Spatially Filtering Strain Sensors for Structural Shape Estimation of Intelligent Structures

by

Mark Sven-Erik Andersson

B.S., Aeronautics and Astronautics,
Massachusetts Institute of Technology (1989)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1993

Author .....
Department of Aeronautics and Astronautics
February 5, 1993

Certified by .....
Professor Edward F. Crawley
Professor of Aeronautics and Astronautics
MacVicar Faculty Fellow
Thesis Supervisor

Accepted by .....
Professor Harold Y. Wachman
Chairman, Departmental Committee on Graduate Students

Spatially Filtering Strain Sensors for Structural Shape Estimation of Intelligent Structures

by

Mark Sven-Erik Andersson

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
ON FEBRUARY 5, 1993, IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

Abstract

The design of discrete, highly distributed spatially averaging sensor systems for intelligent structures is investigated. As a starting point, the functional requirements and design parameters for such a sensor system are formulated. It is found that certain spatially averaging strain sensors can be used to satisfy these requirements. The output and transfer function characteristics of spatially averaging sensors with arbitrary spatial weightings are derived for both sinusoidal and exponential strain fields. Desirable spatial weightings are then identified. The performance of nontruncated sensors, and sensors that have a portion that falls beyond a boundary of the structure is derived. A variety of numerical integration schemes are studied for use in the spatial integration of discrete strain measurements made by a sensor array. Such spatial integration is used to estimate the global shape of a structure. Numerical simulations are performed using point, rectangular and Bartlett sensors on a clamped-free beam under a number of generic static loading conditions. The errors encountered in a physical implementation are modeled. An experiment was conducted to experimentally correlate with the results of the numerical simulations. The dynamic mode shapes of pinned-pinned and clamped-free beams are estimated using a variety of sensor weightings. The rolloff of the observability to the mode shapes as the mode number is increased is investigated. Using an array spatially averaging sensors, it is possible to smoothly roll off the observability of the sensor array to the mode shapes of a pinned-pinned beam. However, it is found that exponential curvature components in the mode shapes of clamped-free beams make it extremely difficult to smoothly roll off this observability.

Thesis Supervisor: **Professor Edward F. Crawley**
Title: **Professor of Aeronautics and Astronautics**
MacVicar Faculty Fellow

Acknowledgements

First of all, I would like to thank my parents who first banished me to a boarding school in England while they lived in Sweden, then sent me back to Sweden to another boarding school when they moved to Italy, and finally financed my undergraduate studies at MIT so they could send me across the Atlantic. I am grateful for all their encouragement and support during my enrollment at various educational institutions around the world.

Second, and icke främst, I måste ringraziare mes parents and la nonna who hanno lärt mig comment to use un mucchio of språk dans the stessa sentence:

Till mina häftiga föräldrar, som har frågat, minst en miljard gånger: "När skall du få den där förbannade mastern?" Här är den. Äntligen färdig. Nu får ni jävlar läsa varenda ord jag har skrivit!

E per la nonna, che anche lei mi ha domandato: "Quando lo finisci, questo master?" Non ti devo più dire boh – eccola qua, questa tesi mostruosa. Sei fortunata, che è tutta in inglese – così non la devi leggere!

I would like to thank my advisor, Ed Crawley, for his his constant patience, illuminating suggestions and guidance in the long period of research which culminated in the production of this thesis.

I would also like to thank the following people:

- Malini Ramaswamy for being such a dear friend during the long time I have known her and for all those wonderful years we shared.
- Roman Hachkowski for those awesome times we had in Unified, 3.094, 16.20, 16.53, 16.62x, 16.85, Project Argus, MCS I & II, and that $10K competition we lost. This year we'll win it!
- Mark Barlow for pushing me to finish this tome, and for all the fun we had in setting up the Suns, getting xview to work, playing all those Apache Strike games, and hanging and taking down MODE a few times. Hey! You're back in Boston and you've changed careers!
- Alice LaPierre for being so cheerful and friendly all the time, for forcing me to learn all those rags, for all the help in producing the images for the cover of the UROP '92 guide, and for letting me illustrate the IAP '93 guide.
- Dan Rey and John Ofori-Tenkorang for also taking 42 months to complete their masters degrees.

Finally, I'd like to thank all those excellent people at SERC who constantly badgered me about when I would be done with this here thing. I have enjoyed, and will keep enjoying the privilege of helping you out on the Sun network.

Biography

Mark Andersson was born in Hjo, Sweden, on August 31, 1967, to Emanuela and Sven-Olof Andersson.

Between 1969 and 1978, the author attended English schools in Portugal, Sweden, South Korea and the United States.

Between 1978 and 1983, the author attended Bedales School in Petersfield, Hampshire, England, where he completed Ordinary Level examinations in Math, Physics, Chemistry, Biology, Design, French, Geography, English Language and English Language.

Between 1983 and 1985, the author attended Sigtuna Skolan Humanistiska Läroverket (SSHL) in Sigtuna, Sweden, where he completed the International Baccalaureate examination in Math, Physics and Chemistry at higher level, and Biology, Swedish, French and English at subsidiary level.

Between 1985 and 1989, the author attended the Massachusetts Institute of Technology, obtaining the degree of Bachelor of Science in Aeronautics and Astronautics.

Between 1989 and 1993, the author attended the Massachusetts Institute of Technology, working under Prof. Edward F. Crawley in the Space Engineering Research Center, obtaining the degree of Master of Science in Aeronautics and Astronautics, of which this document is an integral part.

The author is a student member of the American Institure of Aeronatics and Astronautics, and a member of the Aeronautics and Astronautics honor society Sigma Gamma Tau, the national engineering honor society Tau Beta Pi and and the national scientific honor society Sigma Xi.

Table of Contents

List of Figures

List of Tables

Notation

$\epsilon(x)$	Longitudinal strain
$f(x)$	Spatial weighting function of sensor
$f^{(i)}(x)$	i-th derivative of weighting function, $d^i f/dx^i$
l	Total length of sensor
k	Wave number
$y(k)$	Sensor output
$\mathcal{F}(k)$	Spatial fourier transform of sensor weighting
$\mathcal{T}(k)$	Sensor transfer function
α	Sensor transfer function rolloff wave number
$\bar{k}$	Ratio of wavenumber to rolloff wavenumber, k/α
x_0	Distance between sensor center and strain field origin
n	Number of sensors in sensor array
L	Length of structure
$\delta(x)$	Unit delta function
$u(x)$	Unit step function
$\mathrm{sgn}(x)$	Sign function
σ_o	Standard deviation of tip slope or displacement estimation error due to *optimistic* sensor uncertainty levels
σ_r	Standard deviation of tip slope or displacement estimation error due to *realistic* sensor uncertainty levels

$w(x)$	Tranverse displacement of beam
$\tilde{w}(x)$	Estimated tranverse displacement of beam
$\frac{dw(x)}{dx}$	Tranverse displacement of beam
$w'(x)$	Slope of beam
$\kappa(x)$	Curvature of beam
GSSF	Global Static Shape Function integration scheme
GDSF	Global Dynamic Shape Function integration scheme
LSGSSF-n	Least Squares Global Static Shape Function integration scheme using n shape functions
LSGDSF-n	Least Squares Global Dynamic Shape Function integration scheme using n shape functions
m	Mode number
w_{tip}	Beam displacement at tip of structure, $x = L$
$\mathcal{E}$	Error in tip displacement estimate
$\mathcal{O}$	Sensor array observability
PVDF	Polyvinylidene Flouride; a piezoelectric film

Chapter 1

Introduction

1.1 Background and Objectives

Intelligent structures incorporate sets of highly distributed actuators, sensors and processors [56]. Much work to date has centered on the development of technology for distributed actuators. However, less work has been performed on distributed sensors, and only a few pioneering efforts have dealt with distributed processors. All three of these components must be present in a system in order to implement any closed-loop shape control scheme. The need for additional work on sensors specifically for intelligent structures has motivated this work.

The use of piezoelectric materials in intelligent structures has been studied by de Luis [15] and Anderson [14]. De Luis et al. develop analytical models of piezoelectric actuators for intelligent structures, and verify these models experimentally. Analytic models are developed for static and dynamic actuation of segments of piezoelectric material either embedded in the structure or bonded to its surface. These models allow the response of the structure to actuation of the piezoelectrics to be predicted. The models can therefore help in efficient or optimal actuator placement. It is shown that the experimental response of a structure could be predicted reasonably well using the models developed. In addition, it is shown that although the stiffness of the

structure is not strongly affected by the addition of embedded piezoelectric components, the ultimate strength of the structure is reduced. Anderson et al. also develop models for induced-strain actuation of beam-like structures. Two analytical models and on finite element based model are presented which model extension, bending and shearing deformations. In addition to this work on piezoelectric materials, electrostrictive and magnetostrictive materials as well as shape memory alloys have been investigated.

Work has been done on integrating electronic components in intelligent structures [58]. Warkentin et al. demonstrate that embedding processing and other electronic components in addition to sensors and actuators within an intelligent structure simplifies physical implementation, and a technique for embedding silicon chips within the plies of a graphite/epoxy composite laminated structure is described.

Control schemes suitable for implementation on structures containing distributed actuators, sensors and processors have been developed by de Luis [40] and How [28]. De Luis et al. develop optimal controllers for intelligent structures by assuming that the sensors and actuators are so numerous that is can be assumed they are spatially continuous. Therefore, a functional analysis approach can be used to derive control algorithms. An experimental implementation of a prototype of an intelligent structure is presented, and optimal controllers are derived and implemented for this structure. How et al., on the other hand present a new control architecture specially suited to intelligent structures with many distributed sensors and actuators. This architecture is a hierarchic one with many controllers operating at a local level, and a single global one operating to control overall motions of the flexible structure.

In the area of sensors, work at MIT has been done on a variety of modal, convolving and wave sensors [13, 42]. Collins et al. [13] have developed piezoelectric film sensors for the control of a two-link planar flexible robotic manipulator system. The spatial shape of the sensors is selected such that their output is proportional to a mode

of the structure. These sensors make it possible to directly measure the dynamic states of a flexible structure. Ordinarily, such states must be measured using an estimator and some inherently inaccurate analytical model of the structure. Miller et al. [42] investigate sensors which output a single temporal signal by convolving measurements of structural variables distributed over a finite length of a structure. The shape of the sensor determines how this convolution procedure takes place. This shape can be optimized to ensure that the sensors is specially suited for use in the control of flexible structures, because it can be designed such that the sensor output rolls off without exhibiting phase lag relative to a point sensor at its center. This would allow gain stabilization without the accompanying reduction of available gain margin. Analytical examples and experimental counterparts are presented.

Lee et al. have also worked on modal sensors as well as actuators. Lee and Moon [38] develop a theory for piezoelectric laminates for sensing and actuation, in order to develop modal sensors and actuators. Such sensors, designed for one-dimensional operation are modeled analytically and compared to experimental implementations. If such sensors are manufactured accurately, actuator or observer spillover will not exist and the modal coordinates of a structure can be measured or actuated directly. Lee, Chiang and O'Sullivan [39] construct modal sensors and actuators to critically damn a cantilever plate using PVF_2 piezoelectric film. Plate theory is extended to include piezoelectric actuators and sensors. The sensor geometry is designed such that the sensor signal is proportional to the derivative in time of the modal coordinate of the structure. This allows a derivative feedback controller to be implemented. It is shown that critical damping can be achieved using this controller.

The principal advantage of shape sensors is that, relative to a point sensor at their center, they allow a roll off in amplitude of measurement output without a change in phase. The rolloff of such sensors is not limited by Bode's Gain-Phase Theorem [9], because they are not making causal measurements. Bode's integral

defines a relationship between the magnitude and phase of a dynamic system. The integral holds only for causal dynamics and requires that any magnitude rolloff must be accompanied by some phase lag. Point sensors are limited by Bode's theorem because they are causal and only measure present information. However, any sensor measuring strain over a finite length has access to incoming waves (future information) and outgoing waves (past information) at the same instant. Thus it has information which is noncausal with respect to its midpoint: it can sense future events. This enables it to exhibit magnitude rolloff with *no associated phase lag.*

One of the shortcomings of the work done to date on modal sensors is the fact that, to work well, the sensors must be able to sense deformations from all or large parts of the structure. This means that the sensor must cover a large fraction of the structure. It therefore becomes problematic to implement several modal sensors on the same structure. Another problem is that modal sensors are typically implemented in hardware. This means that the output properties of the sensors as a function of frequency are constant. Therefore, should the shape or mass of the structure change, the mode shapes will change, and the modal sensors will not operate correctly. A third problem with the sensors currently being investigated is it is problematic to extrapolate their design to more complex structures, such as truss-structures or two-dimensional plate-like structures.

The purpose of this work is to develop techniques for the design of discrete distributed sensor systems to accurately estimate the quasistatic and dynamic states of an intelligent structure for the purposes of closed-loop control or identification.

The remainder of this introductory chapter will propose the functional requirements for such sensors, discuss their design parameters, and then preview the contents of the report.

1.2 Functional Requirements

The development of a distributed sensor system for estimating the global shape of a structure is a design process. Therefore, it is appropriate to formally formulate the functional requirements of the sensor system. The functional requirements for distributed sensor systems can be characterized as: those which involve the properties of individual sensors; those which involve the properties of the system of discrete distributed sensors as a whole; and those which relate to practical implementation. The requirements are numbered for ease of reference in the remainder of the document.

The first functional requirements involve the individual sensors.

1. Individual sensors should be able to accurately observe and precisely resolve modes which are targeted for control.

2. The observability of the dynamic modes should roll off quickly in frequency beyond the desired bandwidth.

3. The transfer function from the strain at the center of the sensor to the sensor output, as a function of spatial frequency, should not contain negative regions.

The first functional requirement is required because the sensor system is being done in order to implement closed loop control on an intelligent structure. The second requirement is necessary in order to minimize the effect of mismodeled or unmodeled higher order dynamics and to reduce the effects of aliasing. Aliasing will occur when the wavelength of the mode shape is less than or equal to the length between two successive sensors. The third functional requirement is necessary in order to avoid possible instabilities in a control scheme using these sensor measurements.

The next three functional requirements are for the sensor system as a whole, including the processing of the sensor output to estimate overall shape.

