong transients of about* 2 *Hz.*

Another excerpt corresponding to much calmer sea is shown in Figure 2.7. The nearly periodic strain loads resulting from impacts with waves can clearly be seen. From the peaks in the estimated power spectral density plots we can conclude that there are components in data with frequencies around 2 Hz and 17 Hz, as well as very strong components in the range 0-1 Hz. However, it is not clear *where* the 2 Hz and 17 Hz vibrations are located in time. Since these vibrations are characteristics of the hull, it can be argued that vibrations around these frequencies are present whenever the

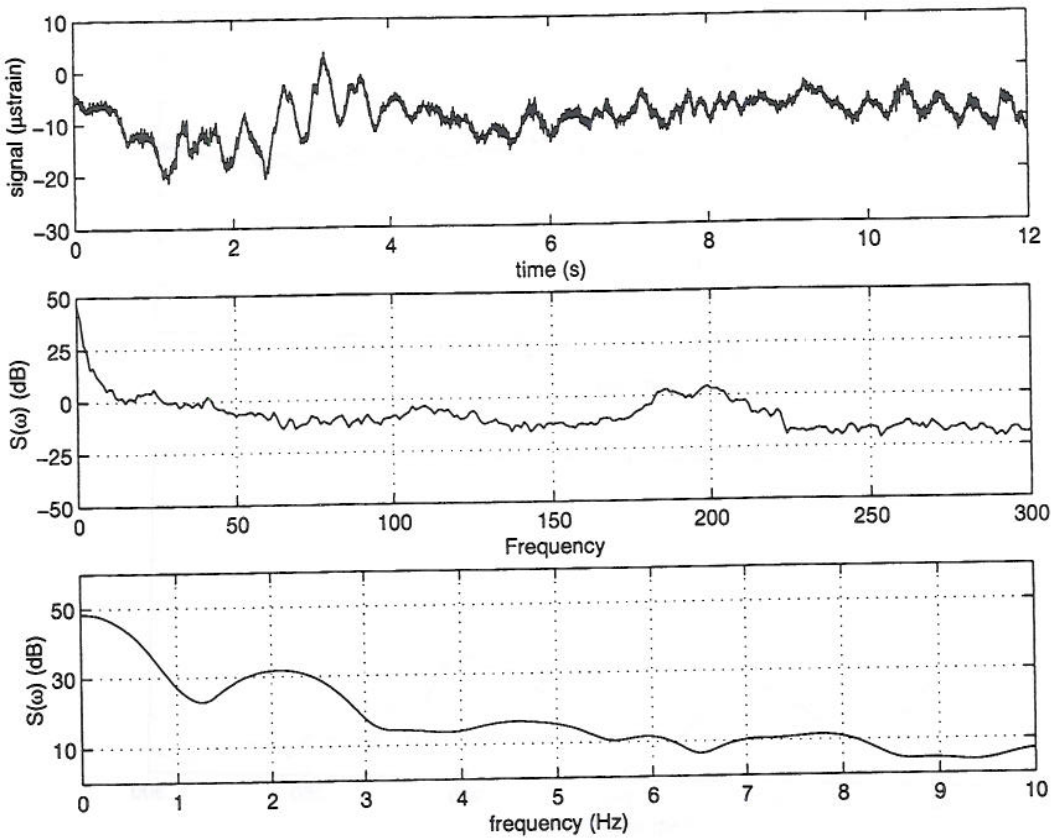


Figure 2.7 *A twelve-second excerpt from the experimental strain measurements on a ship (top), and the estimated spectral density function (middle and bottom).*

hull is subjected to a dynamically varying load [23], but they need not stand clearly out from the background as in Figure 2.6. We say that these vibrational modes are *poorly excited* in Figure 2.7, whereas the 2 Hz vibration is *strongly excited* in Figure 2.6.

It has been hypothezied that the 17 Hz component is also associated with higher harmonics. The small peak around 40 Hz might represent these hypothetical higher harmonics, or it may represent another mode in the ship. The peak around 200 Hz has been identified as belonging to vibrations caused by the hydraulic water pump in the propulsion machinery.

The plot in Figure 2.8 shows a close-up of a strongly excited transient, along with its estimated *spectral density* function, see [27] for details. We see a characteristic top in the spectral density plot at 20 Hz, corresponding to a local panel vibration. The small ripples in the time domain plot of Figure 2.8 are the 200 Hz vibration caused by the water pump mentioned above.

The relative magnitude of the spectral density plot gives us an indication of the energy content in the different frequency, or *spectral*, components of the signal. Knowing that energy is proportional to amplitude squared [28], we can give a rough estimate of the amplitude of these vibrations. In Table 2.1 we show the relative energy and amplitude of different frequency components, as obtained from Figure 2.8. The estimated amplitudes were calculated assuming that the 20 Hz transient has an "average" amplitude of

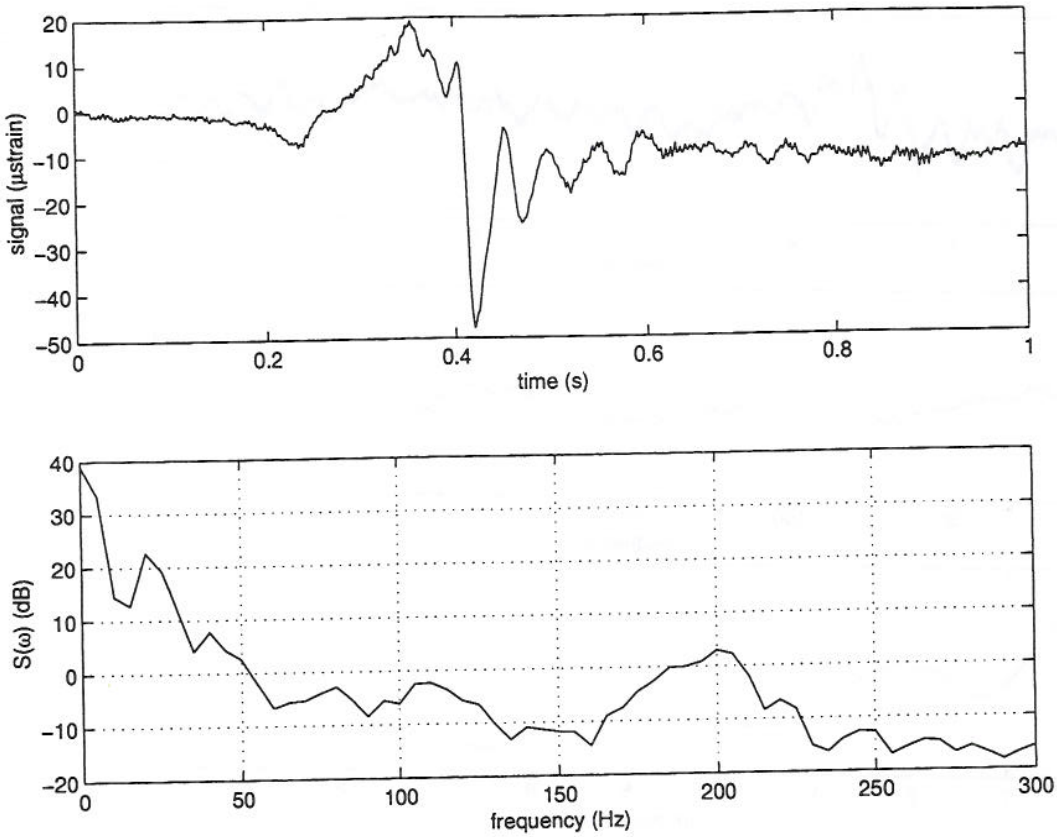


Figure 2.8 *A one-second excerpt from the experimental data showing a* 20 Hz *transient (top) and its estimated spectral density function (below). The spectral density function was estimated by using a Hanning window of approximately the same size as the transient itself.*

Table 2.1 *The table shows the energy and amplitude ratio (per cycle) between peaks in Figure 2.8 and the energy and amplitude of the peak at* 20 *Hz. The estimated amplitudes are calculated from the amplitude ratios assuming that the amplitude of the* 20 *Hz component is of the order of magnitude* 10 *μstrain.*

Peak at	Energy per cycle relative to the energy per cycle at 20 Hz (dB)	Amplitude relative to the amplitude at 20 Hz	Estimated amplitude (μstrain)
20 Hz	0	1	10
40 Hz	-15	0.2	2
200 Hz	-20	0.1	1

the order-of-magnitude 10 μstrain. This coarse estimate was found by visual inspection of the plot.

We see that the estimated amplitude of the 200 Hz components is about 1 μstrain, and this is consistent with the amplitude of the small ripples in Figure 2.8 and earlier findings [30]. The amplitude of the hypothetical harmonic component at 40 Hz should then be about 20% of the fundamental vibration, or about 1–2 μstrain.

Several objections can be raised to these back-of-the-envelope calculations. Some can be overcome by a more careful analysis, and some are of more fundamental nature. Indeed, one may question the appropriateness of using these Fourier methods since our transients are by nature time-localised. This was not a great problem for this signal since we had analysed only a one-second excerpt dominated by a transient. In practice, we will need to analyse long signals with a relative sparse density of transients, and then we will be interested in the *time-localisation* of the transients as well. In the following chapters, we will investigate alternative and complementary techniques.

3 THEORY OF WAVELETS AND WAVEPACKETS

In this chapter we present the basic theory behind wavelet and wavelet packet transforms. We show how wavelet analysis is a part of the same framework as many other *transform techniques* by using the theory of (generalized) frames which is introduced in Section 3.2; We demonstrate how signal processing techniques commonly performed in the time or frequency (Fourier) plane can be equivalently performed in any general transform domain. For our application we are especially interested in doing filtering operations in the *time-scale* plane of the wavelet transform. This is particularly interesting for us, because we may then restrict filtering operations for detection and characterization of transients to those characteristic scales where they occur.

3.1 Wavelet transforms - a first look

This introductory section gives a short overview of the wavelet transform, before it is reintroduced in Section 3.3.1 using the theory of multiresolution. In order to further an intuitive interpretation of this transform, we first give an example of the *short time Fourier transform* (STFT) and the *Gabor transform*, which are similar in spirit, but perhaps easier to understand and interpret. The Gabor transform has also been an important tool for investigating data in the CHESS-project.

3.1.1 The short time Fourier transform

We would like to represent a signal by a set of basis functions which facilitates separation of its spectral content at different times. As discussed in [14], [8] and many others, the Fourier transform is not suited for this because its sinusoidal basis functions are not localized in time, stretching over the entire real line. A natural attempt is then to find a set of basis functions which are simultaneously well localized in both time and frequency. Intuitively, we might try as candidates sinusoidal waves multiplied with a *window function* which serves to localize the sinusoids in time. This approach can lead to the well-known *short-time Fourier transform*, which decomposes functions as linear combinations of time-localized sinusoids;

Definition 1 (Short time Fourier transform) *We define the* short time Fourier transform *(STFT) of a function* $f(t) \in L^2(\mathbb{R})$ *corresponding to the* window function $w(t) \in L^2(\mathbb{R})$ *as*

$$\left(\mathrm{T}^{STF} f\right)(\nu, \tau) := \int dt\, f(t) e^{-2\pi i \nu t} \overline{w(t-\tau)} \tag{3.1}$$

The STFT corresponding to a Gaussian window function is called a Gabor transform. □

The Fourier transform is reviewed in Appendix A.2. An illustration of the Gabor transform is shown in Figure 3.1, computed using the discrete Fourier transform. At the top a function consisting of two sinusoids is plotted together with the Gaussian window (dotted) used to obtain the Gabor transform of f, the magnitude of which is plotted

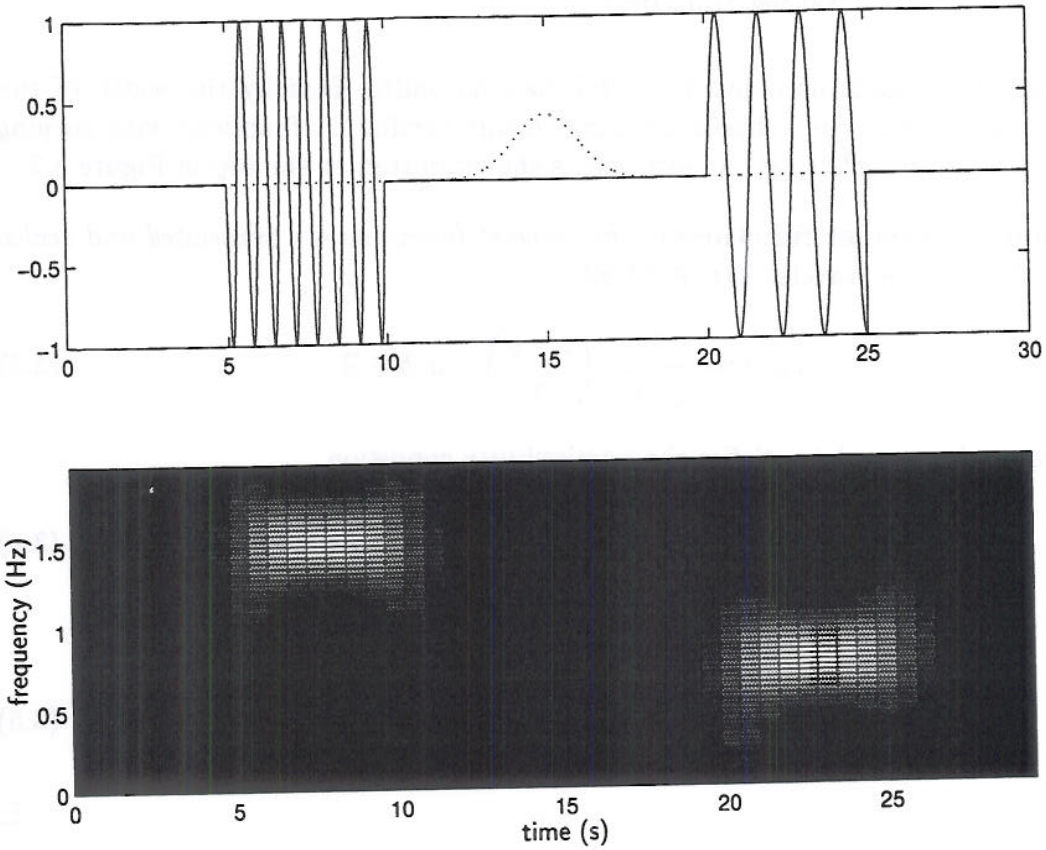


Figure 3.1 *A function consisting of two localized sinusoids of different frequency (top), and a the squared magnitude of its Gabor transform bottom. The Gaussian window used is shown dotted at the top.*

at the bottom. We recognize correctly that $f(t)$ has a 1.5 Hz harmonic component in the interval [5 s, 10 s], and a 0.75 Hz component in the interval [20 s, 25 s]. For more details on Short-time Fourier methods and its similarities to wavelet analysis, we refer to [14]. There it is proved that the f can be recovered by

$$f = \|w\|_{L^2(\mathbb{R})}^{-2} \iint d\nu d\tau \left(T^{STF} f\right)(\nu, \tau) \overline{w(\nu, \tau) e^{-2\pi i \nu \tau}}. \tag{3.2}$$

By examining this formula or Figure 3.1, it should be clear that representing a function by its STFT is *redundant*. By that we mean that we do not in general need to know $\left(\mathrm{T}^{STF} f\right)(\nu, \tau)$ for all ν and τ in order to recover f. See [8] or [14] for a fuller discussion of these considerations.

The STFT is useful in many applications, and it has been used in the CHESS-project as a tool for preliminary analyzes. However, there are several objections to this technique. One of the principal objection is that the width of the window function w is static. In many applications one would be interested in having a more flexible window when analyzing signals that have components of different time duration.

The wavelet transform, in addition to its many other advantages, has such a built-in *zooming capability*. In the following, we concentrate on wavelet transforms.

3.1.2 The continuous wavelet transform

The localized sinusoids used by the STFT had its width *fixed* by the width of the window. Wavelet analysis is based on using certain oscillatory functions with *varying* width. These are *wavelets* and an example is shown dotted at the top in Figure 3.2.

Definition 2 (Wavelet functions) *The wavelet functions are translated and scaled versions of a* mother wavelet $\psi(t) \in L^2(\mathbb{R})$*;*

$$\psi_{a,b}(t) := \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right) \quad a, b \in \mathbb{R}, \tag{3.3}$$

where the mother wavelet satisfies the admissibility condition

$$C_\psi := \int d\nu |\nu|^{-1} |\hat{\psi}(\nu)|^2 < \infty. \tag{3.4}$$

If we let $a := a_0^{-j}$*, and* $b := a_0 b_0 k$ *with fixed* a_0 *and* b_0*, we write*

$$\psi_{j,k}(t) := \frac{1}{\sqrt{|a_0|}}\psi\left(a_0^{-j}t - kb_0\right), \quad j, k \in \mathbb{Z}. \tag{3.5}$$

□

Except for this section, we will mostly consider discrete *dyadic wavelet expansions* where $a_0 = 2$. By rescaling, we may without loss of generality assume that $b_0 = 1$. Also, we will in practice impose far more restrictive conditions on our wavelets than those in Definition 2. We now define the continuous wavelet transform, and find that it is invertible.

Definition 3 (Continuous wavelet transform) *Let* $\psi(t) \in L^2(\mathbb{R})$ *be a mother wavelet. The continuous wavelet transform* T^{CW} *of a function* $f(t) \in L^2(\mathbb{R}; dx)$ *is defined by*

$$\left(\mathrm{T}^{CW} f\right)(a, b) := \int dt\, f(t)\overline{\psi_{a,b}(t)}, \qquad (a, b) \in (\mathbb{R}^+ \times \mathbb{R}). \tag{3.6}$$

□

We note for future reference that we may write the wavelet transform as

$$\left(\mathrm{T}^{CW} f\right)(a, b) = \langle f, \psi_{j,k}\rangle(a, b) \qquad (a, b) \in (\mathbb{R}^+ \times \mathbb{R}). \tag{3.7}$$

In Figure 3.2 we show a plot of the magnitude of the continuous wavelet transform (CWT) of the same function we analyzed with the Gabor transform above. The *analyzing* wavelet used is shown dotted at the top. The CWT is really a *correlation* between the function $f(t)$ and the function $\psi_{a,b}(t)$. This means that the interpretation of the CWT is straightforward; for a given scale and translation (a, b), the magnitude of $\left(\mathrm{T}^{CW} f\right)(a, b)$ is *the degree of similarity* between $f(t)$ and $\psi_{a,b}(t)$ in the area around $t = b$. For large a, the analyzing wavelet will be very stretched out and consequently correlate mainly with the low-frequency parts of the analyzed function. At small scales, the wavelet will be narrow and correlate mainly with high-frequency components in the

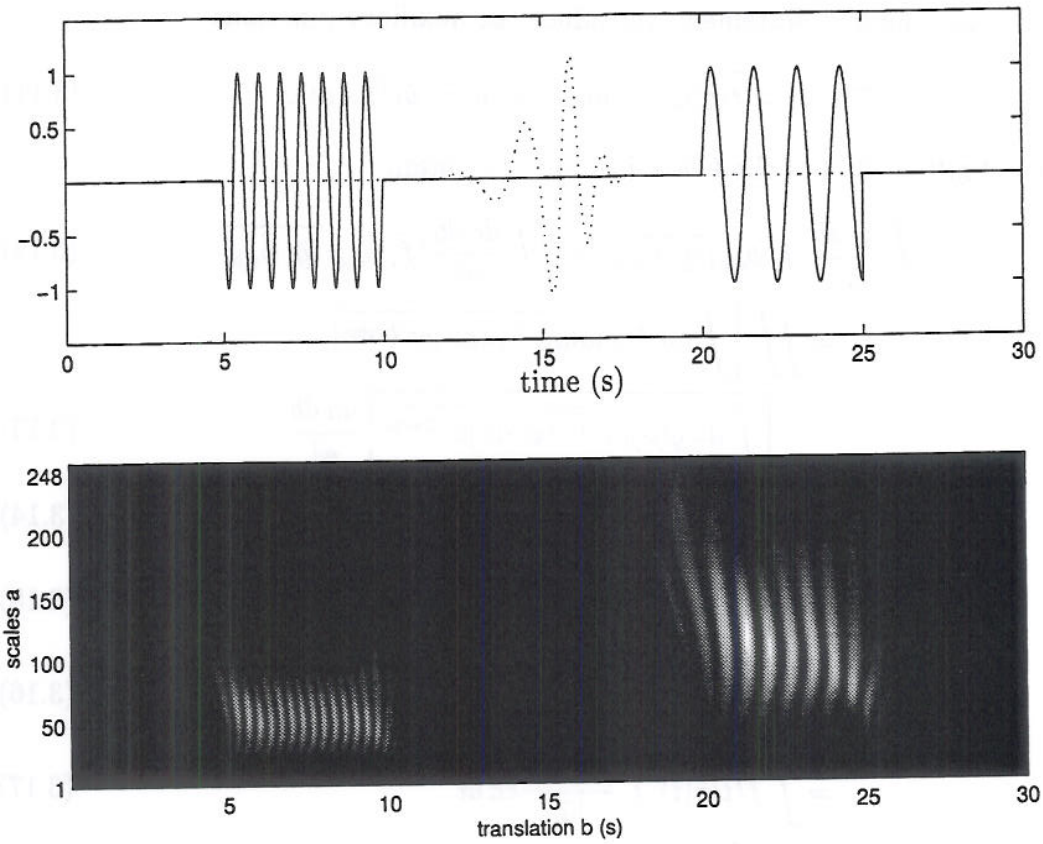


Figure 3.2 *A signal consisting of two localized sinusoids of different frequency and a plot of the squared magnitude of its Continuous wavelet transform.*

analyzed function. These interpretations are illustrated in Figure 3.2, where the translational parameter b gives the time localization of the different components of $f(t)$. We also see that the high frequency components are to be found at low scales and vice versa, as expected.

For future reference, we note the following lemma;

Lemma 1 *The Fourier transform of a wavelet $\psi_{a,b}(t)$ is given by*

$$\hat{\psi}_{a,b}(t) = \sqrt{|a|}e^{-2\pi i\nu b}\hat{\psi}(a\nu). \tag{3.8}$$

□

We now prove that this transform has an inverse, at least in a 'weak' sense;

Theorem 2 (The continuous wavelet transform) *Any two functions $f, g \in L^2(\mathbb{R})$ and their continuous wavelet transforms satisfy, for all $g \in L^2(\mathbb{R})$*

$$\iint \frac{da\,db}{a^2}(\mathrm{T}^{CW}f)(a,b)\overline{(\mathrm{T}^{CW}g)(a,b)} = C_\psi \int dt\, f(t)\overline{g(t)}. \tag{3.9}$$

Furthermore, assume that any $f \in L^2(\mathbb{R})$ satisfy

$$f(t) = C_\psi^{-1} \iint \frac{da\,db}{a^2}(\mathrm{T}^{CW}f)(a,b)\psi_{a,b}(t) \tag{3.10}$$

at all $t \in \mathbb{R}$ where f is continuous and bounded. □

Proof To prove the first statement, introduce the auxiliary functions

$$F(t;a) = \overline{\hat{f}(t)}\hat{\psi}(at) \text{ and } G(t;a) = \overline{\hat{g}(t)}\hat{\psi}(at). \tag{3.11}$$

Using the definition (3.7) of the CWT in (3.9), we obtain

$$\iint \frac{da\,db}{a^2}\langle f, \psi_{a,b}\rangle\overline{\langle g, \psi_{a,b}\rangle} = \iint \frac{da\,db}{a^2}\langle \hat{f}, \hat{\psi}_{a,b}\rangle\overline{\langle \hat{g}, \hat{\psi}_{a,b}\rangle} \tag{3.12}$$

$$= \iint \left[\int d\nu\, \hat{f}(\nu)\overline{a|a|^{-\frac{1}{2}}\hat{\psi}(a\nu)e^{-2\pi ib\nu}}\right] \overline{\left[\int d\tilde{\nu}\, \hat{g}(\tilde{\nu})\overline{a|a|^{-\frac{1}{2}}\hat{\psi}(a\tilde{\nu})e^{-2\pi ib\tilde{\nu}}}\right]}\frac{da\,db}{a^2} \tag{3.13}$$

$$= \iint \overline{\left[\hat{F}(b;a)\right]}\left[\hat{G}(b;a)\right]\frac{db\,da}{|a|} \tag{3.14}$$

$$= \iint \overline{F(t;a)}G(t;a)\frac{dt\,da}{|a|} \tag{3.15}$$

$$= \int \hat{f}(t)\overline{\hat{g}(t)}\int \frac{|\hat{\psi}(at)|^2}{|a|}da\,dt \tag{3.16}$$

$$= \int \hat{f}(t)\overline{\hat{g}(t)}\int \frac{|\hat{\psi}(\tilde{a})|^2}{|\tilde{a}|}d\tilde{a}\,dt \tag{3.17}$$

$$= C_\psi \int f\overline{g}\,dt. \tag{3.18}$$

We have used Parseval's Theorem to get equations (3.12), (3.15) and (3.18). We changed the order of integration in going to (3.17), which is justified by Fubini's Theorem. The second statement follows formally by letting g be a Dirac impulse in 3.10, for more details, see [8]. ■

The proof is typical for many of the Theorems in wavelet theory. In the following, most proofs will be found in the Appendices.

While the continuous wavelet transform has many interesting applications in functional analysis, it seems less suited to digital signal processing. This is primarily because there exist very efficient algorithms for computing the inner product (3.7) when the scale and translation parameters are discretisized as described in Definition 2. Discussing these wavelets, $\psi_{j,k}$, will be the topic of the rest of this chapter.

3.1.3 The discrete wavelet transform

From the discussion in the previous chapter, it should be clear that the wavelets play the role of a basis-like function. In particular, the continuous wavelet transform bears a striking resemblance to the Fourier transform; both can be written as an inner product between a function and an *analyzing function* - $\psi_{a,b}$ in the wavelet case or $e^{2\pi i\nu t}$ in the Fourier setting, although in the latter case problems arise because $e^{2\pi i\nu t} \notin L^2(\mathbb{R})$. For practical applications, we work with sampled data and use the discrete Fourier transform instead of its continuous counterpart.

In order to use wavelet theory for digital signal processing, we face several problems. First of all, our data are discrete, so we do not in general have a function $f(t)$ to analyze, but rather a sequence of numbers. That means that we cannot numerically compute the inner products in (3.7) correctly. Even if we theoretically could do this, it would probably be quite impractical. To calculate the inner products for continuously varying parameters a and b would be very computationally expensive.

It turns out that it is sufficient to calculate the inner products (3.7) at discrete values of a and b only, in effect *sampling* the CWT. This approach will also offer a natural solution to the first problem mentioned; that we only know the data at sampled values.

Clearly, with a and b restricted to discrete values, we cannot expect (3.9) to hold. It seems reasonable that its discrete counterpart would be something similar to

$$f(t) \sim \sum_j \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k}(t), \tag{3.19}$$

and indeed that is the case. First we need to look at the details of how a superposition of wavelets span a space.

3.2 Frames

Frames generalize the familiar concept of a basis, and allow for a unifying formalism which can cover many transform-operations. Frames provide theory for representing a signal in a *redundant* way, and this can in many instances be useful. The window modulated sinusoids used in the STFT are an example of a frame. A Gabor transform of a function is a redundant representation. The advantage of this representation is that one may get a clearer understanding of the constituent parts of the signal, as illustrated by Figure 3.1. Another reason why frames are interesting in conjunction with signal processing is that they in a sense are robust with respect to random errors, for example caused by quantization. The standard example of this is CD-technology, where the signal is band limited to about 20 kHz, thus requiring a minimum sampling frequency of about 40 kHz. In practice, the signal is *oversampled* at a higher rate than this minimum rate prescribed by the Shannon sampling Theorem.

It has been felt that the theory of frames is potentially useful for many applications at FFI, and it is therefore treated in detail here.

3.2.1 General properties of frames

When we speak of frames, we refer to generalized frames as in [14], where a brief introduction to the necessary measure theory is also given. A short note on this topic can also be found in Appendix A. The precise definition of a frame is as follows.

Definition 4 (Frames) *Let H be a Hilbert space and let M be a measure space with measure μ. A frame in H indexed by M is a family of vectors $\{h_m \in H : m \in M\}$ such that*

1. *For every $f \in H$, the function $\tilde{f} : M \to C$ defined by*

$$\tilde{f}(m) = \langle f, h_m \rangle \tag{3.20}$$

is measurable.

2. *There is a pair of constants $0 < A \leq B < \infty$ such that for every $f \in H$,*

$$A\|f\|_H^2 \leq \|\tilde{f}\|_{L^2(\mu)}^2 \leq B\|f\|_H^2 \tag{3.21}$$

We call (3.21) the frame condition, *and the constants A and B the* frame bounds. *The operator $\mathrm{T} : H \to L^2(\mu)$ defined by*

$$f \mapsto \langle f, h_m \rangle \tag{3.22}$$

is called the frame operator *or* the transform *of f with respect to the frame.* □

We can make a large class of transforms, both continuous and discrete, fit into the theory of frames. We will now give some properties of frames, and investigate whether we can find an inverse transform, in some sense, allowing us to recover the original function from its transform (3.22).

Theorem 3 (Properties of frames) *Assuming the notation from Definition 4 the following is valid for any frame.*

1. *The operator* T *has a unique adjoint operator* T^*.

2. *The action of* T^* *can be written*

$$T^*g = \int_M d\mu(m)\, h_m g(m). \tag{3.23}$$

3. *There exists bounded, hermitian operator* $\mathrm{G} := \mathrm{T}^*\mathrm{T}$ *called* the metric operator *satisfying the operator inequality*

$$0 < A\mathrm{Id} \leq \mathrm{G} \leq B\mathrm{Id} < \infty. \tag{3.24}$$

where Id *is the identity operator on H.*

4. *The metric operator has a well-defined bounded inverse which satisfies the operator inequality*

$$0 < B^{-1}\mathrm{Id} \leq \mathrm{G}^{-1} \leq A^{-1}\mathrm{Id} < \infty. \tag{3.25}$$

5. *The action of* G *can be written*

$$f \mapsto \int_M d\mu(m)\, \langle f, h_m \rangle_H h_m. \tag{3.26}$$

□

The proofs of these properties are instructive, and they can be found in Appendix B.1. The question of inverting a transform with respect to some frame will be seen to involve the *dual frame*;

Definition 5 (Dual frame) *Let H be a Hilbert space, and let the family $\mathcal{H}_M = \{h_m \in H | m \in M\}$ be a frame. The* dual frame *of $\mathcal{H}_M$ is the set of vectors*

$$\mathcal{H}^M = \{h^m | m \in M\}, \tag{3.27}$$

where the dual vectors of h_m are defined as

$$h^m := \overline{\mathrm{G}^{-1} h_m}. \tag{3.28}$$

□

We are now in a position to state a second Theorem which summarizes relations between functions and elements in the space of the transformed functions as defined in (3.22). This Theorem also presents a general formula for the inverse of a transform.

Theorem 4 (Properties of frames 2) *Let $\mathcal{H}_M$ be a frame in the Hilbert space H, and let $\mathcal{H}^M$ be the corresponding dual frame. Furthermore, let $f \in H$, $g \in L^2(\mu)$ and denote the transform of f with respect to the frame $\mathcal{H}_M$ by $\tilde{f}(m) := (\mathrm{T}f)(m) \equiv \langle f, h_m \rangle_H$. Then,*

1. *The operator $\mathrm{S} : L^2(\mu) \to H$ defined by*

$$\mathrm{S} := \mathrm{G}^{-1}\mathrm{T}^* \tag{3.29}$$

is a left inverse of T, and is called the synthesizing operator.

2. *Its action is given by*

$$(\mathrm{S}g)(t) = \int_M d\mu(m)\, g(m) \overline{h^m(t)}. \tag{3.30}$$

3. *In the weak sense, f may be recovered from its transform $\tilde{f}$ as follows*

$$f = \langle \tilde{f}, h^m \rangle_{L^2(\mu)}, \tag{3.31}$$

where $h^m \in \mathcal{H}^M$ are the dual frame vectors of $\mathcal{H}_M$.

4. *The operator $\mathrm{P} : L^2(\mu) \to ran\ \mathrm{T} \subset L^2(\mu)$ defined by*

$$\mathrm{P} := \mathrm{TS} \tag{3.32}$$

is the orthogonal projection from $L^2(\mu)$ to the range of T in $L^2(\mu)$

5. *Its action is given by*

$$(Pg)(m) = \int_M d\mu(m')\, K(m, m') g(m'), \tag{3.33}$$

where the reproducing kernel *$K(m, m')$ of the range of T and the frame $\mathcal{H}$ is defined as*

$$K(m, m') := \langle h^m, h_m \rangle \tag{3.34}$$

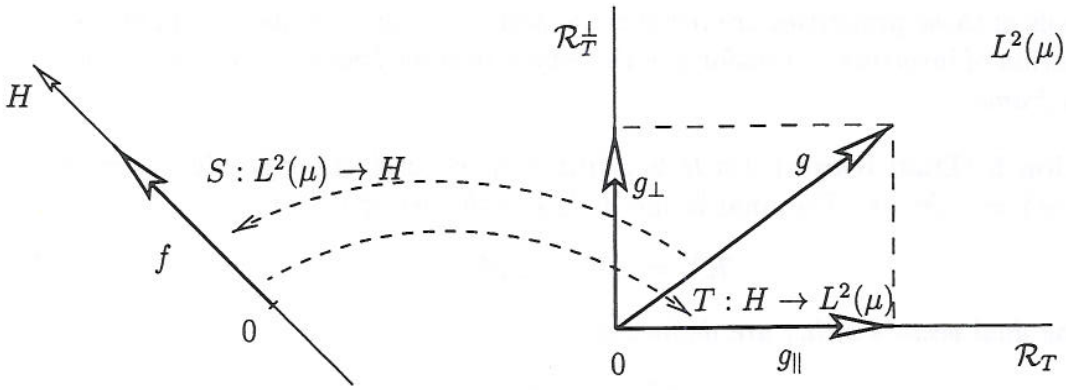


Figure 3.3 *The frame operator T takes a function $f \in H$ onto its range $\mathcal{R}_T$ in $L^2(\mu)$. The synthesizing operator S takes functions $g = g_\perp + g_\parallel$ in $L^2(\mu)$ to H, where $g_\parallel \in \mathcal{R}_T$, and $g_\perp$ is in the orthogonal complement of $\mathcal{R}_T$ in $L^2(\mu)$.*

6. *For any $g \in L^2(\mu)$, there is a unique $f \in H$ for which the discrepancy $\|g - \mathrm{T}f\|^2_{L^2(\mu)}$ is minimized, and it is given by*

$$f = \mathrm{S}g. \tag{3.35}$$

7. *For any $f \in H$, the following inequality holds*

$$\|\mathrm{T}f\|^2_{L^2(\mu)} \leq \|g\|^2_{L^2(\mu)} \tag{3.36}$$

for all $g \in L^2(\mu)$ such that $f = \mathrm{S}g$. Furthermore, (3.36) is an equality if and only if $g = \mathrm{T}f$.

□

The above results hold in the weak sense, but we expect that for reasonable and well-behaved choices of measure space M, frame vectors h_m, and functions $f(t)$, $g(m)$ the results will hold point-wise. A geometric interpretation of the transform and synthesis operators is given in Figure 3.3. We will not need the so-called reproducing kernel in this report, but we have included their relation with the projection operator for completeness. We will later see that the family $\{\psi_{a,b}\}$ corresponding to a mother wavelet constitutes a frame indexed by $m = (a, b)$ with measure space $M = \{(a, b) \in \mathbb{R}^+ \times \mathbb{R}\}$ and a measure $d\mu = da\, db/a^2$.

As we mentioned in the previous section, we would like to discretisize the parameters (a, b). This will involve discrete frames where the index m will be discrete. With μ in (3.21) as the counting measure (Appendix A), the frame condition still reads

$$A\|f\|^2_H \leq \|\tilde{f}\|^2_{L^2(\mu)} \leq B\|f\|^2_H. \tag{3.37}$$

However, $L^2(\mu)$ will be is now a space with a countable number of elements, and norms and inner products will now typically be defined using sums instead of integrals. These aspects are detailed in [8], [14].

3.2.2 Tight Frames

Tight frames are essential for our purpose, because we then have a Parseval's Theorem, as we show below.

Definition 6 (Tight frame) *A frame whose frame bounds are equal is called a* tight frame. □

The frame condition (3.21) for a tight frame reduces to

$$\|\tilde{f}\|^2_{L^2(\mu)} = A\|f\|^2_H, \tag{3.38}$$

where we use the notation from the previous sections.

Theorem 5 (Parseval's Theorem) *Let H be a Hilbert space, and let $\mathcal{H}_M$ be a tight frame with frame bound A indexed by m in the measure space M. Let T be the corresponding frame operator. Then, for any $f, g \in H$,*

$$\langle f, g\rangle_H = A^{-1}\langle \mathrm{T}f, \mathrm{T}g\rangle_{L^2(\mu)}. \tag{3.39}$$

□

The proof is in Appendix B.3. The theorem states, in other words, that the operator T is a partial isometry; it preserves both norms (or energy) and inner products (or angles) between elements of H. By a partial isometry, we mean that its range is not necessarily the whole of $L^2(\mu)$, but this is of less importance to us.

The significance of this Theorem lies in that any operations which may be written as an inner product in a Hilbert space, can equivalently be performed in a transform domain. This opens for alternative representations for a wide range of operators, in signal processing, statistics, quantum mechanics and other fields:

Corollary 1 *Let H be a Hilbert space, and let $\mathcal{H}_M$ be a tight frame with frame bound A indexed by m in the measure space M. Let T be the corresponding frame operator. If $F : H \to \mathbb{C}$ is an arbitrary bounded linear functional, there exist a unique $f \in H$ such that*

$$Fg = \langle g, f\rangle_H \quad \forall\, g \in H. \tag{3.40}$$

This operation may equivalently be written

$$Fg = A^{-1}\langle \mathrm{T}g, \mathrm{T}f\rangle_{L^2(\mu)}. \tag{3.41}$$

□

Proof The statement (3.40) is simply the Riesz representation Theorem, and (3.41) follows directly from Parseval's Theorem above. ■

There could be many reasons for wanting to perform an operation in a transform-domain, and they could vary from transform to transform. Some operations become simpler in the transform domain, such as convolution in time, which is a complicated operation, but amounts to a simple multiplication in the Fourier plane, In other cases one might be interested in performing operations on only certain *parts* or *components* of the function which are singled out in the transform domain. For our application, this latter reason will be important, as will be shown in Section 4.1.2.