4. The sensor system as a whole should resolve the shape of the structure accurately and in detail.

5. Global shape estimates of the structure should roll off quickly with spatial frequency to avoid aliasing involving static modes of high spatial frequency.

6. The roll off toward zero of the sensor system observability to static modes of increasing spatial frequency should be monotonic and should not contain any negatives.

Functional requirement #6 stems from the fact that the static shape typically contributes significantly to the performance metric in any structural control scheme. This is because the truncated series of modes used to represent the dynamics of the structure cannot be superposed to produce an accurate representation of the static structural shape in order to assure good performance. Numerical spatial integration will be performed in order to estimate the global shape from the individual sensor measurements. The next functional requirement means that the observability of the sensor system to static modes of relatively high spatial frequency must decrease. Aliasing will occur when the wavelength of the static mode is less than or equal to the length between two successive sensors. The sixth functional requirement stems from the fact that negatives appreaing in the roll off of the sensor system observability would be capable of instigating unstable interactions in a control scheme using these sensor measurements. Such negatives can be introduced by sensor aliasing.

There are functional requirements concerning the physical implementation of a distributed sensor system.

7. The sensors must be of finite length to be implementable on any practical structure.

8. Sensors with partially negative or highly complex weightings should be avoided.

9. Sensors near structure boundaries must be truncated in such a way as to not degrade performance.

The seventh functional requirement is evident: the required length of the sensor cannot exceed the physical length of the structure. Functional requirement #8 addresses ease of sensor implementation. Ease of implementation is of great concern in order to assure accuracy and the performance predicted analytically. Thus sensors whose weighting is everywhere non-negative are desirable, because fabrication of such sensors is significantly easier than sensors that require regions of negative weighting. In addition, sensors of other complex shapes must be fabricated very accurately to ensure that they exhibit the properties predicted analytically. The last functional requirement requirements that sensors that would ordinarily extend beyond a boundary of the structure must be truncated in some fashion, so that they can be implemented in a practical setting.

The development of a distributed sensor system that meets, to the extent possible, the functional requirements is outlined in the objective of this work.

1.3 Design Parameters

The design parameters are: (1) the spatial and temporal derivative sensed; (2) the sensor weighting; (3) the sensor length; (4) the boundary sensor design; (5) the integration method; (6) the number of sensors. The investigation of these design parameters to meet the functional requirements forms the outlint of this report.

The first design parameter that must be considered deals with the spatial variable of the structure which will be measured by the sensors. The possible spatial derivatives of a structure that could be sensed are absolute displacement, slope or relative displcement, or a continuous variable such as strain. The displacement is difficult to measure without time integration of an inertial acceleration or without a reference

to a relative frame. The easiest spatial derivative to measure, however, is strain, since it is continuous, and easily amenable to weighted measurements. True static shape estimation is best performed using sensors which measure some spatial derivative of displacement such as strain, rather than temporal derivatives of displacement such as velocity or acceleration. By using spatially filtering sensors with appropriate weightings, the observability of such sensors will roll off as the spatial frequency of the spatial derivative being measured increases. Sensors that spatially average strain are more easily implemented than ones that average displacement. Thus the sensor system will be composed of spatially averaging *strain sensors* whose output will be spatially integrated using a numerical integration scheme in order to obtain the quasistatic and dynamic states of the structure.

The spatial weighting of the sensor determines the transfer function of the sensor and how its output will be affected by the spatial frequency of the longitudinal strain field that acts on it. Chapter 2 deals with the performance and design of individual spatially filtering strain sensors. The design of the weighting function used for the sensors is described. These weightings are analyzed and the effect of the sensor shape on the transfer function and output is derived in detail, for both sinusoidally and exponentially varying strain fields. The effect of moving the sensors along the length of the structure on the sensor output and transfer functions is addressed.

All spatially filtering sensors of finite length roll off beyond some spatial frequency with at least -20 dB/decade. The point at which this rolloff begins is defined as the point where the transfer function (the output of the sensor relative to the output of a point sensor at its center) has dropped to $1/\sqrt{2}$ (-3 dB), and is largely controlled by the effective length of the sensor. As the sensor is lengthened the spatial frequency at which rolloff begins decreases. This is because more wavelengths of a strain field at a fixed spatial frequency can be fitted within the length of a longer sensor, and the spatial frequency of the strain field becomes considerably higher than the weighting

of the sensor. The effect of sensor length on the sensor transfer function is discussed in Chapter 2.

Since the length of the sensors employed is finite, it will be impossible to avoid having a sensor which would normally extend beyond a boundary of the structure. The design of such sensors and the method by which the weighting is truncated is very important as it has a significant effect on the performance of the sensors. Attention is given in Chapter 2 to the properties of sensors whose weighting partially falls beyond the boundary of the structure. These properties are derived and analyzed. Two methods of truncation for sensors near a structure boundary are presented.

Once the design of the sensors has been established, it is necessary to decide what scheme should be used to estimate the global slope and deflection of the structure using the outputs of the sensor array. The array of strain sensors will yield curvature measurements along the length of the structure, in the case of bending. These curvatures can be numerically integrated twice to estimate the global deflections of the structure. Chapter 3 introduces a selected set of integration schemes which are derived in Appendix C. The integration schemes include the simple midpoint, trapezoidal and Simpson rules, and two rules based on cubic interpolating splines and second order B-splines. In addition, two optimal quadrature rules, the Gauss and Radau methods, and a conventional interpolation scheme using Chebyshev polynomials are included.

One-dimensional structures, such as beams, are chosen as an example for implementation. The static shape and dynamic modes of such a structure are sensed by a sensor array mounted on the surface, along the length of a beam-like structure. The numerical integration schemes described in Chapter 3 are used to obtain global shape estimates by numerically integrating the measurements made by the sensor array.

The performance of the integration schemes for quasistatic loadings and a varying number of sensors in the array is investigated in Chapter 4. Uncertainties in the gage factor and sensor position are included in an effort to model the errors typically

encountered in a practical implementation. It then becomes possible to determine when is it more beneficial to improve the accuracy of the individual sensors or increase the number of sensors in order to achieve the best shape estimation possible.

In Chapter 5, the dynamic modes of both pinned-pinned and clamped-free beams are used to assess the rolloff behavior of the spatial integral of the sensor outputs. Pinned-pinned and clamped-free beams with an array of nine evenly distributed sensors are simulated. As the mode number is increased, global shape estimation is performed, and the tip deflection is found. The observability of the sensor system to the mode shape is computed to determine how this observability rolls off as the mode number is increased.

The number of sensors distributed along the structure, in conjunction with the integration rules, is critically important in achieving the estimation accuracy required. The convergence properties of the integration rules should guarantee that as the number of sensors is increased, the integrated result should achieve higher accuracy provided the individual sensor measurements are free from error. In order to determine the number of sensors which are required to obtain accurate global shape estimates, the integration rules introduced in Chapter 3 are used to perform simulated static shape estimation of a cantilevered beam and a cantilevered beam with its midspan constrained, both with applied tip displacements. The description of the simulations and the results are presented in Chapter 4.

Chapter 2

Shaped Sensors

In this chapter the output and transfer functions for spatially averaging strain sensors are derived for strain fields that are spatially sinusoidal and those that are spatially exponential. Expressions are obtained for both nontruncated sensors and sensors partially truncated at the boundaries of a structure.

To begin, basic principles of spatially averaging sensors are introduced. Then, a set of relatively simple sensor weightings are introduced and defined. General expressions for the output and transfer function of a nontruncated sensor in a spatially sinusoidal strain field are obtained as a function of the parameters that define the spatial weighting of the sensor. Specific results for the chosen weightings are then obtained. Finally, expressions for the output and transfer function of truncated sensors in a sinusoidal strain field are derived, as well as expressions for the output and transfer functions of both nontruncated and truncated sensors in a spatially exponential strain field.

2.1 Introduction

Spatially averaging sensors can be used in a number of different applications to measure a variety of signals. In fact, such sensors need not be limited to measuring

spatial variables in structures. For example, applications in acoustics and optics can be envisioned. For the purposes of this work, however, it is assumed that the sensors report measurements of spatially averaged extensional strain signals.

The most important characteristic in the design of spatially averaging strain sensors is their spatial weighting. This weighting determines the way in which the magnitude and phase of the sensor output relative to the strain at the center of the sensor are affected by the spatial variation of the strain field. Let $f(x)$ define the spatial weighting of the sensor. Let this weighting be nonzero over a finite region $[-l/2, l/2]$ of the structure. The sensor output is then a filtered measurement of the strain of the form

$$y = \int_{-l/2}^{l/2} f(x)\, \epsilon(x)\, dx \tag{2.1}$$

where $\epsilon(x)$ is the longitudinal strain along the structure and l is the length of the sensor. The weighting function $f(x)$ is implemented by varying the spatial sensitivity of the sensor. Depending on the device used as the sensor, this might be done by:

- varying the width of the sensor;
- varying the thickness of the sensor;
- segmenting the sensor and implementing a weighted sum of the measurements of different sensor segments in the signal processing system;
- varying the distance of the sensor from the elastic axis of a beam in bending.

Note that the weighting function $f(x)$ should be scaled such that

$$\int_{-l/2}^{l/2} f(x)\, dx = 1 \tag{2.2}$$

to assure that strain signals of zero spatial frequency, or infinite spatial wavelength, are measured accurately. This is required in order to satisfy the first functional requirement presented in Chapter 1.

As an example of a physical implementation, take a case where the sensor is made of a piece of thin Polyvinylidene Flouride (PVDF) piezoelectric film bonded to the surface of a beam [13]. The weighting function $f(x)$ is implemented by varying the width of the electrode on the film. The output of the PVDF sensor is a charge proportional to a filtered measurement of the surface strain. The magnitude of this charge depends on the area of the electrode. Therefore, the gain of the sensor can be varied spatially by spatially varying the electrode area. The charge generated by the sensor is given by

$$Q(t) = d_{31} E_p \int_{-l/2}^{l/2} f(x)\, \epsilon(x,t)\, dx \tag{2.3}$$

where E_p is Young's modulus for the piezoelectric film, $f(x)$ is the width of the sensor electrode, $\epsilon(x,t)$ is the surface longitudinal strain, and d_{31} is the piezoelectric constant with units of charge per unit area per unit stress. This constant relates mechanical stress to electric displacement.

2.2 Filter Theory

In this section, traditional filter theory is used as a starting point in the design of spatially averaging strain sensors. Classical filter theory has traditionally focused on the processing of sinusoidal or periodic signals in the time domain. This work has led to the design of filters that exhibit a variety of magnitude rolloff and ripple characteristics in the temporal frequency domain. By a transformation from the temporal frequency domain to the spatial frequency domain (the wave number domain), results from classical filter theory can be applied in the design of sensors that provide spatially filtered measurements of strain fields.

Filter theory considers signals in the Fourier domain. Fourier transforms of temporal signals defined in terms of temporal frequency, ω, are defined classically [50] as:

$$\mathcal{F}(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-j\omega t}\, dt \tag{2.4}$$

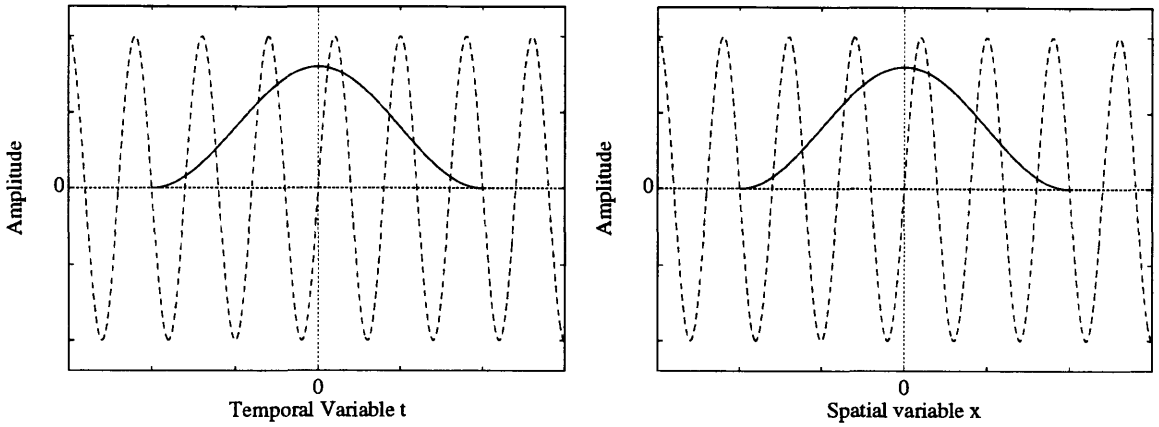

Figure 2.1: The similarity between temporal and spatial Fourier analysis: solid line: temporal window or spatial weighting of filter; dashed line: temporal or spatial signal to be filtered.

Fourier transforms in terms of spatial frequency, k, can be developed in a similar fashion:

$$\mathcal{F}(k) = \int_{-\infty}^{\infty} f(x)\, e^{-jkx}\, dx \tag{2.5}$$

While the spatial frequency k is related to the temporal frequency ω, the spatial sensor weighting acts like a temporal sampling window in time. Figure 2.1 shows this duality clearly. The solid line represents the temporal window $f(t)$ and the spatial weighting $f(x)$, and the dashed line represents the temporal signal $e^{-j\omega t}$ and the spatial signal e^{-jkx}.

Due to these similarities, design techniques available for sampling windows can be extended to applications involving spatial filtering. As with temporal windows, the spatial weighting can be changed to tailor the magnitude response of the sensor. firther, the effective sensor length can be changed to control the spatial frequency at which magnitude rolloff commences. This rolloff should be quick in order to satisfy Functional Requirement #2 presented in Chapter 1. It is critical to note that if $f(x)$ is symmetric about $x = 0$, $\mathcal{F}(k)$ will be purely real, as the imaginary component drops out. This means that $\mathcal{F}(k)$ can exhibit magnitude rolloff without incurring

phase lag. However, $\mathcal{F}(k)$ can still be negative, indicating a 180° phase shift. The third functional requirement presented in Chapter 1 states that the individual sensors must have transfer functions that are nonnegative for all values of k in order to avoid possible instabilities in a control scheme which uses spatially averaging sensors. In Section 2.4.1 it will be seen that such phase uncertainties do exist for certain weightings, and must be avoided.

2.3 Sensor Weightings

As a starting point for investigating possible spatial weightings of sensors, some simple ones which are used in temporal windowing of temporal signals were selected. Such window shapes include rectangular, Bartlett (or triangular) and Hanning windows. In addition, Sinc and Gaussian sensor weightings were also selected because they possess interesting frequency characteristics. Using these simple shapes, more complicated ones were created by multiplying the weightings of two or more simple weightings together. Examples of these include a Sinc weighting multiplied by another Sinc weighting, a Bartlett weighting multiplied by another Bartlett weighting, a Hanning weighting multiplied by another Hanning weighting, and a Gaussian multiplied by a Hanning weighting. A number of such complex weightings were investigated in order to get a sense of the spatial properties a weighting must possess so that its transfer function will have certain desired properties.

In the following sections, the selected sensor weightings are defined. The parameter α is used to select the effective length of the sensor. It will be seen later that this parameter is scaled such that the magnitude response to a sinusoidal field drops to $1/\sqrt{2}$ (-3 dB) exactly, or in the vicinity of $k = \alpha$. In the following three sections the spatial definitions of nine selected sensors will be presented. In Section 2.3.1 the definitions of three spatially finite windows are shown. In Section 2.3.2 the definitions of three spatially infinite weightings are given. Finally, Section 2.3.3 contains defini-

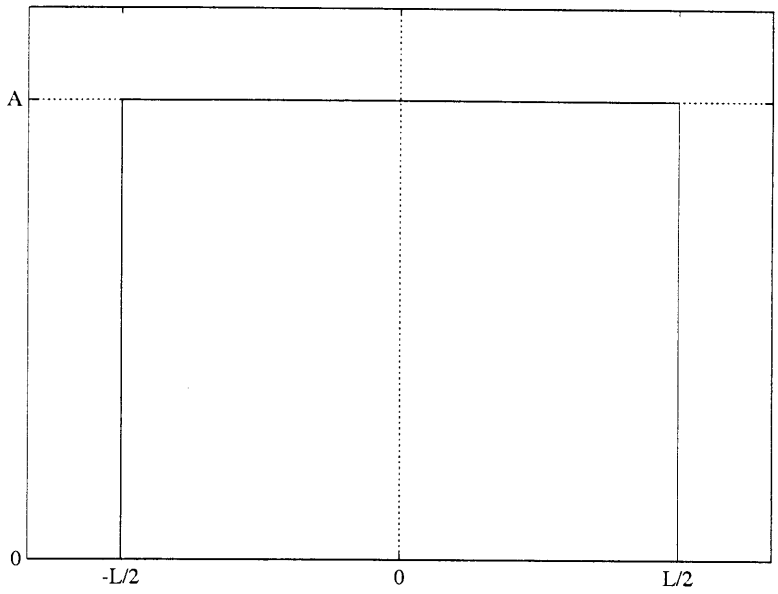

Figure 2.2: The spatial weighting of the rectangular, or box car weighting.

tions of three compound weightings that were obtained by multiplying two weightings together.

2.3.1 Finite Spatial Sensor Weightings

Rectangular Weighting

The rectangular, or box car weighting is shown in Figure 2.2, and is defined as

$$f(x) = \frac{\alpha}{2\sqrt{2}}, \qquad -\frac{\sqrt{2}}{\alpha} \leq x \leq \frac{\sqrt{2}}{\alpha} \tag{2.6}$$

Note that the magnitude of $f(x)$ has been chosen such that $\int_{-\sqrt{2}/\alpha}^{\sqrt{2}/\alpha} f(x)\, dx = 1$. The length l and maximum amplitude A of the rectangular weighting are given by

$$l = \frac{2\sqrt{2}}{\alpha} \tag{2.7}$$

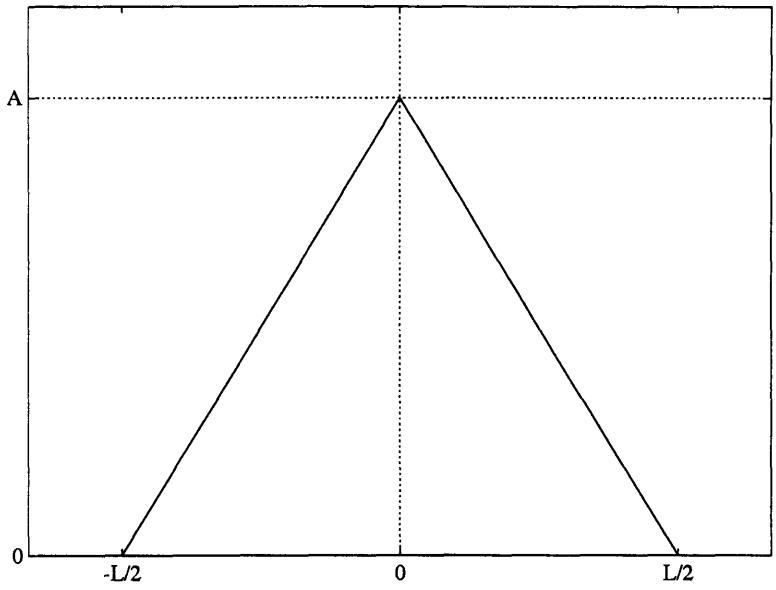

Figure 2.3: The spatial weighting of the Bartlett, or triangular weighting.

$$A = \frac{\alpha}{2\sqrt{2}} \tag{2.8}$$

Bartlett Weighting

The Bartlett, triangular, or Parzen weighting is shown in Figure 2.3, and is defined by

$$f(x) = \frac{\alpha}{2}\left(1 - \left|\frac{\alpha x}{2}\right|\right), \qquad -\frac{2}{\alpha} \leq x \leq \frac{2}{\alpha} \tag{2.9}$$

The length and maximum amplitude of the Bartlett weighting are

$$l = \frac{4}{\alpha} \tag{2.10}$$

$$A = \frac{\alpha}{2} \tag{2.11}$$

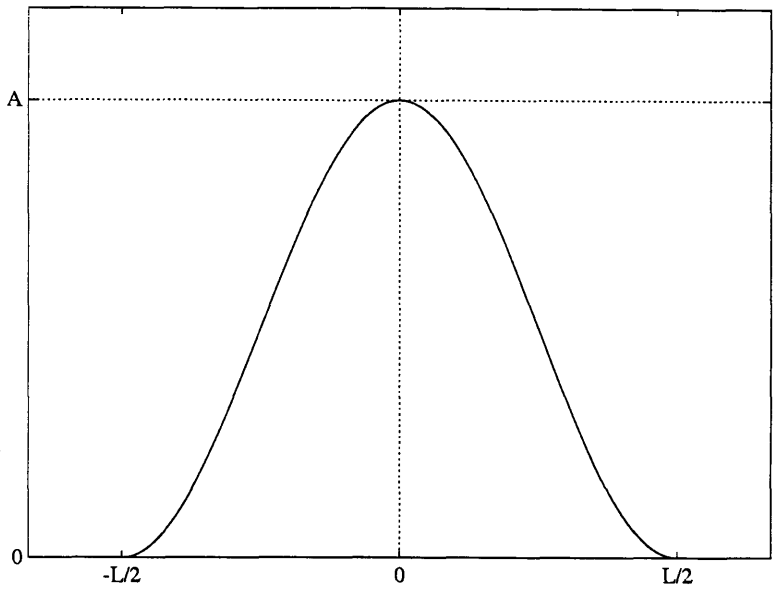

Figure 2.4: The spatial weighting of the Hanning weighting.

Hanning Weighting

The Hanning weighting is defined by

$$f(x) = \frac{\alpha}{\sqrt{2}\pi}\left(1 + \cos\left(\sqrt{2}\alpha x\right)\right), \qquad -\frac{\pi}{\sqrt{2}\alpha} \leq x \leq \frac{\pi}{\sqrt{2}\alpha} \tag{2.12}$$

and is shown in Figure 2.4. The length and maximum amplitude of the Hanning weighting are given by

$$l = \frac{\sqrt{2}\pi}{\alpha} \tag{2.13}$$

$$A = \frac{\alpha}{2\sqrt{2}\pi} \tag{2.14}$$

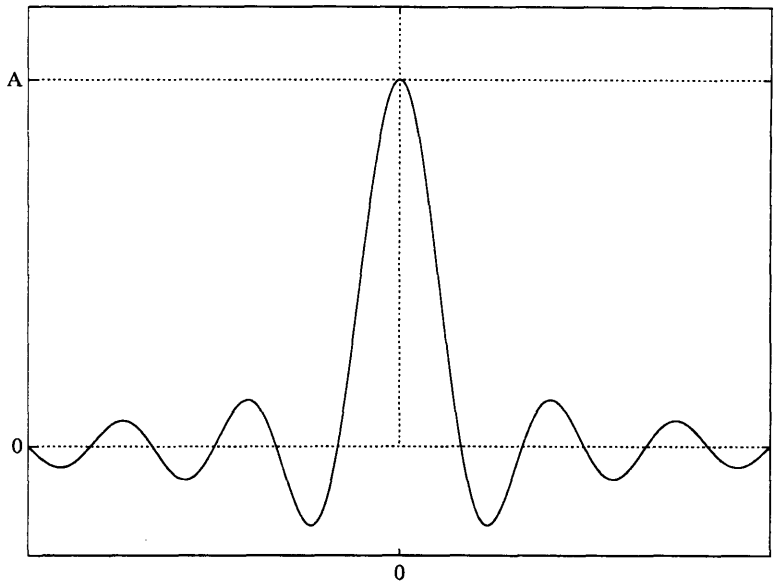

Figure 2.5: The spatial weighting of the Sinc weighting.

2.3.2 Infinite Spatial Sensor Weightings

Sinc Weighting

The Sinc weighting is defined by

$$f(x) = \frac{\alpha}{\pi}\left(\frac{\sin \alpha x}{\alpha x}\right), \qquad -\infty < x < +\infty \tag{2.15}$$

and is shown in Figure 2.5. The Sinc weighting extends over the infinite domain $[-\infty, +\infty]$, and its maximum amplitude A is given by

$$A = \frac{\alpha}{\pi} \tag{2.16}$$

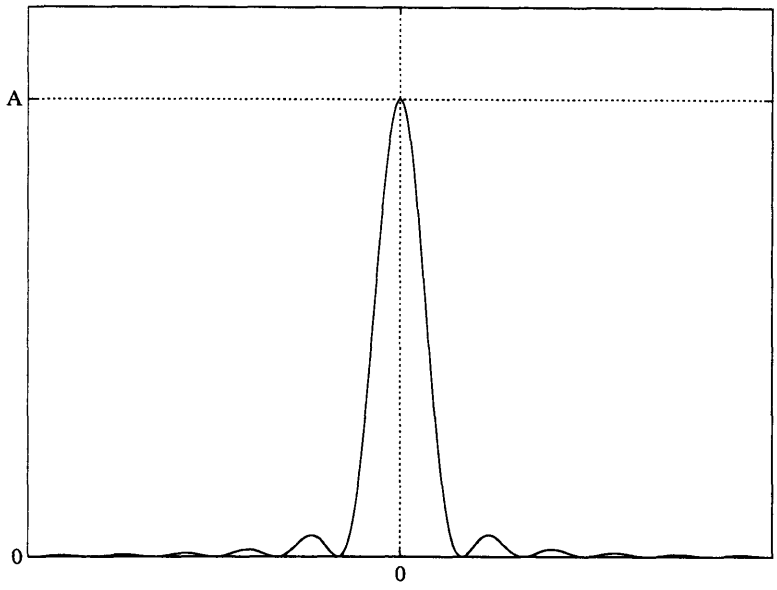

Figure 2.6: The spatial weighting of the Sinc-Sinc weighting.

Sinc-Sinc Weighting

The Sinc-Sinc weighting is produced by multiplying two Sinc functions together. The result is shown in Figure 2.6, and is defined by

$$f(x) = \frac{\sqrt{2}+1}{\sqrt{2}} \frac{\alpha}{\pi} \left[\frac{\sin\left(\frac{\sqrt{2}+1}{\sqrt{2}}\alpha x\right)}{\frac{\sqrt{2}+1}{\sqrt{2}}\alpha x} \right]^2, \qquad -\infty < x < +\infty \tag{2.17}$$

As with the Sinc weighting, the Sinc-Sinc weighting extends over $[-\infty, +\infty]$. Its amplitude A is given by

$$A = \frac{\sqrt{2}+1}{\sqrt{2}} \frac{\alpha}{\pi} \tag{2.18}$$

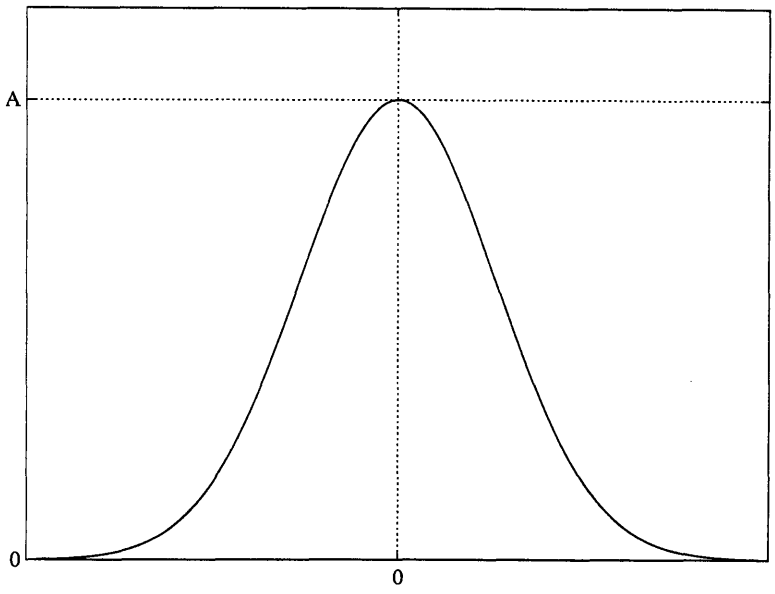

Figure 2.7: The spatial weighting of the Gauss weighting.