3.3 Multiresolution and filter theory

So far, we have formulated the theory of wavelets in an abstract mathematical setting, and we have made use of Hilbert space formalism. We now exclusively concentrate on wavelets whose scaling and translation parameters are discretisized, as in Definition 2. This discretization leads naturally to *multiresolution analysis*, which also provides a connection to filter theory. We will see that the concept of multiresolution leads to a formulation of the basic theory which lends itself to an easy implementation on a computer. The resulting algorithm will be very fast, allowing us to compute the coefficients in the expansion in $O(N)$ time.

3.3.1 Multiresolution

The concept of multiresolution is fundamental to the theory that follows,

Definition 7 (Multiresolution) *A sequence of subspaces is called a* Multiresolution Analysis *if there exists a sequence $\{V_j\}_{j\in\mathbb{Z}}$ of closed subspaces in $L^2(\mathbb{R})$ and a $\phi \in V_0$ satisfying*

$$V_{j+1} \subset V_j \tag{3.42}$$

$$f(\cdot) \in V_{j+1} \Leftrightarrow f(2\cdot) \in V_j \tag{3.43}$$

$$\bigcap_{j\in\mathbb{Z}} V_j = \{0\} \tag{3.44}$$

$$\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R}) \tag{3.45}$$

$$\{\phi(t-k)\}_{k\in\mathbb{Z}} \quad \textit{is an orthonormal basis for } V_0 \tag{3.46}$$

We say that the multiresolution is generated by ϕ. □

We will not discuss to any depth the precise requirements a function ϕ has to satisfy in order to generate a multiresolution; they involve $\hat{\phi}(\nu)$ being bounded for all ν and continuous near $\nu = 0$ and that $\hat{\phi}(0) \neq 0$. This is discussed in [8], Chapter 5, where it is also shown that (3.46) can be relaxed; the family $\{\phi(t-k)\}_{k\in\mathbb{Z}}$ need only constitute a *Riesz basis.* A schematic illustration of a multi-resolution is given in Figure 3.4. The connection between wavelets and multiresolution analysis will now be made in the following two Theorems.

Theorem 6 (The Scaling identity) *Suppose that $\{V_j\}_{j\in\mathbb{Z}}$ constitute a multiresolution analysis generated by ϕ. Introducing the notation*

$$\phi_{j,k}(t) := \sqrt{2}\phi(2^{-j}t - k), \tag{3.47}$$

we have

- *The family*

$$\{\phi_{j,k}(t); k \in \mathbb{Z}\} \tag{3.48}$$

is an orthonormal basis for the space V_j.

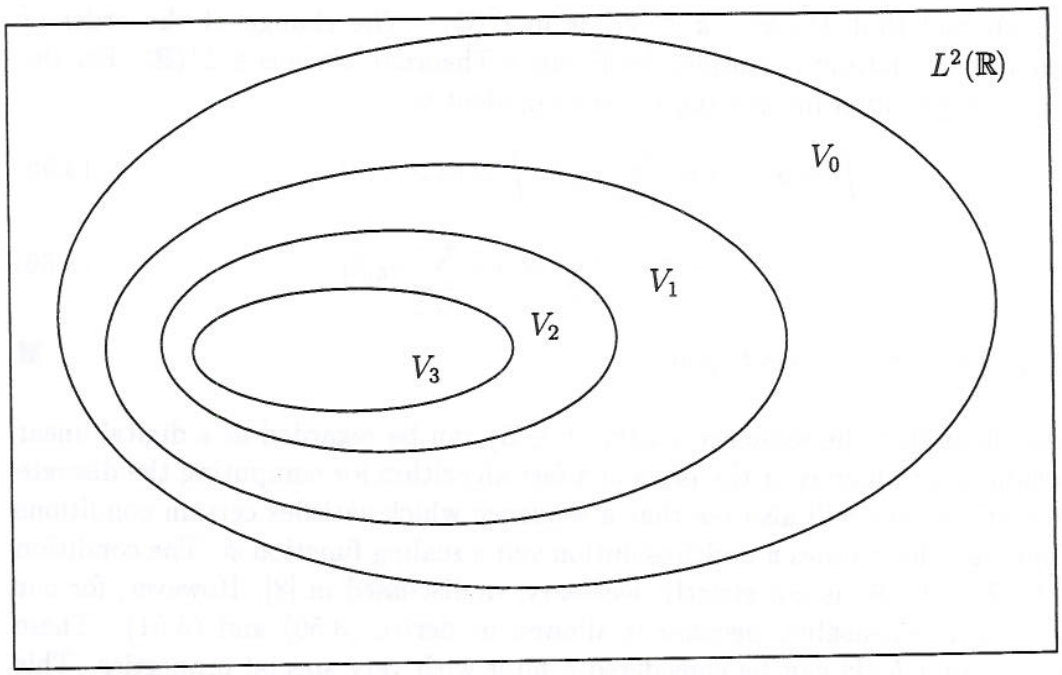


Figure 3.4 *The spaces V_j satisfy $V_{j+1} \subset V_j$. The initial space V_0 is an approximation to $L^2(\mathbb{R})$*

- *There exists a sequence $\{h_0[k] : k \in \mathbb{Z}\} \in \ell^2(\mathbb{Z})$ such that $\phi(t)$ satisfies*

$$\phi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h[k] \phi(2t - k). \tag{3.49}$$

  We refer to (3.49) as the scaling identity.

- *When $\phi \in L^2(\mathbb{R}) \cap L^1(\mathbb{R})$ and $\int dt\, \phi(t) \neq 0$, the sequence $\{h_0[k]\}$ also satisfies*

$$\sum_{k \in \mathbb{Z}} h_0[k] = \sqrt{2}, \tag{3.50}$$

$$\sum_{k \in \mathbb{Z}} h_0[k] \overline{h}_0[k + 2n] = \delta_{0,n}. \tag{3.51}$$

□

Proof The fact that $\{\phi_{j,k}(t) : k \in \mathbb{Z}\}$ is a orthonormal basis for V_j is a consequence of (3.43) and (3.46). Now $\{\phi_{-1,k}(t) : k \in \mathbb{Z}\}$ is an orthonormal basis for V_{-1}, and $\phi(t) \in V_0$ Since $V_0 \subset V_{-1}$ it follows that $\phi(t)$ can be expressed as (3.49), with $h_0[k] = \langle \phi, \phi_{-1,k} \rangle$. Also, since ϕ is orthonormal,

$$\delta_{0,n} = \langle \phi(\cdot - n), \phi(\cdot) \rangle \tag{3.52}$$

$$= \sum_{k \in \mathbb{Z}} \sum_{k' \in \mathbb{Z}} h_0[k] \overline{h}_0[k'] 2 \int dt\, \phi(2t - 2n - k) \overline{\phi}(2t - k') \tag{3.53}$$

$$= \sum_{k \in \mathbb{Z}} h_0[k] \overline{h}_0[k + 2n], \tag{3.54}$$

proving (3.50) and that $\{h_0[k] : k \in \mathbb{Z}\}$ is in $\ell^2(\mathbb{Z})$. The change of the order of integration and summation is justified by Fubini's Theorem, since $\phi \in L^1(\mathbb{R})$. For the same reason, we get upon integrating the scaling identity

$$\int dt\, \phi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_0[k] \int dt\, \phi(2t - k) \tag{3.55}$$

$$= \sqrt{2} \cdot 2^{-1} \int d\tilde{t}\, \phi(\tilde{t}) \sum_{k \in \mathbb{Z}} h_0[k], \tag{3.56}$$

and (3.51) follows, since $\int dt\, \phi(t) \neq 0$. ■

We shall see later that the sequence $\{h_0[k] : k \in \mathbb{Z}\}$ can be regarded as a digital linear time invariant filter which is at the basis of a fast algorithm for computing the discrete wavelet transform. We will also see that a sequence which satisfies certain conditions actually uniquely determines a multiresolution and a scaling function ϕ. The condition that $\phi \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ is not strictly necessary, as discussed in [8]. However, for our purposes, it is indispensable, because it allowed us derive (3.50) and (3.51). These conditions say that $h_0[k]$ can be considered a filter with very special properties. This will be discussed in the next section. Assuming ϕ to be in $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ is for practical purposes not very restrictive; one usually wants a "nice" scaling function, for example in $L^2(\mathbb{R})$, with compact support. Then ϕ will automatically be in $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$.

The usefulness of the following definition will become apparent in the next theorem.

Definition 8 (Wavelet spaces) *Assume that the sequence $\{V_j\}_{j \in \mathbb{Z}}$ constitutes a multiresolution analysis. The wavelet space W_j is then defined as the orthogonal complement of V_j in V_{j-1} so,*

$$V_{j-1} := V_j \oplus W_j, \tag{3.57}$$

where the sum is direct and orthogonal. □

Thus, we can say that the wavelet space at level j contains the extra detail required for going from V_j to the higher resolution space V_{j-1}. Defined in this way the different spaces W_j are orthogonal. If we have a high detail space at level j_0, we may decompose it in mutually orthogonal subspaces by iterating (3.57) as follows:

$$V_{j_0} = V_J \oplus \left(\bigoplus_{k=j_0+1}^{J} W_k \right). \tag{3.58}$$

We stop at some level J, where V_J contains coarse information which we are not interested in splitting further. The starting level j_0 is by convention taken to be 0. Since the spaces V_0 are dense in $L^2(\mathbb{R})$, we can also write

$$L^2(\mathbb{R}) = V_J \oplus \left(\bigoplus_{k=-\infty}^{J} W_k \right). \tag{3.59}$$

It turns out that whenever we have a multiresolution analysis, there exist an orthonormal basis with very desirable properties for each of the wavelet spaces W_j. This basis is a set of wavelets $\psi_{j,k}(t)$, hence the name *wavelet space*. These bases can explicitly be found from the scaling identity. This fact is often referred to as "the wavelet miracle" and is made precise in the following Theorem.

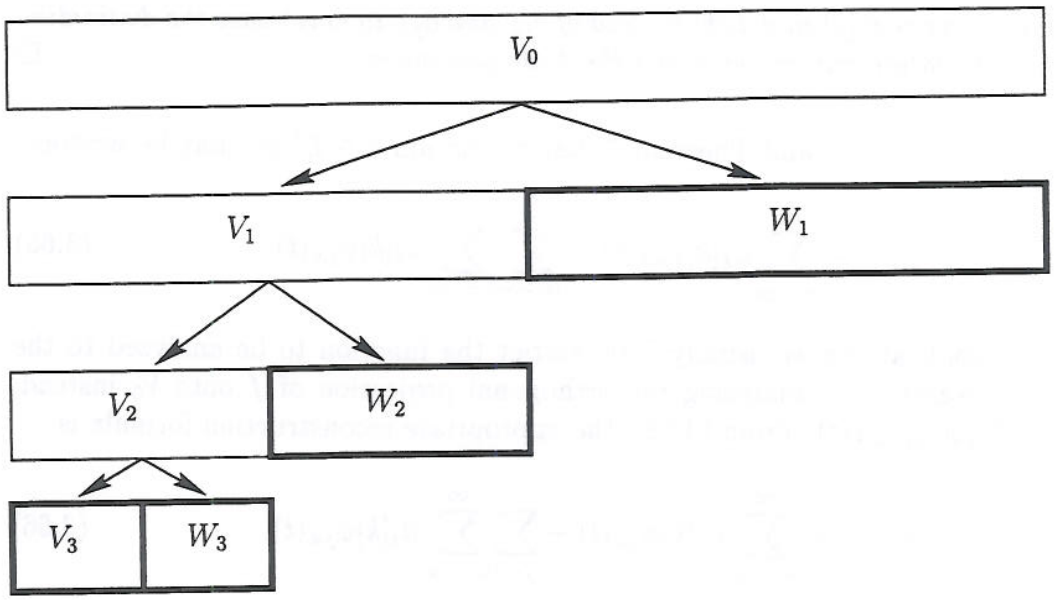


Figure 3.5 A three-level wavelet decomposition of the initial space V_0.

Theorem 7 *Assume that the sequence $\{V_j\}_{j\in\mathbb{Z}}$ constitutes a multiresolution analysis generated by ϕ. Define the function $\psi \in W_{-1}$ by*

$$\psi(t) := \sum_{k\in\mathbb{Z}} h_1[k]\phi_{-1,k}(t), \tag{3.60}$$

$$h_1[k] := (-1)^k h_0[2N+1-k], \quad N \in \mathbb{Z}. \tag{3.61}$$

Then:

- *$\{\psi_{j,k} : k \in \mathbb{Z}\}$ is an orthonormal wavelet basis for W_j,*
- *$\{\psi_{j,k} : j,k \in \mathbb{Z}\}$ is an orthonormal wavelet basis for $L^2(\mathbb{R})$,*
- *$\{\phi_{J,k}, \psi_{j,k} : j = 1,\ldots,J; k \in \mathbb{Z}\}$ is an orthonormal basis for V_0,*
- *The function ψ is not unique since all functions ψ' satisfy the above, where*

$$\hat{\psi}'(\nu) = \rho(\nu)\hat{\psi}(\nu), \tag{3.62}$$

and $\rho(\nu)$ has period 1 and modulus $|\rho(\nu)| = 1$ almost everywhere.

□

The proof of the Theorem is rather long, and can be found in Appendix B. The discrete wavelet transform of a function $f \in V_0$ is the set of Fourier coefficients, or inner products, between f and $\psi_{j,k}(t)$.

Definition 9 (The discrete wavelet transform) *Suppose that $\phi(t)$ generates a multiresolution analysis $\{V_j\}_j$, and that ψ is the corresponding mother wavelet. The J-level* discrete wavelet transform *of a function $f \in L^2(\mathbb{R})$ is defined as the set of coefficients $\{c_{J,k}, d_{j,k} : j = 1,\ldots,J, k \in \mathbb{Z}\}$, where*

$$c_{j,k} := \langle f, \phi_{j,k}\rangle, \tag{3.63}$$

$$d_{j,k} := \langle f, \psi_{j,k}\rangle, \qquad j = 1,\ldots,J. \tag{3.64}$$

We occasionally write $c_j[k]$ and $d_j[k]$ instead of $c_{j,k}$ and $d_{j,k}$ to emphasize the distinction between the translation parameter k and the scale parameter j. □

It now follows from (3.59) and Theorem 7 that a function $f \in L^2(\mathbb{R})$ may be written

$$f(t) = \sum_{k=-\infty}^{\infty} c_J[k]\phi_{J,k}(t) + \sum_{j=-\infty}^{J}\sum_{k=-\infty}^{\infty} d_j[k]\psi_{j,k}(t). \tag{3.65}$$

For practical applications we usually first restrict the function to be analyzed to the space V_0, for example by analyzing the orthogonal projection of f onto V_0 instead, $f_{V_0} := \sum_{k\in\mathbb{Z}}\langle f, \phi_{0,k}\rangle\phi_{0,k}(t)$. From (3.58), the appropriate reconstruction formula is

$$f_{V_0}(t) = \sum_{k=-\infty}^{\infty} c_J[k]\phi_{J,k}(t) + \sum_{j=0}^{J}\sum_{k=-\infty}^{\infty} d_j[k]\psi_{j,k}(t). \tag{3.66}$$

The convergence of the reconstruction formulae is, as usual, in $L^2(\mathbb{R})$-sense. For sufficiently nice functions, they will hold pointwise as well. These considerations are discussed in much more detail in many of the references, e.g. [8] or [6].

3.3.2 Filter theory

The multiresolution approach is useful in order to understand and construct wavelets. In this section we establish the relationship between wavelet analysis and filter theory. The main result in this respect is contained in the following theorem, due to Stéphane Mallat.

Theorem 8 (Fast Wavelet Transform) *Assume that the finest scale coefficients $\{c_0[k] : k \in \mathbb{Z}\}$, as defined by (3.63) are given. The coefficients in a J-level wavelet expansion can then be recursively found for $j = 1, \ldots, J$ by*

$$c_j[k] = \sum_{n\in\mathbb{Z}} \overline{h_0[n-2k]}c_{j-1}[n], \tag{3.67}$$

$$d_j[k] = \sum_{n\in\mathbb{Z}} \overline{h_1[n-2k]}c_{j-1}[n]. \tag{3.68}$$

The fine scale coefficients $\{c_0[k] : k \in \mathbb{Z}\}$ can be recovered recursively from a J-level wavelet expansion by

$$c_{j-1}[n] = \sum_{n\in\mathbb{Z}} h_0[k-2n]c_j[n] + \sum_{n\in\mathbb{Z}} h_1[k-2n]d_j[n], \quad j = 1\ldots J. \tag{3.69}$$

□

This is proved in Appendix B.4, as a special case of Theorem 12 which will be stated later. This result allows the coefficients in the the wavelet expansion (3.65) to be calculated using simple digital filtering operations, assuming we somehow have the fine scale inner products $c_0[k]$. In practice one often approximates these by the *samples* of the analyzed function, $f(kT) \approx c_0[k]$, where T is the sample period.

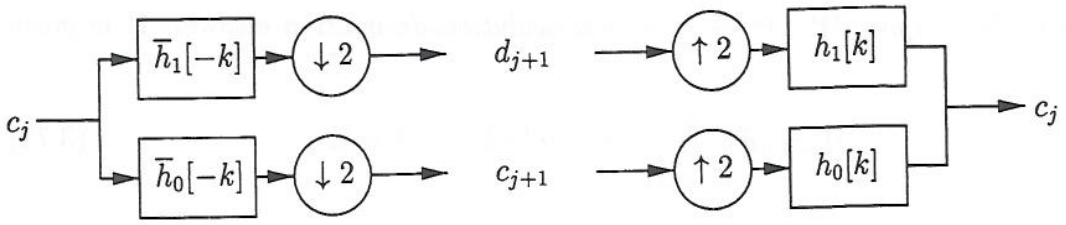


Figure 3.6 One stage in the wavelet decomposition and reconstruction scheme.

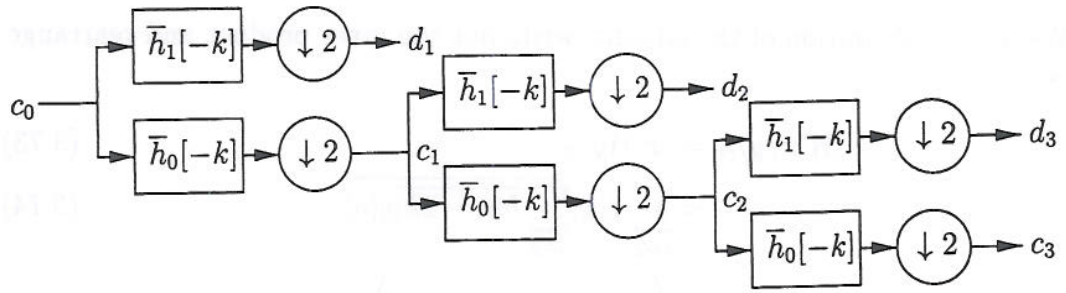


Figure 3.7 A three-stage wavelet decomposition using an iterated filter bank.

One stage in both the decomposition and reconstruction is illustrated by the block diagram in Figure 3.6. The symbol $\downarrow 2$ performs *downsampling* by two, that is, it acts on sequences such that $(\downarrow 2a)[k] = a[2k]$. Similarly, $\uparrow 2$ inserts zeros between every other element:

$$(\uparrow 2a)[k] = \begin{cases} a[k/2], & k \text{is even}, \\ 0, & k \text{ is odd} \end{cases}. \tag{3.70}$$

Such filter structures have been in use for a long time, and the filters are generally referred to as *quadrature mirror filters*. The sequence $\{c_j[k]\}$ is split into two sequences $\{c_{j+1}[k]\}$ and $\{d_{j+1}[k]\}$ from which the original $\{c_j[k]\}$ can be recovered. This is called the *perfect reconstruction property*, and only very special filter sequences h_0 and h_1 allow this. Figure 3.7 shows how the decomposition shown in Figure 3.5 would be implemented in practice using iterated filtering and downsampling.

Although this exactly how how MATLAB's Wavelet Toolbox [25] implements the wavelet transform, it is not optimal. Taken literally, the figures would imply that a convolution is performed at each level, whereupon half the samples are thrown away in the downsampling. This is clearly wasteful, and the computing speed can be doubled by combining the convolution and decimation processes into one operator.

Definition 10 (Convolution-decimation operator) *The* convolution-decimation operator *corresponding to a filter given by the sequence* $\{h[k]\}_{k\in\mathbb{Z}}$ *is given by the operator* $\mathrm{H} : \ell^2 \to \ell^2$ *defined by*

$$x[k] \mapsto \sum_{n\in\mathbb{Z}} \overline{h[n-2k]}x[n] \qquad k \in \mathbb{Z}. \tag{3.71}$$

□

We will need the following lemma.

Lemma 9 *The adjoint* $\mathrm{H}^* : \ell^2 \to \ell^2$ *of a convolution-decimation operator* H *is given by*

$$(\mathrm{H}^* x)[n] = \sum_{k \in \mathbb{Z}} h[n-2k]x[k] \qquad k \in \mathbb{Z}. \tag{3.72}$$

□

Proof We use the definition of the adjoint, write out the inner product and rearrange the sums;

$$\begin{aligned} \langle \mathrm{H}^* x, y \rangle_{\ell^2} &= \langle x, \mathrm{H}y \rangle_{\ell^2} && (3.73)\\ &= \sum_{k \in \mathbb{Z}} x[k] \overline{\sum_{n \in \mathbb{Z}} h[n-2k] y[n]} && (3.74)\\ &= \left\langle \sum_{k \in \mathbb{Z}} h[n-2k]x[k], y \right\rangle_{\ell^2}, && (3.75) \end{aligned}$$

and the theorem is proved. ■

The similarity to the operations in (3.67), (3.68) and (3.69) is apparent.

Definition 11 (Quadrature Mirror Filters) *A set* $\{F_0, F_1\}$ *of convolution-decimation operators are said to be orthogonal* quadrature mirror filters *(QMFs) if they and their adjoints satisfy the conditions below.*

$$\begin{aligned} F_0 F_0^* &= F_1 F_1^* = \mathrm{I}, && (3.76)\\ F_0 F_1^* &= F_1 F_0^* = 0, && (3.77)\\ F_0^* F_0 &+ F_1^* F_1 = \mathrm{I}. && (3.78) \end{aligned}$$

□

The scaling sequence $\{h_0[k]\}$ and the associated wavelet sequence $\{h_1[k]\}$ actually satisfy the conditions above, so they define a pair of QMF. From Theorem 8, we see that we can write

$$\begin{aligned} c_j[k] &= (\mathrm{H}_0 c_{j-1})[k], && (3.79)\\ d_j[k] &= (\mathrm{H}_1 c_{j-1})[k]. && (3.80) \end{aligned}$$

Using the language of convolution-decimation operators, we may write the reconstruction formula (3.69)

$$\begin{aligned} c_{j-1} &= \mathrm{H}_0^* c_j + \mathrm{H}_1 d_j && (3.81)\\ &= (\mathrm{H}_0^* \mathrm{H}_0 + \mathrm{H}_1^* \mathrm{H}_1) c_{j-1},, && (3.82) \end{aligned}$$

which shows that H_0 and H_1 satisfy (3.78). That (3.76) and (3.77) are satisfied follows from (3.51) and the definition of h_1 (3.61). Using these operators, we may state the algorithm for the fast wavelet transform.

Algorithm 1 (Fast wavelet transform)
Denote $c_j := \{c_j[k] : k \in \mathbb{Z}\}$ *and* $d_j := \{d_j[k] : k \in \mathbb{Z}\}$, *and let* J *be the number of scales to be used in the decomposition. Furthermore, let* H_0, H_1 *be two convolution-decimation operators corresponding to a wavelet decomposition.*

1. Input $c_0 := \{c_0[k] : k \in \mathbb{Z}\}$ and $J > 0$
2. for $j = 1, \ldots, J$, {

$$c_j = \mathrm{H}_0 c_{j-1}; \tag{3.83}$$
$$d_j = \mathrm{H}_1 c_{j-1}; \tag{3.84}$$

   }
3. return c_J, d_j, $j = 1, \ldots, J$;

□

3.4 Wavelet packets as generalization of wavelets

In this section, we will generalize the wavelet decomposition scheme we have developed so far. Before proceeding more formally, we will present the basic idea behind the *wavelet packet analysis.*

3.4.1 The idea

A simple extension of the scheme presented in Figure 3.5 would be to split the wavelet spaces W_j as well as the approximation spaces V_j. This results in a decomposition of the initial space V_0 as indicated in Figure 3.8. At a level j, we have 2^{j-1} wavelet packet spaces, indexed by the level j and a *frequency index* f. In Figure 3.8, we have indicated the spaces which correspond to the usual wavelet decomposition $V_0 = V_3 \oplus W_3 \oplus W_2 \oplus W_1$ by a a thicker line. The wavelet packet scheme offer a much greater flexibility because we can choose between several other decompositions of the initial space, and one such choice is shown hatched.

We will in Section 3.4.2 show how we can split the initial space using wavelet packets on a single scale, corresponding to the dotted spaces in Figure 3.8. The next Section 3.4.3 presents conditions for splitting the initial space into spaces at different scales, corresponding to the spaces marked gray in Figure 3.8. Finally, criteria for choosing the *best basis* for any particular application will be given in Section 3.4.4.

3.4.2 Single scale Wavelet packets

The definition of the (single scale) wavelet packets can now be stated.

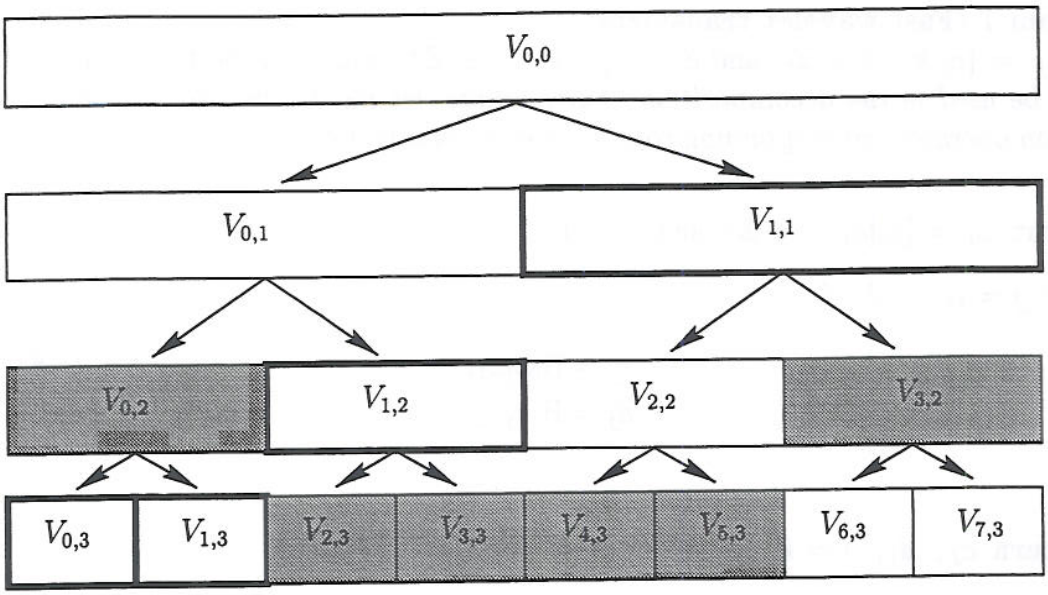


Figure 3.8 *The structure obtained by splitting the initial space $V_{0,0}$ into orthogonal wavelet packet spaces $V_{f,j}$, $f = 1, 2, \ldots 2^{j-1}$ at level j.*

Definition 12 (Single scale wavelet packets) *The single scale wavelet packets are recursively defined as*

$$w_{2f} = \sqrt{2} \sum_{k \in \mathbb{Z}} h_0[j] w_f(2t - k) \tag{3.85}$$

$$w_{2f+1} = \sqrt{2} \sum_{k \in \mathbb{Z}} h_1[j] w_f(2t - k) \tag{3.86}$$

where $\int w_0 \, dt = 1$. □

Clearly, w_0 and w_1 can be identified with the scaling function ϕ and the mother wavelet ψ, respectively. For a quadrature mirror filters that generate a multiresolution analysis, these functions are well-defined. The wavelet packets are thus linear combinations of the scaling function and wavelets, and the eight first wavelet packets derived from the the Daubechies 6 system are shown in Figure 3.9. We note that the wavelet packets number of oscillations tend to increase with increasing f. This observations is the reason for referring to f as a frequency index.

Definition 13 (Wavelet packet spaces) *Given a multiresolution analysis $\{V_j\}_{j \in \mathbb{Z}}$, and the two-scale operators we define the* single scale wavelet packet spaces U_f *as the closed linear span*

$$U_f = \overline{\text{span}\{w_f(t - k) : \quad k \in \mathbb{Z}\}}. \tag{3.87}$$

More generally, the wavelet packet spaces are defined by

$$W_{f,j} = \overline{\text{span}_k\{2^{-j/2} w_f(2^{-j}t - k) : \quad k \in \mathbb{Z}\}}. \tag{3.88}$$

□

In this section we will concentrate on the single scale spaces U_f and defer the treatment of $W_{f,j}$ To the next section. The main result of this section is the following theorem.

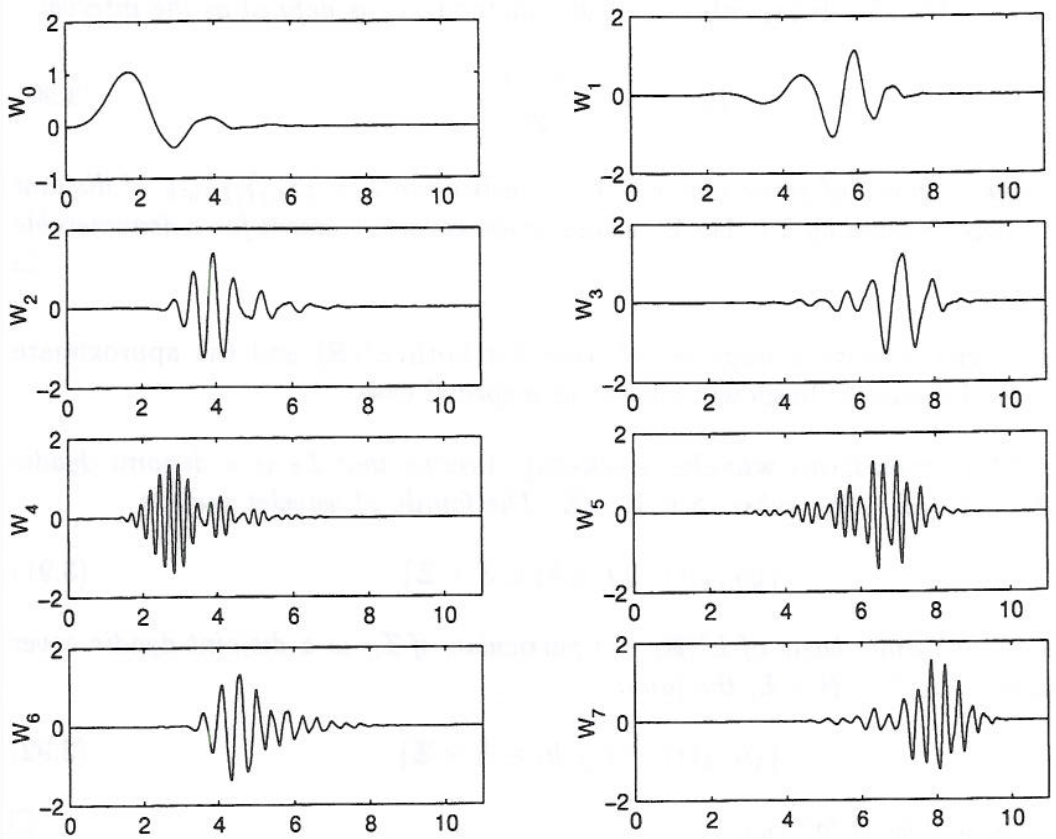


Figure 3.9 The first eight wavelet packets derived from the db6 scaling function.

Theorem 10 *For any $f \geq 0$, the set $\{w_f(t-k) : k \in \mathbb{Z}\}$ is an orthonormal basis for U_f. Furthermore, the set $\{w_f(t-k) : (k,f) \in \mathbb{Z}^2, f \geq 0\}$ is an orthonormal basis for $L^2(\mathbb{R})$.* □

The proof can be found in [7].

3.4.3 Multi scale wavelet packet

We will now translate and *dilate* the single-scale wavelet packets w_f from the previous section. This results in the most general wavelet packets.

Definition 14 (Wavelet packets) *The wavelet packets are functions $\psi_{f,j,k}(t) \in L^2(\mathbb{R})$ indexed by a scaling parameter j, a frequency parameter f and a translation parameter k defined by*

$$\psi_{f,j,k}(t) := 2^{-j/2} w_f(2^{-j}t - k), \tag{3.89}$$

where the functions $w_f(t)$ are single-scale wavelet packets as defined in Definition 12. □

Translations of these functions constitute an orthonormal basis for $L^2(\mathbb{R})$ for certain choices of (f,j). To characterize these sets, we introduce the following definition.

Definition 15 (Dyadic intervals) *A* dyadic interval $I_{f,j}$ *is defined as the interval*

$$I_{f,j} := \left[\frac{f}{2^j}, \frac{f+1}{2^j}\right). \tag{3.90}$$

A disjoint dyadic cover *of some interval A is a collection $\mathcal{I} = \{I_{f,j}\}_{(f,J)\in S}$ of disjoint dyadic intervals, indexed by $S \subset \mathbb{N}\times\mathbb{Z}$, whose union covers A except for a denumerable set.* □

We may now characterize a large set of bases for both $L^2(\mathbb{R})$ and the approximate space V_0, and the wavelet basis will emerge as a special case.

Theorem 11 (Bases from wavelet packets) *Assume that $\mathcal{I}_S$ is a disjoint dyadic cover of $\mathbb{R}^+$ indexed by the subset $S \subset \mathbb{N}\times\mathbb{Z}$. The family of wavelet packets*

$$\{\psi_{f.j,k}(t) : (f,j,k) \in S \times \mathbb{Z}\} \tag{3.91}$$

is then an orthonormal basis of $L^2(\mathbb{R})$. In particular, if $\mathcal{I}_R$ is a disjoint dyadic cover of $[0,1)$ indexed by $R \subset \mathbb{N}\times\mathbb{Z}$, the family

$$\{\psi_{f.j,k}(t) : (f,j,k) \in R \times \mathbb{Z}\} \tag{3.92}$$

is an orthonormal basis for $V_{0,0}$. □

Proof See [7] ■

This Theorem makes it possible to create orthonormal bases form linear combinations of wavelets. However, it does not fully characterize *all* orthonormal bases that may be created from a given family of wavelets and quadrature mirror filters. There are for example orthonormal bases derived from the Haar system of wavelets that do not correspond to a disjoint dyadic cover of $[0,1)$. This is discussed in more depth in [42]

We now have a wavelet packet transform, which consists of coefficients $c_{f,j}[k] := \langle f, \psi_{f,j,k}\rangle$. The coefficients can be computed recursively similar to the wavelet coefficients.

Theorem 12 *Assume that the finest scale wavelet packet coefficients are given. The other coefficients in an wavelet packet decomposition can be found by*

$$c_{2f+\epsilon,j}[k] := \sum_{n\in\mathbb{Z}} \overline{h_\epsilon[n-2k]}\langle f, \psi_{f+\epsilon,j-1,n}\rangle = (\mathrm{H}_\epsilon c_{f+\epsilon,j-1})[k] \tag{3.93}$$

□

The proof is in the Appendix. The question of finding from a collection of bases the *best basis* for a particular signal is addressed next.

3.4.4 Selection of a best basis

We have seen how one may get many different bases out of a wavelet packet scheme. In this section, we show how one can find an *best basis* with respect to some criterion which is to be specified. To give that criterion, we first introduce the concept of information cost.

Definition 16 (Information cost functional) *An* information cost functional *is any functional* $M : \mathbb{R}^N \to \mathbb{R}$ *which can be written*

$$M(u) = \sum_{k \in \mathbb{Z}} \mu(|u[k]|) \tag{3.94}$$

where the function μ *satisfy*

$$\mu : \mathbb{R}^+ \to \mathbb{R}; \quad \mu(0) = 0; \quad \text{and} \sum_{k \in \mathbb{Z}} \mu(|u[k]|) < \infty \quad \forall u[k] \in \mathbb{R}^N. \tag{3.95}$$

□

For practical purposes, a information cost functional should measure the degree of *concentration* of the information in a sequence. To capture the intuitive notion of information cost, $M(u)$ should be large when the elements of the vector u are are of similar magnitude, and small when the energy of the vector is concentrated in a few elements. Some common choices are listed below.

- Shannon entropy
  The entropy of a vector $u \in \mathbb{R}^N$ is commonly defined as

$$H(u) = -\sum_{k=1}^{N} \frac{|u[k]|^2}{\|u\|_2^2} \log \frac{|u[k]|^2}{\|u\|_2^2}. \tag{3.96}$$

  This entropy is minimized when the cost functional

$$M_s(u) = -\sum_{k=1}^{N} |u[k]|^2 \log \left(|u[k]|^2 \right). \tag{3.97}$$

  is minimized.

- Threshold
  We pick a threshold ϵ and define the function

$$\mu(x) = \begin{cases} |x|, & |x| \geq \epsilon, \\ 0, & |x| < \epsilon. \end{cases} \tag{3.98}$$

  The cost functional $M(u)$ then gives the sum of the magnitude of the elements whose magnitude exceeds the threshold ϵ.