Gauss Weighting

The Gauss weighting is defined by

$$f(x) = \frac{\alpha}{\sqrt{2\pi \ln 2}} \exp\left(-\frac{(\alpha x)^2}{2 \ln 2}\right), \qquad -\infty < x < +\infty \tag{2.19}$$

Figure 2.7 shows the shape of the Gauss weighting. As with the Sinc weighting, the Gauss weighting extends over the infinite domain $[-\infty, +\infty]$. The maximum amplitude A, of the Gauss weighting, is given by

$$A = \frac{\alpha}{\sqrt{2\pi \ln 2}} \tag{2.20}$$

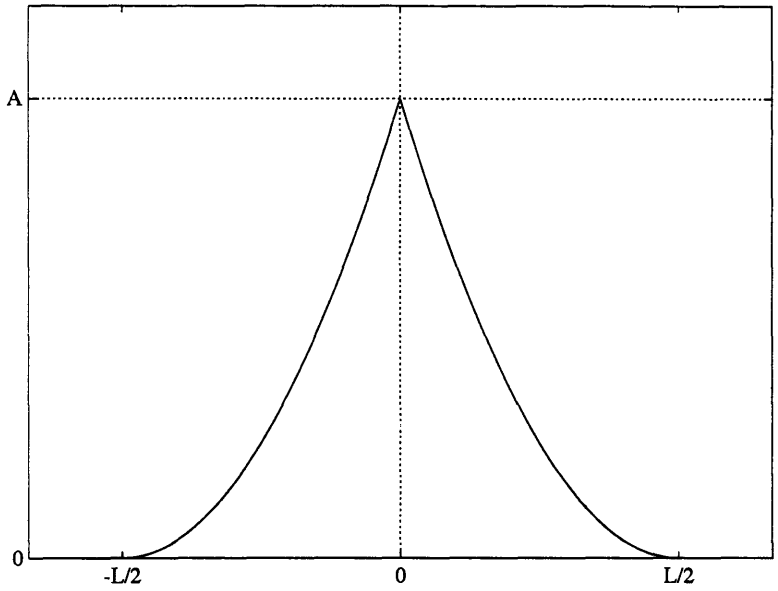

Figure 2.8: The spatial weighting of the Bartlett-Bartlett weighting.

2.3.3 Compound Spatial Sensor Weightings

Bartlett-Bartlett Weighting

The Bartlett-Bartlett weighting is produced by multiplying two Bartlett weightings together. The resultant weighting is shown in Figure 2.8. It is defined by

$$f(x) = \frac{3\alpha}{2\pi}\left(1 - \left|\frac{\alpha x}{\pi}\right|\right)^2, \qquad -\frac{\pi}{\alpha} \leq x \leq \frac{\pi}{\alpha} \tag{2.21}$$

The length and maximum value of the Bartlett-Bartlett weighting are

$$l = \frac{2\pi}{\alpha} \tag{2.22}$$

$$A = \frac{3\alpha}{2\pi} \tag{2.23}$$

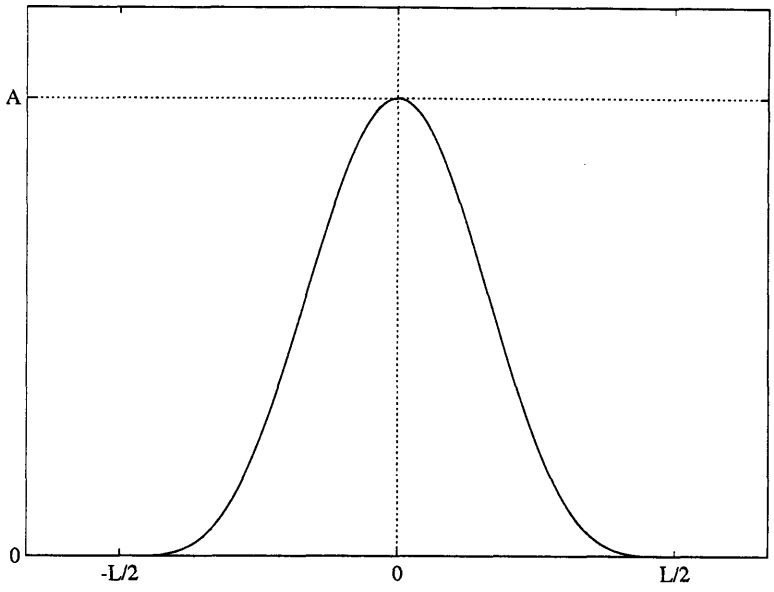

Figure 2.9: The spatial weighting of the Hanning-Hanning weighting.

Hanning-Hanning Weighting

The Hanning-Hanning weighting is constructed by multiplying two Hanning weightings together. The result is a weighting that tapers to zero more smoothly than a pure Hanning weighting. The weighting is defined by

$$f(x) = \frac{\alpha}{3\pi}\left(1 + \cos \alpha x\right)^2, \qquad -\frac{\pi}{\alpha} \leq x \leq \frac{\pi}{\alpha} \tag{2.24}$$

The Hanning-Hanning weighting is shown in Figure 2.9. The length and maximum amplitude of the Hanning-Hanning weighting are given by

$$l = \frac{2\pi}{\alpha} \tag{2.25}$$

$$A = \frac{\alpha}{6\pi} \tag{2.26}$$

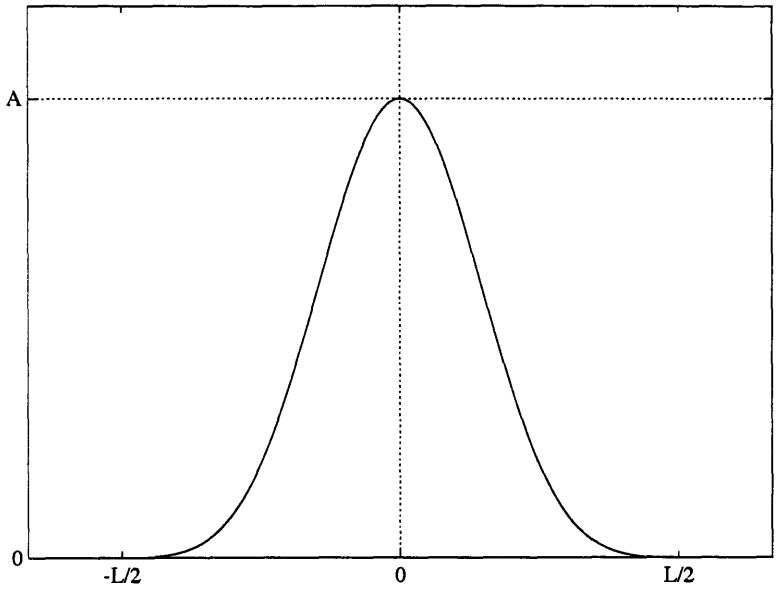

Figure 2.10: The spatial weighting of the Gauss-Hanning weighting.

Gauss-Hanning Weighting

The Gauss-Hanning weighting is produced by truncating a Gaussian weighting by multiplying it by a Hanning weighting. The Gauss-Hanning weighting is defined by

$$f(x) = M(\alpha)\ \exp\left(-\frac{(\alpha x)^2}{2\ln 2}\right)\left[1+\cos\left(\sqrt{2}\alpha x\right)\right], \qquad -\frac{\pi}{\sqrt{2}\alpha} \le x \le \frac{\pi}{\sqrt{2}\alpha} \tag{2.27}$$

and is shown in Figure 2.10 The amplitude parameter $M(\alpha)$ is found by requiring that $\int_{-l/2}^{l/2} f(x)\,dx = 1$. $M(\alpha)$ is given by

$$M(\alpha) = \frac{4\alpha}{\sqrt{2\pi\ln 2}}\left[\frac{1}{\operatorname{erf}\left(\frac{\pi + j\,2\ln 2}{2\ln 2}\right) - \operatorname{erf}\left(\frac{-\pi + j\,2\ln 2}{2\ln 2}\right) + 4\operatorname{erf}\left(\frac{\pi}{2\sqrt{\ln 2}}\right)}\right] \tag{2.28}$$

$$M(\alpha) \approx \frac{0.68985}{\sqrt{2\pi\ln 2}}\alpha \approx 0.33056\ \alpha \tag{2.29}$$

Table 2.1: Spatial properties of selected strain-averaging sensors.

Sensor Type	Sensor Weighting	Negatives in x?	Finite in x?	Length l
Finite	Rectangle	No	Yes	$2\sqrt{2}/\alpha$
	Bartlett	No	Yes	$4/\alpha$
	Hanning	No	Yes	$\sqrt{2}\pi/\alpha$
Infinite	Sinc	Yes	No	∞
	Sinc-Sinc	No	No	∞
	Gauss	No	No	∞
Compound	Bartlett-Bartlett	No	Yes	$2\pi/\alpha$
	Hanning-Hanning	No	Yes	$2\pi/\alpha$
	Gauss-Hanning	No	Yes	$\sqrt{2}\pi/\alpha$

where $j = \sqrt{-1}$, and $\mathrm{erf}(z)$ is the error function given by

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \tag{2.30}$$

The length and maximum amplitude of the Gauss-Hanning weighting are given by

$$l = \frac{\sqrt{2}\pi}{\alpha} \tag{2.31}$$

$$A = \frac{M(\alpha)}{2} \tag{2.32}$$

2.3.4 Summary of Spatial Sensor Weightings

Table 2.1 summarizes some of the spatial properties of the set of weightings presented in Sections 2.3.1, 2.3.2 and 2.3.3. The third column shows which of the sensor

weighting functions contain some negative regions. Clearly, of the weightings selected, only the Sinc sensor contains regions of negative sensitivity. The fourth column of the table specifies whether or not the weighting is spatially finite. It is interesting to note that the Sinc, Gauss and Sinc-Sinc weightings, which will be shown to offer the best performance, are not spatially finite. The last column of Table 2.1 shows the total length of the sensor.

The results in Table 2.1 can now be compared to the functional requirements presented in Chapter 1. The third column of Table 2.1 shows that only the Sinc weighting must contain negative regions. Functional Requirement #8 stated that such negative regions should be avoided as they complicate the fabrication process.

Functional Requirement #7 requires that the weighting be finite in x so as to be implementable in a physical system. Thus, all the spatially infinite weightings (Sinc, Sinc-Sinc and Gauss) cannot be used without some sort of spatial truncation. Functional Requirement #9 states that such truncation should be done in such a way so as not to degrade sensor performance. One such truncation method is implemented with the compound Gauss-Hanning weighting, where the infinite Gauss weighting is smoothly truncated using a finite Hanning weighting.

In the following two sections, output and transfer functions for spatially averaging strain sensors are derived for both sinusoidally and exponentially varying strain fields.

2.4 Transfer Functions for Sinusoidal Strain

At this point, the properties of the weightings of sensors measuring spatially sinusoidal strain signals are investigated. The placement of sensors relative to the point of symmetry in the strain field is important. This point of symmetry or antisymmetry is defined as the point about which a sinusoidal strain signal remains an odd function and a cosinusoidal strain signal remains an even function, no matter

how the spatial frequency is changed. For example, for the strain fields $\sin(kx)$ and $\cos(kx)$, the point of antisymmetry and symmetry, respectively, is simply $x = 0$. This point has the additional property that the strain remains independent of the spatial frequency k.

First, in Section 2.4.1, the properties of a sensor whose center is positioned at the point of symmetry of the strain field are investigated. Then, in Section 2.4.2, the sensor is moved and the properties of a sensor placed with its center at a known distance x_0 from the point of symmetry are derived. Further, some sensors will inevitably be placed close enough to a boundary of the structure that a portion of their weighting falls beyond it. In this case the weighting is truncated at the boundary. The properties of a truncated sensor are therefore investigated in Section 2.4.3. Finally, in Section 2.4.4, in an effort to satisfy Functional Requirement #9, an alternative truncation scheme (folded truncation) is introduced and its properties are analyzed.

2.4.1 Centered Sensors

Since a spatially averaging sensor acts as a spatial filter, it is important to investigate the properties of its output and transfer function. The transfer function is defined as the ratio of the filtered strain measurement the sensor reports, to the strain at the center of the sensor. For a spatially constant strain field, the transfer function should therefore be unity as required by Functional Requirement #1. In the following discussion it is assumed that the center of the sensor lies at the point of symmetry in the strain field. Assuming that the weighting function $f(x)$ is non-zero only over the closed interval $[-l/2, l/2]$, its output as a function of the spatial frequency k of the strain field is given by

$$y(k) = \int_{-l/2}^{l/2} f(x)\, \epsilon(k, x)\, dx \tag{2.33}$$

where $\epsilon(k, x)$ is the extensional strain along the structure as a function of the spatial variable x and the wave number k. Since $\epsilon(k, 0)$ is the strain at the center of the

sensor, the transfer function of the sensor can be found from

$$\mathcal{T}(k) = \frac{y(k)}{\epsilon(k,0)} = \frac{\int_{-l/2}^{l/2} f(x)\ \epsilon(k,x)\ dx}{\epsilon(k,0)} \tag{2.34}$$

For the case of cosinusoidal strain, where $\epsilon(k,x) = \cos(kx)$, this expression simplifies to

$$y(k) = \mathcal{T}(k) = \int_{-l/2}^{l/2} f(x)\cos(kx)\ dx \tag{2.35}$$

since $\cos(0) = 1$.

The strain field has been assumed to be of the form $\cos(kx)$ and not $\sin(kx)$ to avoid a zero strain condition at the center of the sensor. It will be shown in Section 2.4.3 that when the sensor weighting $f(x)$ is symmetric (that is, when $f(x) = f(-x)$), the transfer function is the same for both sinusoidal and cosinusoidal strain fields. This means simply that their sine and cosine transforms are identical.

It is now shown that when $f(x)$ is symmetric, the transfer function (2.35) is also the spatial Fourier transform of the weighting function $f(x)$. The spatial Fourier transform of the spatial weighting $f(x)$ is defined as

$$\mathcal{F}(k) = \int_{-\infty}^{\infty} f(x)\ e^{-jkx}\ dx \tag{2.36}$$

Since $f(x)$ is nonzero only over $[-l/2, l/2]$, we have

$$\mathcal{F}(k) = \int_{-l/2}^{l/2} f(x)\ e^{-jkx}\ dx \tag{2.37}$$

Expanding the complex exponential e^{-jkx} yields

$$\mathcal{F}(k) = \int_{-l/2}^{l/2} f(x)\ \cos(kx)\ dx - j\int_{-l/2}^{l/2} f(x)\ \sin(kx)\ dx \tag{2.38}$$

Since $f(x)$ has been assumed to be symmetric, it is an even function. The product $f(x)\sin(kx)$ is therefore odd, since $\sin(kx)$ is odd. The second integral in (2.38) is therefore zero. Equation (2.38) then equals (2.35):

$$\mathcal{F}(k) = \int_{-l/2}^{l/2} f(x)\ \cos(kx)\ dx = \mathcal{T}(k) = y(k) \tag{2.39}$$

By repeatedly integrating Equation (2.39) by parts, an infinite series solution for the output of a sensor mounted in the interior of the structure, away from its boundaries, centered at the point of symmetry in a cosinusoidal strain field can be written as

$$\begin{aligned} y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{2i+1}}\Big[f^{(2i)}(l/2) + f^{(2i)}(-l/2)\Big]\sin(kl/2) \\ & + \frac{(-1)^i}{k^{2i+2}}\Big[f^{(2i+1)}(l/2) - f^{(2i+1)}(-l/2)\Big]\cos(kl/2)\Bigg\} \end{aligned} \tag{2.40}$$

where $f^{(i)}(x)$ represents the i-th derivative of the weighting function $f(x)$ with respect to the spatial variable x, $d^i f(x)/dx^i$. The derivation of (2.40) is presented in Section A.1 of Appendix A. In the derivation, it is assumed that $f(x)$ has a continuous derivative and is integrable over the interval $[-l/2, l/2]$.

Typically, the weighting function $f(x)$ is symmetric. In this case, we have

$$f^{(2i)}(x) = f^{(2i)}(-x) \qquad (i = 0, 1, \ldots, \infty) \tag{2.41}$$

$$f^{(2i+1)}(x) = -f^{(2i+1)}(-x) \qquad (i = 0, 1, \ldots, \infty) \tag{2.42}$$

and (2.40) simplifies to

$$y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{2(-1)^i}{k^{2i+1}} f^{(2i)}(l/2)\sin(kl/2) + \frac{2(-1)^i}{k^{2i+2}} f^{(2i+1)}(l/2)\cos(kl/2) \right\} \tag{2.43}$$

Table 2.2: Summary of rolloff properties of the sensor transfer function $\mathcal{T}(k)$ for $f(x)$ with a continuous derivative.

$f^{(i)}(\pm l/2)$	Rolloff		Comments
$f(\pm l/2) \neq 0$	$1/k$	-20 dB/decade	Weighting function nonzero at either end.
$f(\pm l/2) = 0$	$1/k^2$	-40 dB/decade	Weighting function zero at both ends.
$f(\pm l/2) = 0$ $f'(\pm l/2) = 0$	$1/k^3$	-60 dB/decade	Weighting function and slope zero at both ends.
$f(\pm l/2) = 0$ $f'(\pm l/2) = 0$ $f''(\pm l/2) = 0$	$1/k^4$	-80 dB/decade	Weighting function, slope and curvature zero at both ends.

This result shows that the transfer function for a weighting function with derivative continuity depends only on the value of the weighting function and its derivatives at the two ends of the sensor, modulated by a sine or cosine function. As higher derivatives are considered, the contributions of the weighting function and its derivatives decreases faster as a function of k. Further, as the weighting function $f(x)$ is tapered more smoothly towards zero at the ends, $f^{(i)}(\pm l/2) = 0$ for higher and higher values of i, and rolloff rate increases. Table 2.2 summarizes this behavior, which is critical to satisfying Functional Requirement #2. When the weighting is nonzero at either or both ends, as for the rectangular weighting, only $1/k$ rolloff is possible. When the weighting goes to zero at both ends, as for the Bartlett weighting, $1/k^2$ rolloff is possible. When the weighting and its slope got to zero at both ends, as for the Hanning weighting, $1/k^3$ rolloff is possible. The rolloff increases as more and more derivatives of the weighting are made zero at the ends.

Since the weighting function $f(x)$ must have a continuous derivative for (2.40) and Table 2.2 to be valid, the Bartlett (triangular) and Bartlett-Bartlett weightings must be considered differently. For a weighting function with a derivative discontinuity at

$x = 0$, for example, the integral (2.35) must be written

$$y(k) = \mathcal{T}(k) = \int_{-l/2}^{0^-} f(x)\cos(kx)\,dx + \int_{0^+}^{l/2} f(x)\cos(kx)\,dx \tag{2.44}$$

Equation (2.40) is modified, and the result is

$$\begin{aligned} y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{2i+1}}\Big[f^{(2i)}(-l/2) + f^{(2i)}(l/2)\Big]\sin(kl/2) \\ & + \frac{(-1)^i}{k^{2i+2}}\Big[f^{(2i+1)}(0^-) - f^{(2i+1)}(0^+) \\ & \qquad + \Big[-f^{(2i+1)}(-l/2) + f^{(2i+1)}(l/2)\Big]\cos(kl/2)\Big]\Bigg\} \end{aligned} \tag{2.45}$$

where $x = 0^-$ is a point just before the discontinuity, and $x = 0^+$ is a point just after the discontinuity. The derivation of this expression is shown in Section A.1 of Appendix A.

Assuming that $f(x)$ is symmetric, and Equations (2.41) and (2.42) apply, (2.45) simplifies to

$$\begin{aligned} y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{2(-1)^i}{k^{2i+1}}\Big[f^{(2i)}(l/2)\sin(kl/2)\Big] \\ & + \frac{2(-1)^i}{k^{2i+2}}\Big[-f^{(2i+1)}(0^+) + f^{(2i+1)}(l/2)\cos(kl/2)\Big]\Bigg\} \end{aligned} \tag{2.46}$$

This result shows the appearance of an extra term proportional to the value of the odd derivatives of $f(x)$ at the point of derivative discontinuity. Table 2.3 summarizes the rolloff behavior of sensors with weightings containing a derivative discontinuity. Again, $1/k$ rolloff is assured for any weighting function $f(x)$. To obtain $1/k^2$ rolloff, we must have $f(\pm l/2) = 0$, as for a weighting with no derivative discontinuity. However, to achieve $1/k^3$ rolloff, we need not only $f(\pm l/2) = 0$ and $f'(\pm l/2) = 0$, but also $f'(0^{\pm}) = 0$. That is, the slope of $f(x)$ at the point of higher derivative discontinuity

Table 2.3: Summary of rolloff properties of the sensor transfer function $\mathcal{T}(k)$ for $f(x)$ with a derivative discontinuity.

$f^{(i)}(\pm l/2)$	Rolloff		Comments
$f(\pm l/2) \neq 0$	$1/k$	-20 dB/decade	Weighting function nonzero at either end.
$f(\pm l/2) = 0$	$1/k^2$	-40 dB/decade	Weighting function zero both ends.
$f'(\pm l/2) = 0$ $f'(0^{\pm}) = 0$	$1/k^3$	-60 dB/decade	Weighting function zero at both ends and slope zero at discontinuity and both ends.
$f(\pm l/2) = 0$ $f'(0^{\pm}) = 0$ $f''(\pm l/2) = 0$	$1/k^4$	-80 dB/decade	Weighting function and curvature zero at both ends and slope zero at discontinuity and both ends.

must be zero. Since this requirement can only be satisfied when $f(x)$ has no first derivative (slope) discontinuity, it is clear that no more than $1/k^2$ rolloff is obtainable for such weightings. In general, for weightings with a discontinuity in the i-th derivative, only $1/k^{i+1}$ rolloff is possible.

At this point the properties of transfer functions of the weightings in a spatially cosinusoidal strain field can be investigated in more detail. Table 2.4 shows the predicted rolloff rates of the selected weighting functions introduced earlier. The rectangular weighting has $f(\pm l/2) \neq 0$, and should therefore have $1/k$ rolloff by Table 2.2. The Bartlett window has $f(\pm l/2) = 0$ but also a slope discontinuity at $x = 0$, and should therefore have $1/k^2$ rolloff by Table 2.3. The Hanning window tapers smoothly to zero and has $f'(\pm l/2) = 0$. This means it should have $1/k^3$ rolloff. All the derivatives of the infinite windows approach zero at $\pm\infty$, which suggests that their rolloff rate should be $1/k^{\infty}$ far above the rolloff frequency. For the compound weightings, the Bartlett-Bartlett weighting has a slope discontinuity at $x = 0$, and should therefore have $1/k^2$ rolloff. For the Hanning-Hanning weighting, all derivatives up to and including the third are zero. This suggests a rolloff rate of $1/k^5$. The Gauss-Hanning weighting simply tapers to zero as a Hanning weighting, and should therefore

Table 2.4: Predicted frequency properties of selected strain-averaging sensors.

Sensor type	Sensor Weighting	Predicted rolloff	
Finite	Rectangle	$1/k$	-20 dB/Decade
	Bartlett	$1/k^2$	-40 dB/Decade
	Hanning	$1/k^3$	-60 dB/Decade
Infinite	Sinc	$1/k^\infty$	$-\infty$ dB/Decade
	Sinc-Sinc	$1/k^\infty$	$-\infty$ dB/Decade
	Gauss	$1/k^\infty$	$-\infty$ dB/Decade
Compound	Bartlett-Bartlett	$1/k^2$	-40 dB/Decade
	Hanning-Hanning	$1/k^5$	-100 dB/Decade
	Gauss-Hanning	$1/k^3$	-60 dB/Decade

have the same rolloff rate, $1/k^3$.

Analytic expressions for the transfer functions of the weightings presented in Table 2.4 will now be derived and discussed in more detail.

Finite Spatial Sensor Weightings

Rectangular Weighting: The simplest weighting function is the rectangular weighting. It simply averages a signal evenly over a certain domain. The transfer function for the rectangular sensor is a Sinc function [50]. The transfer function is easily derived from (2.40). From a mathematical viewpoint, the weighting function $f(x)$ is nonzero at the endpoints ($x = \pm l/2$), but all the higher derivatives of the rectangular weighting are zero at the ends of the sensor. Thus only a single term remains from

the infinite series in (2.40):

$$\mathcal{F}(k) = \frac{1}{k}\Big[f(-l/2) + f(l/2)\Big] \sin\frac{kl}{2} \tag{2.47}$$

For the rectangular weighting, $f(-l/2) = f(l/2) = \alpha/(2\sqrt{2})$ and $l/2 = \sqrt{2}/\alpha$ as shown in Equations (2.7) and (2.8). The transfer function then becomes

$$\mathcal{T}(k) = \frac{1}{k}\left[\frac{\alpha}{2\sqrt{2}} + \frac{\alpha}{2\sqrt{2}}\right] \sin\frac{\sqrt{2}k}{\alpha} \tag{2.48}$$

$$\mathcal{T}(k) = \frac{\alpha}{\sqrt{2}k} \sin\frac{\sqrt{2}k}{\alpha} \tag{2.49}$$

Using $\bar{k} = k/\alpha$, we find

$$\mathcal{T}(k) = \frac{\sin\left(\sqrt{2}\bar{k}\right)}{\sqrt{2}\bar{k}} \tag{2.50}$$

Figure 2.11 shows the transfer function and phase for the rectangular weighting. The transfer function begins to roll off for $\bar{k}$ slightly less than unity. The rolloff point is defined as the point where the magnitude of the transfer function goes below $1/\sqrt{2}$, or about -3 dB. Note that due to the logarithmic scales employed, it is unclear from the figure that the transfer function is indeed a Sinc function. Equation (2.50) and Table 2.4 show that the rolloff rate of the rectangular weighting must be $1/k$, or -20 dB/decade, as shown in Figure 2.11. From the phase plot, it is clear that there are sign reversals at each zero of the transfer function shown in (2.50). The first side lobe of the transfer function has a height of -13.26 dB [21].

Bartlett Weighting: The Bartlett weighting is used to provide better side lobe response than the rectangular weighting. This weighting can be expressed as a convolution of two rectangular weightings of half the width of the Bartlett weighting [37].

The transfer function for the Bartlett or triangular weighting is the square of a

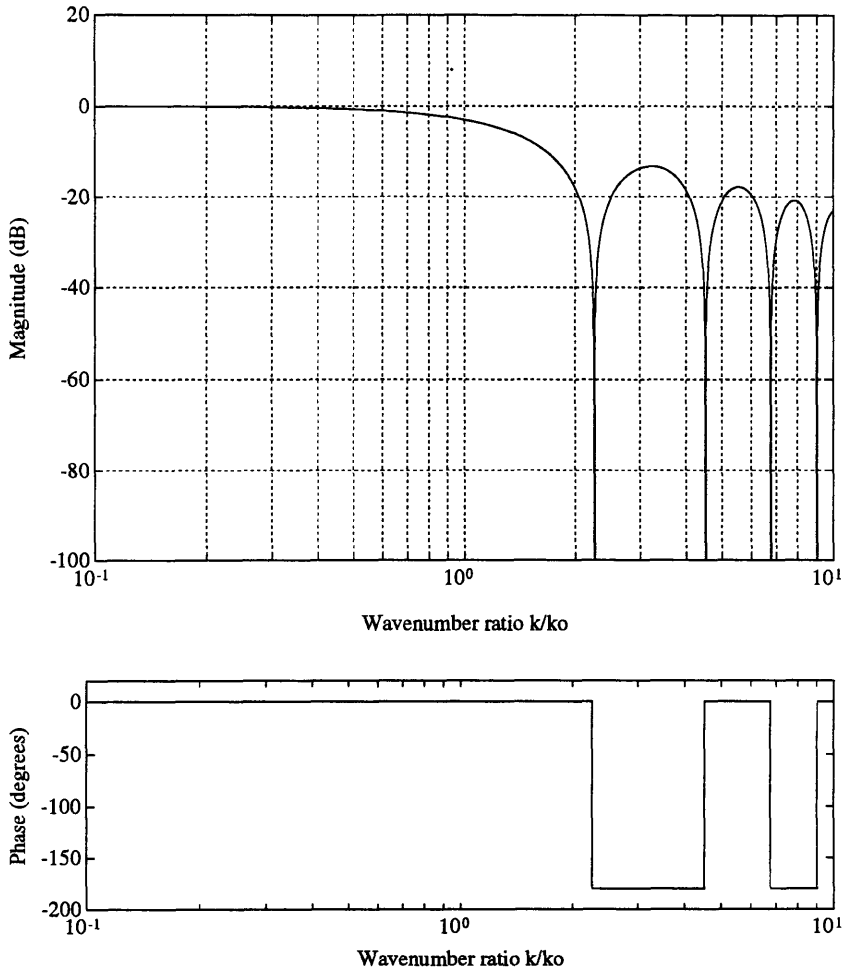

Figure 2.11: The transfer function and phase of the rectangular, or box car weighting.