- Concentration in l^p-norm
  The cost functional is here defined as

$$M_{l^p}(u) = \sum_{k=1}^{N} |u[k]|^p = \|u\|_p^p, \quad p \in [1,2). \tag{3.99}$$

  This cost measures concentration of energy. If the two vectors $u, v \in \mathbb{R}^N$ have the same energy ($\|u\|_{\ell^2} = \|v\|_{\ell^2}$) but $M(u) < M(v)$, then u has a greater part of its energy concentrated into fewer coefficients.

For other information measures and a discussion of the relative merits of these information costs we refer to [42]. We can now give precise meaning to the concept *best basis*.

Definition 17 (Best basis) *Let $\mathcal{B}$ be a collection of countable bases for a separable Hilbert space H, and let $B = \{b_k\}_k \in \mathcal{B}$ be one of these bases. Furthermore, let M be a information cost functional, and define the coefficients of the representation of the vector $v \in H$ in the basis B by $c := \langle v, b_k \rangle$. Letting $c := \{c_k\}_{k \in \mathbb{Z}}$, we define the* M-information cost of v in the basis B *by*

$$\mathcal{M}_v : \mathcal{B} \to \mathbb{R}; \quad B \mapsto M(c_k). \tag{3.100}$$

The best basis *for v with respect to the information cost M and the collection $\mathcal{B}$ is a basis set B which minimizes the M-information cost $\mathcal{M}$ over $\mathcal{B}$.* □

A recursive algorithm for finding an optimal basis for a vector v from a library $\mathcal{B}$ with the structure of a wavelet packet decomposition may now be derived. This algorithm, originally by Wickerhauser and Coifman, thus *concentrates* the information contained in the vector v. For the purpose of detection and characterization of signals, this means that we can find a basis which optimally concentrates the class of signals we are looking for into a few characteristic wavelet or wavelet packet spaces. This means that when analyzing data, we can restrict the analysis to those spaces where there is a significant contribution by signals similar to the the signals we are looking for.

Algorithm 2 (Best basis)
Let $\{W_{j,f} : j = 0 \dots J, f = 0 \dots 2^{J-1}\}$ be the spaces corresponding to a full J-level wavelet packet decomposition. Denote the wavelet packet basis for the space $W_{j,f}$, by

$$B_{j,f} := \{\psi_{j,f,k}(t) : k \in \mathbb{Z}\}. \tag{3.101}$$

Let $v \in W_{0,0}$ and let $\mathcal{M}_v$ be an M-information cost (Definition 17). The best basis algorithm can be summarized as follows.

1. `let` $j = J > 0$;
   `for` $f = 0 \dots 2^j - 1$, {`let` $A_{J,f} = B_{J,f}$;}

2. `for` $f = 0 \dots 2^{j-1} - 1$, {`let`

$$A_{j-1,f} = \begin{cases} A_{j,2f} \oplus A_{j,2f+1}, & \texttt{if } \mathcal{M}_v(A_{j,2f} \oplus A_{j,2f+1}) < \mathcal{M}_v(B_{j-1,f}), \\ B_{j-1,f}, & \texttt{otherwise}. \end{cases} \tag{3.102}$$

   }

3. while $j > 0$, {let $j = j - 1$; goto 2;}

4. return $A_{0,0}$;

□

In step 1, the algorithm is initialized. In step 2 we find the find the least expensive representation of v projected onto the span of $B_{j-1,f}$, which is the wavelet packet space $W_{j-1,f}$. The returned basis $A_{0,0}$ is the best basis, and we prove this in the following Theorem.

Theorem 13 *The Best Basis algorithm finds the best basis with respect to the information cost M and the library tree $\mathcal{B}$ for any fixed $v \in W_{0,0}$* □

Proof The proof is by induction on the depth J of the decomposition. Let $A'_{i,j}$ be any basis for $W_{i,j}$.

1. For a tree of depth $J = 0$, the only basis in the library is the basis for $W_{0,0}$. This is clearly the best basis.

2. For the induction step, suppose that that the algorithm gives the best algorithm for any tree of depth J.

3. We prove that the $A_{0,0}$ returned by the algorithm applied to tree of depth $J+1$ satisfies $\mathcal{M}_v(A_{0,0}) < \mathcal{M}_v(A'_{0,0})$, so $A_{0,0}$ is a best basis.

   By the construction of the library $\mathcal{B}$, $A'_{0,0}$ either equals $B_{0,0}$ or $A'_{1,0} \oplus A'_{1,1}$. The induction hypothesis states that the algorithm applied to each of the two subtrees of depth J starting from $V_{1,0}$ and $V_{1,1}$ returns the best bases for these two spaces. These bases, $A_{1,0}$ and $A_{1,1}$, then satisfy, for any $A'_{1,0}, A'_{1,1}$,

$$\mathcal{M}_v(A_{1,0}) \leq \mathcal{M}_v(A'_{1,0}), \quad \mathcal{M}_v(A_{1,1}) \leq \mathcal{M}_v(A'_{1,1}). \tag{3.103}$$

   The basis $A_{0,0}$ returned by the algorithm applied to the whole tree satisfies, by (3.101) and the above result,

$$\begin{aligned} \mathcal{M}_v(A_{0,0}) &= \min\{\mathcal{M}_v(B_{0,0}),\ \mathcal{M}_v(A_{1,0}) + \mathcal{M}_v(A_{1,1})\} && (3.104)\\ &\leq \min\{\mathcal{M}_v(B_{0,0}),\ \mathcal{M}_v(A'_{1,0}) + \mathcal{M}_v(A'_{1,1})\} && (3.105)\\ &\leq \mathcal{M}_v(A'_{0,0}). && (3.106) \end{aligned}$$

Consequently, $A_{0,0}$ is a best basis. ■

3.5 Shift properties

It is well known that the wavelet and wavelet transforms are *shift variant.*

Theorem 14 (Shift invariance) *Let* S *be a shift operator, such that* $\mathrm{S}x(t) = x(t+1)$. *Then, for a fixed scale and* $q = q_0 + 2^j q_1$, $q_0 = 0, 1, \ldots, 2^j - 1$, *the wavelet packet transform of* $S^q x$ *is*

$$\left(\mathrm{TS}^{q_0+2^j q_1} x\right)(f, j, k) = \left(\mathrm{TS}^{q_0} x\right)(f, j, k + q_1) \tag{3.107}$$

□

Proof

$$\begin{aligned}
\left(\mathrm{TS}^{q_0+2^j q_1} x\right)(f, j, k) &= \int dt\, x(t + q_0 + 2^j q_1)\overline{\psi_{f,j,k}(t)} && (3.108)\\
&= \int dt\, x(t + q_0 + 2^j q_1)\overline{2^{-j/2} w_f(2^{-j}t - k)} && (3.109)\\
&= \int dt\, x(\tilde{t} + q_0)\overline{2^{-j/2} w_f(2^{-j}(\tilde{t} - 2^j q_1(j)) - k)} && (3.110)\\
&= \int d\tilde{t}\, x(\tilde{t} + q_0)\overline{2^{-j/2} w_f(2^{-j}\tilde{t} - (q_1(j) + k))} && (3.111)\\
&= \left(\mathrm{TS}^{q_0} x\right)(f, j, k + q_1(j)) && (3.112)
\end{aligned}$$

■

4 THEORY OF ESTIMATION

This chapter reviews theory for detecting and characterizing signals in a noisy background. We discuss the possibilities for using wavelet-based techniques for detection and characterization purposes.

4.1 The signal model

The problem of detecting whether a signal is present in a data set will be referred to as *the detection problem*, and can be formalized as follows;

Definition 18 (The Detection Problem) *Let $x(t)$ be data observed for a period T over an interval $I = [t, t+T]$. The problem of determining at time $t+T$ whether a partially known waveform $s(t;\theta)$ which is parameterized by a vector θ of unknown parameters is present in the data for $t \in I$ can be written as a hypothesis test;*

$$H_0 : x(t) = n(t) \tag{4.1}$$

$$H_1 : x(t) = s(t;\theta) + n(t) \tag{4.2}$$

Here, $n(t)$ is a stochastic process, modelling everything in the data which is not a part of the signal $s(t)$. We assume that $n(t)$ is Gaussian white noise with covariance $\mathrm{E}[n(\tau)n(\tau')] = N_0/2\delta(\tau-\tau')$. We refer to this hypothesis test as the detection problem. □

Such hypothesis tests are frequently used in communication theory, where one typically is interested in determining whether a certain signal is or is not present. In our application we are interested in detecting characteristic transient vibrations in a ship hull. Because these vibrations are *characteristic* of the hull, there are reason to believe that they are continuously excited. However, a heavy wave impact will excite a single, relatively well-defined transient. and it is these transients we are most interested in detecting.

We now turn towards quantitative methods for distinguishing between the two competing hypotheses.

4.1.1 Matched filter

Our concern is to be able to detect signals with unknown arrival times. To do this, we will need to solve the detection problem above for all possible arrival times t. Detection problems of this type is commonly handled by matched filter techniques, and two classic reference are [36] and [37]. Briefly, the matched filter technique is based on running the data $x(t)$ through a linear time-invariant filter h_{mf} which is designed to give a large output value for those parts of the data which belongs to the signal we wish to detect. With reference to Figure 4.1, we define the output signal-to-noise ratio as

$$\mathrm{SNR}_o = \frac{|s_o(T)|^2}{\mathrm{E}[n_o^2(t)]}, \tag{4.3}$$

E being the expectation operator.

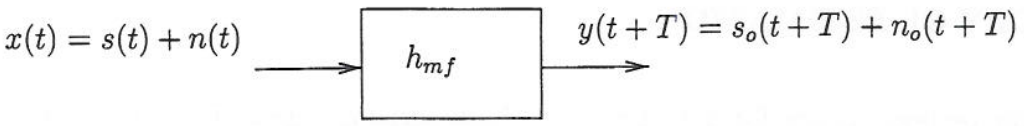


Figure 4.1 Schematic representation of a matched filter

Definition 19 (Matched filter) *A matched filter is a linear time invariant filter which maximizes SNR_o at a time T corresponding to an arrival at $t = 0$.* □

The impulse response of the matched filter is easily found if the noise process is white Gaussian noise;

Theorem 15 *When the noise process $n(t)$ is white, the impulse response given by*

$$h_{MF}(t) = \begin{cases} k \cdot s(T-t) & t \in [0,T] \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

defines a matched filter for arbitrary non-zero constants k. □

Proof If $h(t)$ is the impulse response of any causal linear time-invariant filter, then, for any input signal s, the output is $s_O(t) = (s * h)(t)$. We find, regarding t as a fixed parameter,

$$\begin{aligned} |(s*h)(t)|^2 &= \left| \int_{-\infty}^{\infty} d\tau\, s(\tau) h(t-\tau) \right|^2 \\ &= |\langle \tilde{s}_t, \overline{h} \rangle|^2, \end{aligned}$$

where we have introduced $\tilde{s}(\cdot) = -s(t - \cdot)$. We also find for $n_o(t) \equiv (n * h)(t)$

$$\begin{aligned} \mathrm{E}[|n_o(t)|^2] &= \mathrm{E}\left[\int_{-\infty}^{\infty} d\tau\, n(\tau) h(t-\tau) \int_{-\infty}^{\infty} d\tau'\, n(\tau') h(t-\tau') \right] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} d\tau\, d\tau' \underbrace{\mathrm{E}\left[n(\tau) n(\tau')\right]}_{N_o \delta(\tau-\tau')/2} h(t-\tau) \overline{h(t-\tau')} \\ &= \frac{N_0}{2} \|h\|^2, \end{aligned}$$

where we have used standard properties of the covariance of Gaussian white noise. Now, using the Cauchy-Schwartz inequality, we can find an upper bound for the signal-to-noise ratio;

$$\mathrm{SNR}_o = \frac{|\langle \tilde{s}_T, \overline{h} \rangle|^2}{\frac{N_0}{2}\|h\|^2} \leq \frac{\|\tilde{s}_T\|^2 \|h\|^2}{\frac{N_0}{2}\|h\|^2}. \tag{4.5}$$

Thus, $\mathrm{SNR}_o \leq 2\|\tilde{s}_{t_a+T_0}\|^2/N_0$, with equality if and only if $\overline{h}$ is proportional to $\tilde{s}_T$ Since s is assumed real-valued, the SNR attains its upper bound when

$$h(\cdot) = \text{const} \cdot s(T - \cdot),$$

and the theorem is proved. ■

To calculate the detector output, we find

$$y(t+T) = \int_{-\infty}^{\infty} d\tau\, x(\tau) h_{mf}(t+T-\tau) \tag{4.6}$$

$$= \int_{-\infty}^{\infty} d\tau\, x(\tau) s(T-t-T+\tau) \tag{4.7}$$

$$= \int_{0}^{T} d\tau\, x(\tau+t) s(\tau), \tag{4.8}$$

where we used the fact that s is supported on $[0,T]$. Now, if $y(t+T)$ is larger than some predetermined threshold, we say that we have a detection at time t.

4.1.2 A wavelet based matched filter

As we mentioned in Section 3.2.2, one of the advantages with tight frames is that any operation which can be represented as a inner product, may equivalently be performed in the transform-domain. This leads us to a wavelet domain representation of a matched filter.

Theorem 16 (Wavelet-domain matched filter) *Assume that $\mathcal{I}_G$ is a disjoint dyadic cover of $[0,1)$ indexed by G, so that the collection*

$$\{\psi_{f,j,k}(t) : (f,j,k) \in S \times \mathbb{Z}\} \tag{4.9}$$

is an orthonormal basis for $V_{0,0}$. If the functions $x(t), s(t) \in V_{0,0}$, we may write (4.6) as

$$y(t+T) = \sum_{(f,j)\in G} \sum_{k=-\infty}^{\infty} (Tx)(f,j,k+q_1(j)) \overline{(\mathrm{TS}^{q_0(j)} s)(f,j,k)}. \tag{4.10}$$

□

Proof Introducing the shift operator S, we shift the reference transient $\mathrm{S}^{-t}s = s(\cdot - t)$. Let T be a wavelet packet transform corresponding to the dyadic cover $\mathcal{I}_G$. The matched filter output is, using Corollary 1,

$$y(t+T) = \langle x(\cdot), \mathrm{S}^{-t} s(\cdot) \rangle_{L^2(\mathbb{R})} \tag{4.11}$$

$$= \langle \mathrm{T}x, \mathrm{TS}^{-t} s \rangle_{\ell^2(\mathbb{Z}^3)} \tag{4.12}$$

$$= \sum_{(f,j)\in G} \sum_{k=-\infty}^{\infty} (\mathrm{T}x)(j,k) \overline{(\mathrm{TS}^{-t} s)(j,k)}. \tag{4.13}$$

Now, we get by using Theorem 14 and taking $t = q_0(j)+2^j q_1(j)$ where $q_0 \in \{0, \ldots, 2^j - 1\}$,

$$y(t+T) = \sum_{(f,j)\in G} \sum_{k=-\infty}^{\infty} (\mathrm{T}x)(j,k)\overline{(\mathrm{TS}^{-t}s)\,(j,k)} \tag{4.14}$$

$$= \sum_{(f,j)\in G} \sum_{k=-\infty}^{\infty} (\mathrm{T}x)(j,k)\overline{(\mathrm{TS}^{q_0(j)}s)\,(j,k-q_1(j))} \tag{4.15}$$

$$= \sum_{(f,j)\in G} \sum_{k=-\infty}^{\infty} (\mathrm{T}x)(j,k+q_1(j))\overline{(\mathrm{TS}^{q_0(j)}s)\,(j,k)} \tag{4.16}$$

$$\tag{4.17}$$

■

As (4.10) stands, it is equivalent to (4.6). However, if we restrict the index set G to be those frequencies f and those scales j where most of the energy in s are located, we effectively perform a selective matched filtering. In our experience, this can give improved results over the conventional time-domain matched filter. However, the results are sensitive as to which scales or to which analyzing wavelet packet is used.

4.2 Parameter estimation

There is a large literature on estimating parameters and characterizing signals. The by far most popular method uses as *maximum likelihood approach* to the estimation. It is not necessarily optimal in the sense that it that it is unbiased and has a minimum expected square error. However, it is simple to find the *maximum likelihood estimator* (MLE) of an unknown parameter, and for large data records its performance is near optimal. Consequently, it is the favoured estimator for practical applications [15]. We define the likelihood function and the likelihood ratio.

Definition 20 (Maximum likelihood terminology) *Let $X(t)$ be a random variable with probability density $p(x;\theta)$, and let $x(t)$ be a realization of $X(t)$. The* maximum likelihood function *corresponding to the realization is*

$$L(x;\theta) = p(x;\theta) \tag{4.18}$$

The maximum likelihood estimate *of θ corresponding to the realization $x(t)$*

$$\theta^* = \arg\max_{\theta} L(x;\theta) \tag{4.19}$$

□

With our detection problem,

$$H_0 : x(t) = n(t) \tag{4.20}$$

$$H_1 : x(t) = s(t;\theta) + n(t), \tag{4.21}$$

we wish to find θ, by maximizing the likelihood function $L(x;\theta) = p(x|H_1;\theta)$. Since $p(x|H_0)$ is independent of θ we may instead maximize the likelihood ratio

$$\Lambda := \frac{p(x|H_1)}{p(x|H_0)}. \tag{4.22}$$

This following theorem establishes an important link between the matched filter output and this likelihood ratio.

Theorem 17 *The likelihood ratio and the matched filter output are related by*

$$N_0 \ln \Lambda = 2y(t_a + T_0) - E, \tag{4.23}$$

where $E = \|s\|^2$. □

This is proved in [36]. As a particular case of (4.19), we have maximum likelihood estimates as follows.

Theorem 18 (Maximum likelihood) *We find the following MLE estimates.*

- *The MLE of the amplitude as given by* $s(t;A) = As(t)$ *is given by*

$$\hat{A} = \frac{y(t+T)}{E}. \tag{4.24}$$

  The estimator is unbiased when we have white noise.

- *The MLE of the arrival time* t_0 *is given by the time when the matched filter output is at maximum;*

$$\hat{t}_0 = t_m \text{ where } \max_t y(t) = y(t_m). \tag{4.25}$$

□

The proof of this was given in the earlier CHESS-report [38], and will not be repeated here.

5 SYSTEM THEORY

System theory is a vast subject which has applications in many different areas. The purpose of this chapter is to give a very brief introduction to this field, and to establish a formal framework which can, in principle, describe the dynamical behaviour of a vibrating structure. This will result in a *state-space* formulation of the governing equations. We show how physical parameters such as damping and vibrational frequency can be extracted from these equations in a natural way. The main reason for our interest in the state-space formalism is that these equations may be estimated from measurements, as will be described in Chapter 6.

5.1 Introductory remarks

In the previous chapters we have described general signal processing techniques for detecting and characterizing the signals from strain sensors. However, our interest is not so much the signals themselves, but rather what can be inferred about the hull on the basis of the signals. Determining the properties of a system from measurements is generally referred to as system identification, which is an established discipline within system theory. During the present work it became clear that system theory might be the correct framework solving some of the unresolved problems in the CHESS-project.

- The signals we are trying to detect are not arbitrary, but represent the response of the hull to an applied load, and this response is governed by physical laws. Incorporating this fact into the methods we use to characterize the signals seems sensible.

  As a contrast, the output from the matched filter is merely the degree of similarity between the data and a reference transient. Apart from how the reference transient was constructed, it does not reflect the underlying physics in any obvious way. It turned out that in practice, the matched filter technique gave good results when trying to detect the strong and well known transients at about 2 Hz and 20 Hz. However, it was difficult to get unambiguous results when trying to find other characteristic vibrational frequencies. It was especially difficult to reliably distinguish between vibrations with almost the same frequencies. In fact, it is not at all clear what we really mean by assigning a frequency to a short transient signal. This motivated the search for complementary techniques.

- When using matched filters to characterize a signal, we run the data through a bank of matched filters, each filter being matched to transients with slightly different characteristics. In order to obtain good accuracy in the parameter estimates, we need to have many similar matched filters. This is a computationally intensive brute-force method, and one may well ask if it is possible to estimate parameters in a more elegant and efficient way.

- We have several sensors, so we will eventually end up with multivariable signals. The techniques we employ for detection and characterization should reflect and take advantage of this. While it is not obvious how this should be done in the matched filter formalism, multivariable systems is a well documented area in systems theory.

- At some later stage, we will need to find the optimal placement of the sensors on the hull. This calls for a systematic investigation of signals obtained from different sensors. This problem naturally fits within a system framework [11].

We have already used the term system, and many different definitions can be given. The essential aspect of a system is that it has a *state*. Given knowledge of this state at a time t_0 and subsequent input up to time $t > t_0$ should be enough to determine the state at time t. A general and axiomatic presentation is given in [34], and stochastic aspects are detailed in [4]. We restrict ourselves to shorter and less general approach which is more tailored to our particular application.

5.2 Linear systems

We will exclusively consider linear systems, and a classic reference on this topic is the book by Kailath, [13]. There are several reasons for concentrating on linear systems, and we mention some of them below.

- A linear system can be regarded as a linearization of an underlying non-linear system around a working point. Experience has shown that this approximation gives accurate results in many situations. In particular, we have reasons to believe that the most prominent vibrations in the hull and the panels are described by linear equations.
- Non-linearities and complex behaviour can generally be modelled by linear models of sufficiently high *order*.
- We are eventually interested in *estimating* the equations that govern our system. Since a variety of reliable algorithms for estimating, or *identifying*, linear systems exist, it is reasonable to consider a linear approximation, at least at an early stage. Other non-linear identification methods, making use of neural nets, fuzzy logic or adaptive algorithms are perhaps unnecessary complicated for our purpose.

Although non-linear and adaptive methods may be attractive in conjunction with automated damage control, it was decided that linear systems should first be examined.

5.2.1 Continuous time

We are only interested in systems can be represented in a *state space* form to be defined below. Therefore we take that as a definition of a system. Other, more abstract definitions exist, but they will not be considered here [34].

Definition 21 (Continuous-time linear time-invariant system) *A* continuous-time linear time-invariant system with outputs *can be represented by the following set of equations*

$$\sigma: \begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{cases} . \tag{5.1}$$

Here, $(A, B, C, D) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times n} \times \mathbb{R}^{p\times m}$. *We call* $x(t) \in \mathbb{R}^n$ *the* state *of the system at time* t, $u(t) \in \mathbb{R}^m$ *the* input *to the system at time* t *and* $y(t) \in \mathbb{R}^p$ *the* output *or measurements on the system at time* t. *The* order *of the system is given by* n, *and (5.1) is a* state space representation *of the system.* □

We see that the future state of the system, or rather, the rate of change of the state at time t, is determined by the current state $x(t)$ and the input. Thus future states are determined by input and the current state. The effect of $x(t)$ propagates through the system matrix A. Consequently, we say that the matrix A determines the dynamics of the system.

The state x often has some physical meaning, but it does not need to. Frequently, we are able to observe x only indirectly through the measurements y, which are modelled as a linear transformation of the state and the input u. The input u is an external influence acting on the system.

It is well known that the state space representation is not unique. That is, the measurements $y(t)$, may be generated by several configurations of state vectors. If $T \in \mathbb{R}^{n\times n}$ is an non-singular matrix, we may always define a new state vector by $z = Tx$. The system defined by

$$\sigma_T : \begin{cases} \dot{z}(t) &= TAT^{-1}z(t) + TBu(t) \\ y(t) &= CT^{-1}z(t) + Du(t) \end{cases} \tag{5.2}$$

is indistinguishable from σ with respect to its input-output behaviour, given by the observable quantities u and y. This can be verified by direct substitution of Tx for z. The main observation we make is that the matrix TAT^{-1} is a similarity transform of A. Consequently, A and TAT^{-1} have common eigenvalues (Appendix A.3). We say that the systems σ and σ_T are equivalent to within a similarity transform.

5.2.2 Discrete time

The definition of a discrete-time system is similar to the continuous-time case.

Definition 22 (Discrete-time linear time-invariant system) *A* discrete-time linear time-invariant system with outputs *can be represented by the following set of equations*

$$\sigma : \begin{cases} x_{k+1} &= \Phi x_k + \Lambda u_k \\ y_k &= Hx_k + Gu_k \end{cases} . \tag{5.3}$$

Here, $(\Phi, \Lambda, H, G) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times n} \times \mathbb{R}^{p\times m}$. *We call* $x_k \in \mathbb{R}^n$ *the* state *of the system at time* k, $u_k \in \mathbb{R}^m$ *the* input *to the system at time* k *and* $y_k \in \mathbb{R}^p$ *the* output *or measurements on the system at time* k. *The* order *if the system is given by* n, *and (5.3) is a* state space representation *of the system.* □

The same remarks concerning the non-uniqueness of the system and the effect of the system matrix Φ can be made as for the continuous time system. An illustration of a discrete-time system is given in Figure 5.1.

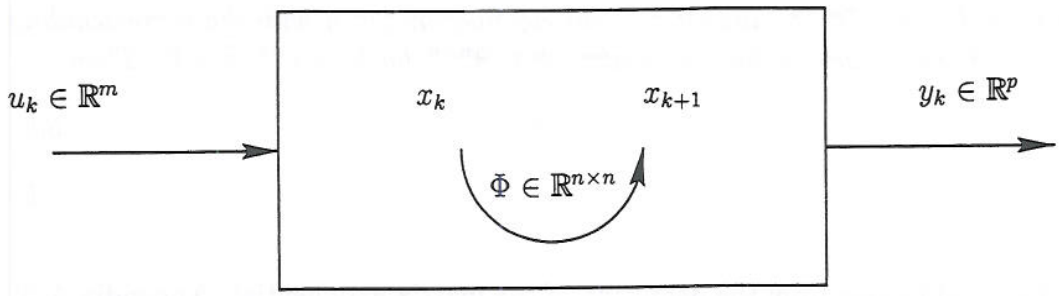


Figure 5.1 *A schematical representation of a linear discrete-time system. The next state x_{k+1} of the system is determined by the current input u_k and the current state x_k through the the state transition matrix Φ. The state x_k of the system cannot in itself be observed; the measurements, or outputs, of the system are found in the vector y_k.*

5.2.3 Relations between continuous and discrete time systems

When modelling a physical process from physical laws, one usually works with differential equations in continuous time. The resulting equations may often be written as a continuous-time system. However, when using digital control systems, the continuous-time system must be approximated by a discrete-time system. There are many ways of obtaining this discrete version, and this is a standard topic in any modern book on control engineering. We start by solving (5.1) for x. The solution is unique, and given by [34], [44]

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau, \tag{5.4}$$

where $x(t_0)$ is an initial condition. For comments on the matrix exponential e^{At}, see Appendix A.3. We consider a sampling interval δ and let $t_0 = \delta h$ and $t = (k+1)\delta$, and assume that the input $u(t)$ is constant on the sampling intervals, so $u(t) = u(\delta k) =: u_k$ for $t \in [k\delta, (k+1)\delta)$. Letting $x(k\delta) =: x_k$, we get

$$\begin{aligned} x_{k+1} &= x((k+1)\delta) && (5.5)\\ &= e^{A\delta}x(\delta k) + \int_{\delta k}^{\delta(k+1)} e^{A(\delta k-\tau)}Bu(\tau)d\tau && (5.6)\\ &= e^{A\delta}x_k + \int_{0}^{\delta} e^{A(\delta-\tau)}d\tau Bu_k && (5.7)\end{aligned}$$

Thus, we see that the continuous-time system with piecewise constant inputs has a discrete-time equivalent, with $\Phi = e^{A\delta}$ and $\Lambda = \int_0^\delta e^{A(\delta-\tau)}d\tau B$. Provided the sampling rate is sufficiently small, the assumption of u being piecewise constant is not very limiting.

It is well known that the eigenvalues of A determine the characteristic behaviour of the solutions $x(t)$. We are therefore interested in getting information about the eigenvalues of A, where A describes the response of the hull of our ship. We will do this by estimating a matrix Φ as if our system were discrete, and then use this matrix to extract information about the underlying continuous physical system.

Theorem 19 *Let $A \in \mathbb{R}^{n\times n}$, and let x be an eigenvector for A with the corresponding eigenvalue λ. Furthermore, define the matrix $\Phi \in \mathbb{R}^{n\times n}$ by $\Phi := e^{A\delta}$, $\delta > 0$. Then*

$$\Phi x = \lambda_\delta x \tag{5.8}$$

where $\lambda_\delta := e^{\lambda\delta}$. □

Proof The proof follows from the definition of the matrix exponential (Appendix A.3) and the fact that for $n = 0, 1, \ldots$, we have $A^n x = \lambda^n x$.

$$\Phi x = \left(\sum_{n=0}^{\infty} \frac{\delta^n A^n}{n!}\right) x \tag{5.9}$$

$$= \left(\sum_{n=0}^{\infty} \frac{\delta^n \lambda^n}{n!}\right) x \tag{5.10}$$

$$= e^{\delta\lambda} x \tag{5.11}$$

■

A slight problem is caused by the fact that we will go in a direction opposite from this theorem. We will first find an eigenvalue of Φ, and then find the corresponding eigenvalue λ for the continuous system. The problem is caused by the fact that if $\lambda_\delta = e^{\lambda\delta}$, then we also have, for all $n \in \mathbb{Z}$, $\lambda_\delta = e^{(\lambda + \frac{2\pi i n}{\delta})\delta}$. Thus, λ is not unique.

The non-uniqueness is resolved by introducing the principal value of the logarithm of complex numbers (Appendix A.1). Then we have $\lambda = \mathrm{Ln}\,(\lambda_\delta)/\delta$, or

$$\lambda = \frac{\ln|\lambda_\delta|}{\delta} + i\frac{\arg(\lambda_\delta)}{\delta}. \tag{5.12}$$

where $-\pi < \arg\lambda \leq \pi$. Restricting the argument to be between $-\pi$ and π may seem a bit contrived. However, this has a nice interpretation in terms of the Nyquist frequency, as will be noted in Section 5.4.2.

5.3 The Kalman filter

In practice, any system will be subjected to inputs that we cannot measure. The measurements of the output will also be subjected to noise. The Kalman filter is an efficient algorithm for estimating the state of a dynamical system, even in the presence of noise. It is fundamental in a wide variety of estimation problems, and it has proven its worth in practice.

For our purposes, it will suffice to consider only the discrete-time Kalman filter. Details can be found in [2], or more mathematical in [32]. We will consider the following extension of model (5.3), where we assume that the state and the measurements are disturbed by unmeasurable additive noise v_k and w_k, respectively.

$$\sigma: \begin{cases} x_{k+1} &= \Phi x_k + \Lambda u_k + v_k \\ y_k &= Hx_k + Gu_k + w_k \end{cases}. \tag{5.13}$$

We assume that the covariance matrix between v_k and w_k is independent of time and given by

$$\mathrm{E}\left[\begin{pmatrix} w_j \\ v_j \end{pmatrix} \begin{pmatrix} w_k^T & v_k^T \end{pmatrix}\right] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \delta_{j,k}. \tag{5.14}$$

Furthermore, it is assumed that the initial state is Gaussian distributed with mean $\mathrm{E}x_0 = m = \text{const}$ and covariance matrix R_0.

Definition 23 (Discrete-time Kalman filter) *Consider the state-space model (5.13). The Kalman filter estimate $\hat{x}_{k+1|k}$ of the state x_{k+1} at a time $k+1 \in \mathbb{Z}$ using the measurements and inputs $\{y_j, u_j; j \leq k\}$ up to time k is defined recursively by*

$$\hat{x}_{k+1|k} = \Phi\hat{x}_{k|k-1} + \Lambda u_k + K_k(y_k - H\hat{x}_{k|k-1} - Du_{k-1}) \tag{5.15}$$

$$\tilde{P}_{k+1|k} = \Phi\tilde{P}_{k|k-1}\Phi^T + Q - K_k(\Phi\tilde{P}_{k|k-1}\Phi^T + S)^T \tag{5.16}$$

$$K_k = (\Phi\tilde{P}_{k|k-1}H^T + S)(H\tilde{P}_{k|k-1}H^T + R)^{-1} \tag{5.17}$$

$$\tag{5.18}$$

where $P_{0|-1} = R_0$ is an initial condition. The matrix $\tilde{P}_{k|k-1}$ is the covariance of the estimation error $\tilde{x}_k := x_k - \hat{x}_{k|k-1}$,

$$\tilde{P}_{k|k-1} = \mathrm{E}\left[(\tilde{x}_k - \mathrm{E}\tilde{x}_k)(\tilde{x}_k - \mathrm{E}\tilde{x}_k)^T\right]. \tag{5.19}$$

The matrix (5.17) is the Kalman gain. □

We mentioned that the Kalman filter is an efficient method for estimating the state. We now give precise meaning to this statement.

Theorem 20 (Properties of the Kalman filter) *The Kalman filter is optimal in the sense that the Kalman filter state estimate $\hat{x}_k$ is the best linear unbiased estimator, since it minimizes the Bayesian mean square error*

$$\mathrm{E}\left[(x_{k+1} - \hat{x}_{k+1|k})(x_{k+1} - \hat{x}_{k+1|k})^T | \{y_j, u_j; j \leq k-1\}\right] \tag{5.20}$$

among all linear estimators. *Furthermore, when w_k and v_k are Gaussian noise processes, this is the minimum mean square error estimator among all possible estimators, and $\hat{x}_{k|k-1}$ is the minimum variance unbiased estimator.* □

The proof of this theorem is in [15]. The important point is that the Kalman filter is optimal for estimating the state when the noise is Gaussian, both in terms of minimum variance (MVUE) and mean-square. It is also robust, in the sense that it is the best linear estimate, regardless of the nature of the noise process.

We have discussed how to estimate the state, provided we know the system matrices Φ, Λ, H, G. We now turn to interpreting the state, before we in Chapter 6 show how both the state and the system matrices can be estimated from the input and the output.

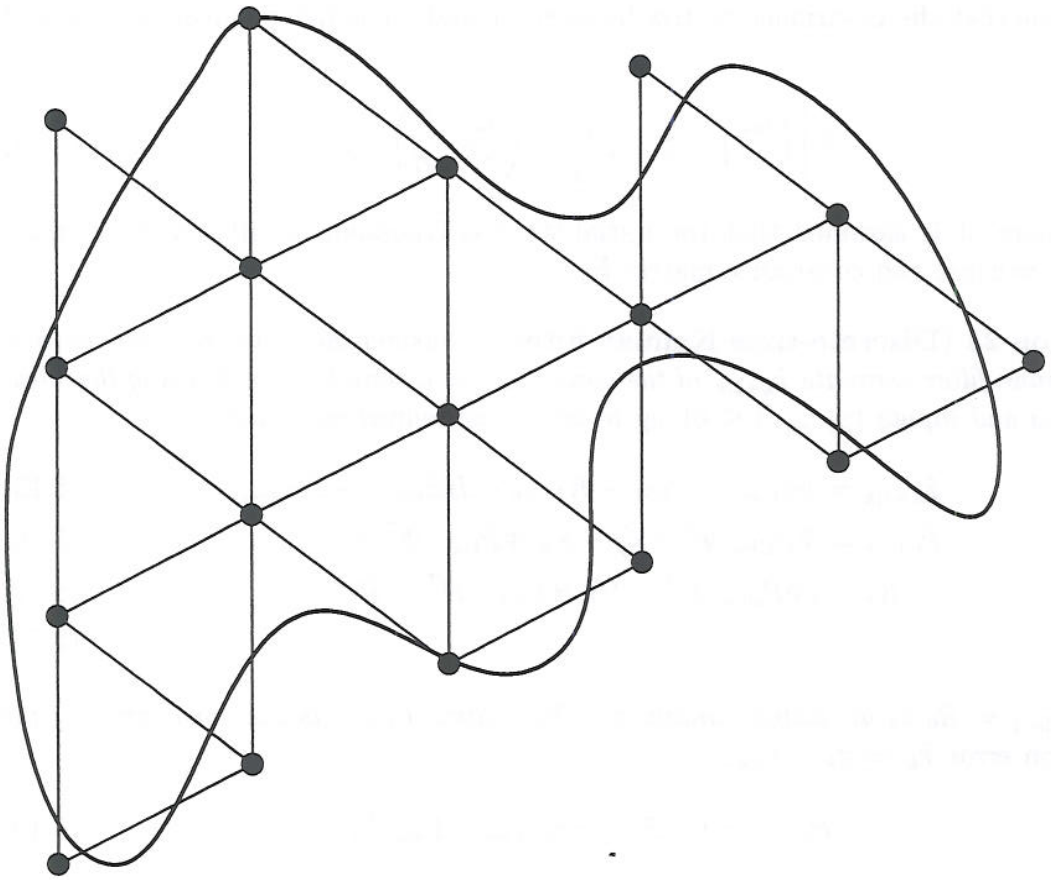


Figure 5.2 *A finite element discretization of a structure with complicated geometry using triangular elements.*

5.4 A state-space formulation of vibrational problems

5.4.1 Finite element method

The governing field equations for the response of structure to applied loads are in general well known. Unfortunately, analytic solutions to these equations can only be found for very simple loading mechanisms and geometric boundary conditions. Numerical methods are then called for, and the by far most popular method is the finite element method (FEM). In structural analysis, one usually discretisize the geometry of the structure using *finite elements* as in Figure 5.2. An approximate description of the original problem can be found by applying the governing equation to interpolating functions between each *node*, as indicated by a black dot in Figure 5.2. The details of this procedure need not concern us [23]. If we designate the displacements of node i in Figure 5.2 by $(v_{i,x}, v_{i,y}, v_{i,z})$, we may form a displacement vector

$$q = \left(v_{1,x}, v_{1,y}, v_{1,z} \ldots v_{k,x}, v_{k,y}, v_{k,z}\right)^T. \tag{5.21}$$

The finite element method is based on interpolating the displacements between these nodes in a special way. We may think of the method as approximating the behaviour of a continuous material region by discrete point-masses located at the nodes, connected with springs of different stiffness and subjected to damping. Essentially by Newton's second law, a set of coupled equations describing the motion of each node is then

obtained,

$$M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = f(t), \tag{5.22}$$

where the matrix M is called the mass matrix, C expresses damping and K models the stiffness. This is standard, and it is treated in much more detail in almost any book on the finite element method, e.g [23]. The end result is that the problem of solving the partial differential equations that govern the vibration is approximated by a set of coupled ordinary second order equations. In this way only the time-dependence, or dynamics, of the original problem is present, and the spatial derivatives are gone as a result of the discretization.