Sinc function. The transfer function can be derived from (2.45) which takes into account the fact that the Bartlett weighting does not possess first derivative (slope) continuity over the interval $[-l/2, l/2]$. Only $f(x)$ and $f'(x)$ are nonzero for the Bartlett weighting.

Thus only two terms in the infinite series of (2.45) remain, and we obtain:

$$\begin{aligned}\mathcal{T}(k) &= \frac{1}{k}\Big[f(-l/2)\sin(kl/2)\Big] \\ &+ \frac{1}{k^2}\Big[-f'(-l/2)\cos(kl/2) + f'(0^-)\Big] \\ &+ \frac{1}{k}\Big[f(l/2)\sin(kl/2)\Big] \\ &+ \frac{1}{k^2}\Big[-f'(0^+) + f'(l/2)\cos(kl/2)\Big] \end{aligned} \tag{2.51}$$

$$\begin{aligned}\mathcal{T}(k) &= \frac{1}{k}\Big[f(-l/2) + f(l/2)\Big]\sin(kl/2) \\ &+ \frac{1}{k^2}\Big[f'(0^-) - f'(0^+) + \Big[-f'(-l/2) + f'(l/2)\Big]\cos(kl/2)\Big] \end{aligned} \tag{2.52}$$

For the Bartlett weighting, $f(-l/2) = f(l/2) = 0$, $f'(-l/2) = f(0^-) = \alpha^2/4$, and $f'(l/2) = f(0^+) = -\alpha^2/4$ as shown in (2.9). In addition, $l/2 = 2/\alpha$, from (2.10). Thus (2.52) simplifies to

$$\mathcal{T}(k) = \frac{1}{k^2}\left(\frac{\alpha^2}{4} + \frac{\alpha^2}{4} + \left(-\frac{\alpha^2}{4} - \frac{\alpha^2}{4}\right)\cos\frac{2k}{\alpha}\right) \tag{2.53}$$

$$\mathcal{T}(k) = \frac{\alpha^2}{k^2}\left(\frac{1}{2} - \frac{1}{2}\cos\frac{2k}{\alpha}\right) \tag{2.54}$$

$$\mathcal{T}(k) = \frac{1}{\bar{k}^2}\left(\frac{1 - \cos 2\bar{k}}{2}\right) \tag{2.55}$$

using $\bar{k} = k/\alpha$. Further, using the fact that $1 - \cos 2\theta = 2\sin^2\theta$, the transfer function (2.55) simplifies to

$$\mathcal{T}(k) = \frac{\sin^2\bar{k}}{\bar{k}^2} \tag{2.56}$$

Figure 2.12 shows plots of the magnitude and phase of the transfer function for the Bartlett sensor. In Table 2.4 it was predicted that the rolloff of the Bartlett weighting would be $1/k^2$. From Figure 2.12 and Equation (2.56) it is clear that this is indeed the case. In addition, the transfer function is non-negative for all values of

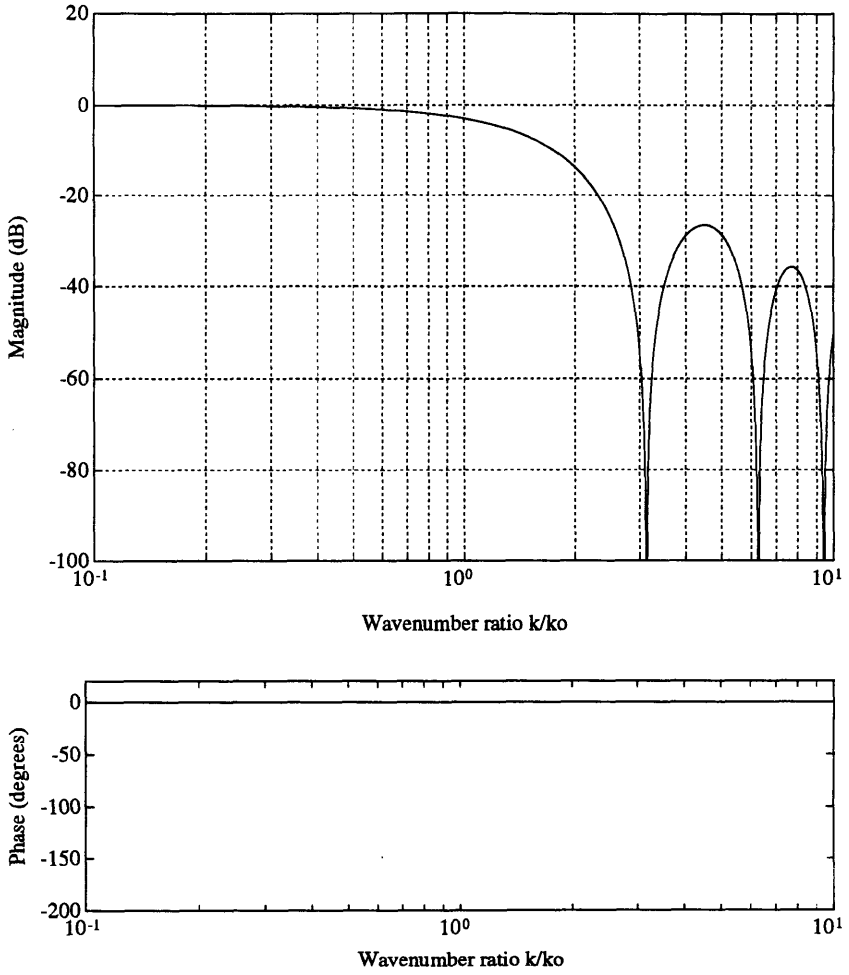

Figure 2.12: The transfer function and phase of the Bartlett weighting.

k, which indicates that the rolloff shown in Figure 2.12 is not accompanied by any phase flips of $\pm 180°$.

A convolution in the spatial domain corresponds to a multiplication in the spatial frequency domain. Thus since the Bartlett weighting is the convolution of two rectangular weightings, the transfer function of the Bartlett window is the square of the transfer function of the rectangular weighting. As can be seen from Figure 2.12, the side lobes of the transfer function for the Bartlett weighting have been attenuated significantly compared to those for the transfer function of the rectangular weighting.

The first side lobe is at -26.52 dB, exactly two times lower than for the rectangular weighting.

Hanning Weighting: The side lobes of the Bartlett weighting can be attenuated further by using the principle of superposition [37]. The result is the Hanning window. Its transfer function is

$$\mathcal{T}(k) = \frac{1}{2}\frac{\sin\left(\pi(\bar{k}/\sqrt{2}+1)\right)}{\pi(\bar{k}/\sqrt{2}+1)} + \frac{1}{2}\frac{\sin\left(\pi(\bar{k}/\sqrt{2}-1)\right)}{\pi(\bar{k}/\sqrt{2}-1)} + \frac{\sin(\pi\bar{k}/\sqrt{2})}{\pi\bar{k}/\sqrt{2}} \qquad (2.57)$$

Figure 2.13 shows the magnitude and phase of the transfer function for the Hanning weighting. The Hanning weighting yields a rolloff rate of -60 dB/decade, as predicted in Table 2.4, an improvement of -20 dB/decade over the Bartlett weighting. However, from the phase plot in Figure 2.13 is can be seen that there are now phase reversals at every zero of the transfer function. The first side lobe has a height of -31.47 dB, a fair improvement over the Bartlett weighting.

Infinite Spatial Sensor Weightings

Sinc Weighting: A Sinc sensor is an example of a sensor which satisfies some, but not all the functional requirements introduced in Chapter 1. Its weighting is that of a Sinc function, and its transfer function is like a perfect lowpass filter. The magnitude of this transfer function is unity for all frequencies below the cutoff, and zero at all frequencies above the cutoff. The magnitude immediately drops from unity to zero at the cutoff frequency, and no phase lag is introduced. This is seen to satisfy Functional Requirement #2, but since the weighting is negative for some values of x, Functional Requirement #8 is not satisfied. In addition, Functional Requirement #7 is not satisfied because the weighting is infinitely long.

The transfer function and phase of the Sinc weighting is the perfect lowpass filter

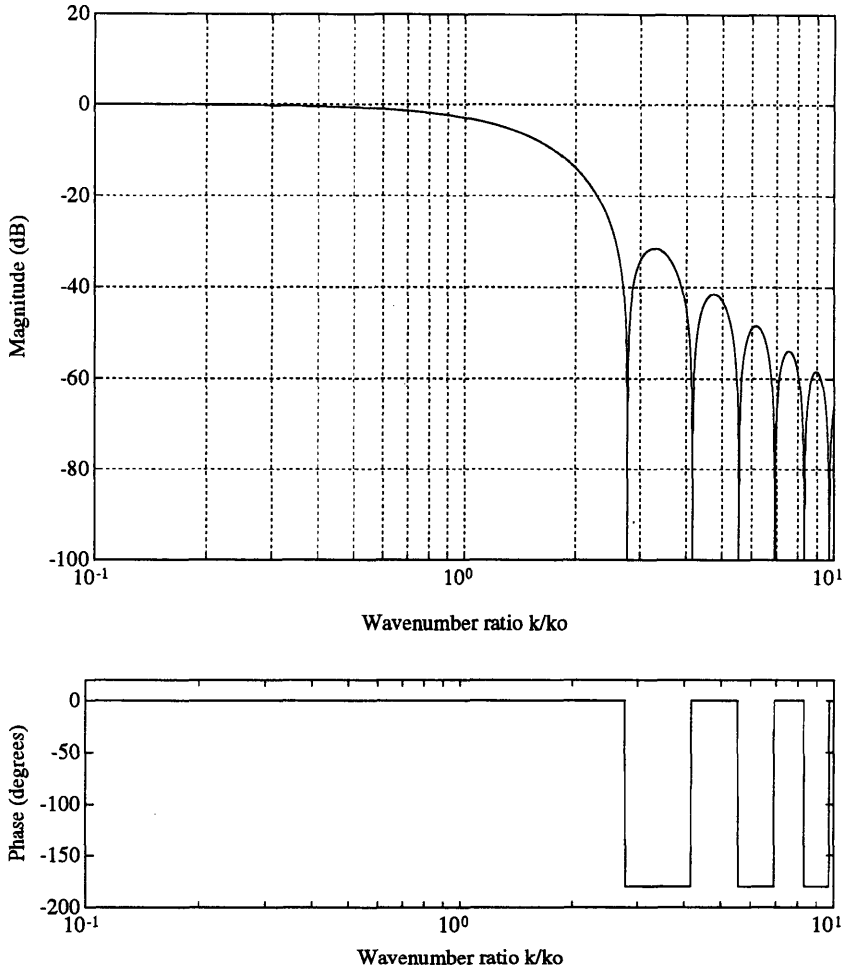

Figure 2.13: The transfer function and phase of the Hanning weighting.

shown in Figure 2.14. It is defined by

$$\mathcal{F}(k) = \frac{1 + \text{sgn}(1 - \bar{k})}{2} \tag{2.58}$$

where $\bar{k} = k/\alpha$, and where $\text{sgn}(x)$ is the sign function, defined by

$$\text{sgn}(x) = \begin{cases} 1, & x > 0 \\ -1, & x < 0 \end{cases} \tag{2.59}$$

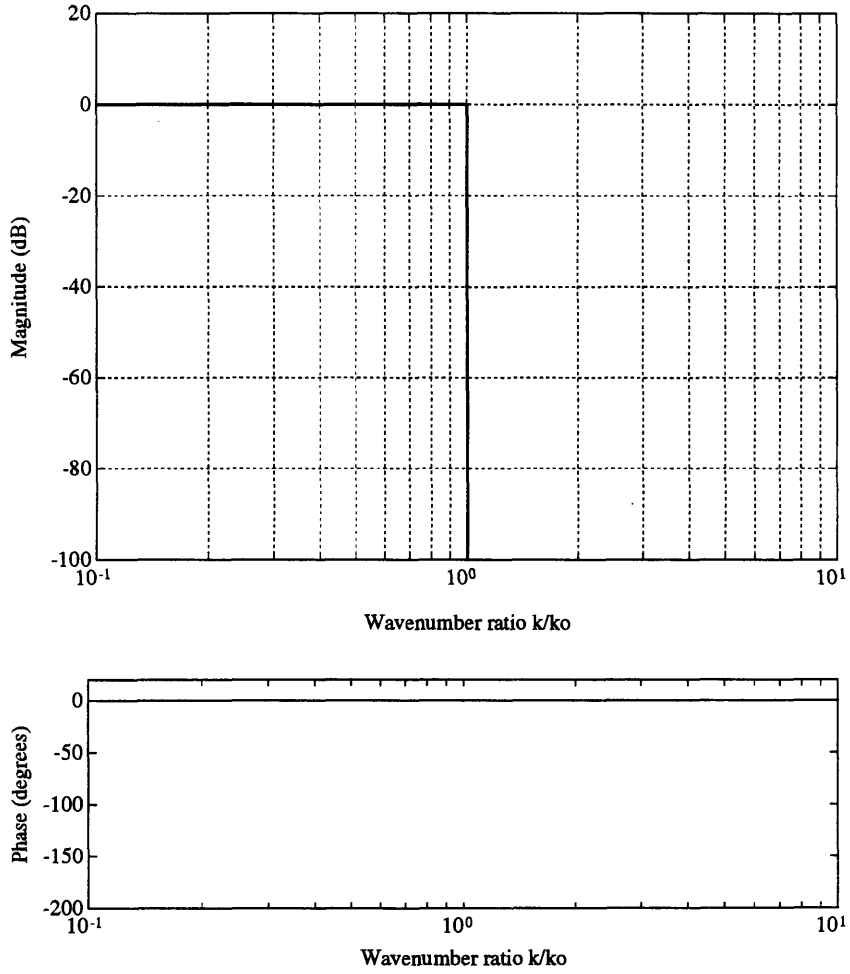

Figure 2.14: The transfer function and phase of the Sinc weighting.

The parameter α introduced in (2.15) is used to define the spatial frequency k at which the sensor rolls off. In this case, rolloff occurs at exactly $\bar{k} = 1$, or when $k = \alpha$.

The transfer function is unity for all wave numbers k less than the cutoff wave number, and zero for all k above it. The transition from perfect transmission to perfect impedance is instantaneous, and no phase lag is introduced. This behavior is only exhibited by a Sinc weighting that extends over an infinite domain. For implementation to be possible, the sensor must be spatially truncated at the boundaries of the structure. Such a truncated sensor can accurately observe modes in the bandwidth

and will roll off relatively fast, satisfying Functional Requirements #1 and #2. However, the magnitude rolloff of the output of the truncated sensor will contain negative regions, where the sign of the output of the sensor is opposite to that of the strain at the center of the sensor. In space, truncation is a multiplication of the sensor weighting by a rectangular weighting. In the wave number domain, the Fourier transform of the truncated sensor is the result of convolving the perfect low pass filter with the transfer function of the rectangular weighting. The result contains negative regions, i.e. regions where the phase has shifted by $\pm 180°$. Functional Requirement #3 is therefore violated.

Sinc-Sinc Weighting: Since the Sinc-Sinc weighting is obtained by multiplying two Sinc functions together, it follows that the transfer function for the Sinc-Sinc weighting must be the convolution of the transfer functions of two Sinc functions. Thus the transfer function of the Sinc-Sinc weighting is triangular. It is shown in Figure 2.15 and is given by:

$$\mathcal{T}(k) = \begin{cases} 1 - \dfrac{\bar{k}}{2+\sqrt{2}}, & \bar{k} < 2+\sqrt{2} \\ 0, & \bar{k} \geq 2+\sqrt{2} \end{cases} \tag{2.60}$$

The Sinc-Sinc weighting was investigated because it does not require regions of negative sensitivity, and yet yields excellent performance. As with the Sinc weighting, it suffers from the fact that it must be spatially truncated for the purposes of implementation, violating Functional Requirement #7.

Gauss Weighting: In general, the side lobes can be reduced by having the ends of the weighting approach zero more and more smoothly [44]. This is shown in Equation (2.40) and in Table 2.2 in that the rolloff rate is faster as higher and higher derivatives of the weighting function $f(x)$ are zero at the ends. A Gaussian is infinite in length and can be shown to approach zero infinitely smoothly as x approaches

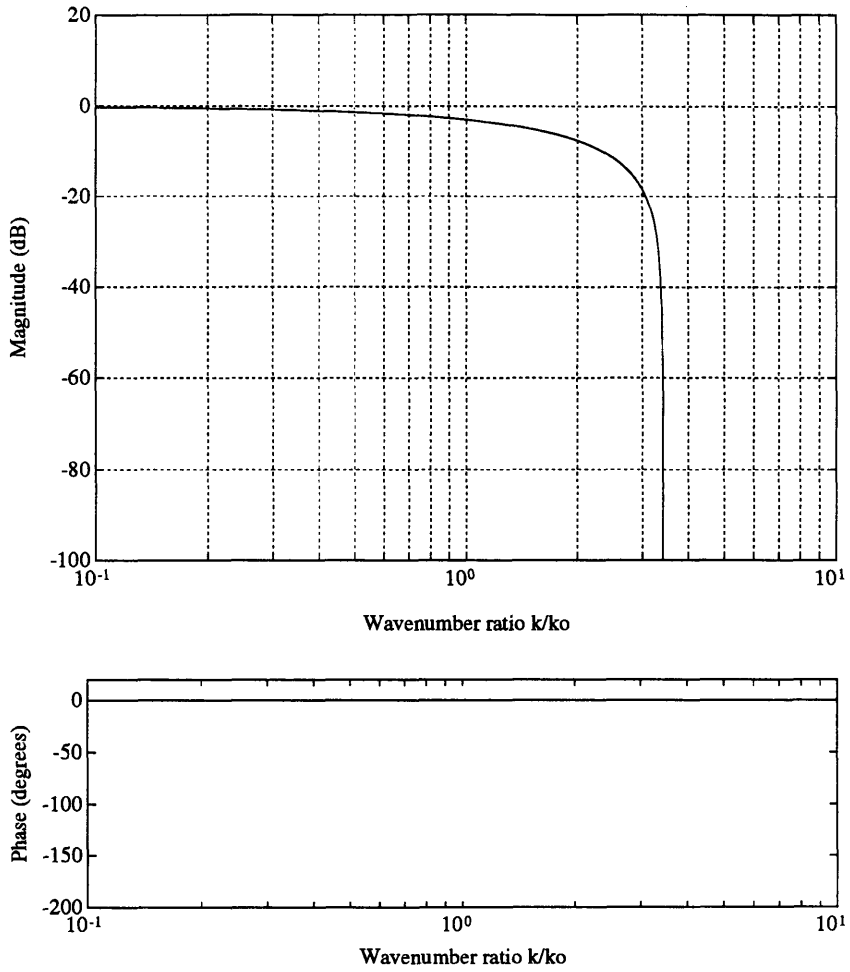

Figure 2.15: The transfer function and phase of the Sinc-Sinc weighting.

$\pm\infty$. The transfer function of a Gaussian is also a Gaussian:

$$\mathcal{T}(k) = \exp\left(-\frac{\ln 2}{2}\,\bar{k}^2\right) \tag{2.61}$$

Figure 2.16 shows the magnitude and phase of the transfer function for a Gaussian weighting. It can be seen from this plot and (2.61), that the transfer function of a Gaussian sensor has no side lobes at all, and the rolloff becomes increasingly fast. In fact, at $k = \infty$ the rolloff is infinite, as predicted in Table 2.4. As shown in (2.61),

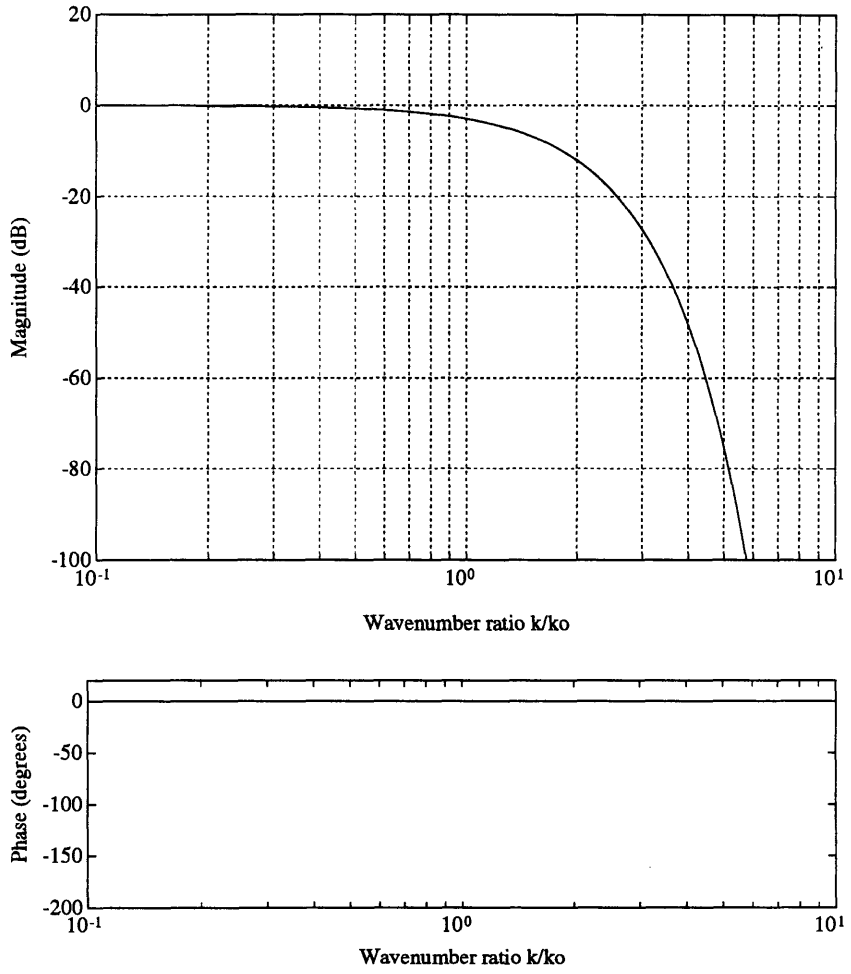

Figure 2.16: The transfer function and phase of the Gauss weighting.

the magnitude is everywhere nonnegative, implying that the transfer function of the Gaussian weighting will not exhibit any phase lag. Note again that the Gaussian weighting must be spatially truncated for implementation purposes.

Compound Spatial Sensor Weightings

Bartlett-Bartlett Weighting: The transfer function of the Bartlett-Bartlett weighting is derived from (2.45). From (2.21) it is clear that all derivatives $f^{(i)}(x)$ for $i > 2$

are zero. Thus (2.45) simplifies to

$$\begin{aligned}\mathcal{T}(k) &= \frac{1}{k}\Big[f(-l/2)+f(l/2)\Big]\sin(kl/2) \\ &+ \frac{1}{k^2}\Big[f'(0^-)-f'(0^+)+\Big[-f'(-l/2)+f'(l/2)\Big]\cos(kl/2)\Big] \\ &+ \frac{1}{k^3}\Big[-f''(-l/2)-f''(l/2)\Big]\sin(kl/2)\Big] \end{aligned} \tag{2.62}$$

Knowing that for the Bartlett-Bartlett weighting, $f(-l/2) = f(l/2) = 0$, $f'(-l/2) = f'(l/2) = 0$, $f'(0^-) = -f'(0^+) = 3\alpha^2/\pi^2$, $f''(-l/2) = f''(l/2) = 3\alpha^3/\pi^3$, and $l/2 = \pi/\alpha$, the expression for the transfer function simplifies to

$$\mathcal{T}(k) = \frac{1}{k^2}\left(\frac{3\alpha^2}{\pi^2}+\frac{3\alpha^2}{\pi^2}\right)+\frac{1}{k^3}\left(-\frac{3\alpha^3}{\pi^3}-\frac{3\alpha^3}{\pi^3}\right)\sin\frac{k\pi}{\alpha} \tag{2.63}$$

$$\mathcal{T}(k) = \frac{6\alpha^2}{\pi^2k^2}-\frac{6\alpha^3}{\pi^3k^3}\sin\frac{k\pi}{\alpha} \tag{2.64}$$

Using $\bar{k} = k/\alpha$, we find

$$\mathcal{T}(k) = 6\left(\frac{\pi\bar{k}-\sin(\pi\bar{k})}{(\pi\bar{k})^3}\right) \tag{2.65}$$

Figure 2.17 shows the magnitude and phase of the transfer function for the Bartlett-Bartlett weighting. As with the Bartlett weighting, the Bartlett-Bartlett weighting provides a rolloff rate of only $1/k^2$, as predicted in Table 2.4. However, unlike the Bartlett weighting, the transfer function of the Bartlett-Bartlett weighting approaches zero monotonically as k approaches infinity. Thus it possesses no side lobes or zeros, and introduces no phase lag.

Hanning-Hanning Weighting: In order to improve the side lobe structure of the Hanning weighting, the Hanning-Hanning weighting was created by multiplying two Hanning weightings together. This has the effect of convolving the transfer functions

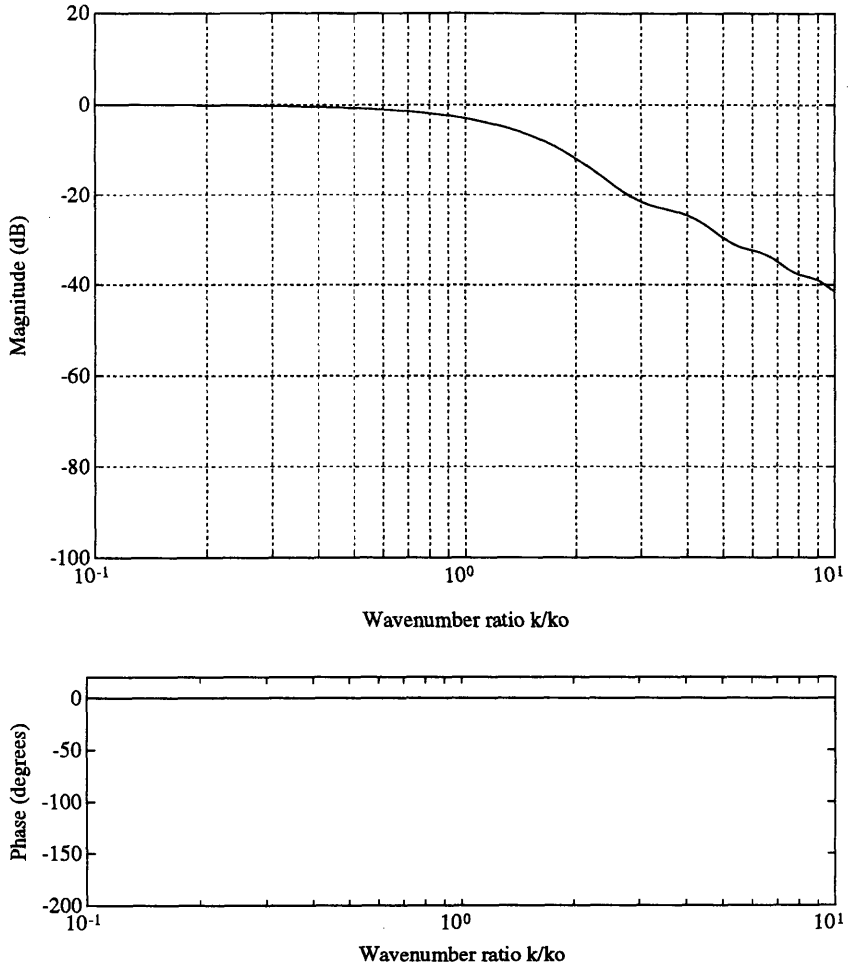

Figure 2.17: The transfer function and phase of the Bartlett-Bartlett weighting.