If the load is constant and the structure is in static equilibrium, there are no dynamics. The equation reduces to $Kq = f$, and this can be solved by standard techniques, e.g. Gauss-Jordan elimination.

We are interested in the dynamic response of the structure, and recast (5.22) into a state-space form as follows.

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{5.23}$$

where

$$A = \begin{pmatrix} 0 & I_n \\ -M^{-1}K & -M^{-1}C \end{pmatrix} \in \mathbb{R}^{2n\times 2n}, \tag{5.24}$$

$$x(t) = \begin{pmatrix} q(t) \\ \dot{q}(t) \end{pmatrix} \in \mathbb{R}^{2n} \tag{5.25}$$

$$B = \begin{pmatrix} 0 & 0 \\ 0 & I_n \end{pmatrix} \tag{5.26}$$

$$u(t) = \begin{pmatrix} 0 \\ f(t) \end{pmatrix} \tag{5.27}$$

We have assumed that the mass matrix M is invertible, and that condition can always be fulfilled by a proper discretization of the structure. Many other aspects concerning this state-space formulation and the validity of such a model under varying conditions can be found in the recent books [21] and [23]. For our purpose, it will suffice to know that it is in principle possible to consistently model our system by means of a state-space description. We will in this report not set up such a model from physical principles; instead we will estimate the model from measured data. We now describe how physical parameters may be recovered form the system matrix A.

5.4.2 Extraction of physical parameters

We now discuss the system (5.1). For concreteness, we may consider the state variables to be displacements an velocities of different points of structure, such as the nodes in Figure 5.2. This assumption is not necessarily correct, but it is introduced to ease the interpretation. Because the state vector x consists of both the displacements q and velocities $\dot{q}$, a model with n degrees of freedom corresponds to a state-space representation of order $2n$.

The system (5.1) is continually excited by the input $u(t)$. We investigate the behaviour of the system for $t > t_0$ if the input goes to zero for time $t > t_0$. We call this response the *unforced response* of the system.

Theorem 21 *Assume that the matrix A in (5.1) is real and has a set of $2n$ distinct eigenvalues $\mathcal{E} = \{\lambda_j\}_{j=1}^{2n}$, and let $2c$ of these be complex conjugates of each other. Without loss of generality, group complex conjugate eigenvalues in such a way that $\lambda_j = \bar{\lambda}_{c+j}$ so we can write*

$$\mathcal{E} = \{\lambda_j, \bar{\lambda}_j\}_{j=1}^{c} \cup \{\lambda_j\}_{j=2c+1}^{2n}. \tag{5.28}$$

Denote the jth component of the state vector $x(t)$ by $x_j(t)$, $j = 1 \ldots 2n$. The unforced response of the system for $t > t_0$ corresponding to the initial condition $x(t_0)$ can be written

$$\begin{aligned} x_j(t) = &\sum_{k=1}^{c} 2|z_{j,k}a_j(t_0)|e^{\alpha_k(t-t_0)}\cos(\beta_k t + \phi_j(t_0) + \theta_{j,k}) \\ &+ \sum_{k=2c+1}^{2n} z_{j,k}a_j(t_0)e^{\alpha_k(t-t_0)}. \end{aligned} \tag{5.29}$$

Here,

- *z_k is the eigenvector corresponding to λ_k, and $z_{j,k}$ is the jth component of z_k. Furthermore, $\phi_{j,k} := \operatorname{Arg} z_{j,k}$, so we may write $z_{j,k} = |z_{j,k}|e^{i\phi_{j,k}}$.*
- *$a_j(t_0) := w_j^T x(t_0)$, where w_j is the jth left eigenvector of A satisfying $A^T w_j = \lambda_j w_j$. Furthermore, $\theta_j := \operatorname{Arg}(w_j^T x(t_0))$, so we may write $a_{j,k} = |a_{j,k}|e^{i\theta_j}$.*
- *α_k and β_k are real and imaginary parts of λ_k, respectively, i.e. $\lambda_k = \alpha_k + i\beta_k$.*

□

Proof We are asked to solve $\dot{x} = Ax$ for $t > t_0$ given $x(t_0)$.

1. Since the $2n$ eigenvalues are distinct, we have $2n$ linearly independent eigenvectors of A. Denote these eigenvectors by z_j, and introduce a matrix of eigenvectors $Z = \begin{pmatrix} z_1 & z_2 & \cdots & z_{2n} \end{pmatrix}$. Similarly, introduce the matrix of left eigenvectors $W = \begin{pmatrix} w_1 & w_2 & \cdots & w_{2n} \end{pmatrix}$. We may then write

   $$W^T A Z = \Lambda, \text{ or } Z^{-1} A Z = \Lambda, \tag{5.30}$$

   where $ZW = WZ = I$ or $Z^{-1} = W$.

2. Since λ_i are distinct, we can diagonalize the matrix A, as $A = Z\Lambda W^T$. The matrix exponential is then $e^{A(t-t_0)} = Ze^{\Lambda(t-t_0)}W^T$. From (5.4) we can write the solution as

$$x(t) = Ze^{\Lambda(t-t_0)}c \tag{5.31}$$

$$= \left(\sum_{k=1}^{2n} z_k e^{\lambda_k(t-t_0)}\right) c, \tag{5.32}$$

where $c = W^T x(t_0)$.

3. We now look at each individual component $x_j(t)$ of $x(t)$. We write $\lambda_k =: \alpha_k + i\beta_k$, $k = 1, \ldots, 2n$, so that $\lambda_{k+c} =: \alpha_k - i\beta_k$, $k = 1, \ldots, c$. Also, the corresponding eigenvectors come as conjugated pairs, $z_k = \bar{z}_{c+k}$ and $w_k = \bar{w}_{c+k}$, $k = 1, \ldots, c$, since A is real.

   Furthermore, We let $z_{j,k}$ be component j of the kth eigenvector of A, and in polar form we write this number $|z_{j,k}|e^{i\phi_{j,k}}$. For the product between the jth left eigenvector w_j and $x(t_0)$ we introduce the polar form $|a_j(t_0)|e^{i\theta_j(t_0)} := w_j^T x(t_0)$.

$$x_j = \sum_{k=1}^{2n} z_{j,k} e^{\lambda_k(t-t_0)} w_j^T x(t_0) \tag{5.33}$$

$$= \sum_{k=1}^{2n} |z_{j,k}| e^{i\phi_{j,k}} e^{\lambda_k(t-t_0)} |a_j(t_0)| e^{i\theta_j(t_0)} \tag{5.34}$$

$$= \sum_{k=1}^{c} |z_{j,k} a_j(t_0)| e^{\alpha_k(t-t_0)} \left(e^{i[\beta_k(t-t_0)+\phi_{j,k}+\theta_j(t_0)]} + e^{-i[\beta_k(t-t_0)+\phi_{j,k}+\theta_j(t_0)]}\right)$$

$$+ \sum_{k=2c+1}^{2n} z_{j,k} a_j(t_0) e^{\alpha_k(t-t_0)} \tag{5.35}$$

$$= \sum_{k=1}^{c} |z_{j,k} a_j(t_0)| e^{\alpha_k(t-t_0)} 2\cos\left(\beta_k(t-t_0) + \phi_{j,k} + \theta_j(t_0)\right)$$

$$+ \sum_{k=2c+1}^{2n} z_{j,k} a_j(t_0) e^{\alpha_k(t-t_0)}. \tag{5.36}$$

■

The significance of this theorem is that we have have obtained expressions which describe the subsequent unforced real response of a system which is excited to a state $x(t_0)$. Furthermore, the motion consists of uncoupled damped sinusoids and exponentials, and these components can be analyzed separately. These considerations were inspired by Hoen [11], [12].

If the matrix A is symmetric, the eigenvalues and eigenvectors are real. We then have a straightforward interpretation of the eigenvectors, and this is treated in many books in physics or structural mechanics. However, it is rarely mentioned how the complex eigenvectors should be interpreted. Strang [10] has noted that

> I am not sure about economics, but physics and engineering and statistics are usually kind enough to produce symmetric matrices in their eigenvalue problems.

In many cases the matrix A will be non-symmetric, for example when A is estimated from data, and eigenvalues and eigenvectors will in general be complex. The present approach seems to provide some insight into the real dynamics of the system directly from the eigenvalues and the eigenvectors.

Remarks

- All the eigenvalues of A should satisfy $\Re(\lambda) < 0$, i.e. the response should be stable. If we think of the components of the state vector $x(t)$ to be the displacements of different parts of the structure, we expect an initial excitation to eventually die out.

  When estimating A and interpreting its eigenvalues, we will simply ignore any eigenvalues whose real part is non-negative on physical grounds.

- With the above condition, we have written the response as a series of damped sinusoids and damped exponentials. The imaginary part of the complex eigenvalues play the role of frequency, and the real part is the damping. Conventionally, we define the undamped frequency ω and the damping ratio ξ corresponding to the eigenvalue λ by the two equations $\Re(\lambda) = \alpha = \xi\omega$ and and $\Im(\lambda) = \beta = \sqrt{1-\xi^2}\omega$, where $\Re$ and $\Im$ denote real and imaginary parts, respectivly. We will not say anything about the interpretation and physical meaning of these parameters, but refer to [23].

  This interpretation does away with the somewhat artificial engineering concept of "instantaneous frequency". Instead, the frequencies are given by the imaginary part of the eigenvalues of the system matrix.

- Most importantly for our purpose, the response of the system is described by a series of decoupled equations in the phase space. If we are able to estimate the state at time t_0, we may recover the amplitude of the subsequent free response. Furthermore, we have isolated the contributions from each of the individual vibrational frequencies to the overall motion.

- When recovering the eigenvalues of the underlying continuous system from a discrete approximation, we restricted the continuous eigenvalues λ to have $\Im(\lambda) \in (-\pi/\delta, \pi/\delta]$, where δ is the sampling period. This means that when finding β_k from a discrete system, $\beta_k \in [0, \pi/\delta]$, which is less than or equal to the highest frequency we could estimate with a sampling frequency of $1/\delta$, by the Nyquist criterion.

6 SUBSPACE IDENTIFICATION

System identification has been defined as *the field of mathematical modelling of systems from experimental data.* [33]. We will restrict ourselves to dealing with linear models, even when we have reason to believe that the system is non-linear. The rationale for this was already mentioned in Section 5.2. The main purpose of this chapter is therefore to present methods for estimating the system matrices (A, B, C, D) from input-output measurements. In practice, the available measurements will be discrete, so we will actually estimate the discrete system matrices. Their continuous counterparts can be found using the results from Section 5.2.3.

We restrict ourselves to present the main results behind the identification schemes commonly called *subspace identification methods*, and do not discuss other identification methods. Proofs are generally not given, but can be found in the references.

6.1 Preliminaries

In this section we first state the problem we wish to solve. Then we present the geometrical and statistical tools we will be needing in developing a solution to this problem. It will be seen that the subspace methods can be given a geometrical interpretation, and this will be helpful in developing an intuitive understanding of the methods.

However, due to the large number of variables involved in the derivation and formulation of the theory, the subject may seem unnecessarily complicated. For this reason we have collected a number of preliminary results and notational definitions in this section.

6.1.1 Problem definition

We formally state the problem we will solve, and introduce some notation which will be used throughout this chapter.

Definition 24 (Subspace identification problem) *Let $\mathcal{S} \subset \mathbb{Z}$ be a time set, and assume that we have a set of measurements $\mathcal{M}_{\mathcal{S}} = \{(u_k, y_k) \in \mathbb{R}^m \times \mathbb{R}^p | k \in \mathcal{S}\}$ of the input u_k and the associated output y_k generated by the unknown system σ given by*

$$\sigma: \begin{cases} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Dy_k + v_k \end{cases}, \qquad \mathrm{E}\left[\begin{pmatrix} w_j \\ v_j \end{pmatrix} \begin{pmatrix} w_k^T & v_k^T \end{pmatrix}\right] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \delta_{j,k} \ . \tag{6.1}$$

Here, E is the expectation operator, and v_k and w_k are noise processes. We conceptually split the state x_k and the output y_k in a stochastic and a deterministic part indicated by superscript s and d, respectively, as follows:

$$x_k = x_k^d + x_k^s, \qquad y = y_k^d + y_k^s, \tag{6.2}$$

where the different parts of the state and output satisfy the deterministic and stochastic subsystems σ^d and σ^s given by

$$\sigma^d: \begin{cases} x_{k+1}^d &= Ax_k^d + Bu_k \\ y_k^d &= Cx_k^d + Dy_k^d \end{cases} \qquad \sigma^s: \begin{cases} x_{k+1}^s &= Ax_k^s + w_k \\ y_k^s &= Cx_k^s + v_k \end{cases} . \tag{6.3}$$

On the basis of the measurements $\mathcal{M}_S$, we wish to determine

- *The order n of the system,*
- *The system matrices $(A, B, C, D) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times n} \times \mathbb{R}^{p\times m}$ to within a similarity transform,*
- *The spectral density matrices $(Q, R, S) \in (\mathbb{R}^{n\times n} \times \mathbb{R}^{p\times p} \times \mathbb{R}^{n\times p})$ associated with the noise processes w_k, v_k such that the output that would result from applying a constant zero input $u_k \equiv 0$ to the system σ has the same second order statistics as y_k^s.*

□

We restrict ourselves to finding only those matrices which are relevant to the stochastic system σ^s. The deterministic problem can be solved in a similar way, but refer the reader to [40] for more thorough treatment than can be given here.

6.1.2 Block Hankel matrices

Matrices with Hankel structure occur frequently in system theory. This section will establish some notation concerning these matrices.

Definition 25 (Input and output block Hankel matrices) *The* input block Hankel matrix *of order j corresponding to an output sequence $\{y_k\}_k$, $y_k \in \mathbb{R}^m$ is defined as*

$$Y_{0:2i-1} := \left(\begin{array}{ccccc} y_0 & y_1 & y_2 & \cdots & y_{j-1} \\ y_1 & y_2 & y_3 & \cdots & y_j \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{i-1} & y_i & y_{i+1} & \cdots & y_{i+j-2} \\ \hline y_i & y_{i+1} & y_{i+2} & \cdots & y_{i+j-1} \\ y_{i+1} & y_{i+2} & y_{i+3} & \cdots & y_{i+j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{2i-1} & y_{2i} & y_{2i+1} & \cdots & y_{2i+j-2} \end{array}\right) = \left(\begin{array}{c} Y_{0:i-1} \\ \hline Y_{i:2i-1} \end{array}\right). \tag{6.4}$$

The line drawn between block rows $i-1$ and i in the first matrix indicate the partitioning of $Y_{0:2i-1}$ into $Y_{0:i-1}$ and $Y_{i:2i-1}$ as shown. □

For heuristic reasons, we refer to this partitioning as splitting the matrix into *future outputs* Y_f and *past outputs* Y_p.

$$Y_{\mathrm{p}} := Y_{0:i-1} \qquad Y_{\mathrm{f}} := Y_{i+1:2i-1} \tag{6.5}$$

The dependence on the parameters i and j are in all cases implicitly understood. For notational convenience, we also defined Y_p^+ and Y_f^-, two matrices obtained by shifting the delimiter in the partitioning one block row down;

$$Y_{\mathrm{p}}^+ := Y_{0:i} \qquad Y_{\mathrm{f}}^- := Y_{i:2i-1} \tag{6.6}$$

Thus, the matrix $Y_{0:2i-1}$ may be partitioned in a number of ways;

$$Y_{0:2i-1} = \begin{pmatrix} Y_{0:i-1} \\ \hline Y_{i:2i-1} \end{pmatrix} = \begin{pmatrix} Y_p \\ \hline Y_f \end{pmatrix} = \begin{pmatrix} Y_p^+ \\ \hline Y_f^- \end{pmatrix} = \begin{pmatrix} Y_{0:i} \\ \hline Y_{i+1:2i-1} \end{pmatrix}. \tag{6.7}$$

6.1.3 Oblique projections

In statistical signal processing, it often proves fruitful to regard signals as vectors in a signal space. Removing certain parts (e.g. noise) from a data set can sometimes be seen as projecting the data onto a signal space which is orthogonal to the space containing the unwanted elements. We have already seen examples of this in Chapter 4. Now we introduce projections as a linear algebraic concept for use in a system identification setting.

Definition 26 (Orthogonal projections) *The matrix that orthogonally projects the row space of a matrix onto the row space of another matrix $B \in \mathbb{R}^{q \times j}$ is given by*

$$\Pi_B := B^T(BB^T)^{\#}B, \tag{6.8}$$

where $\#$ denotes the Moore-Penrose pseudoinverse. We introduce the notation

$$A/B := A\Pi_B \tag{6.9}$$

for the projection of the row space of $A \in \mathbb{R}^{p \times j}$ onto the row space of B. □

The pseudoinverse is reviewed in Appendix A.3. If we let $B^\perp$ denote a matrix whose rows span the orthogonal complement of the row space of B, we can then express the matrix A as a linear combination

$$A = A\Pi_B + A\Pi_{B^\perp}. \tag{6.10}$$

At the heart of subspace identification as presented here lies a more general projection, the *oblique projection*, which will be defined below. It will not be needed in the following, but it is needed when estimating the deterministic system σ^d. Briefly, it is that part of the orthogonal projection of the row space of A onto the joint row space of B and C which lies in the row space of C. An illustration of these two projections is given in Figure 6.1. The orthogonal projection on the joint row space of B and C is given from (6.8);

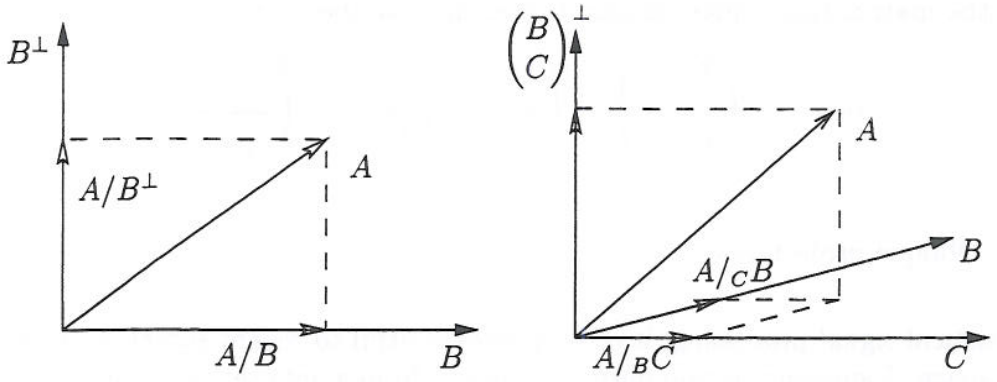
$$A/\begin{pmatrix} B \\ C \end{pmatrix} = A\begin{pmatrix} C \\ B \end{pmatrix}^T \left[\begin{pmatrix} CC^T & CB^T \\ BC^T & BB^T \end{pmatrix}^{\#} \right] \begin{pmatrix} C \\ B \end{pmatrix}.$$

Of this, we need to take the part in the row space of C, as follows.

Definition 27 (Oblique projections) *The* oblique projection *of the row space of $A \in \mathbb{R}^{p \times j}$ along the row space of $B \in R^{j \times r}$ onto the row space of $C \in \mathbb{R}^{r \times j}$ is defined as*

$$A/_B C = A\begin{pmatrix} C \\ B \end{pmatrix}^T \left[\begin{pmatrix} CC^T & CB^T \\ BC^T & BB^T \end{pmatrix}^{\#} \right]_{\textit{first } r \textit{ columns}} C. \tag{6.11}$$

Thus $A/_B C \in \mathbb{R}^{r \times j}$ lies in the row space of C. □



(a) Schematical illustration of the orthogonal projection A/B of the row space of the matrix A on the row space of B. The orthogonal projection of A on the orthogonal complement of the row space of B is denoted $A/B^\perp$.

(b) Schematical illustration of the oblique projection $A/_B C$ of the row space of the matrix A along on the row space of B onto the row space of C.

Figure 6.1 Oblique and orthogonal projections.

6.1.4 Statistical analysis by means of projections

The projections we just defined are very useful in a statistical setting, if our data series are *ergodic*.

Definition 28 (Ergodicity) *The random variables* $\{X_k\}_{k\in\mathbb{Z}}$ *are said to be* ergodic *if*

$$\mathrm{E}[X_k] = \mathrm{E_j}\left[\sum_{k=0}^{j} X_k\right], \tag{6.12}$$

where we have introduced the expectation operator E *and the* averaging operator

$$\mathrm{E_j}[\cdot] := \lim_{j\to\infty} \frac{1}{j}[\cdot]. \tag{6.13}$$

□

The mean of X_k for a fixed k may be regarded as an average over all possible outcomes of the process at time k. The assumption of ergodicity implies that we equivalently may consider a single outcome, averaged over all times k.

The averaging operator allows us to treat stochastic and deterministic systems with much of the same formalism [33]. In practice, we will always have only a finite number of samples j available, so we have to approximate the averaging operator by the sample mean. We now define the covariance between matrices whose columns are ergodic processes. We will use the expectation operator on the columns, so the number of columns can be regarded as going towards infinity. However, this does not pose any significant problems, since the expectation operator will only be applied to products of semi-infinite matrices. These products will always have finite dimensions, and the convergence is guaranteed because of the assumption of ergodicity.

Definition 29 (The covariance between two matrices) *The covariance matrix between the two stochastic matrices $A \in \mathbb{R}^{p\times j}$ and $B \in \mathbb{R}^{q\times j}$ is defined as*

$$R_{[A,B]} := \mathrm{E_j}\left[AB^T\right], \tag{6.14}$$

If the matrices are finite dimensional, we will use the estimator

$$\hat{R}_{[A,B]} := \frac{1}{j}AB^T. \tag{6.15}$$

□

The connection between covariance and projections will now be made in the following (re-) definition of the the projections.

Definition 30 (Stochastic oblique projections) *Let $A \in \mathbb{R}^{p\times j}$, B and $C \in \mathbb{R}^{r\times j}$ be matrices whose elements are a samples from a realization of a stochastic processes. The* oblique projection *of the row space of A along the row space of B onto the row space of C is defined as*

$$A/_B C := \begin{pmatrix} R_{[A,C]} \\ R_{[A,B]} \end{pmatrix}^T \left[\begin{pmatrix} R_{[C,C]} & R_{[C,B]} \\ R_{[B,C]} & R_{[B,B]} \end{pmatrix}^{\#} \right]_{\textit{first } r \textit{ columns}} C. \tag{6.16}$$

The orthogonal projection of A onto the row space of B is defined as

$$A/B := R_{[A,B]}R^{\#}_{[B,B]}B. \tag{6.17}$$

□

For finite dimensional matrices, we use the estimator (6.15) for R. We write R for both R and $\hat{R}$, because we will always have only a finite number of data, so no confusion should arise. Both projections then reduce to the ordinary, non-stochastic version (6.11) as follows,

$$\begin{aligned} A/B &= R_{[A,B]}R^{\#}_{[B,B]}B && (6.18)\\ &= \left(\frac{1}{j}AB^T\right)\left(\frac{1}{j}BB^T\right)^{\#}B && (6.19)\\ &= \frac{1}{j}AB^T j\left(BB^T\right)^{\#}B && (6.20)\\ &= AB^T\left(BB^T\right)^{\#}B, && (6.21) \end{aligned}$$

which is equivalent to (6.8).

6.1.5 Notation

We the define the notation we will use in the main results in the next section.

Definition 31 (Notation and assumptions) *We use the following notation and assumptions.*

- *A subscript i on a matrix usually indicates the number of block columns or rows, and quantities relating to the stochastic subsystem σ^s or the deterministic subsystem σ^d are marked with a superscript s or superscript d, respectively.*

- *We define the following covariance matrices*

$$\Sigma^s := \mathrm{E}[x_{k+1}x_{k+1}^T], \qquad \Lambda[j] := \mathrm{E}[y_{k+j}y_k^T], \qquad G := \mathrm{E}[x_{k+1}^s y_k^T] \tag{6.22}$$

We call Σ^s the stochastic state covariance matrix and $\Lambda[k]$ the output covariance matrix.

- *We assume that x^s is stationary, so*

$$\mathrm{E}[x_k^s] = 0, \qquad \mathrm{E}[x_k^s(x_k^s)^T] := \Sigma^s, \tag{6.23}$$

where Σ^s is independent of the time k.

- *The state is uncorrelated with the noise processes;*

$$\mathrm{E}[x_k^s w_k^T] = 0, \qquad \mathrm{E}[x_k^s v_k^T] = 0, \tag{6.24}$$

- *Δ^{rs}, the reversed stochastic controllability matrix is defined by*

$$\Delta_i^{rs} = \begin{pmatrix} A^{i-1}G & A^{i-1}G & \cdots AG & G \end{pmatrix} \tag{6.25}$$

- *Γ_i, the observability matrix is defined as*

$$\Gamma_i = \begin{pmatrix} B \\ BA \\ BA^2 \\ \vdots \\ BA^{i-1} \end{pmatrix} \in \mathbb{R}^{pi \times n}. \tag{6.26}$$

- *We define two block Toepliz matrices which are constructed from the output covariance matrices Λ_i.*

$$L_i := \begin{pmatrix} \Lambda_0 & \Lambda_{-1} & \Lambda_{-2} & \cdots & \Lambda_{1-i} \\ \Lambda_1 & \Lambda_0 & \Lambda_{-1} & \cdots & \Lambda_{2-i} \\ \Lambda_2 & \Lambda_1 & \Lambda_0 & \cdots & \Lambda_{3-i} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Lambda_{i-1} & \Lambda_{i-2} & \Lambda_{i-3} & \cdots & \Lambda_0 \end{pmatrix}, \tag{6.27}$$

$$C_i := \begin{pmatrix} \Lambda_i & \Lambda_{i-1} & \Lambda_{i-2} & \cdots & \Lambda_1 \\ \Lambda_{i+1} & \Lambda_i & \Lambda_{i-1} & \cdots & \Lambda_2 \\ \Lambda_{i+2} & \Lambda_{i+1} & \Lambda_i & \cdots & \Lambda_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Lambda_{2i-1} & \Lambda_{2i-2} & \Lambda_{2i-3} & \cdots & \Lambda_i \end{pmatrix} \tag{6.28}$$

□

The quantities defined above satisfy the following standard relationships, which are starting points for the development of identification algorithms.

Theorem 22 (Properties) *We have the following*

- *The covariance matrices of the different processes satisfy*

$$\Sigma^s = A\Sigma^s A^T + Q \qquad G = A\Sigma^s C^T + S \tag{6.29}$$

$$\Lambda_0 = C\Sigma^s C^T + R; \qquad \Lambda_j = \begin{cases} CA^{j-1}G, & j = 1, 2, \dots \\ (CA^{j-1}G)^T, & j = -1, -2, \dots \end{cases} \tag{6.30}$$

- *We can find Λ_i by*

$$\Lambda_i = \mathrm{E}_j \left[\sum_{k=0}^{j-1} y_{k+i} y_k^T \right] = R_{[Y_{i|i}, Y_{0|0}]}, \tag{6.31}$$

where $Y_{i|i}$ and $Y_{0|0}$ are output block Hankel matrices. The block Toepliz matrices L_i and C_i can be estimated as

$$L_i = R_{[Y_f, Y_f]} = R_{[Y_p, Y_p]} \quad \textit{and} \quad C_i = R_{[Y_f, Y_p]}. \tag{6.32}$$

□

The simple proofs of this can be found in [40].

6.2 Stochastic identification

Our goal is to find a way to recover the system matrices from the input/output data $\mathcal{M}_S$. This section will show how that can be done.

6.2.1 Kalman filter state sequences

We will use a special form of the Kalman filter to estimate the state sequence, and it is elaborated in Appendix B.5. Briefly, it will split the *error covariance matrix* $\tilde{P}_k$ into the the state estimate covariance matrix Σ^s and another matrix P_k;

$$\tilde{P}_k = \Sigma^s - P_k \tag{6.33}$$

We state a theorem which gives us a relation between input-output data and the Kalman filter state estimates.

Theorem 23 (Kalman filter states) *Consider the system (6.1), and let*

$$P_k := \Sigma^s - \tilde{P}_k,$$

where $\Sigma^s = \mathrm{E}[x_k x_k^T]$ and the state error $\tilde{P}_k = \mathrm{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$. The equations

$$\hat{x}_{k+1} = A\hat{x}_k + K_{k-1}(y_k - C\hat{x}_k) \tag{6.34}$$

$$P_{k+1} = AP_kA^T + K_k(G - AP_kC^T) \tag{6.35}$$

$$K_k = (G - AP_kC^T)(\Lambda_0 - CP_kC^T)^{-1} \tag{6.36}$$

$$\tag{6.37}$$

define a Kalman filter for estimating the states by $\hat{x}_k$. *Given the initial estimates* $\hat{x}_0 = 0$, $P_0 = 0$ *and the data* y_k, $k = 0, 1, \ldots, k-1$ *the Kalman filter state estimate* $\hat{x}_k$, $k > 0$ *is*

$$\hat{x}_k = \Delta_k^{rs} L_k^{-1} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{k-1} \end{pmatrix} \tag{6.38}$$

The covariance matrix P_k *can be found from*

$$P_k = \Delta_k^{rs} L_k^{-1} (\Delta_k^{rs})^{-1} \tag{6.39}$$

□

This Kalman filter looks slightly different from the conventional one, but this is only due to the notation. We show in Appendix B.5 that the the present Kalman filter is equivalent to the more conventional one.

By forming a vector of j consecutive state estimates we obtain an estimated state *sequence* $\hat{X}_i$ from (6.38) as follows:

$$\hat{X}_i := \begin{pmatrix} \hat{x}_i & \hat{x}_{i+1} & \cdots & \hat{x}_{i+j-1} \end{pmatrix} \tag{6.40}$$

$$= \Delta_i^{rs} L_i^{-1} Y_p \tag{6.41}$$

As before, the dependence on the number of block columns j is implicit throughout.

Thus, we may estimate the states from input-output data, if we already know the system matrices A, B, C, D, Q, R, S. We will now see that the the Kalman filter states estimates can in be estimated directly from the input-output data set $\mathcal{M}_S$, without knowing the system matrices. We will then use the estimated states to find the system matrices, which is our final goal.

6.2.2 The stochastic identification theorem

We are now ready to state a theorem which will provide a recipe for computing state estimates, the system order and the observability and controllability matrices. From these, the system matrices themselves can be extracted.

Theorem 24 *We make the following assumptions:*

1. *The process noise* w_k *and measurement noise* v_k *are identically zero;*
2. *The number of measurements goes to infinity,* $j \to \infty$;
3. *The arbitrary weighting matrices* W_1 *and* W_2 *are such that* W_1 *is if full rank and* $\text{rank} W_p = \text{rank}(Y_p W_2)$, *where* W_p *is the combined input-output block Hankel matrix.*
4. *The matrix* $T \in \mathbb{R}^n$ *is any non-singular similarity transformation of the system;*

Defining the orthogonal projection

$$\mathcal{O}_i := Y_f / Y_p \tag{6.42}$$

and the SVD of $W_1 \mathcal{O}_i W_2$ *as*

$$W_1 \mathcal{O}_i W_2 := \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}, \tag{6.43}$$

we then have have:

1. *The projection* $\mathcal{O}_i$ *in (6.42) equals the the observability matrix multiplied with* $\hat{X}_i$*;*

$$\mathcal{O}_i = \Gamma_i \hat{X}_i. \tag{6.44}$$

2. *The order of the system is equal to the number of singular values in (6.43).*

3. *The extended observability matrix* Γ_i

$$\Gamma_i = W_1^{-1} Q_1 S_1^{1/2} T \tag{6.45}$$

$$\Delta_i^{rs} = \Gamma^{\#} R_{[Y_f, Y_p]}. \tag{6.46}$$

4. *The projection of the state sequence* $\hat{X}_i$ *onto the column space of* W_2 *can be recovered from*

$$\hat{X}_i W_2 = T^{-1} S_1^{1/2} V_1^T \tag{6.47}$$

5. *The state sequence* $\hat{X}_i$ *can be found from*

$$\hat{X}_i = \Gamma_i^{\#} O_i. \tag{6.48}$$

□

why these results are true. Proofs and further discussion concerning these results can be found in [39] or [40]

6.2.3 Extraction of the system matrices

Our main goal is not to find the Kalman state sequence (6.48), but rather the system matrices describing the dynamical properties of the system. We present one simple algorithm based on Theorem 24, and comment upon some of the steps below. Other, more sophisticated algorithms are possible, see [40].

Algorithm 3 (Extraction of system matrices)

1. *Calculate the orthogonal projections*

$$\mathcal{O}_i = Y_f / Y_p \quad \text{and} \quad \mathcal{O}_{i-1} = Y_f^- / Y_p^+ \tag{6.49}$$

2. *Calculate the singular value decomposition of the oblique projection*

$$W_1 \mathcal{O}_i W_2 = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}^T. \tag{6.50}$$

3. *Determine the order n of the system by deciding on the number of "significant" singular values. Collect the n significant SVD in the diagonal of the matrix S_1, and partition the SVD into* signal *and* noise *submatrices as indicated in (6.50).*

4. *Determine the observability matrices by*

$$\Gamma_i = W_1 Q_1 S_1^{1/2}, \quad \Gamma_{i-1} = \underline{\Gamma_i}, \tag{6.51}$$

   where $\underline{\Gamma_i}$ has the last p (number of outputs) rows removed.

5. *Determine the sequences of Kalman state estimates*

$$\hat{X}_i = \Gamma_i^{\#} \mathcal{O}_i \tag{6.52}$$

$$\hat{X}_{i+1} = \Gamma_{i-1}^{\#} \mathcal{O}_{i+1}. \tag{6.53}$$

6. *In the least squares sense, solve*

$$\begin{pmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A \\ C \end{pmatrix} \hat{X}_i, \tag{6.54}$$

   to get

$$\begin{pmatrix} A \\ C \end{pmatrix} = \begin{pmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{pmatrix} \hat{X}_i^{\#}. \tag{6.55}$$

□

The SVD in (6.50) mirrors the one in (6.43). However, with a finite number of data, there will be non-zero singular values in S_2. Thus, we must decide on some arbitrary threshold which determines whether a singular value is "significant" or not. In step 4, the observability matrix Γ_i is then found as in (6.45), and we find the Kalman state estimates in step 5. In the last step, the system matrices are found. We do not elaborate further on these aspects, but refer to [40] instead. In the next chapter, we give examples of how the estimated matrix A can be used to extract information about the underlying continuous-time system.

7 ANALYSIS OF CHESS DATA

The theory described in the previous chapters has been implemented and tested both in C++ and MATLAB. The main part of the present work has been to develop wavelet based techniques capable of analyzing large amounts of data. Despite this fact, the larger part of this chapter is dedicated to presenting results from system identification. The reason for this is that we by now have some experience with wavelet analysis, but the system identification approach is new in the CHESS-project.

The intention with this chapter is not to present a detailed analysis of the data, but rather to demonstrate how the methods perform on actual experimental data. A detailed analysis of the data is outside the scope of this report.

7.1 Short description

The CHESS-project and the signal processing system was described in Chapter 2. In Section 7.2 we show how subspace identification methods can be used to obtain estimates of vibrational frequencies, damping and the number of freedoms needed to characterize a structure. In Section 7.3 we mention some typical results that can be obtained with the programs developed (Appendix D).

7.2 Analysis by subspace identification

In the following, we illustrate how it is possible to extract information about the structural behaviour of the hull using subspace methods. The purpose is not to do a very thorough analysis of the data, but rather to illustrate how the methods can be used in practice. The topics we wish to illustrate are summarized below.

- *Identification*
  We identify vibrational frequencies and damping ratios for different modes.
- *Stability*
  We investigate how the modal parameters vary with respect to both *time* and *order* of the model we are using.
- *Statistics*
  The need to have statistics describing the natural variation of the estimated parameters interpreting the data is discussed. We show how the accuracy of the estimates can be characterized by suitable statistical measures.

7.2.1 A single short time series

We first analyze the data shown in Figure 7.1. This data set is is convenient to work with because it contains several strong transients. From the spectral density plot, we identify strong components around 2.2 Hz, corresponding to a long-ship vibration.

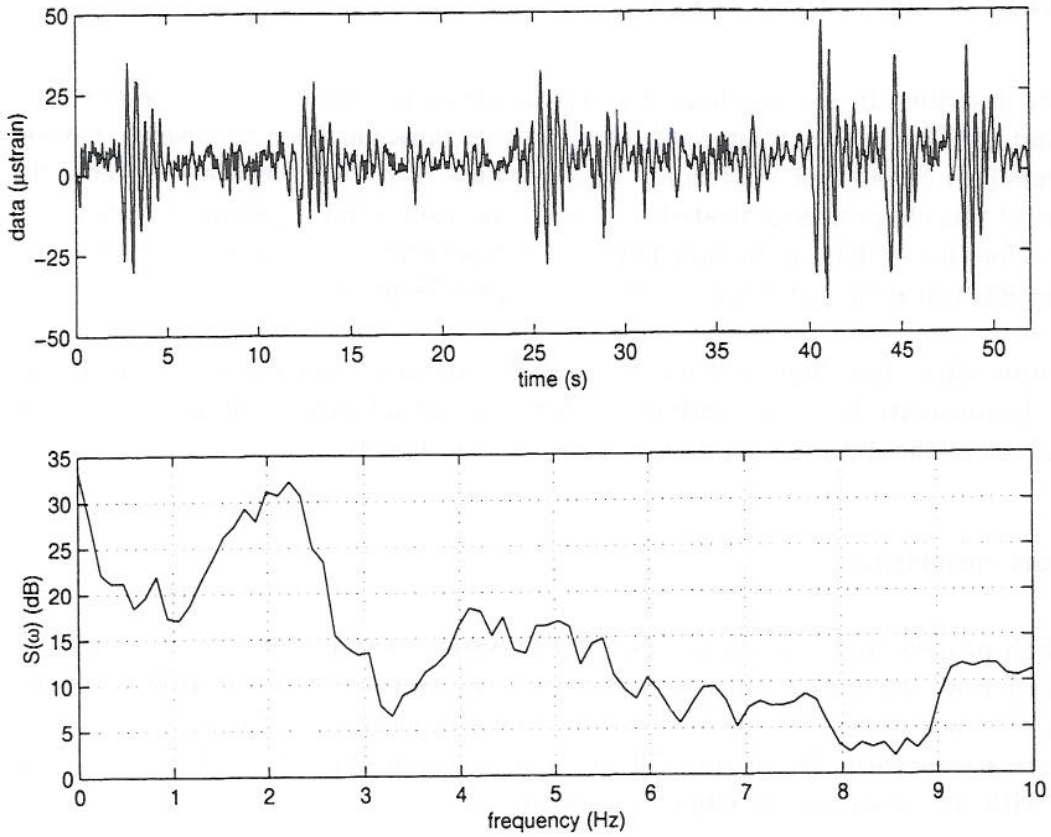


Figure 7.1 *Data containing several strong 2 Hz transients (top) and its estimated spectral density (bottom).*

To use subspace identification, we need to find the *order* n of the discrete system matrix Φ. As described in Chapter 6, n can be estimated by the number of singular values different from zero. With finite data sets corrupted by noise, we do not expect the singular values to be exactly zero, so we take n to be the number of singular values greater than some lower limit. Applying the subspace identification methods to the data in Figure 7.1, we calculate the singular value decomposition (6.50). We have plotted 150 sorted singular values in Figure 7.2 which illustrates that the choice of model order in this case is not a trivial matter. By examining the ratios of consecutive singular values, we notice that there is a relatively large reduction in the magnitudes of the singular values between the third and fourth singular value. The same is true for the sixth and seventh. We expect that the model order has to be least 10 to capture most of the dynamical behaviour. As discussed in Section 5.4.1, this correspond to five degrees of freedom.

We conclude that for complex structures, one cannot expect the model order to be clearly apparent from the singular values alone. Fortunately, it turns out that the exact choice of model order is not very critical.

In Figure 7.3 we have plotted the estimated structural eigenfrequencies for orders in the range $[1, 20]$. In this and in the following plots, eigenfrequencies are estimated by first estimating a discrete-time system matrix, and then taking the imaginary part of the eigenvalues of the underlying continuous-time matrix, as discussed in Chapter 5. The units are Hertz, that is, they have been divided by 2π rad. Several comments

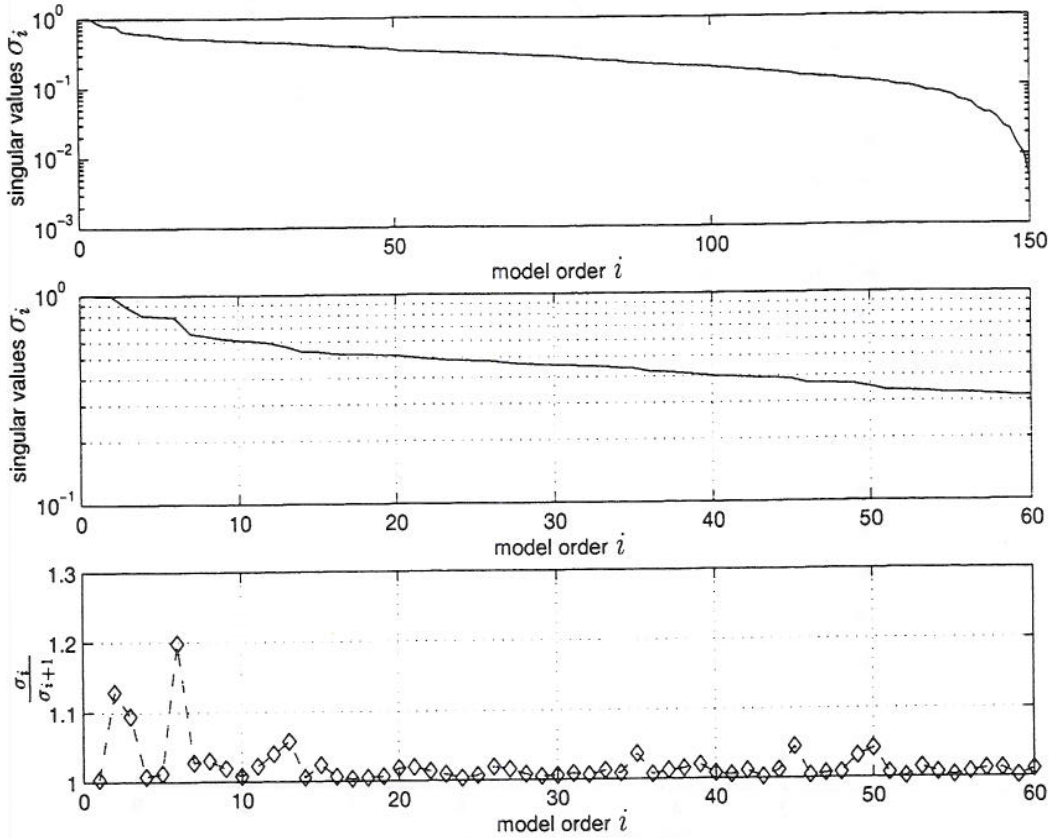


Figure 7.2 *Singular values σ_i for the data in Figure 7.1, sorted after magnitude. The lower plot shows the relative magnitude of consecutive eigenvalues σ_i.*

can be made in conjunction with this plot. We note first of all that it is consistent with the power spectral density in Figure 7.1. By this we mean that peaks appearing in the spectral density plot also show up as eigenfrequencies in Figure 7.3. At higher orders more and more detail are included. For example, the two highest frequency peaks below 1 Hz in Figure 7.1 does not show up in Figure 7.3 at orders less than $n = 9$. The frequency which is identified at the lowest order ($n = 3$) is at around 2 Hz. There are also two "vibrational modes" at less than 1 Hz which appear to be stable at increasing model order. These show up as small peaks in the spectral density plot, and can probably be identified with the load process. Therefore, they do not really represent *structural modes*, but rather characteristics of the load process. It is clearly helpful to have physical knowledge of the process when interpreting such data.

The *stability with respect to model order* is illustrated in Figure 7.4. The estimate is quite close at model order 3 and remains stable for very large model orders. This is reassuring because it indicates that the identified frequency is not a result of just the choice of model order. Increasing the model order results in new state variables wich are fitted to the noise and load processes. One might regard this as increasing the dimension of the state-space, and moving the noisy parts of the data away from the subspaces which correspond to structural modes. The state variables which describe physical modes should therefore remain stable. These considerations are also discussed in [11] and [19].

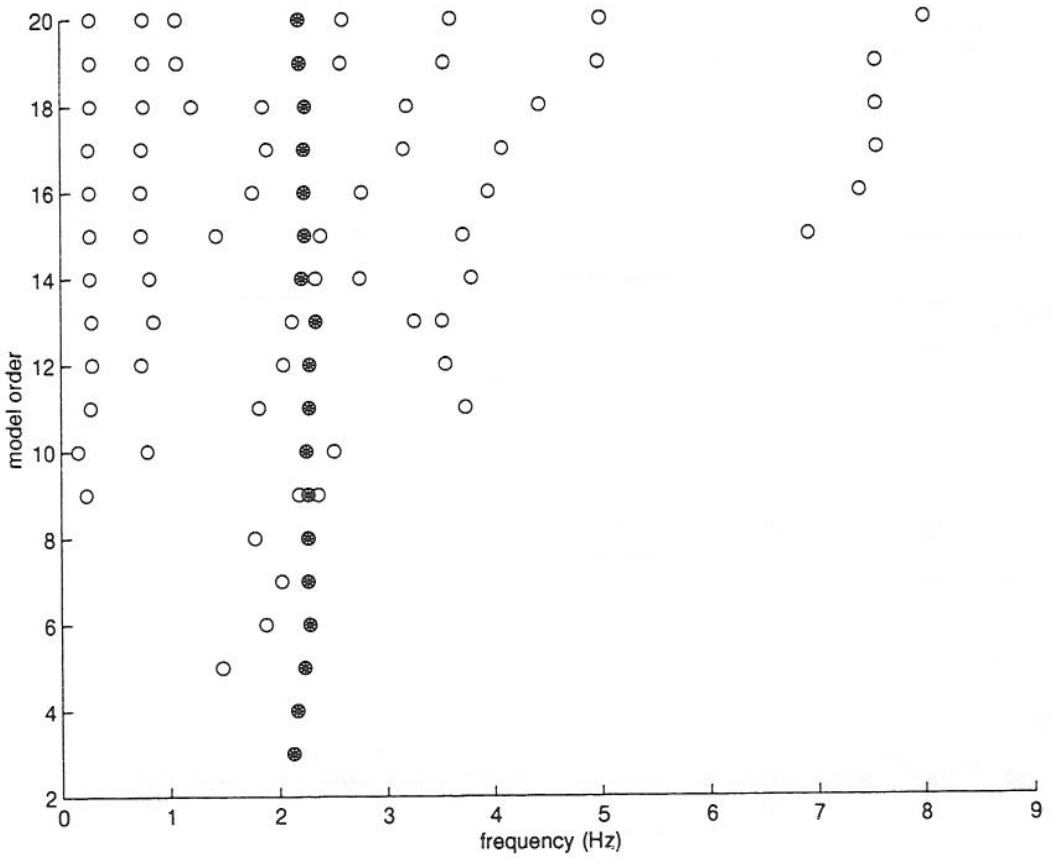


Figure 7.3 *Estimated damped vibrational frequencies at varying model orders. The asterices indicate the mode corresponding to vibrations at 2.2 Hz.*

We also note that there are in Figure 7.3 some higher-frequency estimates which are drifting as the order increases. Increasing the model order up to 60, we see that most of these estimates reaches a more or less steady state, as in Figure 7.5. However, it is now even less clear which of these modes correspond to structural modes, and which appear only as a result of this special data set. To answer this, one would have to analyse other data sets and see which modes frequently show up as stable modes.

The subspace methods can be used even when strong transients, as in Figure 7.1 are not present. We will illustrate this by considering the measurements shown in Figure 7.6. This data set signal is from the L sensor (Section 2.2.4), and it is filtered to avoid aliasing and downsampled from 600 to 200 Hz. It turns out that even with such a short data set, we will be able to identify several vibrational modes.

The modal eigenfrequencies are plotted in Figures 7.7 and 7.8. Several possible structural modes are indicated, and they remain stable with increasing model orders, as indicated in Figure 7.8. Other candidates for structural eigenfrequncies also exist. We summarise our findings in Table 7.1, where we have estimated some vibrational modes and their variation. These parameters were obtained by analysing a short eight-second excerpt of the data. As yet, we cannot say much about the accuracy if these results. If we had access to a database with previously estimated parameters, we could compare our measurements to standard values. Any discrepancies could be checked to see if they are within the previously established limits of natural variation.

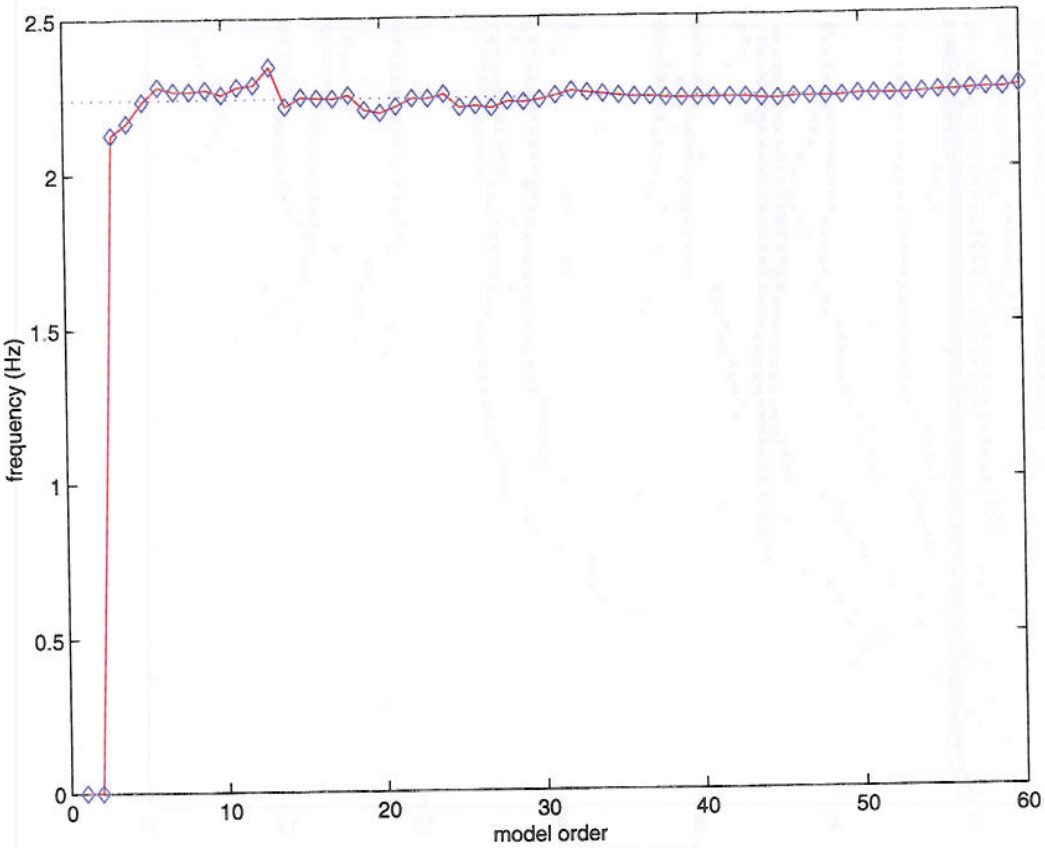


Figure 7.4 *Estimated long-ship vibrational frequency at different orders.*

Table 7.1 *Estimated modal frequencies for the data shown in Figure 7.6. The order ranged indicate the orders used when calculating the the empirical means and standard deviations.*

Type of vibration	$\hat{\mu}$ (Hz)	$\hat{\sigma}$ (Hz)	order range
Long-ship	2.39	0.01	[3, 60]
Plate vibration	19.3	0.2	[12, 60]
Superharmonic	41.6	0.3	[13, 60]

There is, for example, a possibility for some of the identified modes in the present data to actually be the result of a periodicity, or "mode", in the excitation process. Since we do not have the possibility to independently measure the load process separatly, this can be difficult to determine. One way would be to investigate the identified modes in different time series. The vibrations of the hull are expected to remain constant in time. The load processes on the other hand should vary more, since they depend on sea-state, speed of the vessel and other parameters.

7.2.2 Analysis of a long series

We have investigated a 50 min data sequence by first partitioning it into ten 5 minute sequences. We then identified structural eigenfrequencies in each of these sequences for model orders in the range $[1, \ldots, 60]$. By looking for structural parameters that remain stable with both time and increasing order, we are better able to judge if the identified parameter really is a characteristic of the ship. The data series used were from the

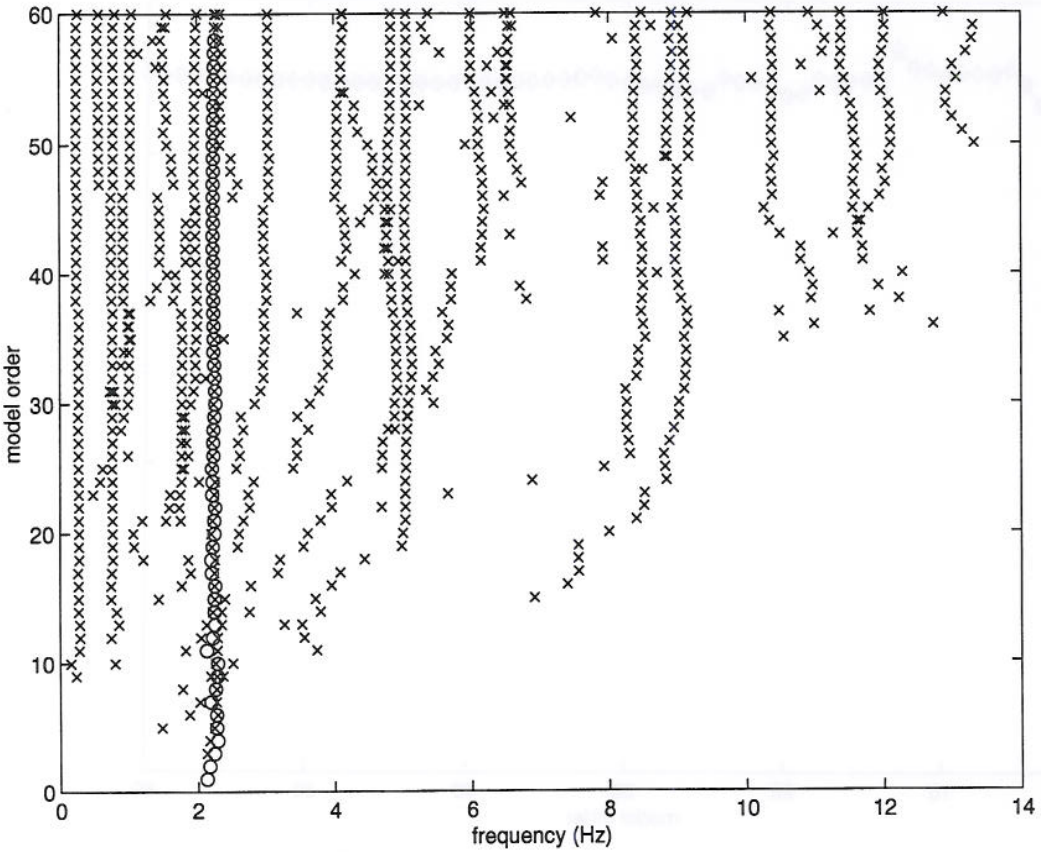


Figure 7.5 *Estimated damped vibrational frequencies at varying model orders. The asterices indicate the mode corresponding to vibrations at 2.2 Hz.*

L sensor in Figure 2.4 and it was filtered and downsampled to 100 Hz. A Nyquist frequency of 100 Hz allows for identification of frequencies up to 50 Hz. Although the majority of dynamic modes are expected to be found at lower frequencies than this, there are indications that modes associated with higher frequencies exist. The spikes between $50 - 60$ Hz in Figure 2.4 could be a result of this.

Frequency

The estimated vibrational frequencies resulting from this analysis are shown in Figures 7.9 and 7.10. Each plot shows estimated vibrational frequencies at varying model orders, as obtained by separate consecutive 5 minute excerpts of data. Visual inspections reveal several interesting characteristics of these two Figures.

- There is a clearly identified mode at 2.2 Hz. It is very stable with respect to order, and it shows up in all the plots. It corresponds to the long-ship vibrational mode.
- Another possible mode close to the previous one show up at a slightly lower frequency. It is much more unstable with respect to order, and in some of the plots it is hardly discernible, but it can be seen in all the plots in Figure 7.10.
- Two possible modes at around 5 Hz show up in all the plots. These are quite well-defined, so we assume as a working hypthesis that they represent intrinisic structural properties. It is not known for certain what kind of vibration they

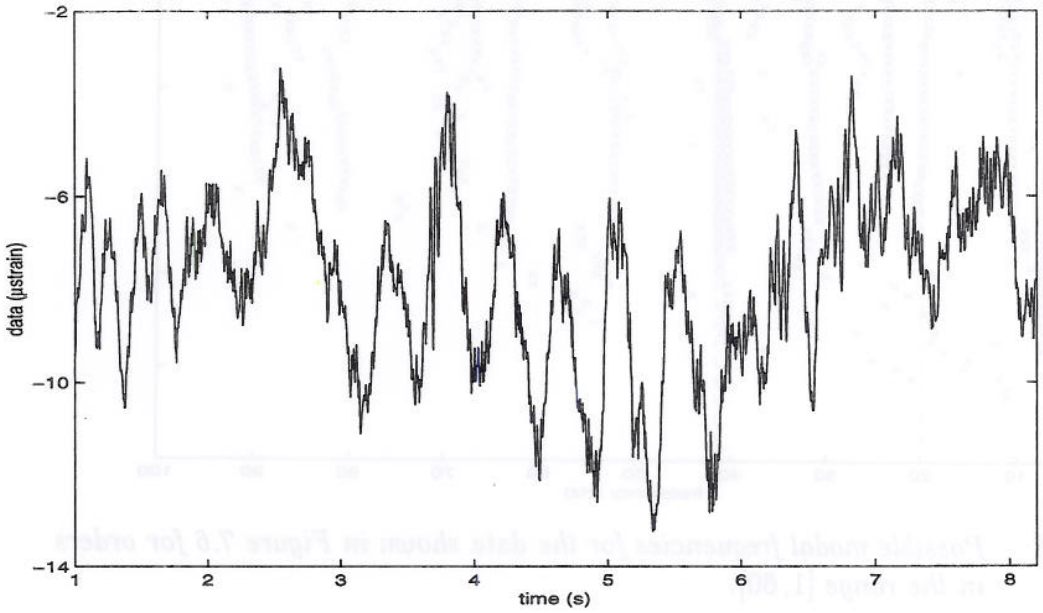


Figure 7.6 *A short data set showing typical strain loads typical for calm sea states.*

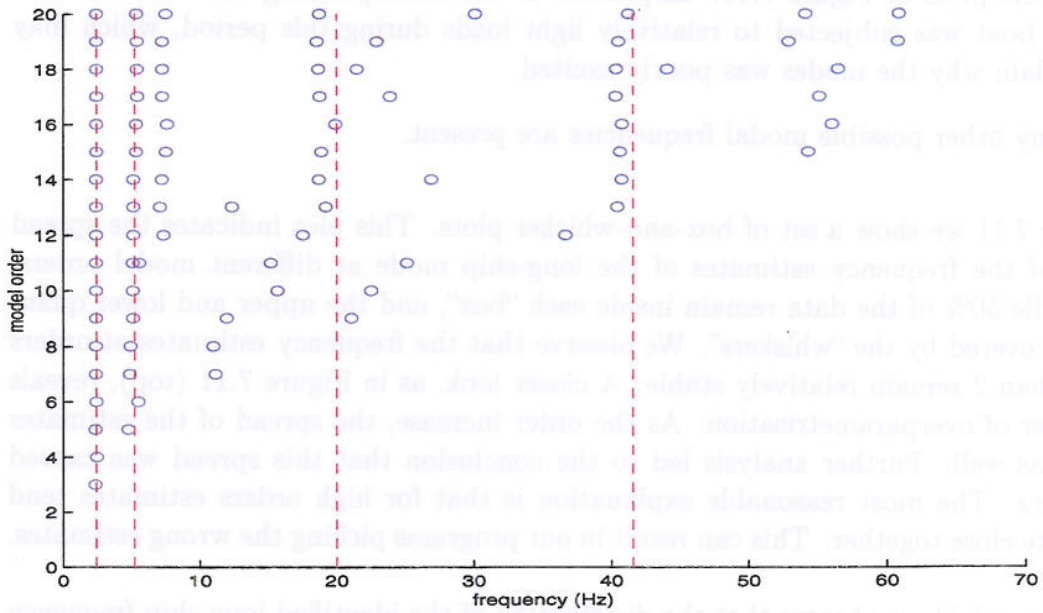


Figure 7.7 *Possible modal frequencies for the data shown in Figure 7.6 for orders in the range* $[1, 20]$.

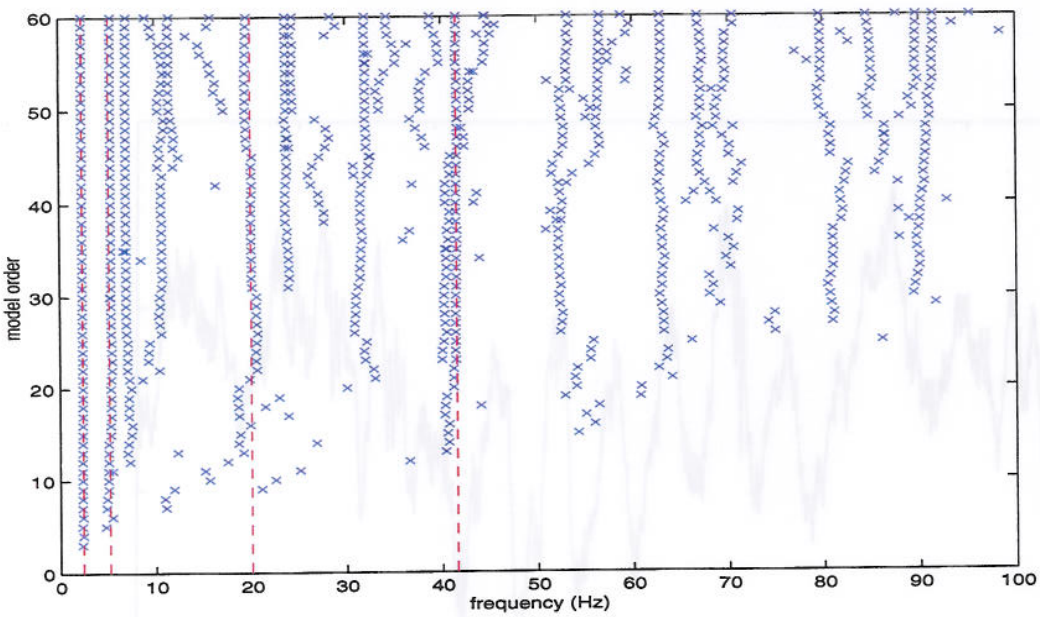


Figure 7.8 *Possible modal frequencies for the data shown in Figure 7.6 for orders in the range* $[1, 60]$.

represent. They might for example be higher-order vibrations connected to long-ship mode mentioned above.

- A stable mode at about 17 Hz is present, and it is identified with a local plate vibration. In some plots, a vibration around 19 Hz is present as well. This vibration is quite poorly excited.

- A mode at about 34 Hz show up in most of the plots. It is poorly excited in the middle plots of Figure 7.10. Inspection of the corresponding data showed that the boat was subjected to relatively light loads during this period, which may explain why the modes was poorly excited.

- Many other possible modal frequencies are present.

In Figure 7.11 we show a set of box-and-whisker plots. This plot indicates the spread in time of the frequency estimates of the long-ship mode at different model orders. The middle 50% of the data remain inside each "box", and the upper and lower quantiles are covered by the "whiskers". We observe that the frequency estimates at orders greater than 2 remain relatively stable. A closer look, as in Figure 7.11 (top), reveals the danger of overparametrisation: As the order increase, the spread of the estimates increase as well. Further analysis led to the conclusion that this spread was caused by *outliers*. The most reasonable explanation is that for high orders estimates tend to lie quite close together. This can result in our programs picking the wrong estimates.

From Figure 7.11 we observe that the distribution of the identified long ship frequency is rather narrowly centered around 2.2 Hz. A preliminary analyis show that much of the spread is caused by a systematic error in the selection of estimates. The development of more robust extraction routines has been considered outside the scope of this thesis.

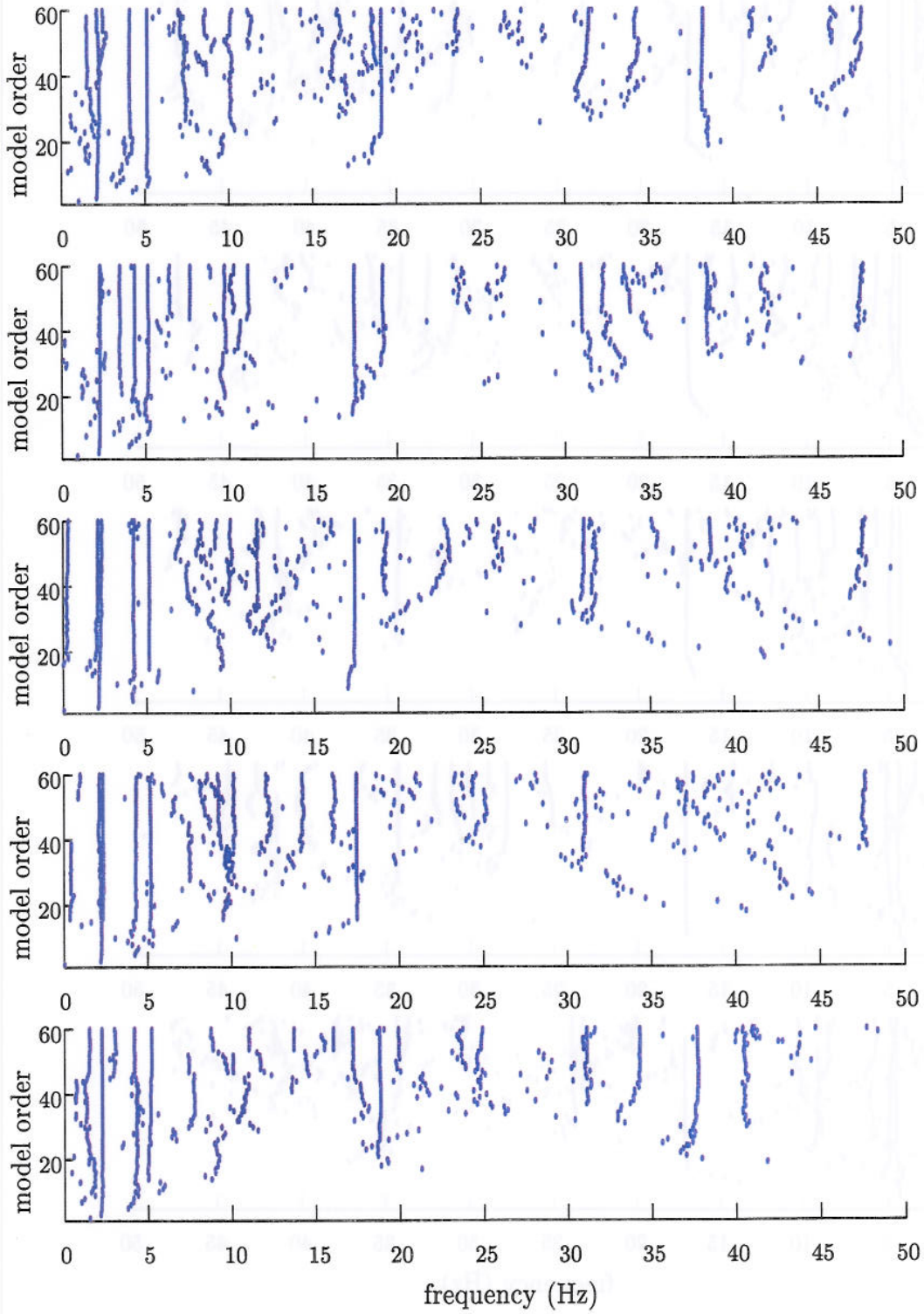


Figure 7.9 *Estimated vibrational frequencies at varying model orders. From top to bottom, the plots show frequency estimates based on consecutive* 5*-minute intervals of data from a data set of length* 25 *minutes.*

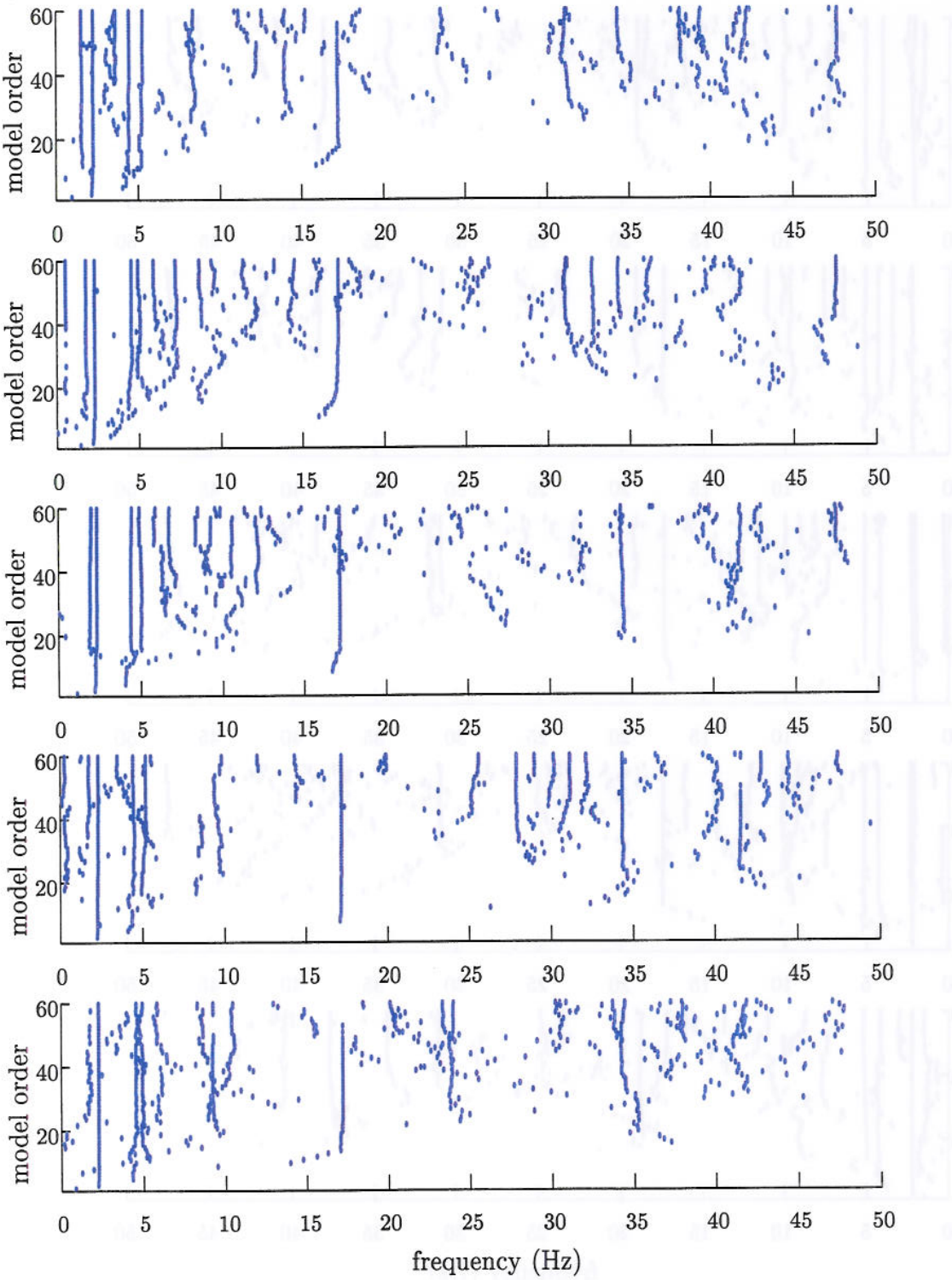


Figure 7.10 *Estimated vibrational frequencies at varying model orders. From top to bottom, the plots show frequency estimates based on consecutive 5-minutes intervals of data. The data used are the 25 minutes following the data used in Figure 7.9.*

Table 7.2 *Estimated modal damped modal frequencies and modal damping for the long-ship bending mode, plate vibration and a hypothetical super-harmonic mode. The estimates are based on the mean of the relevant data after removing the highest and lowest 25% values. The errors is estimated as the interquartile range.*

Type of vibration	frequency (Hz)	damping ratio	order range
Long-ship	2.25 ± 0.05	0.08 ± 0.03	$[3, 60]$
Long-ship2	5.2 ± 0.2	0.05 ± 0.03	$[15, 60]$
Plate vibration	17.3 ± 0.3	0.002 ± 0.0015	$[18, 60]$
"Superharmonic"	34.3 ± 0.2	0.001—0.01	$[22, 60]$

A similar analysis can be done for other vibrational modes. In Figures 7.9 and 7.10 we notice that a there is a fairly stable node around 17 Hz. It is known that there is a local plate vibration at approximatly this frequency, so we identify this this mode as a local plate vibration. Another line close to 20 Hz also appears at times, and this may also represent some kind of plate vibration. These two modes are actually discernible from the spectral density plot in Figure 2.5. We show in Figure 7.12 a box-and-whisker plot for the 17 Hz component. It is seen to be relativly stable, both with respenct to time and order, expecially in the order range $[19, 27]$. The problem with outliers is also visible. Other analyses are also performed, and and Appendix C contains some of the relevant plots.

Modal damping ratio

We now turn to estimation of the modal damping ratio. In general, this is known to be a difficult problem. One of the main reasons for this is that in practical situations, a significant amount of *non-proportional damping* is present. We expect that the damping of the boat is non-constant and non-proportional because the amount of water present varies with different wave impacts. Much research is currently being put into obtaining good estimates of the structural damping when non-proportional damping is present, see e.g. [1].

The estimates show much greater variability, Figures 7.12-7.18, but we are still able to obtain resonably stable results. The results obtained for natural frequencies and damping ratios are summarised in Table 7.2.

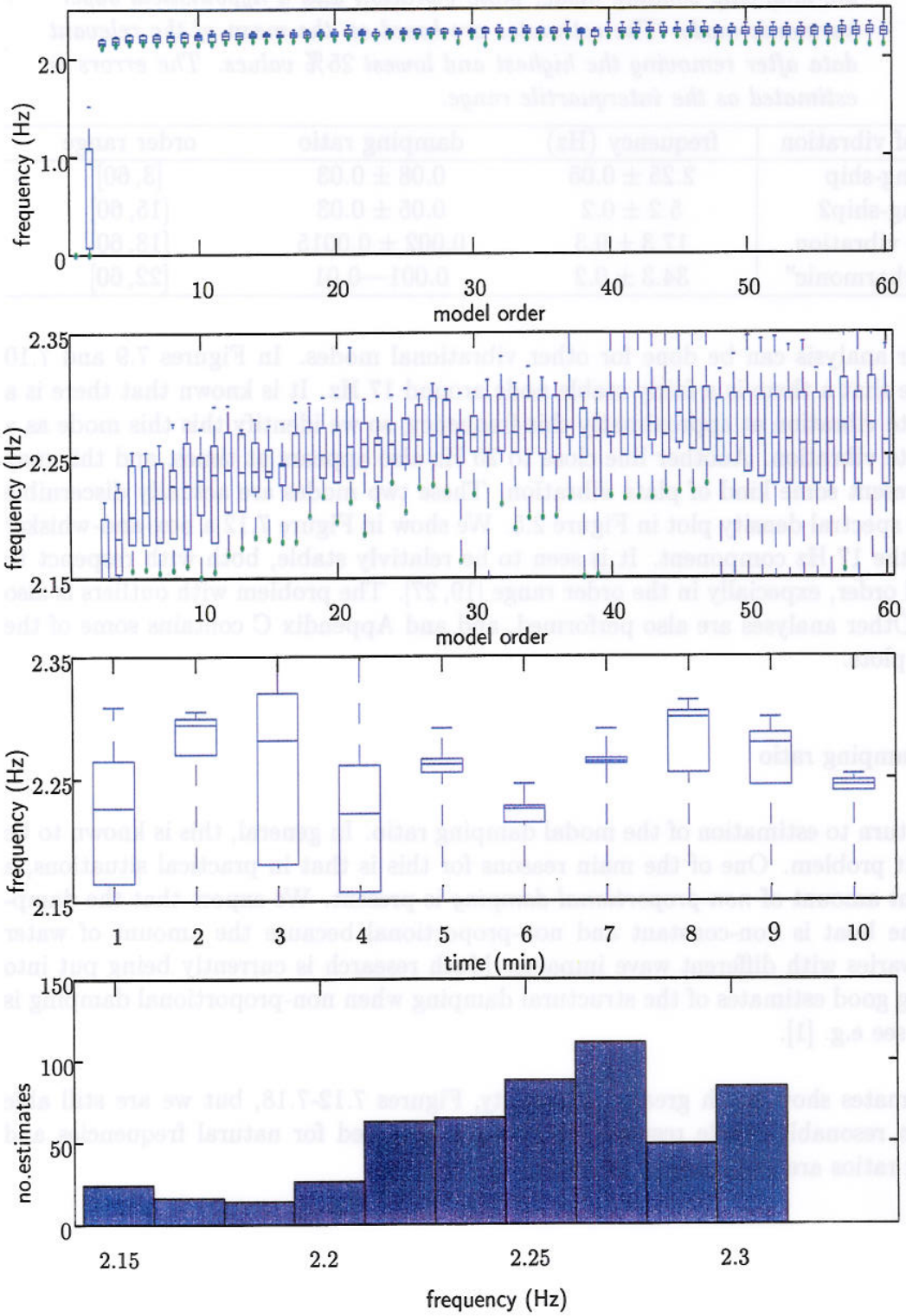


Figure 7.11 *Box-and-whisker plot showing the spread of the estimated frequency of the 2.2 Hz long-ship vibration in 10 separate time series at model orders in the range* $[1, 60]$ *(top and second). For each of these 10 time series, the spread of the estimates for model orders in the range* $[18, 60]$ *at different times is shown in the third plot. The distribution of the middle* 90% *of the data is shown at the bottom.*

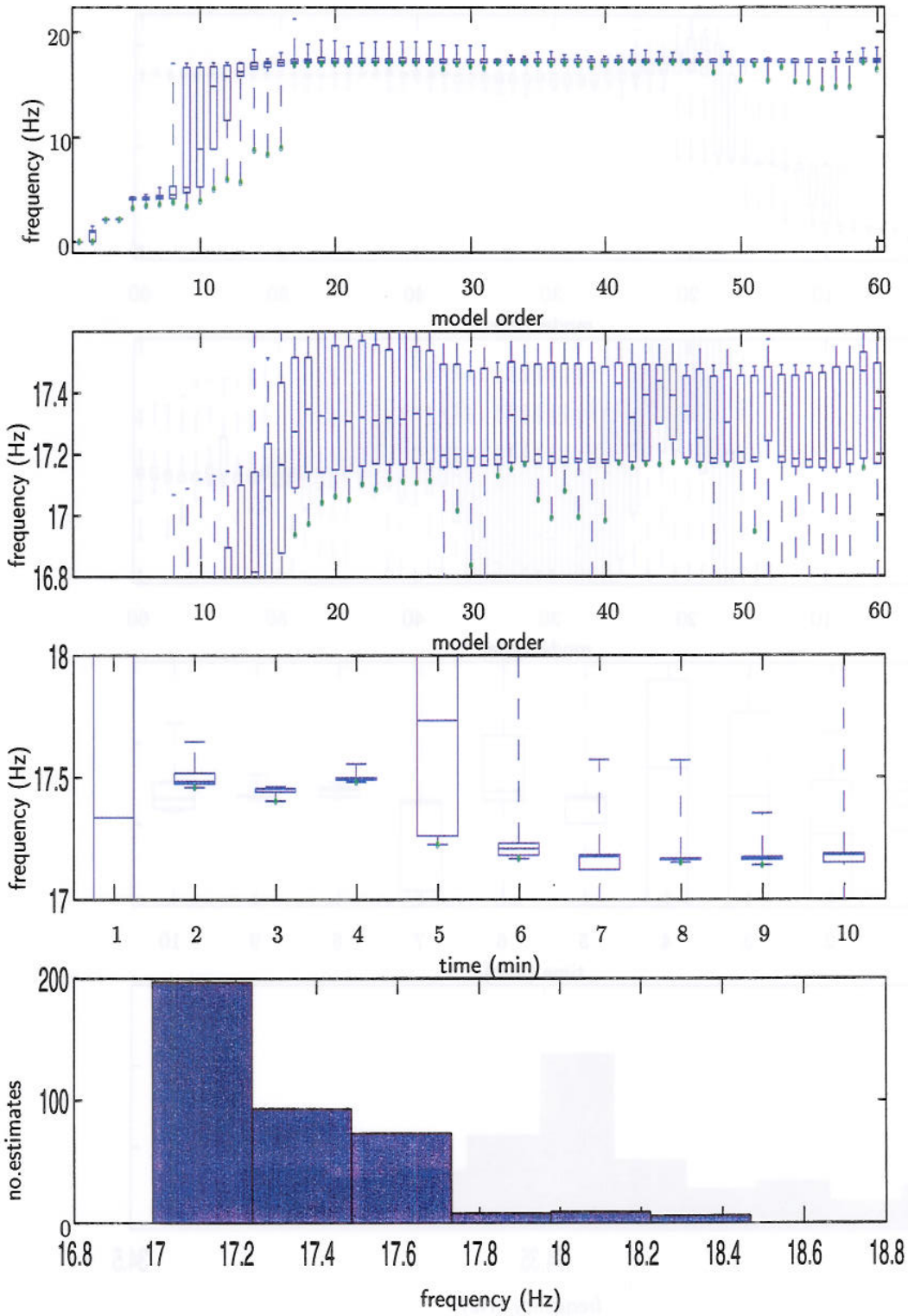


Figure 7.12 *Box-and-whisker plot showing the spread of the estimated frequency of the 17 Hz plate vibration in 10 separate time series at model orders in the range* $[1, 60]$ *(top and second). For each of these 10 time series, the spread of the estimates for model orders in the range* $[18, 60]$ *at different times is shown in the third plot. The distribution of the middle* 90% *of the data is shown at the bottom.*

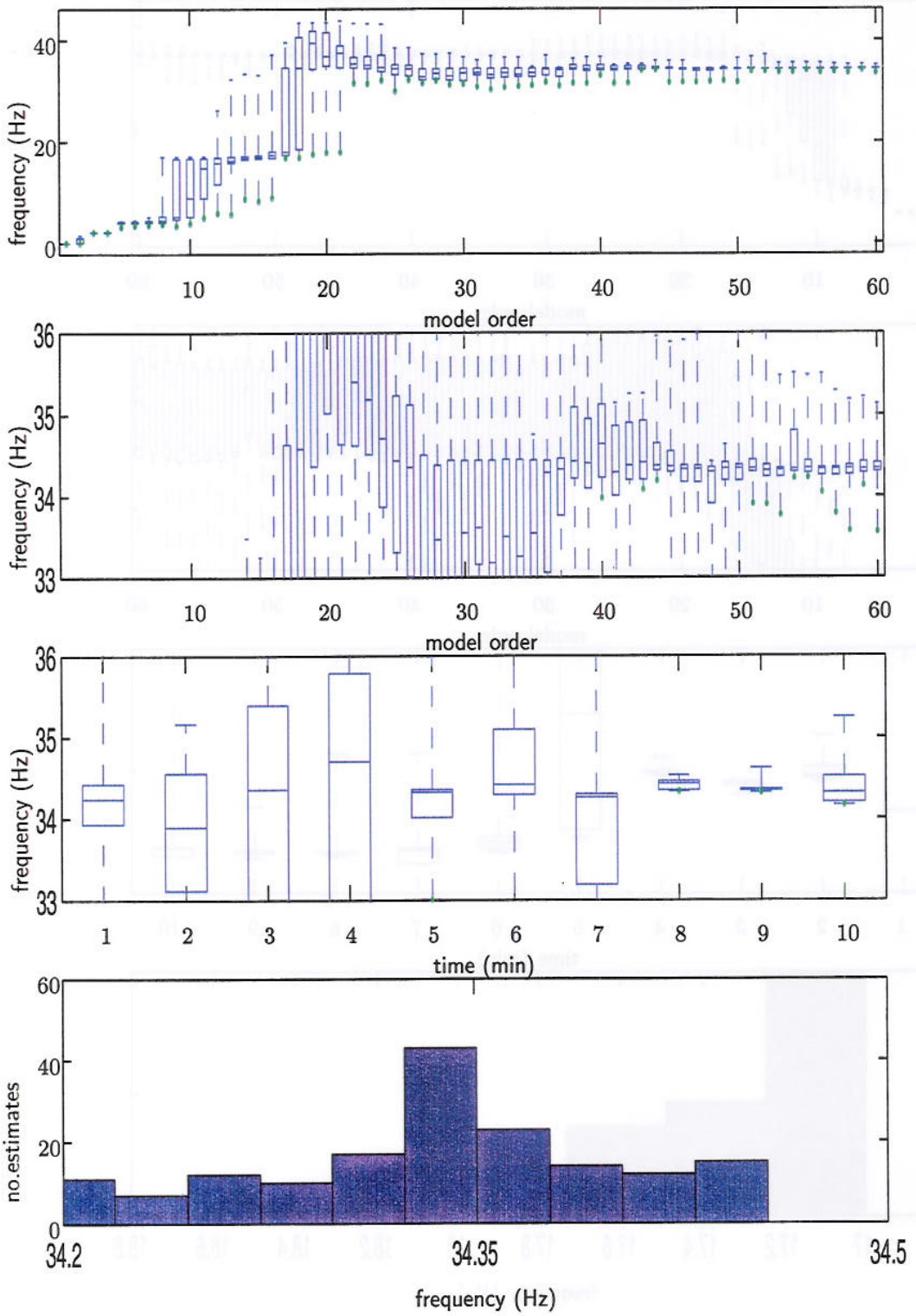


Figure 7.13 *Box-and-whisker plot showing the spread of the estimated frequency of the 34 Hz plate vibration in 10 separate time series at model orders in the range* $[1, 60]$ *(top and second). For each of these 10 time series, the spread of the estimates for model orders in the range* $[22, 60]$ *at different times is shown in the third plot. The distribution of the middle* 50% *of the data is shown at the bottom.*

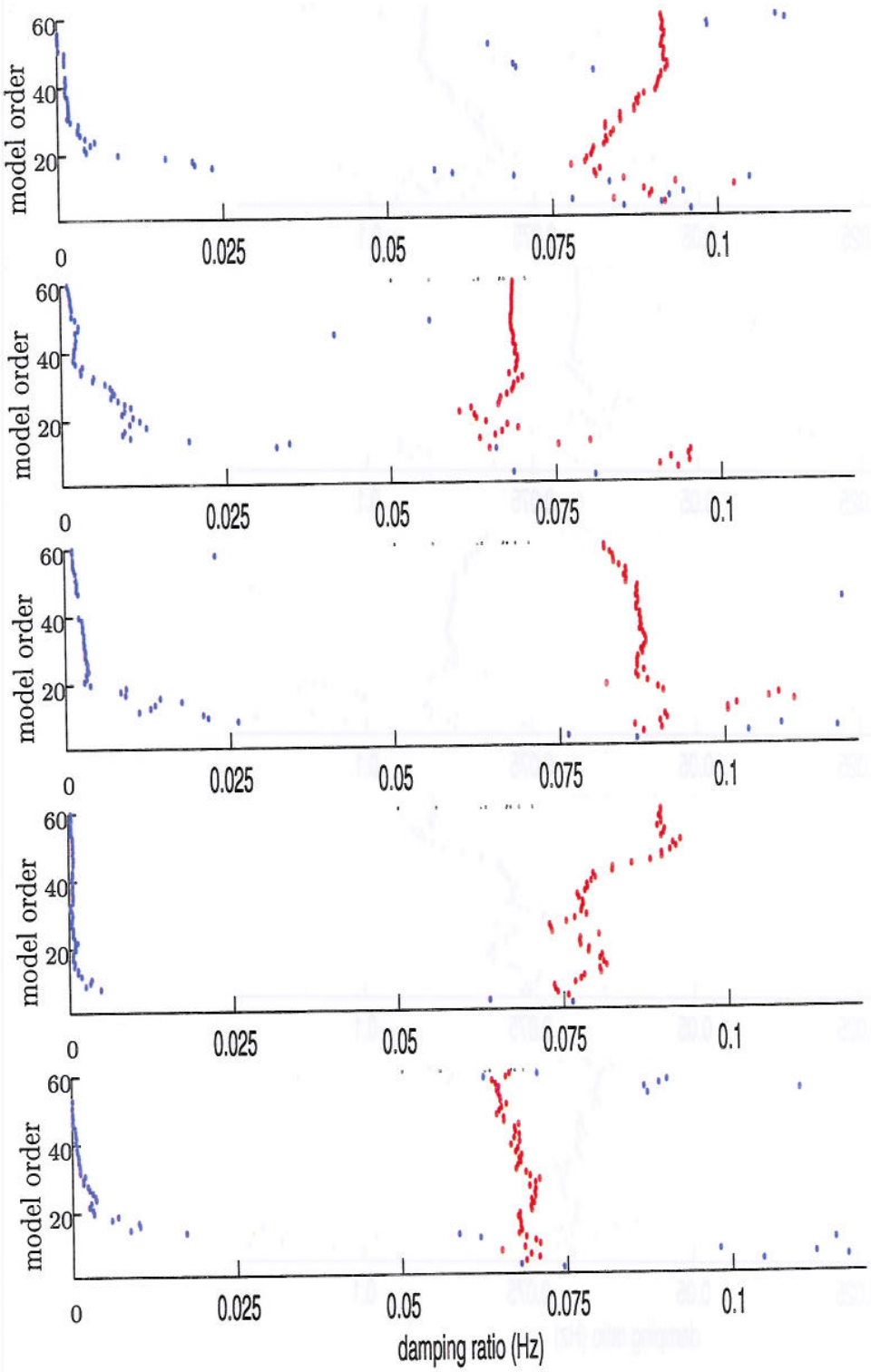


Figure 7.14 *Estimated damping ratios for the modes at 2.2 Hz (red) and 17 Hz (blue) at varying model orders. From top to bottom, the plots show frequency estimates based on consecutive 5-minute intervals of data from a data set of length 25 minutes.*

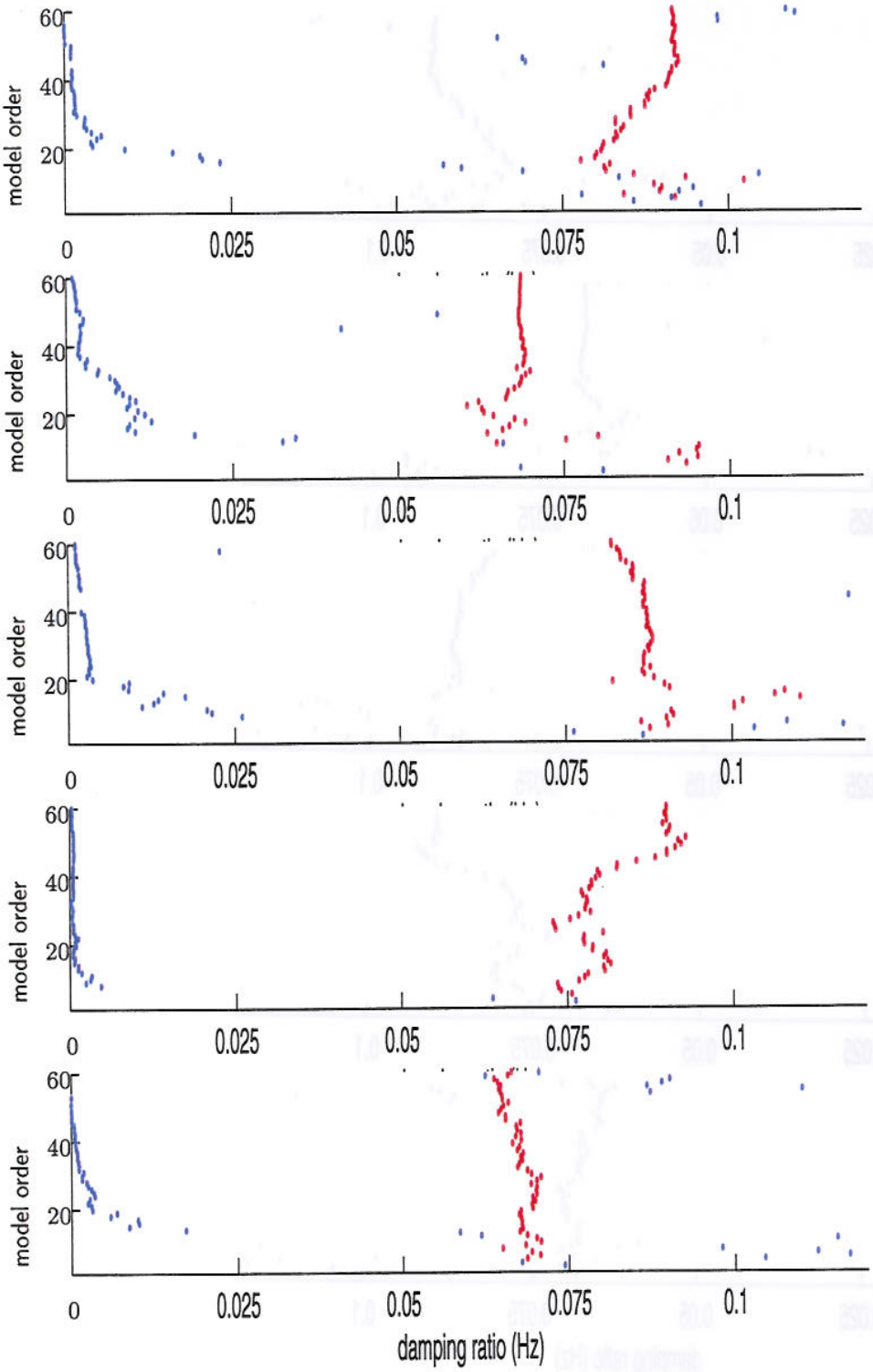


Figure 7.15 *Estimated damping ratios for the modes at 2.2 Hz (red) and 17 Hz (blue) at varying model orders. From top to bottom, the plots show frequency estimates based on consecutive 5-minutes intervals of data. The data used are the 25 minutes following the data used in Figure 7.9.*

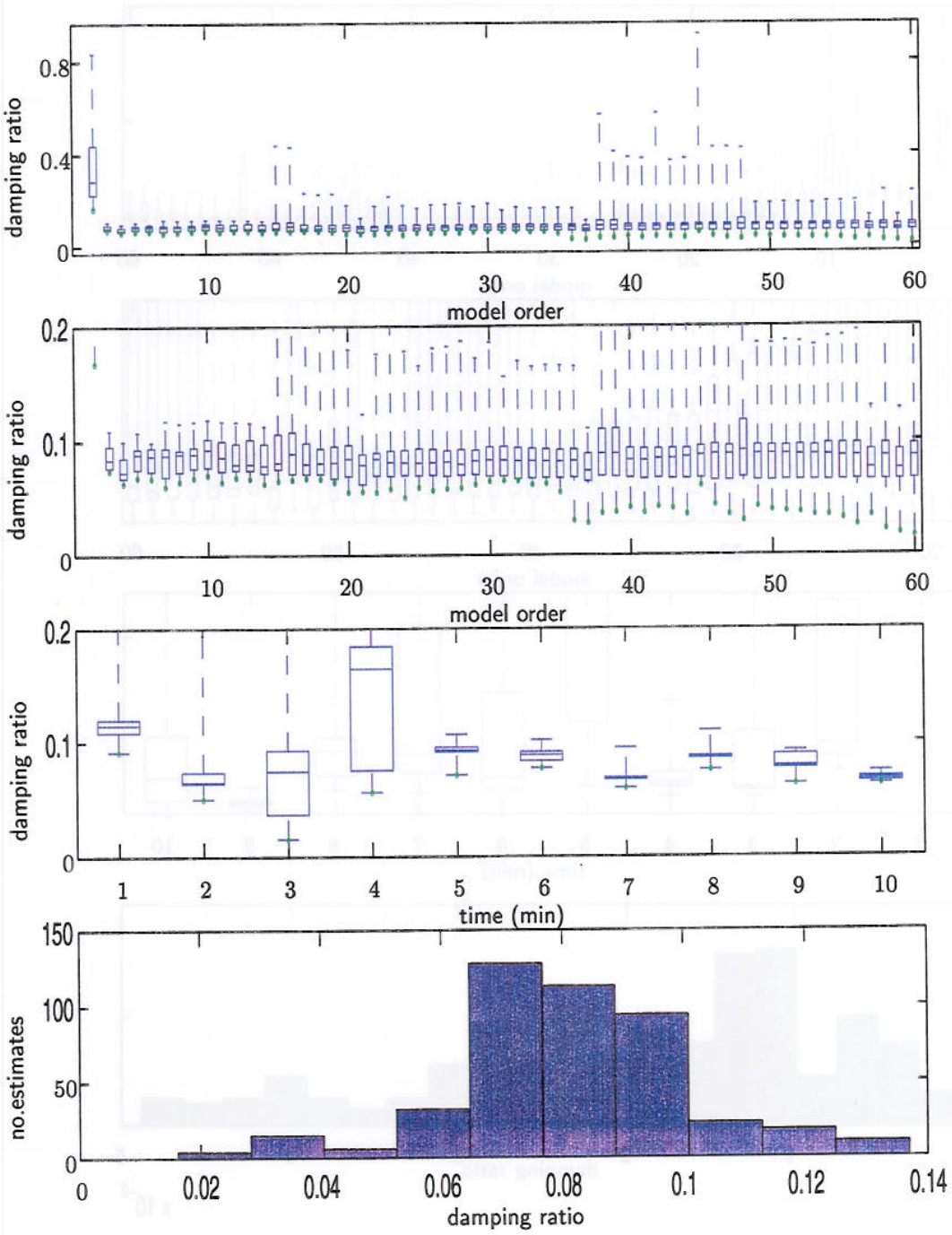


Figure 7.16 *Box-and-whisker plot showing the spread of the estimated damping ratio in the long-ship vibrational mode in 10 separate time series at model orders in the range* $[1, 60]$ *(two first plots). For each of these 10 time series, the spread of the estimates for model orders in the range* $[3, 60]$ *at different times is shown in the third plot. The distribution of the* 90% *lowest estimates is shown at the bottom.*

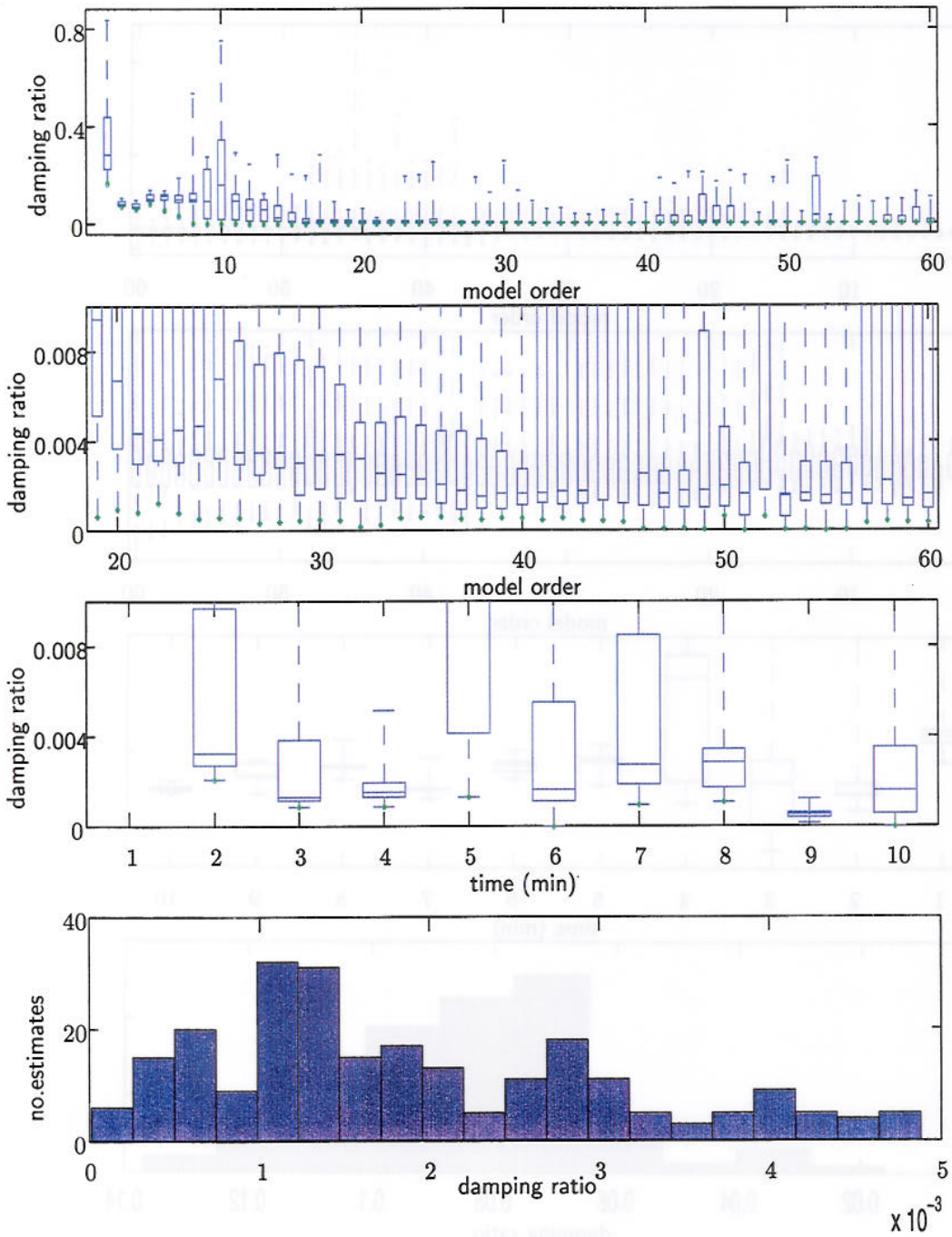


Figure 7.17 *Box-and-whisker plot showing the spread of the estimated damping ratio in the 17 Hz plate vibration mode in 10 separate time series at model orders in the range* $[1, 60]$ *(top). The second plot shows the spread of the estimates using model orders in the range* $[19, 60]$. *For each of these 10 time series, the spread of the estimates for model orders in the range* $[19, 60]$ *at different times is shown in the third plot. The spread of all the estimates is shown at the bottom.*

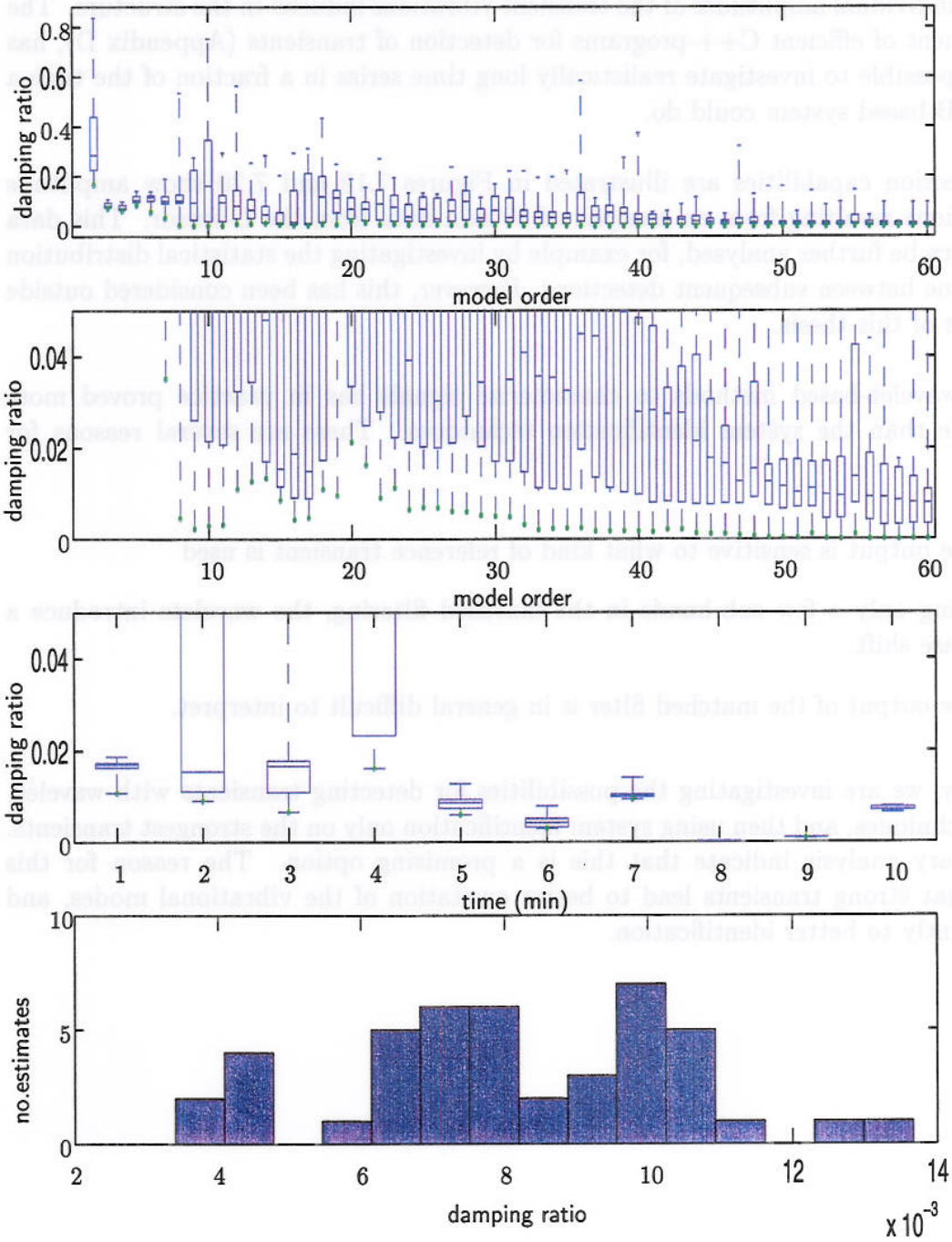


Figure 7.18 *Box-and-whisker plot showing the spread of the estimated damping ratio in the 34 Hz vibration mode in 10 separate time series at model orders in the range* $[1, 60]$ *(top) and middle. For each of these 10 time series, the spread of the estimates for model orders in the range* $[50, 60]$ *at different times is shown in the third plot. The spread of all the these estimates is shown at the bottom.*

7.3 Analysis by wavelet-based matched filtering

One of the most important tasks of a structural health monitoring system is to measure the individual amplitudes of the transient vibrations induced in the structure. The development of efficient C++-programs for detection of transients (Appendix D), has made it possible to investigate realistically long time series in a fraction of the time a MATLAB-based system could do.

The detection capabilities are illustrated in Figures 7.19 and 7.20 show amplitude distributions resulting from an analysis of 50 min data from the L-sensor. This data can in turn be further analysed, for example by investigating the statistical distribution of the time between subsequent detections. However, this has been considered outside the scope of this thesis.

To use wavelet-based methods to characterize signals has in practice proved more unreliable than the system identification techniques. There are several reasons for this.

- The output is sensitive to what kind of reference transient is used
- Using only a few sub-bands in the matched filtering, the wavelets introduce a phase shift.
- The output of the matched filter is in general difficult to interpret.

Currently, we are investigating the possibilities for detecting transients with wavelet-based techniques, and then using system identification only on the strongest transients. Preliminary analysis indicate that this is a promising option. The reason for this lies in that strong transients lead to better excitation of the vibrational modes, and consequently to better identification.

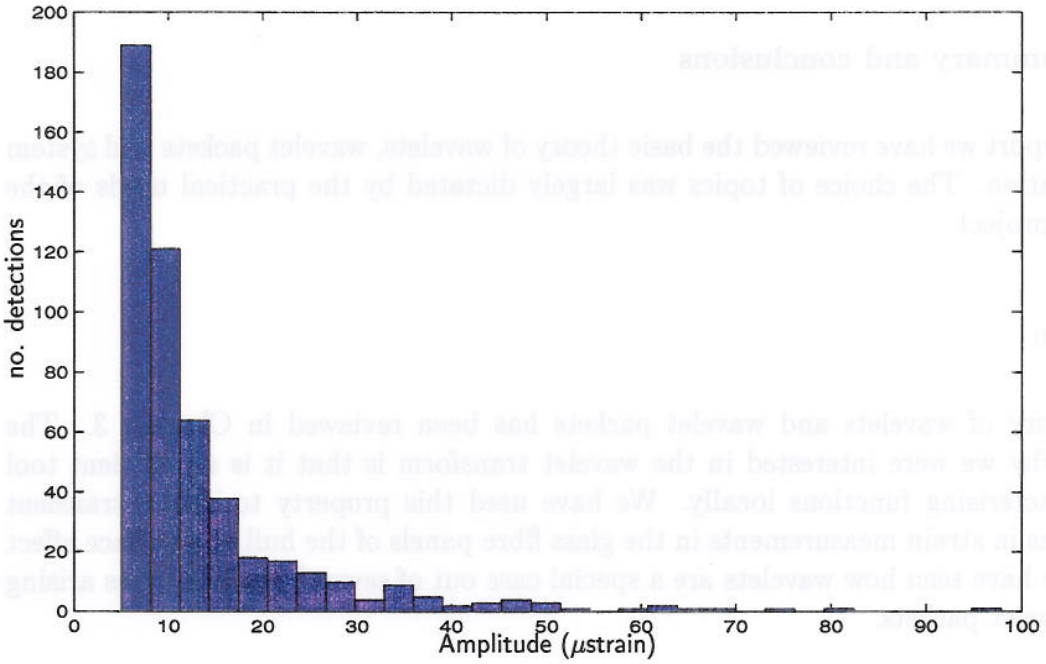


Figure 7.19 *Distribution of detected long-ship vibrational amplitudes greater than 5 μstrain in 50 minutes of data from the L-sensor. The detection routines were based on wavelet domain matched filtering in detail levels seven and eight, using a Daubechies 6 system.*

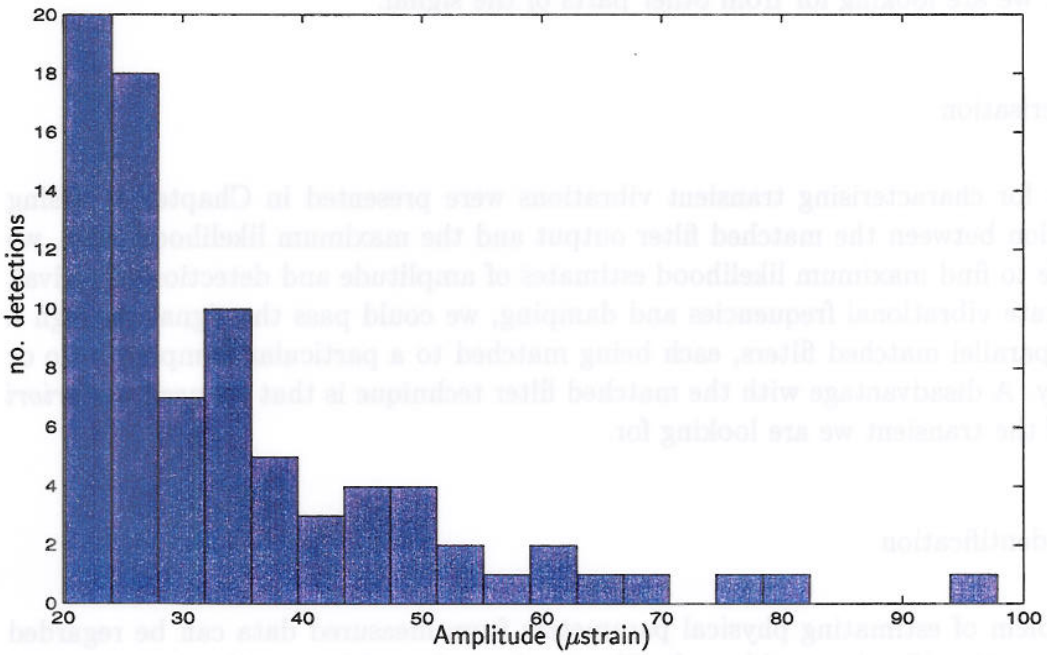


Figure 7.20 *Distribution of detected long-ship vibrational amplitudes greater than 20 μstrain in 50 minutes of data from the L-sensor. The detection routines were based on wavelet domain matched filtering in detail levels 7 and eight, using a Daubechies 6 system.*

8 CONCLUDING REMARKS

8.1 Summary and conclusions

In this report we have reviewed the basic theory of wavelets, wavelet packets and system identification. The choice of topics was largely dictated by the practical needs of the CHESS-project.

Detection

The theory of wavelets and wavelet packets has been reviewed in Chapter 3. The reason why we were interested in the wavelet transform is that it is an efficient tool for characterising functions locally. We have used this property to detect transient vibrations in strain measurements in the glass fibre panels of the hull of a surface effect ship. We have seen how wavelets are a special case out of several possible bases arising from wavelet packets.

Both the wavelets and wavelet packet bases are presented as special cases *frames*. We also showed how any linear operator on a Hilbert space may be performed in any reasonable *transform-domain*. As a special case of this result, matched filtering can be performed in the wavelet packet transform domain. It was also shown that it was possible to perform an approximate matched filtering on a subset of the transform domain. By choosing this subset to be those parts of the ambient signal space where the greater part of the energy of our transients are located, we effectively separate the transient we are looking for from other parts of the signal.

Characterisation

Methods for characterising transient vibrations were presented in Chapter 4. Using the relation between the matched filter output and the maximum likelihood ratio, we were able to find maximum likelihood estimates of amplitude and detection of arrival. To estimate vibrational frequencies and damping, we could pass the signal through a bank of parallel matched filters, each being matched to a particular damping ratio or frequency. A disadvantage with the matched filter technique is that we need a *a priori* model of the transient we are looking for.

System identification

The problem of estimating physical parameters from measured data can be regarded as a system identification problem. In Chapter 5 we showed how a vibrating structure could be considered as a system with inputs and outputs. We reviewed basic system theory, and showed how a non-linear system may be approximated by a linear, time invariant system. We established the connection between the physical continuous-time system and a discrete-time approximation. In Chapter 6 we reviewed methods for

estimating the intrinsic dynamics properties of a linear time-invariant system from measured data and showed how these results could be interpreted physically.

Implementation

The methods described above have been implemented. The fast wavelet transform, detection algorithms and matched filtering have been implemented in a C++ program, and this has made it possible to analyse longer data series. This is also an important step towards making a working prototype of the signal processing system.

We demonstrated how system identification could be used for estimating vibrational frequencies and damping. The algorithms are implemented in MATLAB, and they are mainly based on the implementations in [40].

8.2 Suggestions for further work

The methods developed in this report are general. For this reason, many aspects which are relevant to the CHESS-project have only been touched lightly upon, if at all. In this section we list some of these areas and point out specific possibilities for further work. The topics are organised in several broad and partially overlapping categories.

Wavelet analysis

- In analysing the signals by wavelets, we have hitherto considered them as independent one-dimensional signals. An wavelet-based approach which is able to take the strong correlation between the different channels into consideration would be useful.
- We have not considered bi-orthogonal wavelets in this report. However, if we use wavelets mainly as a prefilter to separate different scales, this is an interesting option. The reason for this is that we may then then use wavelets with better *filtering* performance, such as symmetric wavelets with linear phase.
- There are many generalisations of our wavelet scheme. Instead of dilating by 2, one may use an M-band scheme. Dilating by *matrices* is also possible. Investigating the possibilities offered by these methods in more detail should be done.

Damage detection

- Damage detection has not been a topic *per se* in this report, but many of the methods seem to be applicable for this purpose. Finding relevant methods from the field of structural reliability and incorporating them into ours will be then be necessary. Reliability parameters for the material in the hull must be obtained.
- One idea is to use a wavelet-based detector for detecting strong transients, and then basing the system identification algorithms on only the strongest transients.

- In Chapter 5, we saw how the individual amplitudes of the vibrations could be found from the state vector and the eigenvectors of the system matrix. Implementing this, and comparing the resulting amplitude estimates with matched filter estimates would be interesting.

Data analysis

- A systematic and thorough analysis of the available data should be performed. On the basis of this analysis, one could identify which wavelets and wavepackets are more useful for separating the different transients. We have concentrated on the Daubechies wavelets, but many interesting alternatives exists.

  It is not known in detail how the signals corresponding to the transient vibrations vary. Non-linear responses will most likely exhibit considerable variation in the characteristic mode shapes. The measurements will most likely depend on the sensor location. It is known that the mode shapes of the local panel vibrations are strongly influenced by the panel geometry.

  These aspects should be resolved before a particular wavelet basis is chosen.

- The data analysis could also be performed with the objective of establishing *reference values* for different structural parameters. Their variations should also be quantified, possibly with confidence intervals.

  The amount of data available for analysis is huge. One should take advantage of this and estimate relevant statistical properties of the load and response processes.

Programming aspects

- A recursive real-time implementation of the System identification routines may be considered. The routines we have used are all Matlab programs, and they make extensive use of Matlab's linear algebra capabilities. If these are to be implemented in C++, one should base the programs on existing linear algebra libraries such as LINPACK. Alternatively, one could as a use a Matlab-to-C++ compiler [22].

- The programs implemented are still undergoing development. As they become increasingly complex and more people take part in the programming, care must be taken to make the programs robust, efficient and maintainable. One should seek to formalise program specification and development according to industry standards [35]. In particular, emphasis should be placed on documenting the development process.

List of Figures

List of symbols and acronyms

Symbol	Meaning
$*$	Convolution; Adjoint.
$\downarrow 2$	Downsampling by two, removes every second component of a sequence.
$\#$	Moore-Penrose pseudo-inverse.
$:=$	Is defined as
$\equiv$	Is identical to
$!$	Factorial
$\hat{f}$	Fourier transform of f, $\hat{f}(\omega) = \int dt\, f \exp(2\pi i \omega t)$.
$\langle \cdot, \cdot \rangle$	Inner product.
$\Re$, $\Im$	Real and imaginary parts.
$\tilde{f}$	STFT of f; general transform of f.
$\oplus$	Orthogonal direct sum.
$\psi(t)$	Mother wavelet.
$\psi_{j,k}(t)$	Scaled and translated wavelet, $\psi_{i,j}(t) = 2^{-j/2}\psi(2^{-j}t - k)$.
$\psi_{f,j,k}(t)$	Scaled and translated wavepacket of frequency f.
$\psi_k(t)$	$\psi_{0,k}(t)$.
$\phi(t)$	Scaling function.
$\phi_{j,k}(t)$	Scaled and translated scaling function, $\phi_{j,k}(t) = \phi(2^{-j}t - k)$.
$\phi_k(t)$	$\phi_{J,k}(t)$.
$\overline{z}$	Complex conjugate of z.
a	Continious scaling parameter.
b	Continious shift parameter.
$c_{J,k}, c_J[k]$	Level J approximation coefficients.
$c_{f,j,k}, c_{f,j}[k]$	Wavelet packet coefficients, level j, frequency f.
$d_{j,k}, d_j[k]$	Level j detail coefficients.
f	Function; Frequency index of wavepacket.
$\mathcal{F}$	Fourier transform operator.
h_{MF}	Impulse transfer function of a matched filter.
Id	Identity operator.
$L^2(\mu)$	Space of functions which are quadraticly integrable with measure μ.
$\ell^p(\mathbb{Z})$	Space of sequences u such that $\sum_{k\in\mathbb{Z}} \lvert u_k \rvert^p < \infty$.
$n(t)$	Noise.
S	Shift operator, $\mathrm{S}f(n) = f(n+1)$; Synthesis operator
$\mathbb{R}, \mathbb{R}^+$	Real numbers, non-negative real numbers.
$s(t)$	Signal, transient we are looking for.
$\mathrm{T}, \mathrm{T}_\phi$	Transform, transform with respect to the funciton ϕ.
V_j	Scaling space j.
$V_{j,n}$	Decomposition space at scale j, frequency n.
W_j	Wavelet space j.
$w(t)$	Window function.
$x(t)$	Input to detector or filter.

Acronym	Meaning
CHESS	Composite Hull Embedded Sensor System
db$n$	Daubechies wavelet family n.
DFT	Discrete Fourier transform.

Acronym	Meaning
DWT	Discrete wavelet transfrom.
FEM	Finite element method.
FFI	Forsvarets Forskningsinstitutt
FWT	Fast wavelet transform.
NLR	Naval Research Laboratory
QMF	Quadratic mirror filters.
SES	Surface Effect Ship.
STFT	Short-time Fourier transform.

References

[1] K. F. Alvin, L. D. Peterson, and K. C Park. Extraction of normal modes and full modal damping from complex modal parameters. *AIAA Journal*, 35, 1997.

[2] Karl J. Åström and Björn Wittenmark. *Computer-Controlled Systems. Theory and Design.* Prentice-Hall, Upper Saddle River, New Yersey, 1997.

[3] Amund Solvi Bremer. Wavelets for detection of transient signals. FFI/RAPPORT-97/05153, Forsvarets Forskningsinstitutt, 1997.

[4] Peter Caines. *Linear Stochastic Systems.* Series in Probability and Mathematical Statistics. Wiley, New York, 1988.

[5] Fu-Kuo Chang, editor. *Structural Health Monitoring, Current Status and Perspectives*, Stanford University, Stanford, California, 1997. International Workshop on Structural Health Monitoring.

[6] Albert Cohen and Robert D. Ryan. *Wavelets and Multiscale Signal Processing.* Chapman & Hall, London, 1995.

[7] Ronald R. Coifman, Yves Meyer, and Wickerhauser Mladen Victor. Size properties of wavelet packets. In *Wavelets and their applications.* Jones and Bartlett, Boston, 1992.

[8] Ingrid Daubechies. *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[9] Roger Eriksen. Realtime system for transient detection and characterization. FFI/RAPPORT-96/03149, Forsvarets Forskningsinstitutt FFI, 1996.

[10] Strang Gilbert. *Linear Algebra and its Applications.* Harcourt Brace Jovanovitch, Inc., San Diego, 1988.

[11] Christopher Hoen. *System Identification of Structures Excited by Stochastic Load Processes.* PhD thesis, Norwegian Institute of Technology, 1991.

[12] Christopher Hoen. OFFTek as, Stortingsgaten 12, N-0161 Oslo, Norway, 1998.

[13] Thomas Kailath. *Linear Systems.* Prentice-Hall Information and System Sciences Series. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1980.

[14] Gerald Kaiser. *A Friendly Guide to Wavelets.* Birkhäuser, Boston, 1994.

[15] Stephen Kay. *Fundamentals of statistical signal processing.* PTR Prentice-Hall, Inc., 1993.

[16] David Kincaid and Ward Cheney. *Numerical Analysis.* Brooks/Cole Publishing Company, Pacific Grove, California, 1990.

[17] Jelena Kovačevic and Ingrid Daubechies, editors. *Proceedings of the IEEE, Special Issue on Wavelets*, volume 84. IEEE, April 1996.

[18] Erwin Kreyszig. *Advanced Engineering Mathematics.* John Wiley & Sons, Inc., 7 edition, 1993.

[19] Rolf Magne Larssen. *Estimation of Structural Parameters from Response Measurements on Submerged Floating Tunnels.* PhD thesis, Norwegian University of Science and Technology, 1996.

[20] Lennart Ljung. *System Identification. Theory for the user.* Prentice Hall, Englewood Cliffs, New Jersey, 1987.

[21] Nuno Manuel Mendes Maia, Júlio Martins Montalvão e Silva, Jimin He, Nicholas Andrew John Lieven Lieven, Rong Min Lin, Graham William Skingle, Wai-Ming To, and António Paulo Vale Urugeria. *Theoretical and Experimental Modal Analysis.* Research Studies Press Ltd., Somerset, England, 1997.

[22] MathTools Ltd, P.O.Box 855, Horsham, Pennsylvania 19044-0855, USA. *Matcom*

Matlab to c++ compiler.

[23] Leonard Meirovitch. *Principles and techniques of vibrations.* Prentice-Hall, Inc., 1997.

[24] Yves Meyer. *Wavelets: algorithms and applications.* Society for Industrial and Applied Mathematics, Philadelphia, 1993.

[25] Michel Misti, Yves Misti, Georges Oppenheim, and Jean-Michel Poggi. *Wavelet Toolbox User's Guide.* The MathWorks, Inc., 24 Prime Park Way, Natick, MA 01760-1500, USA, 1996.

[26] D.E. Newland. *Introduction to random vibrations, spectral and wavelet analysis.* Addison Wesley Longman Limited, Essex, 1993.

[27] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[28] Herbert John Pain. *The physics of vibrations and waves.* John Wiley & Sons Ltd., Chichester, 1993.

[29] Karianne Pran. Design of optical fibre bragg gratings. Master's thesis, Norwegian University of Science and Technology, 1995.

[30] Karianne Pran. Personal communication. Forsvarets Forskningsinstitutt, 1998.

[31] Karianne Pran, Emil Urnes, Amund Solvi Bremer, Geir Bjarte Havsgård, and Gunnar Wang. Application of wavelets for transient detection and characterization in health monitoring systems. In *Mathematics and Control in Smart Structures*, volume 3323 of *Proceedings of SPIE.* SPIE, 1998.

[32] Peter A. Ruymgaart and Tsu T. Soong. *Mathematics of Kalman-Bucy Filtering.* Springer-Verlag, Berlin Heidelberg, 1988.

[33] Torsten Söderström and Peter Stoica. *System Identification.* Series In Systems and Control Engineering. Prentice-Hall International, 1989.

[34] Eduardo Sontag. *Mathematical Control Theory.* Text in Applied Mathematics. Springer-Verlag New York, Inc., 1990.

[35] Bjarne Stroustrup. *The C++ Programming Language.* Addison-Wesley, 1997.

[36] George L. Turin. An introduction to matched filters. *IRE Transaction on Information Theory*, IT-6:311–329, 1960.

[37] George L. Turin. An introduction to digital matched filter. *Proceedings of the IEEE*, 64:1092–1112, 1976.

[38] Emil Urnes. Wavelets and their use in the detection and characterization of transient signals. FFI/RAPPORT-97/03066, Forsvarets Forskningsinstitutt, 1997.

[39] Peter Van Overschee and Bart De Moor. A unifying theorem for three subspace system identification algorithms. *Automatica, Special Issue on Trends in System Identification*, 31, 1995.

[40] Peter Van Overschee and Bart De Moor. *Subspace Identification for Linear Systems.* Kluwer Academic Publishers, 1996.

[41] G. Wang, G. B. Havsgård, E. Urnes, K. Pran, and S. Knudsen. Detection wavelets. In *Optical Fibers.* OFS, 1997.

[42] Mladen Victor Wickerhauser. *Adapted Wavelet Analysis from Theory to Software.* A K Peters, Ltd., Wellesley, Massachusets, 1994.

[43] Nicholas Young. *An introduction to Hilbert space.* Cambridge University Press, 1988.

[44] Jerzy Zabczyk. *Mathematical control theory: an introduction.* Systems & Control: Foundations & Applications. Birkhäuser, Boston, 1992.

A PRELIMINARIES

In this Appendix we present some background material and establish some conventions and notation.

A.1 General

Measure theory

A *measure* μ on a set X assigns non-negative values to subsets of X.

For example, we may define a measure on $X = \mathbb{R}^2$ by $\mu(A) = \iint_A a^{-2}\, dadb$, which can conveniently be written $d\mu(a,b) = a^{-2}\, dadb$. Our main use of measure theory is purely formalistic, in that it allows us to write continuous and discrete inner products with the same integral symbol. We denote by $L^2(\mu)$ the Hilbert space of functions which are squarely integrable with respect to the measure μ. The inner product on this space is given by

$$\langle f, g\rangle_{L^2(\mu)} = \int d\mu(t) f(t)\overline{g(t)}. \tag{A.1}$$

For example, the familiar $L^2(\mathbb{R})$ has measure $d\mu = dx$. The Lesbegues integral is defined using measure theory, and by choosing μ as the *counting measure* in (A.1), we get

$$\langle a, b\rangle_{L^2(\mu)} = \int d\mu(n) f(t)\overline{g(n)} = \sum_n a(n)\overline{b(n)}, \tag{A.2}$$

which is the ordinary inner product on ℓ^2. Details and discussions can be found in [14].

The complex logarithm

Let $e^w = z$, so that the logarithm of z is defined to be w. Let z in polar form be $z = |r|e^{i\theta}$. We know that θ is not unique, since we have $z = re^{i(\theta+2\pi n)}$ for all $n \in \mathbb{Z}$. The number w may always be written $w = u + iv$, and this gives

$$e^w = e^{u+iv}e^u e^{iv} = z = |r|e^{i(\theta+2\pi n)}. \tag{A.3}$$

Equating the magnitude and argument, we get

$$e^u = |r|, \quad v = \theta + 2\pi n. \tag{A.4}$$

Since u and $|r| > 0$ are real, we get $w = u + iv = \ln|z| + \theta + 2\pi n$. Thus,

$$\ln z = \ln|z| + \theta + 2\pi n, \quad \forall\, n \in \mathbb{Z}, \tag{A.5}$$

which is not unique. If we pick n in such a way that we always have

$$-\pi < \theta + 2\pi n \leq \pi, \tag{A.6}$$

we obtain the principal value of $\ln z$, denoted $\mathrm{Ln}\, z$. See [18]

A.2 Signal processing

Fourier transform

We will use the following conventions:

$$\hat{f}(\nu) = \int dt\, f(t) e^{-2\pi i \nu t} \tag{A.7}$$

$$f(t) = \int dt\, \hat{f}(\nu) e^{2\pi i \nu t} \tag{A.8}$$

Discrete-time Fourier transform

The discrete-time Fourier transform of a sequence $\{f[n]\}_{n\in\mathbb{Z}}$ is

$$F(\nu) = \sum_{n\in\mathbb{Z}} f[n] e^{-2\pi i \nu n}. \tag{A.9}$$

The inverse transform is

$$f[n] = \int_0^1 d\nu\, F(\nu) e^{2\pi i \nu n}. \tag{A.10}$$

Discrete Fourier transform

The discrete Fourier transform of the (finite) sequence $\{f[n]\}_{n=0}^{N-1}$ is defined as

$$F[k] = \sum_{n=0}^{N-1} f[n] e^{-2\pi i k n/N}. \tag{A.11}$$

The inverse transform is

$$f[n] = \frac{1}{N} \sum_{k=0}^{N-1} F[k] e^{2\pi i k n/N}. \tag{A.12}$$

A.3 Linear algebra

We list some standard topics from linear algebra that we will need. Proofs and more detail can be found in [16] or [10].

Notation

The transpose of a matrix A is denoted by A^T, and the Hermitian conjugation is denoted by A^*; $A^* \equiv \overline{(A^T)}$.

Eigenvalues

An eigenvalue of a square matrix is a constant λ such that

$$Ax = \lambda x. \tag{A.13}$$

Here, x is the corresponding eigenvector. If T is a nonsingular matrix of same size as A, and $B := TAT^{-1}$, then A and B have the same eigenvalues.

Matrix exponentials

The matrix exponential is defined as

$$e^A := \sum_{j=0}^{\infty} \frac{A^j}{j!}, \quad A \in \mathbb{R}^{n\times n} \tag{A.14}$$

The sum always converges.

Singular value decomposition

Any matrix $A \in \mathbb{R}^{m\times n}$ with $r = \text{rank}A$ (say) may be factored into $A = USV^*$. where $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$, $S \in \mathbb{R}^{m\times n}$. The matrix S is diagonal and the first r entries on the diagonal is called singular values. The singular values are the square roots of the non-zero eigenvalues of A^TA, and the remaining $n-r$ diagonal entries are zero. The columns of U are the eigenvectors of AA^* and the columns of V are the eigenvectors of A^*A. Both U and V are unitary. We can write the SVD

$$A = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1 & V_2 \end{pmatrix}^* \tag{A.15}$$

where $U \in \mathbb{R}^{m\times r}$, $U_2 \in \mathbb{R}^{m\times(m-r)}$, $V_1 \in \mathbb{R}^{n\times t}$ and $V_2 \in \mathbb{R}^{n\times(n-r)}$. The columns of V_1 is an orthonormal basis for the row space of A, and the columns of V_2 is an orthonormal basis for the nullspace of A. Similarly, the columns of U_1 is an orthonormal basis for the column space of A, and the columns of U_2 is an orthonormal basis for the left nullspace of A.

Pseudo-inverse

The pseudoinverse of a diagonal matrix, such as

$$S = \begin{pmatrix} s_1 & & & & & & \\ & s_2 & & & & & \\ & & \ddots & & & & \\ & & & s_r & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{pmatrix} \tag{A.16}$$

is given by

$$S^{\#} = \begin{pmatrix} 1/s_1 & & & & & & \\ & 1/s_2 & & & & & \\ & & \ddots & & & & \\ & & & 1/s_r & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{pmatrix} \tag{A.17}$$

Let A have the singular value decomposition

$$A = USV^* \tag{A.18}$$

The pseudoinverse of A is denoted by $A^{\#}$, and it is given by

$$A = VS^{\#}U^{*}. \tag{A.19}$$

The pseudoinverse has its main application in *regularization* of ill-posed problems. For example, let $A \in \mathbb{R}^{m \times r}$, $x \in \mathbb{R}^{n \times 1}$, and $b \in \mathbb{R}^{m \times 1)}$. It is well-known that the solution of the system of linear equations $Ax = b$ is unique only in special cases. If we define $x := A^{\#}b$, x has the following properties:

- If the equations are consistent and has a unique solution, it is given by x.
- If the equations are consistent but has a set of solutions, x is the element of this set with least Euclidean norm.
- If the system is inconsistent, but has a unique least-square solution, this solution is given by x.
- If the system is inconsistent and has a set of least-squares solutions, x is the element of this set with least Euclidean norm.

Thus, the pseudoinverse gives us the exact solution if it exists and is unique, just as the ordinary inverse.

B PROOFS

Many of the proofs below are based on the fact that if

$$\langle f, g\rangle = \langle h, g\rangle \qquad \text{then we say } f = h, \tag{B.1}$$

with the understanding that the equality does not need to hold point wise. For 'nice' functions, we expect that it will, however.

B.1 Theorem 3

Proof

1. The operator $T : H \to L^2(\mu)$ is an operator between Hilbert spaces. It is a fundamental property of Hilbert spaces that such operators have a unique adjoint T^*. A proof of this is given in [43].

2. For any $f \in H$, $g \in L^2(\mu)$,

$$\langle \mathrm{T}^* g, f\rangle_H = \langle g(m), \mathrm{T}f\rangle_{L^2(\mu)} \tag{B.2}$$

$$= \int d\mu(m)\, g \overline{\int dt\, f(t)\overline{h_m(t)}} \tag{B.3}$$

$$= \langle \int d\mu(m) h_m g, f\rangle_H, \tag{B.4}$$

   proving the theorem.

3. From above we see that the operator $\mathrm{G} = \mathrm{T}^*\mathrm{T}$ exist. Since $\|\tilde{f}\|^2_{L^2(\mu)} = \langle \mathrm{T}f, \mathrm{T}f\rangle_{L^2(\mu)} = \langle \mathrm{T}^*\mathrm{T}f, f\rangle_H$, the frame condition (3.21) can be written

$$A\langle f, f\rangle_H \le \langle \mathrm{T}^*\mathrm{T}f, f\rangle_H \le B\langle f, f\rangle_H.$$

   By the argument in (3.24), the theorem follows. Now, $\|Gf\|^2_H = \langle \mathrm{T}f, \mathrm{T}f\rangle \le B\|f\|^2_H$, its norm is bounded;

$$\|G\| = \sup_f\{\|Gf\| : f \in H, \|f\| \le 1\} \le B.$$

4. Above, we found that $A\mathrm{Id} \le \mathrm{G}$, and $\mathrm{G} \le B\mathrm{Id}$. These inequalities imply $\mathrm{G}^{-1} \le A^{-1}\mathrm{Id}$ and $B^{-1}\mathrm{Id} \le \mathrm{G}^{-1}$.

5. We find

$$\begin{aligned}\langle \mathrm{G}f, g\rangle_H &= \langle \mathrm{T}^*\mathrm{T}f, g\rangle_H = \langle \mathrm{T}f, \mathrm{T}g\rangle_{L^2(\mu)} \\ &= \int d\mu(m)\, \langle f, h_m\rangle_H \langle h_m, g\rangle_H \\ &= \left\langle \int d\mu(m)\, \langle f, h_m\rangle_H h_m, g \right\rangle_H,\end{aligned}$$

   where we changed the order of integration in the last step.

■

B.2 Theorem 4

Proof

1. That the operator $S = \mathrm{G}^{-1}\mathrm{T}^*$ is a left inverse of T follows from

$$\mathrm{ST} = (\mathrm{T}^*\mathrm{T})^{-1}\mathrm{T}^*\mathrm{T} = \mathrm{Id}, \tag{B.5}$$

where we used the definition of the metric operator G.

2. By using (3.23), moving G^{-1} under the integral sign and using the definition of the dual frame vectors,

$$\begin{aligned}
\langle \mathrm{S}g, f\rangle_{L^2(\mu)} &= \langle \mathrm{G}^{-}\mathrm{T}^*g, f\rangle_{L^2(\mu)} \\
&= \left\langle \mathrm{G}^{-1}\int d\mu(m)g(m)h_m, f \right\rangle_{L^2(\mu)} \\
&= \left\langle \int d\mu(m)g(m)\overline{h^m}, f \right\rangle_{L^2(\mu)}.
\end{aligned}$$

3. Let $f \in H$, and $\tilde{f}(m) = (\mathrm{T}f)(m) = \langle f, h_m\rangle_H$, Then, since S is a left inverse of T, it follows from the results above that

$$f = \mathrm{ST}f = \mathrm{S}\tilde{f} = \langle \tilde{f}, h^m\rangle_{L^2(\mu)}. \tag{B.6}$$

4. By the closest point property of Hilbert spaces [43], we may write any $g \in L^2(\mu)$ uniquely $g = g_\perp + \tilde{f}$, where $g_\| \in \mathcal{R}_T$, and $g_\perp \in \mathcal{R}_T^\perp$.

   - We show that $\mathrm{P}g_\| = g_\|$: Since $g_\| \in \mathcal{R}_T$, there is a $f \in H$ such that $g_\| = \mathrm{T}f$. Thus $\mathrm{P}g_\| = \mathrm{PT}f = \mathrm{T}(\mathrm{T}^*\mathrm{T})^{-1}\mathrm{T}^*\mathrm{T}f = \mathrm{T}f = g_\|$.
   - We show that $\mathrm{P}g_\perp = g_\perp$: Since $g_\perp \in \mathcal{R}_T^\perp$, we have for all $\mathrm{T}f = \tilde{f} \in \mathcal{R}_T$,

$$0 = \langle g_\perp, \tilde{f}\rangle_{L^2(\mu)} = \langle \mathrm{T}^*g_\perp, f\rangle_H. \tag{B.7}$$

   Hence, $\mathrm{T}^*g_\perp = 0$.

So $\mathrm{P}g = g_\|$, and P is the orthogonal projection onto $\mathcal{R}_T$.

5.

$$\begin{aligned}
(\mathrm{P}g)(m) &= (\mathrm{TS}g)(m) = \langle \mathrm{S}g, h_m\rangle_H(m) \\
&= \left\langle \int d\mu(m')\, g(m')\overline{h^{m'}}, h_m \right\rangle_H (m) \\
&= \int_M d\mu(m')\, g(m')\langle h_m, h^{m'}\rangle_H \\
&= \int_M d\mu(m')\, g(m')K(h_m, h^{m'})
\end{aligned}$$

6. The discrepancy $\|g - g_{\mathcal{F}}\|^2_{L^2(\mu)}$ is minimized when $g_{\mathcal{F}}$ is the unique orthogonal projection of g onto the range of T, by the closest point property of Hilbert spaces, [43]. Thus, $g_{\mathcal{F}} = \mathrm{P}g = \mathrm{TS}g$. The unique vector $f \in H$ whose transform equals $g_{\mathcal{F}}$ satisfies

$$\mathrm{T}f = \mathrm{TS}g, \tag{B.8}$$

which gives $f = \mathrm{S}g$.

7. Any g satisfying $\mathrm{S}g = f$ can be written $g = \mathrm{T}f + g_\perp$, $g_\perp \in \mathcal{R}_T^\perp$. Hence, $\mathrm{T}f \perp g_\perp$, so we can write

$$\|g\|^2_{L^2(\mu)} = \|\mathrm{T}f + g_\perp\|^2_{L^2(\mu)} = \|\mathrm{T}f\|^2_{L^2(\mu)} + \|g_\perp\|_{L^2(\mu)} \geq \|\mathrm{T}f\|^2_{L^2(\mu)}, \tag{B.9}$$

with equality only if $g_\perp \equiv 0$.

■

B.3 Theorem 5 (Parseval)

The proof uses the following standard result [43];

Theorem 25 (The polarization identity) *Let $H = (U, \langle\cdot,\cdot\rangle_U)$ and denote the norm on H induced by the inner product by $\|\cdot\|_H$. Then, for any $f, g \in H$,*

$$\langle f, g\rangle_H = \frac{1}{4}\sum_{n=0}^{3} i^n \|f + ig\|^2_H \tag{B.10}$$

□

Proof of Parseval's Theorem
First note that any Hilbert space is also necessarily a inner product space, so the polarization identity is valid in any Hilbert space. The frame condition reads

$$\|\mathrm{T}(f+g)\|^2_{L^2(\mu)} = A\|f+g\|^2_H, .$$

Using the polarization identity,

$$\langle f, g\rangle_H = \frac{1}{4}\sum_{n=0}^{3} i^n \|f + ig\|^2_H \tag{B.11}$$

$$= \frac{A^{-1}}{4}\sum_{n=0}^{3} i^n \|T(f + ig)\|^2_{L^2(\mu)} \tag{B.12}$$

$$= \frac{A^{-1}}{4}\sum_{n=0}^{3} i^n \|Tf + iTg\|^2_{L^2(\mu)} \tag{B.13}$$

$$= A^{-1}\langle \mathrm{T}f, \mathrm{T}g\rangle_{L^2(\mu)}, \tag{B.14}$$

where we used the frame condition to get (B.12), the linearity of T to get (B.13) and the polarization identity a second time to get (B.14). ■

B.4 Theorem 12

Proof By the definition of the wavelet packets, we have for $\epsilon \in \{0,1\}$,

$$\psi_{2f+\epsilon,j,k} = 2^{-j/2}\psi_{2f+\epsilon}(2^{-j}t-k) \tag{B.15}$$
$$= 2^{-j/2}2^{-1/2}\sum_{n\in\mathbb{Z}} h_\epsilon[n]\psi_{f+\epsilon}(2(2^{-j}t-k)-n) \tag{B.16}$$
$$= s^{-(j-1)/2}\sum_{n\in\mathbb{Z}} h_\epsilon[n]\psi_{f+\epsilon}(2^{-(j-1)}t-(2k+n)) \tag{B.17}$$
$$= \sum_{n\in\mathbb{Z}} h_\epsilon[n-2k]\psi_{f+\epsilon,j-1,n}(t) \tag{B.18}$$

Using this result, we can recursively find the wavelet packet coefficients.

$$c_{2f+\epsilon,j,k} = \langle f, \psi_{2f+\epsilon,j,k}\rangle \tag{B.19}$$
$$= \int dt\, f(t)\overline{\sum_{n\in\mathbb{Z}} h_\epsilon[n-2k]}\psi_{f+\epsilon,j-1,n}(t) \tag{B.20}$$
$$= \sum_{n\in\mathbb{Z}} \overline{h_\epsilon[n-2k]}\langle f, \psi_{f+\epsilon,j-1,n}\rangle \tag{B.21}$$
$$= (\mathrm{H}_\epsilon c_{f+\epsilon,j-1})[k] \tag{B.22}$$

Setting $f \equiv 0$, this also prove Theorem 8 if we identify $\psi_{0,j,k}$ with the scaling function $\phi_{j,k}$ and $\psi_{1,j,k}$ with the wavelet $\psi_{j,k}$, and the sequences $c_{0,j}$ and $c_{1,j}$ with c_j and d_j, respectively. ∎

B.5 Kalman filter

The conventional Kalman filter is given by [2]

$$\hat{x}_{k+1} = A\hat{x}_k + K_k(y_k - C\hat{x}_x) \tag{B.23}$$
$$\tilde{P}_{k+1} = A\tilde{P}_k A^T + Q - K_k(A\tilde{P}_k C^T + S)^T \tag{B.24}$$
$$K_k = (A\tilde{P}_k C^T + S)(C\tilde{P}_k C^T + R)^{-1} \tag{B.25}$$
$$\tag{B.26}$$

when there is no input ($u \equiv 0$). We have written $\tilde{P}_k$ for the more common $\tilde{P}_{k|k-1}$ and, similarly $\hat{x}_k$ instead of $\hat{x}_{k|k-1}$. Now, by replacing $\tilde{P}_{k+1}$ by $\Sigma^s - P_k$, we get

$$K_k = (\underbrace{A\Sigma^s C^T + S}_{=G} - AP_k^T C)(\underbrace{C\Sigma^s C^T + R}_{=\Lambda_0} - CP_k C^T)^{-1} \tag{B.27}$$
$$= (G - AP_k C^T)(\Lambda_0 - CP_k C^T)^{-1}, \tag{B.28}$$

which is identical to (6.36). We have used some of the relations in Theorem 22. For $\tilde{P}_k$ we find from (B.24) using $Q = \Sigma^s - A\Sigma^s A^T$,

$$\Sigma^s - P_{k+1} = A\Sigma^s A^T - AP_k A^T + Q - K_k(\underbrace{A\Sigma^s C^T + S}_{=G} - AP_k C^T)^T \tag{B.29}$$
$$= A\Sigma^s A^T - AP_k A^T + \Sigma^s - A\Sigma A^T - K_k(G - AP_k C^T)^T \tag{B.30}$$

So

$$P_{k+1} = AP_kA^T + K_k(G - AP_kC^T)^T, \tag{B.31}$$

which is identical to (6.35).

C OTHER RESULTS

In this Appendix we present the results from some of the analyses not included in the main text.

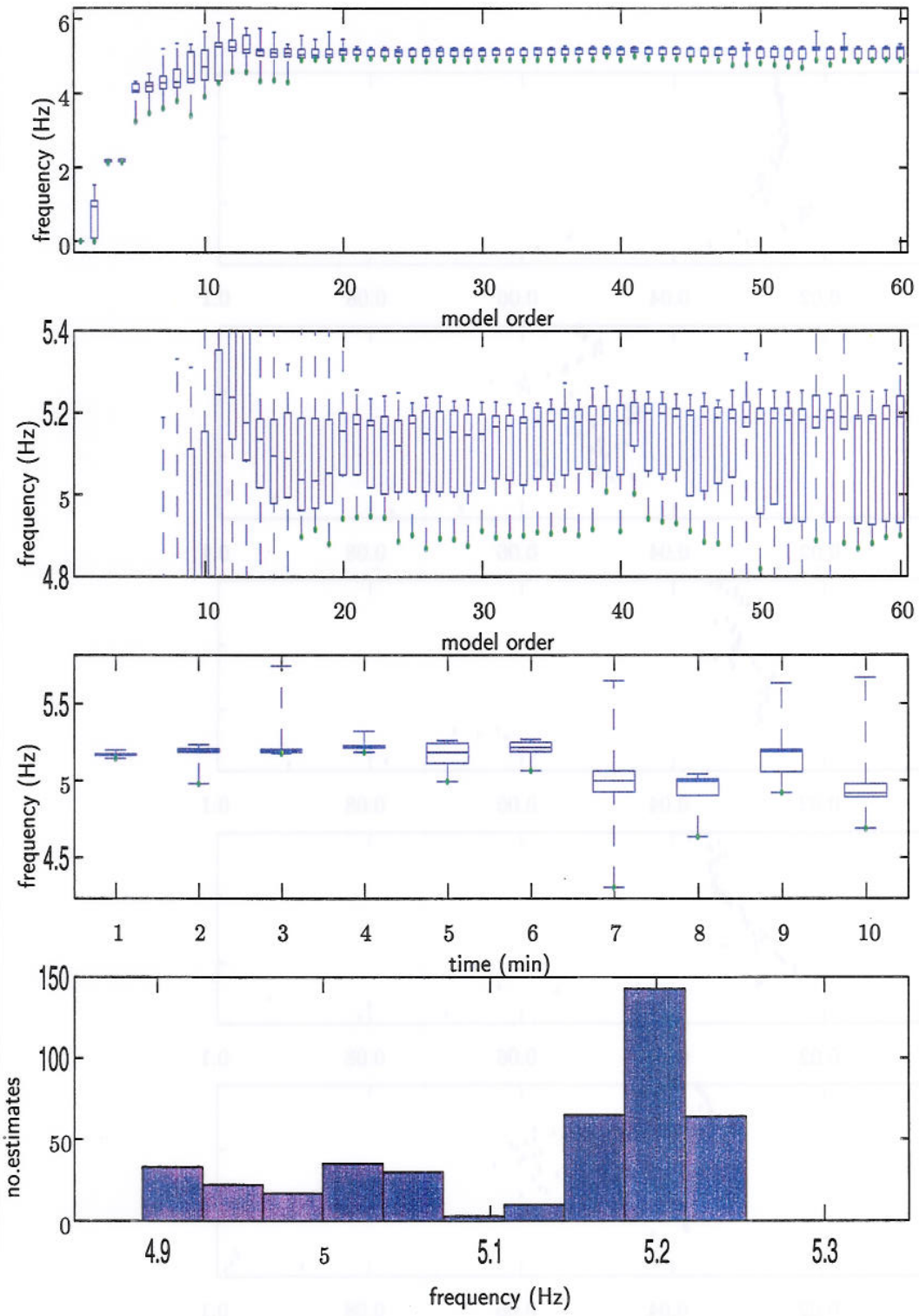


Figure C.1 *Box-and-whisker plot showing the spread of the estimated frequency of the 5.2 Hz vibration in 10 separate time series at model orders in the range* $[1, 60]$ *(top and second). For each of these 10 time series, the spread of the estimates for model orders in the range* $[18, 60]$ *at different times is shown in the third plot. The distribution of the middle* 90% *of the data is shown at the bottom.*

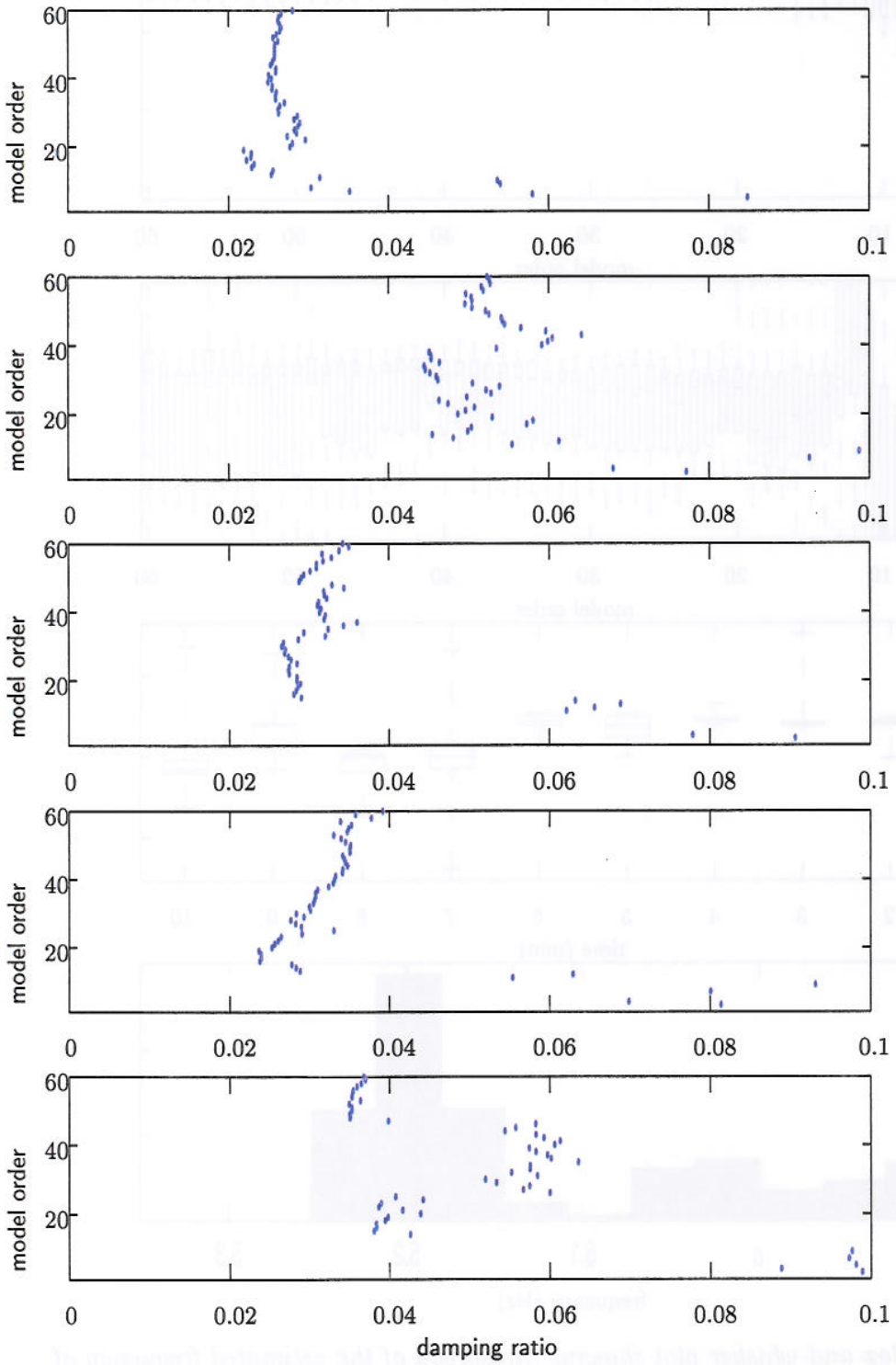


Figure C.2 *Estimated damping ratios for the mode at 5.2 Hz varying model orders. From top to bottom, the plots show damping estimates based on consecutive 5-minutes intervals of data.*

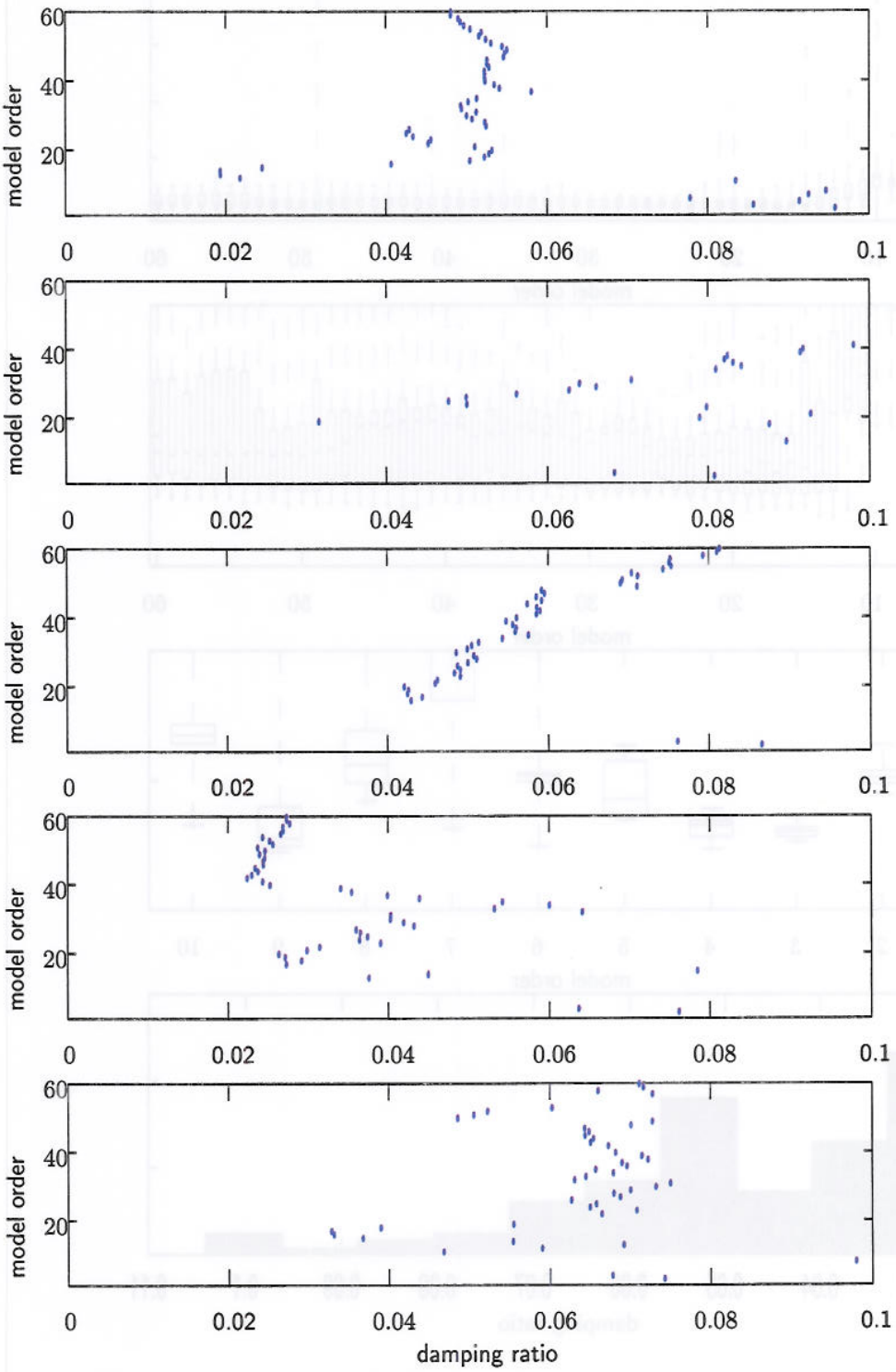


Figure C.3 *Estimated damping ratios for the mode at 5.2 Hz varying model orders. From top to bottom, the plots show damping estimates based on consecutive 5-minutes intervals of data. The data used are the 25 minutes following the data used in Figure C.2.*

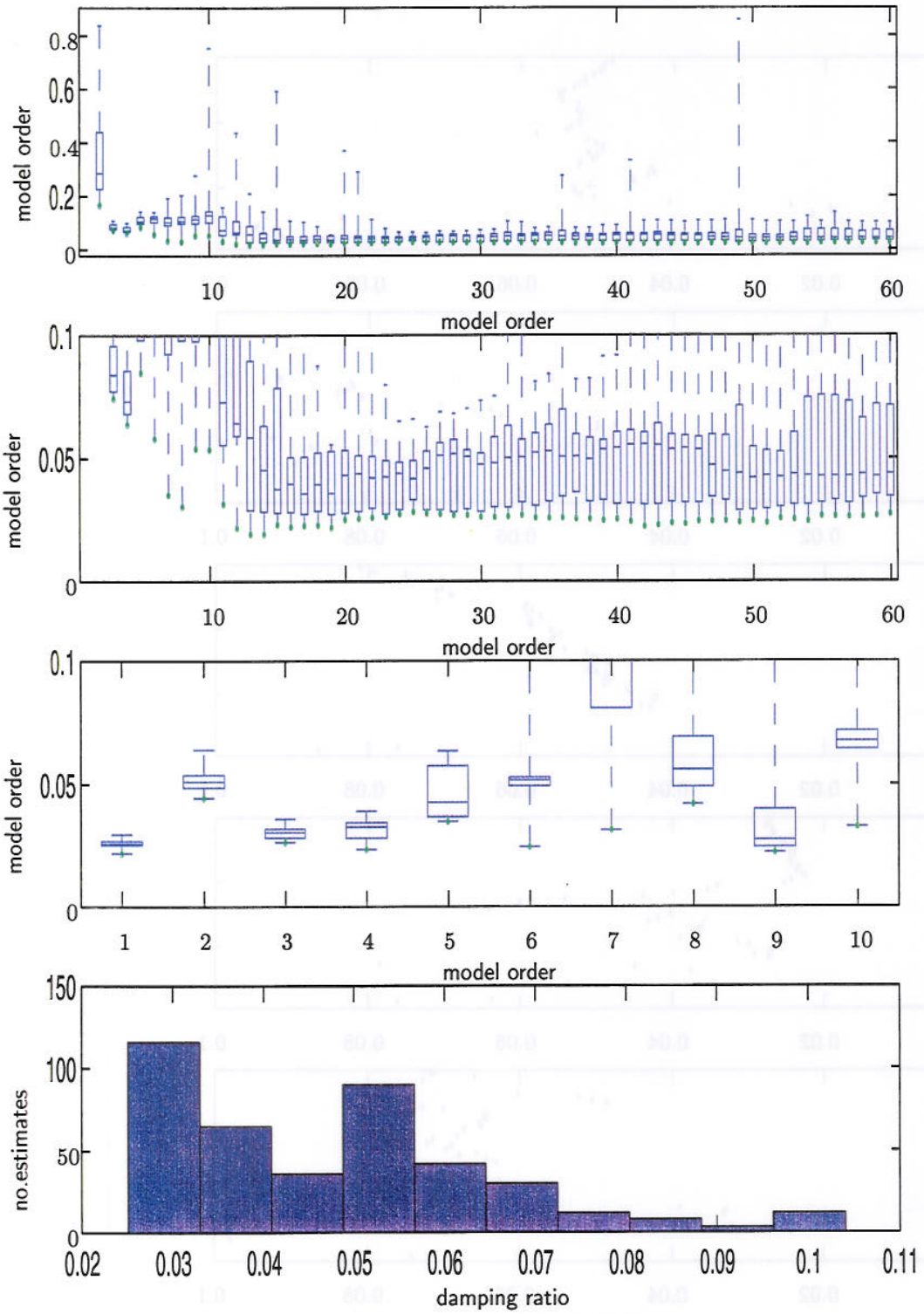


Figure C.4 *Box-and-whisker plot showing the spread of the estimated damping ratio of the 5.2 Hz vibration in 10 separate time series at model orders in the range* $[1, 60]$ *(top and second). For each of these 10 time series, the spread of the estimates for model orders in the range* $[15, 60]$ *at different times is shown in the third plot. The distribution of the middle* 90% *of the data is shown at the bottom.*

D `wdmf` - WAVELET-BASED MATCHED FILTERING

This appendix presents the first version of general stand-alone C++ program, which performs matched filtering in the wavelet domain. This first prototype has since been further developed by both the present author and Emil Urnes, but this program is not included here.

D.1 Syntax

Given a data file and a reference signal, the program `wdmf` detects occurences of signals with a waveform similar to the given reference signal. It returns the output of a wavelet domain mathced filter and, optionally, estimates of the amplitude and time of arrival for each detected signal-

The command line syntax for the is

```
wdmf wavelet data signal levels outfile [threshold index amplitude].
```

The meaning of the parameters are

`wavelet` is a file containing the wavelet filter coefficients specifying the type of wavelet to be used. Its contents must be of the format`h0(1) h0(2) ... h0(n) * h1(0) h1(1) ... h1(n)`, where h0 are the coefficients in the low-pass filter and h1 the corresponding high-pass filter. The asterix * serves as a delimiter.

`data` is a file containing the data to be analysed.

`signal` is a file containing the the reference signal to be used in the matched filter.

`levels` is a sequence of 0s and 1s with *no spaces* specifying which levels of the wavelet decomposition are to be used in the matched filtering. The first entry refers to the approximation level, and for $n > 1$, the nth entry refers to detail level $n - 1$. A 1 means that the corresponding level should be used in the matched filter, whereas a 0 means that the level should be included.

`outfile` is the name of the file to which the detector output should be written. The detector output is scaled in such a manner that at any instant, it can be regarded as an estimate of the amplitude of a signal similar to that contained in the file `signal` *as if it ocurred* at that instant.

Optionally, three additional parameters may be specified. They all relate to estimating the time of arrival and the amplitude of signals similar to the one specified in `signal`. Their meaning are as follows.

`threshold` is a number defining the lowest detector output the program should register in the output files below.

`index` is the name of the file to which the element numbers (indices) of the detector output which correspond to a local maxima should be written. The nth element in this file corresponds to the nth element in in `amplitude`.

`amplitude` is the name of the file to which the estimated amplitudes should be written. The nth element in this file corresponds to the nth element in in `index`.

D.2 Example of use

An example of the use is as follows.

```
tdwf db6 sensor_a.dat trans.dat 00011 detector.dat 12 i.dat a.dat
```

Here, we search for for waveforms similar to `trans.dat` in `sensor_a.dat` using the two highest detail levels levels in a 4-level wavelet decomposition. We record the detector outputs which correspond to detection of signals with amplitudes greater than 12 in the files `i.dat` and `a.dat`. In MATLAB, we may write

```
>> Ts=0.011; % Define your own sampling period Ts here
>> load i.dat a.dat detector.dat
>> subplot(2,1,1)
>> bar([( i(1)-1) : ( i(length(i))-1 )]*Ts, a)
>> xlabel('Time of arrival (s)')
>> ylabel('Estimated amplitude')
>> subplot(2,1,2)
>> plot([0:length(detector)-1]*Ts,abs(detector))
>> xlabel('Time of arrival (s)')
>> ylabel('Estimated amplitude')
```

D.3 Program listing

The program for performing wavelet-based matchewd filtering is called `wdmf.C` (short for wavelet domain matched filtering). It needs the module `Signal.h`, which contains general classes and methods for signal processing.

D.3.1 `Signal.h` - Signal processing module

```
#include<vector>
#include<iostream.h>
#include<math.h>
typedef double Tsample ;
typedef vector<Tsample> Tsignal;
typedef Tsignal::size_type size_type;

class Signal{
```

```
private:
  Tsignal s_data;
//  size_type length;
public:
  // CONSTRUCTORS AND OPERATORS
  Signal();
  Signal(size_type initial_size): s_data(initial_size){};
  //,length(initial_size){};
  Signal(size_type initial_size, Tsample data)
    : s_data(initial_size,data){};                    //,length(initial_size){};
  Signal(Tsample * from, size_type elements);        //double a[10];Signal s(a,7);
  ~Signal(){};

  Tsample& operator[]( size_type i){return s_data[i];};
  Tsample operator[]( size_type i)const{return s_data[i];};
  Signal& operator=(Signal& rhs);
  // DATA HANDLING FUNCTIONS
  Signal& downsample2(const Signal & in);
  Tsignal::iterator begin() {return s_data.begin();};                    //not
  Tsignal::iterator const end(){return s_data.end();};                  //  const!
  size_type size()const{return s_data.size();};
  Tsample vector_product ( Signal& b);                              //, Signal& b);
  Tsample vector_product ( Signal& b, size_type start_a,
                                  size_type end_a,size_type start_b);
  Signal& convolute (const Signal& h, const Signal& x,size_type q0);
  friend ostream& operator<<(ostream& out, const Signal& in);
};                                                                  // End Class Signal

inline Signal& Signal::operator=(Signal& rhs){

  if (this==&rhs) return *this;
  s_data=rhs.s_data;
  return rhs;
  };

Signal& Signal::downsample2(const Signal & in){

  /*Down-samples the signal in by two, keeping the _odd_ indices. It is equal
    to MATLAB's dyaddown(x), which keeps the even indices on vectors starting
    with one. */

  size_type l_d=in.size()/2;
            //Length of downsampled signal (integer division).
  Tsignal tmp(l_d);
  for(size_type i=0;i<l_d;i++){
    tmp[i]=in[2*i+1];
  }
  s_data=tmp;       //There is probably a better way, for instance  vector.resize()
  return (*this);
```

```
}

Signal::Signal(Tsample* from, size_type n_elements):s_data(n_elements){
  for(int i=0;i<n_elements;i++)
    s_data[i]=*from++;
}
Signal::Signal(){}

inline Tsample Signal::vector_product( Signal& b){
  // length(a)<=length(b)
  Tsample out=0.0;
  Tsignal::iterator first1=begin();
  Tsignal::iterator last1=end();
  Tsignal::iterator first2=b.begin();
  while (first1 ≠ last1)
    out +=*first1++ * *first2++;
  return out;
}



inline Tsample Signal::vector_product( Signal& b,size_type start_a, size_type
end_a,
                size_type start_b=0){
  // length(a)<=length(b)
//cout<<"a= "<<*this<<endl;
  Tsample out=0.0;
  size_type first1=start_a;
  size_type last1=end_a;
  size_type first2;
  first2=start_b;
//cout<<"b= "<<b<<endl;
  while (first1 ≤ last1) {
//     cout<<"vp:first1, first2= "<<first1<<", "<<first2<<": ";
    if(first1≥s_data.size()) {cout≪"errorfirst1>=a.size";exit(1);}
    if(first2≥b.size()) {cout≪"error:first2>b.size";exit(1);}
//     cout<<s_data[first1]<<" * "<<b[first2]<<"="<<s_data[first1] * b[first2];
    out +=s_data[first1++] * b[first2++];
//   cout<<" out="<<out<<endl;

  }
  return out;

}

ostream& operator≪(ostream& out, Signal& in){
  Tsignal::iterator first=in.begin();
  Tsignal::iterator last=in.end();
if (first≥last) cout≪"error in cout"≪endl;
```

```
  while(first≠last)
    out≪*first++≪" ";
  return out;
}

Signal& Signal::convolute(const Signal& h, const Signal& x, size_type offset=0){
  size_type l_y=h.size()+x.size()-1-offset;
  size_type l_h=h.size();
  size_type l_x=x.size()-offset;
  Signal tmp(l_y,0.0);
  s_data=tmp.s_data;
  size_type t=0;
  size_type k=0;
  for(t=l_h-1;t<l_x;t++){
    for(k=0; k<l_h;k++){
      this→s_data[t]+=h[k]* x[t-k+offset];
     }
  }
  for( t=0;t<l_h-1;t++){                          //FIRST PART OF s_data
    for( k=0; k<l_h-(l_h-1-t);k++){
      this→s_data[t]+=h[k]*x[t-k+offset];
    }
  }
  for( t=l_x;t<l_y;t++){                          //LAST PART OF s_data
    for( k=t+l_h-l_y; k<l_h;k++){
      this→s_data[t]+=h[k]*x[t-k+offset];
    }
  }
  return *this;
}

/*************************************************/

class W_filter{
public:
  Signal high;
  Signal low;
  W_filter(Signal a,Signal b){high=b;low=a;};
  W_filter(){};
  ~W_filter(){};
};

class WT{
  vector<Signal*> ptable;

public:
  size_type original_size()const{return L[L.size()-1];}
  vector<size_type> L;
  size_type n_scales;              //number of detail scales in the decomposition
```

```
  WT(){n_scales=0;};
  WT(size_type n){vector<Signal*> tmp(n+1);ptable=tmp;};
  ~WT();
  Signal& cA()const{return *(ptable[ptable.size()-1]);};
  Signal& cD(size_type j)const{if(j< ptable.size()) return *(ptable[j-1]);
cout<<"feil";};
  void one_scale_decomposition(W_filter& h, Signal &s, size_type
                                        scale,Signal &approx);
  void wavelet_transform(const W_filter& h, const Signal &w, size_type scales,
                  size_type q0);
  friend ostream& operator<<(ostream& out, const Signal& in);
};

ostream& operator<<(ostream& out, const WT& in){
  cout<<"Approximation"<<endl;          //To cout because out might be to a file
  out<<in.cA();              //If out is a file, we only want data to be written to it
  cout<<endl;
   //for(size_type i=1; i<=in.n_scales;i++){
   for(size_type i=in.n_scales; i>1;i--){
    cout<<"Details "<<i<<endl;
    out<<in.cD(i);
    cout<<endl;
   }
  return out;
}

WT::~WT(){
  cout<<"~WT"<<endl;
  for(size_type i=0;i<ptable.size();i++){
    delete ptable[i];
  }
}

void WT::wavelet_transform (const W_filter& h, const Signal &w, size_type
scales,
                  size_type q0=0) {
  Signal approx;
  Signal c_s(w.size()-q0);                              //Current signal to work on
  for(size_type i=0;i<c_s.size();i++)c_s[i]=w[i+q0];
  vector<Signal*> init(scales+1);
  ptable=init;
  vector<size_type> L_init(scales+2,33);
  //approximation + scales + original length
  L=L_init;
  L[L.size()-1]=w.size()-q0;
  n_scales=scales;
  for(size_type c_scale=0; c_scale<ptable.size()-1;c_scale++){     //current scale
      ptable[c_scale]=new Signal;
      ptable[c_scale]=&ptable[c_scale]→
```

```
        downsample2(ptable[c_scale]→convolute(h.high,c_s));
      L[L.size()-2-c_scale]=ptable[c_scale]→size();                    //
      c_s=approx.downsample2(approx.convolute(h.low,c_s));
  }
  ptable[ptable.size()-1]=new Signal;
  (*ptable[ptable.size()-1])=c_s;

  L[0]=ptable[ptable.size()-1]→size();
}

inline size_type pow_i(double d, int i){
  return static_cast<size_type>(pow(d,i));
}
double log2(double x){
  return log(x)/log(2);
}
inline size_type qfactor(size_type q0){
  double out;
  if (q0 > 1)
    out=floor(log2(0.5+q0));
  else if (q0==1||q0==0)
    out=0;
  return static_cast<size_type>(out);
}


//CLASS DECLARATION TIWT
class TIWT{

 vector< vector<Signal*> > ptable;
 size_type org_length;                          //Length of original signal
 size_type* whichscales;


public:
 void ti_wavelet_transform(const W_filter& h, const Signal & x,
                           size_type nscales);
 void qfactor2j(size_type& q0, size_type& q1,size_type t,size_type
                scale)const;
  Signal& cD(size_type shift,size_type scale)const;
 Signal& cA(size_type shift)const;
 TIWT(){};
 ~TIWT();
 TIWT(int nscales, Tsample* W,size_type length);
 vector<Signal*>* wt(const W_filter& h, const Signal &w, size_type scales,
                size_type q0);
 void tiwt_mf(Signal& y, const WT& s,const size_type *const whichscales,
                size_type n_scales);
 friend ostream& operator≪(ostream& out,const TIWT& in);
```

```
 double norm(size_type nscales,const size_type* w_scales);
};
//END CLASS TIWT DECLARATION//END CLASS TIWT DECLARATION



inline Signal& TIWT::cD(size_type shift, size_type scale)const{
  if (scale==0) {cout≪"TIWT::cD error.  Scale=0 undefined"≪endl;exit(1);}
  size_type scale_index=ptable[0].size()-scale;
    //Approximation + Detail levels -the scale we want=correct index
 if (scale_index≥ptable[shift].size()) {
    cout≪"TIWT::cD error.  Too high index"≪endl;
    exit(1);
 }
  return *(ptable[shift][scale_index]);
}

inline Signal& TIWT::cA(size_type shift)const{
  if (shift≥ptable.size()) {
     cout≪"TIWT::cA error.  To high shift"≪endl;
     exit(1);
  }
  return *(ptable[shift][0]);
}

void TIWT::tiwt_mf(Signal& out, const WT& s,
        const size_type* const whichscales, size_type n_scales){
  char tmp;
  size_type ly=s.L[s.L.size()-1]-org_length;
  size_type q0; size_type q1;
  cout≪"org_length="≪org_length≪endl;

  Signal y(s.L[s.L.size()-1],0.0);   //Detector output initialised with length and data
  //cout<<"y="<<y<<endl;
  for(int t=0;t≤ly;t++){
    for(int j=1;j≤n_scales;j++){
      if (whichscales[j]){
        qfactor2j(q0, q1, t, j);
        size_type ls=this→cD(q0,j).size();
//        cout<<"mf:q0="<<q0<<", q1="<<q1<<", scale="<<j;
//        cout<<" fra(q1),til(q1+ls-1)  "<<q1<<","<<q1+ls-1<<endl;
        //y[t]+=s.cD(j).vector_product((*this).cD(q0,j),0,s.cD(j).size()-1,q1);
          y[t]+=this→cD(q0,j).vector_product(s.cD(j),0,
                    ls-1,q1);
      }
    }
  }
  cout≪"Now approximations.."≪endl;
  if (whichscales[0]≠0){
```

```
    for(size_type t=0;t≤ly;t++){
      qfactor2j(q0, q1, t, n_scales);
      size_type ls=this→cA(q0).size();
      y[t]+=this→cA(q0).vector_product(s.cA(),0,ls-1,q1);
    }
  }
  Tsample nrm_sq=pow(this→norm(n_scales, whichscales),2 );
  for(size_type i=0;i<y.size();i++){
    y[i]=y[i]/nrm_sq;
  }
  out=y;
}

void TIWT::qfactor2j(size_type& q0, size_type& q1,size_type t,size_type scale)
  const{
  q1=static_cast<int>(floor(t/pow_i(2.0,scale)));
  q0=t-q1*pow_i(2.0,scale);
}

TIWT::TIWT();


TIWT::~TIWT(){
  for(int j =0; j<ptable.size();j++){
    for(int q0=0;q0<ptable[j].size();q0++){
      delete ptable[j][q0];
    }
  }
}

ostream& operator≪(ostream& out, const TIWT& in){

  for(size_type q0=0;q0<in.ptable.size();q0++){
    out≪"Shift "≪q0≪endl;
    out≪"W.cA("≪q0≪")="≪in.cA(q0)≪endl;
    for(size_type j=in.ptable[0].size()-1;j>in.ptable[0].size()-in.ptable[q0].size();j--){
      out≪"W.cD("≪q0≪","≪j≪")="≪in.cD(q0,j)≪endl;
    }
  }
  return out;
}

vector<Signal*>* TIWT::wt(const W_filter& h, const Signal &w, size_type scales,
                  size_type q0=0){
  Signal approx;
  Signal c_s(w.size()+q0,0.0);              //Current signal to work on- to +, 0.0
  for(size_type i=0;i<w.size();i++)c_s[i+q0]=w[i];

  vector<Signal*>* peker;
```

```
  peker = new vector<Signal*> (scales+1);
  Signal* ptmp;
  for(size_type c_scale=peker→size()-1; c_scale≥1;c_scale--){      //current scale
      ptmp = new Signal;
      *ptmp=( ptmp→
        downsample2( ptmp→convolute(h.high,c_s)) );
      (*peker)[c_scale]=ptmp;
      //cout<<"wt:scale= "<<c_scale<<" c_s= "<<c_s<<endl;
      c_s=approx.downsample2(approx.convolute(h.low,c_s));
  }
  //cout<<"wt: c_s= "<<c_s<<endl;
  ptmp=new Signal;
  *ptmp=c_s;
  (*peker)[0]=ptmp;
//for(int i =0;i<(*peker).size();i++) cout<<"—"<<*((*peker)[i])<<endl;
  return peker;
}

void TIWT::ti_wavelet_transform(const W_filter& h,const Signal & x,
                    size_type nscales){

  char tmpc;
  ptable=vector<vector<Signal*> > (pow_i(2.0,nscales));
  vector<Signal*>* peker;                    //pointer to wavelet transformation
  org_length=x.size();
  for(int q0=0;q0<pow_i(2.0,nscales);q0++){
    //cout<<"q0="<<q0<<endl;
    size_type qf= qfactor(q0);
    cout≪"Now transforming shift q0="≪q0≪endl;
    ptable[q0]=vector<Signal*>(nscales+1-qf);  //details + approximation - factor
    peker=wt(h,x,nscales,q0);                  //q0 is left shiftRIGHT!!!
    if (!peker) cout≪"error"≪endl;

    for(int j=0;j<nscales+1;j++){                                  //details

      if (j≤nscales-qf){
        ptable[q0][j]=(*peker)[j];
        //cout<<"scala="<<nscales+1-j<<" ";
        //cout<<"*ptable["<<q0<<"]["<<j<<"]="<<*ptable[q0][j]<<endl;
      }
      else {
        //cout<<"delete "<<j<<endl;
        delete (*peker)[j];
      }
    }
    delete peker;
  }
}
```

```
double TIWT::norm(size_type nscales, const size_type* wscales){

  Tsample norm_sq=0.0;
  for(size_type j=1;j≤nscales;j++){
    if(wscales[j]){
      norm_sq+=this→cD(0,j).vector_product(this→cD(0,j));
    }
  }
  if(wscales[0]){
    norm_sq+=this→cA(0).vector_product(this→cA(0));
  }
  return sqrt(norm_sq);
}
```

D.3.2 `wdmf.C` - Main program

```
#include"signalr.h"
#include<fstream.h>
#include<iostream.h>
//#include<stream.h>


void error(const char* const message_1,const char * const message_2=""){
  cerr≪message_1≪' '≪message_2≪endl;
  exit(1);
}

size_type file_size(const char * const filename){

  ifstream file(filename);

  if (!file){
    error("error in opening",filename);
```

```
  }

  double temp;
  size_type i=0;

  while(file≫temp){
    i++;
  }
  cout≪"f_s";
 return i;
}
void matlab_wavelet( WT& wx, const char* const data_name,
                const char* const L_name){

  /* Writes the wavelet transform wx to disk in MATLAB's format; i.e.
     to the two files specified by data_name and L_name, which can be used
     as [C L] as output by MATLAB's wavedec. */

  ofstream to1(data_name);                    //Open output file for writing
  if (!to1){
    error("error in opening",data_name);
  }
  ofstream to2(L_name);                       //Open output file for writing
  if (!to2){
    error("error in opening",L_name);
  }
  to1≪wx;
  for(size_type i =0;i<wx.L.size();i++){
    to2≪wx.L[i]≪" ";
  }
}
Tsample maximum(const Signal& in, const size_type i_a, const size_type i_b,
                size_type& i_max){

  if( (i_a≥in.size())||(i_b≥in.size()) ){
    error("maximum:  index overflow");
  }
  if( (i_a<0)||(i_b<0) ){
    error("maximum:  index underflow");
  }
  if( (i_b<i_a) ){
    error("maximum:  not a legal index range");
  }

  Tsample cand=in[i_a];
  i_max=i_a;
  for(size_type i=i_a;i≤i_b;i++){
    if(in[i]>cand){
      cand=in[i];
```

```
      i_max=i;
    }
  }
  return cand;
}

void extract(vector<Tsample>& out_amp, vector<size_type>& out_ind,
             const Signal& in, const Tsample thresh, const size_type width){
  vector<size_type> tmps;
  vector<Tsample> tmpt;
  out_amp=tmpt;
  out_ind=tmps;
  size_type i_lower=0;
  size_type ii=0;
  Tsample current_highest=0.0;
  size_type i=0;
  while( i<in.size()-1-width){
                    //for the length of the signal-width do this
    if (in[i]>thresh){                                          //Transient here
      if (i<width){                    //Preventing negative indices into maximum
        i_lower=0;
      }
      else{
        i_lower=i-width;
      }
      if (in[i]≥maximum(in,i_lower,i,ii)){
        current_highest=in[i];                                  //Largest so far
        while( (i<in.size()-1-width)&&
             (maximum(in, i, i+width, ii)>current_highest)) {
          i=ii;                    //this is the index of the highest value so far
          current_highest=in[i];                                //Largest so far
        }
        out_amp.push_back(current_highest);                  //Store amplitude
        out_ind.push_back(i);                                  //Store index
        cout≪"i="≪i≪endl;
      }  //
      i++;
    }  //transient
    else
      i++;
  }
}

void ampest(const WT& wx, const TIWT& ws, const size_type* whichscales,
            const vector<size_type>& index, vector<Tsample>& ahat){

 size_type q0;size_type q1;
 Tsample norm=0.0;
  ahat=vector<Tsample>(index.size(),0.0);
```

```
  for(size_type i=0;i<index.size();i++){//FOR ALL TIMES WITH TRANSIENT
    size_type t=index[i];
    for(size_type j=1;j≤wx.n_scales;j++){
      if (whichscales[j]){
        cout≪"(t,j)=("≪t≪", "≪j≪endl;
        ws.qfactor2j(q0, q1, t, j);
        size_type ls=ws.cD(q0,j).size();
        ahat[i]+=ws.cD(q0,j).vector_product(wx.cD(j),0,
                    ls-1,q1);
      norm+=ws.cD(0,j).vector_product(ws.cD(0,j),0,
                    ls-1);
      }
    }
    if (whichscales[0]){
      ws.qfactor2j(q0, q1, t, wx.n_scales);
      size_type ls=ws.cA(q0).size();
     ahat[t]+=ws.cA(q0).vector_product(wx.cA(),0,ls-1,q1);
     norm+=ws.cA(wx.n_scales).vector_product(ws.cA(wx.n_scales),0,
                    ls-1);
    }
  }  //end loop over t
  for(size_type j=1;j≤wx.n_scales;j++){
      if (whichscales[j]){
        size_type ls=ws.cD(0,j).size();
        norm+=ws.cD(0,j).vector_product(ws.cD(0,j),0,ls-1);
      }
    }
    if (whichscales[0]){
      size_type ls=ws.cA(0).size();
      norm+=ws.cA(0).vector_product(ws.cA(0),0,ls-1);
    }
    for(size_type i=0;i<index.size();i++){ahat[i]=ahat[i]/norm;}
    cout≪"norm="≪norm≪endl;
}


/* —- MAIN PROGRAM —- */


int main(int argc, char* argv[]){

  size_type nscales=4;                                          //
  size_type whichscales[]={0,0,0,0,1};
  Tsample threshold=5;
  size_type width=950;                  //Should be about half the length of the mf
  if (argc≠7){
    error("Format:  wdmf [w-filter][source][m-filter][output][i][a] ");
  }
```

```
/* - Processing command line inputs - */
size_type lwfilter=file_size(argv[1]);                    //Wavelet filter coefficients
ifstream wfilter(argv[1]);
if (!wfilter){
  error("error in opening",argv[1]);
}

size_type lsource=file_size(argv[2]);                              //Source data
ifstream source(argv[2]);
if (!source){
  error("error in opening",argv[2]);
}
cout≪"source file length is "≪lsource≪endl;
double* source_data = new double[lsource];                        //allocate memory
if (!source_data){
  error("Error in allocating array");
}

size_type lmfilter=file_size(argv[3]);              //Matched filter impulse response
ifstream mfilter(argv[3]);
if (!mfilter){
  error("error in opening",argv[3]);
}
double* mfilter_data = new double[lmfilter];                      //allocate memory
if (!mfilter_data){
  error("Error in allocating array");
}

ofstream outind(argv[5]);                              //Open output file for writing
if (!outind){
  error("error in opening",argv[5]);
}
ofstream outamp(argv[6]);                              //Open output file for writing
if (!outamp){
  error("error in opening",argv[6]);
}

ofstream out(argv[4]);                                 //Open output file for writing
if (!out){
  error("error in opening",argv[4]);
}
size_type i=0;

/* - Read input data - */
while(source≫source_data[i]){                                      //Source data
  i++;
}
Signal x(source_data,lsource);
delete[] source_data;
```

```
  i=0;
  while(mfilter≫mfilter_data[i]){                    //Matched filter impulse response
     i++;
  }
  Signal h_mf(mfilter_data,lmfilter);
  delete[] mfilter_data;
  i=0;                                                          //Create wavelet filter
  size_type filterlength;
  double* tmp=new double[lwfilter];
  while(wfilter≫tmp[i]){          //The l∧2-norm is 1 for all filters, so coeffs.<=1.
    if(tmp[i]==66.6) {filterlength=i;}   //66.6 separates lowpass and highpass
    i++;
  }
  Signal h_0(tmp,filterlength);
  Signal h_1(tmp+filterlength+1,filterlength);
  W_filter h(h_0,h_1);
  delete[] tmp;

/*- Compute wavelet transforms and filter - */
  cout≪"Taking WT of input signal..."≪endl;
  WT wx;
  wx.wavelet_transform(h,x,nscales);
  cout≪"Taking TIWT of filter..."≪endl;
  TIWT tiwh_mf;
  tiwh_mf.ti_wavelet_transform(h,h_mf,nscales);

  cout≪"Matched filtering..."≪endl;
  Signal detector;
  tiwh_mf.tiwt_mf(detector,wx,whichscales,nscales);

/* - Clean up and write to disk - */
  cout≪"Writing detector to disk..."≪endl;
  out≪detector;
  //matlab_wavelet(wx,argv[5],argv[6]);
  //cout<<tiwh_mf;

/* - Estimate amplitude - */
  vector<size_type> index;
  vector<Tsample> ampdet;
  cout≪"Extracting..."≪endl;
  extract(ampdet, index, detector,threshold,width);


/* - write to disk - */
  cout≪"Writing exctract to disk..."≪endl;
  cout≪"
index.size()="≪index.size()≪endl≪"ampdet.size()="≪ampdet.size()≪endl;
 if ( index.size()≠ampdet.size() ){
    cout≪"detect:  index and amplitude are of incompatible sizes"≪endl;
```

```
    }
    for (i=0;i<index.size();i++){
      outamp≪ampdet[i]≪" ";
      outind≪index[i]≪" ";
    }
}
```