of two Hanning weightings. The transfer function is given by

$$\begin{aligned} \mathcal{T}(k) &= \frac{2}{3}\frac{\sin\big(\pi(\bar{k}+1)\big)}{\pi(\bar{k}+1)} + \frac{2}{3}\frac{\sin\big(\pi(\bar{k}-1)\big)}{\pi(\bar{k}-1)} \\ &+ \frac{1}{6}\frac{\sin\big(\pi(\bar{k}+2)\big)}{\pi(\bar{k}+2)} + \frac{1}{6}\frac{\sin\big(\pi(\bar{k}-2)\big)}{\pi(\bar{k}-2)} + \frac{\sin\pi\bar{k}}{\pi\bar{k}} \end{aligned} \tag{2.66}$$

The magnitude and phase of the transfer function for the Hanning-Hanning weighting is shown in in Figure 2.18. It can be seen that the rolloff rate has been increased

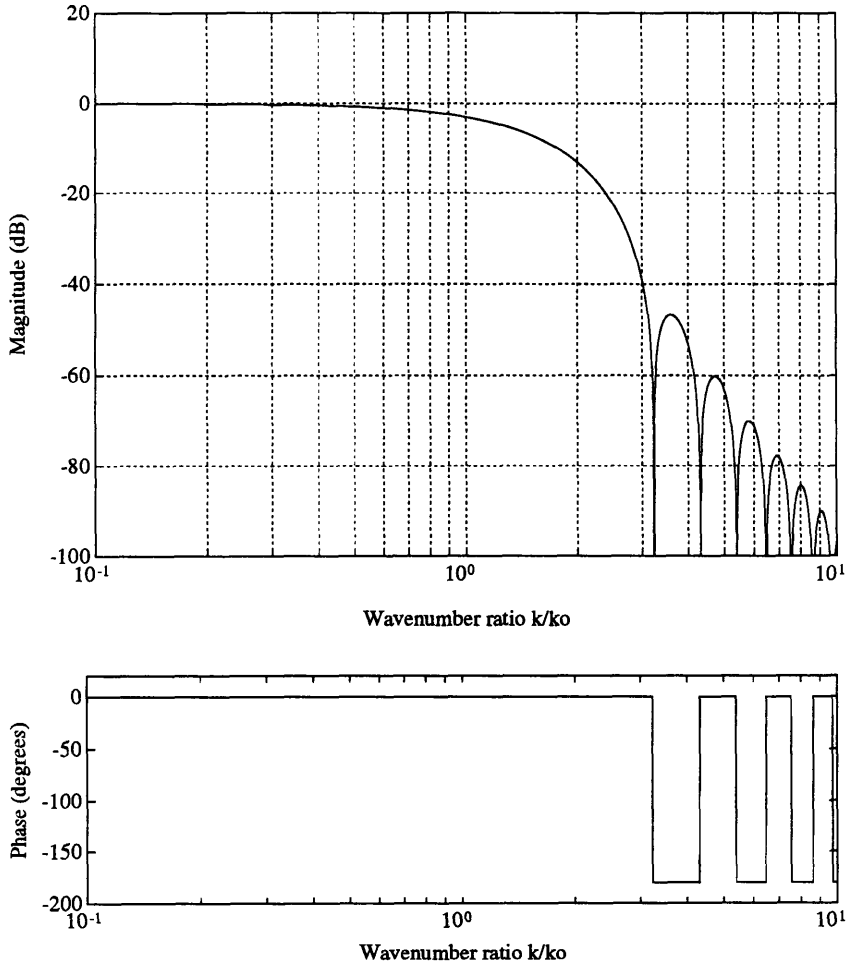

Figure 2.18: The transfer function and phase of the Hanning-Hanning weighting.

to -100 dB/decade (or $1/k^5$), as predicted in Table 2.4. The sign reversals at the zeros of the transfer function can be seen in the phase plot. The height of the first side lobe has been decreased to -46.74 dB, an improvement over the simple Hanning weighting. It can be seen that the improvement in rolloff rate and side lobe height has come at the expense of a spatially longer weighting.

Gauss-Hanning Weighting: In an effort to obtain a finite weighting with rapid rolloff, a Gaussian weighting was made finite by truncating it smoothly with a Han-

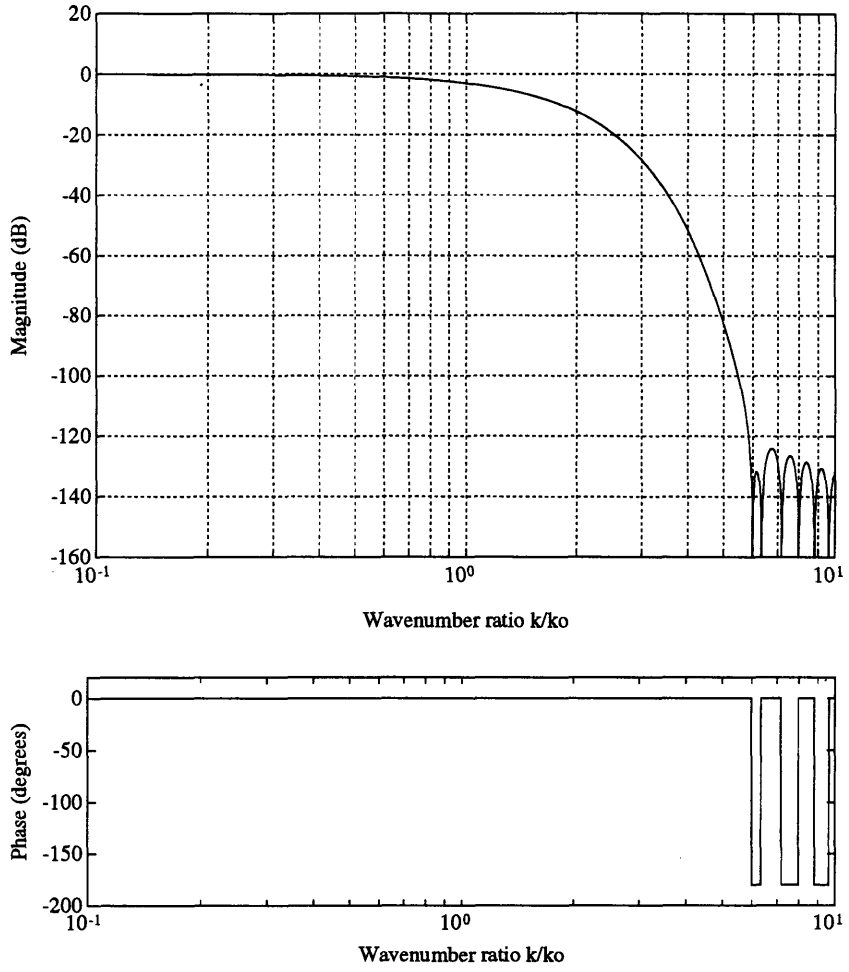

Figure 2.19: The transfer function and phase of the Gauss-Hanning weighting.

ning weighting. The result is a weighting that has very good rolloff in the decade beyond the rolloff point and yet is finite. The magnitude and phase of the transfer function is shown in Figure 2.19. The analytic expression for the transfer function is highly complex and is given in Appendix B. As predicted in Table 2.4, the magnitude eventually rolls off at the same rate as a simple Hanning window, $1/k^3$.

Table 2.5: Frequency properties of selected strain-averaging sensors.

Sensor Weighting	Rolloff in first decade (dB)	Rolloff rate (dB/decade)	Negatives in k?	Several zeroes?	First side lobe height (dB)
Rectangle	-20	-20	Yes	Yes	-13.26
Bartlett	-40	-40	No	Yes	-26.52
Hanning	-60	-60	Yes	Yes	-31.47
Sinc	$-\infty$	$-\infty$	No	No	-
Sinc-Sinc	$-\infty$	$-\infty$	No	No	-
Gauss	-300	$-\infty$	No	No	-
Bartlett-Bartlett	-40	-40	No	No	-
Hanning-Hanning	-100	-100	Yes	Yes	-46.74
Gauss-Hanning	-140	-60	Yes	Yes	-131.83

2.4.2 Summary

At this point the properties of the transfer functions of the weightings are summarized. Table 2.5 summarizes the actual spatial frequency properties of the selected weighting functions introduced earlier. In the second column, the rolloff point is defined as the -3 dB point, where the magnitude of the transfer function has fallen to $1/\sqrt{2}$, and the rolloff in the first decade is estimated from the magnitude decrease in one decade beyond the rolloff wave number. The Sinc, Gaussian and Sinc-Sinc weightings yield the most rolloff within one decade up from the -3 dB point, and yield infinite rolloff rate far above the rolloff point. The Gauss-Hanning weighting approaches this performance, and is spatially finite and thus implementable without further spatial truncation. Thus it satisfies not only Functional Requirement #2, but Functional Requirements #7 and #8. However, its eventual rolloff rate is that of a Hanning weighting, as predicted in Table 2.4. The third column gives the rolloff rate

the transfer function exhibits far above the rolloff point. The fourth column indicates whether the transfer function of the weighting contains any negative portions. Functional requirement #3 states that such negative regions should be avoided in order to avoid instabilities in a control system using spatially averaging strain sensor measurements. The fifth column of Table 2.5 shows whether there are multiple zeros in the transfer function. Those that do not have multiple zeros have one at $k = \infty$. It should be noted that transfer functions of the Sinc and Sinc-Sinc weightings go to zero at a finite value of k and remain zero for all larger values of k. The last column of Table 2.5 shows the height of the first side lobe, when such a lobe exists. The size or height of the side lobes determines the amount of leakage the window suffers from [47]. It is responsible for ripples in the pass- and stop-bands of the sensor's transfer function $\mathcal{T}(k)$.

The results shown in Table 2.5 indicate that only relatively complex weightings with stringent weighting continuity requirements at the ends offer high rolloff rates. In further investigations in this work, Bartlett and Gauss-Hanning sensors will be used, as well as point and rectangular sensors for reference.

2.4.3 Noncentered Sensors

Being able to center a sensor at the point of symmetry in a strain field represents an ideal condition. In general, sensors are not placed at such ideal locations. Therefore, the output of a sensor mounted in the interior of a structure, away from its boundaries, centered at a distance x_0 from the point of symmetry in a sinusoidal strain field is investigated. It is now assumed that the strain field is spatially sinusoidal and of the form

$$\epsilon(x) = \sin(kx) \tag{2.67}$$

where k is the wave number. The output of a sensor with weighting function $f(x)$, centered at $x = x_0$ is then given by

$$y(k) = \int_{-l/2+x_0}^{l/2+x_0} f(x - x_0)\sin(kx)\, dx \tag{2.68}$$

By repeatedly integrating this expression by parts, an infinite series solution for the output of a sensor mounted in the interior of a structure, away from its boundaries, centered at a distance x_0 from the point of symmetry in a sinusoidal strain field can be written as

$$\begin{aligned} y(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{(2i+1)}}\Big[-f^{(2i)}(l/2)\cos\big[k(l/2+x_0)\big] + f^{(2i)}(-l/2)\cos\big[k(-l/2+x_0)\big]\Big] \\ & + \frac{(-1)^i}{k^{(2i+2)}}\Big[f^{(2i+1)}(l/2)\sin\big[k(l/2+x_0)\big] - f^{(2i+1)}(-l/2)\sin\big[k(-l/2+x_0)\big]\Big]\Bigg\} \end{aligned} \tag{2.69}$$

To obtain the transfer function from the strain at the center of the sensor to its output, (2.69) must be divided by the strain at the center of the sensor, $\sin(kx_0)$:

$$\begin{aligned} \mathcal{T}(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{(2i+1)}}\left[\frac{-f^{(2i)}(l/2)\cos\big[k(l/2+x_0)\big] + f^{(2i)}(-l/2)\cos\big[k(-l/2+x_0)\big]}{\sin(kx_0)}\right] \\ & + \frac{(-1)^i}{k^{(2i+2)}}\left[\frac{f^{(2i+1)}(l/2)\sin\big[k(l/2+x_0)\big] - f^{(2i+1)}(-l/2)\sin\big[k(-l/2+x_0)\big]}{\sin(kx_0)}\right]\Bigg\} \end{aligned} \tag{2.70}$$

In Section A.1 of Appendix A, this expression is simplified by expanding sine and cosine terms, leaving

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \left[-f^{(2i)}(l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} - \sin(kl/2) \right) \right. \right.$$
$$\left. + f^{(2i)}(-l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} + \sin(kl/2) \right) \right]$$
$$+ \frac{(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \left(\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right.$$
$$\left. \left. - f^{(2i+1)}(-l/2) \left(-\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right] \right\} \tag{2.71}$$

It is now shown that if the sensor if symmetric, (2.71) simplifies to (2.43), showing that the transfer function between the strain at the center or the sensor and its output is independent of x_0, and hence where the sensor is centered with respect to the point of symmetry. Using the identities (2.41) and (2.42) that are valid when $f(x)$ is symmetric, (2.71) is simplified to the form

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \left[-f^{(2i)}(l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} - \sin(kl/2) \right) \right. \right.$$
$$\left. + f^{(2i)}(l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} + \sin(kl/2) \right) \right]$$
$$+ \frac{(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \left(\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right.$$
$$\left. \left. + f^{(2i+1)}(l/2) \left(-\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right] \right\} \tag{2.72}$$

which further simplifies to

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{2(-1)^i}{k^{(2i+1)}} f^{(2i)}(l/2)\sin(kl/2) + \frac{2(-1)^i}{k^{(2i+2)}} f^{(2i+1)}(l/2)\cos(kl/2) \right\} \tag{2.73}$$

This expression is clearly the same as (2.40), indicating that the transfer function is indeed independent of x_0 as long as $f(x)$ is symmetric.

This result can also be derived for weightings containing a derivative discontinuity, such as the Bartlett weighting. For such a weighting, (2.68) is first modified:

$$y(k) = \int_{-l/2+x_0}^{x_0^-} f(x-x_0)\sin(kx)\,dx + \int_{x_0^+}^{l/2+x_0} f(x-x_0)\sin(kx)\,dx \tag{2.74}$$

where x_0^- is a point just before the discontinuity in $f(x)$, and x_0^+ is a point just after it. Equation (2.74) is integrated repeatedly to obtain:

$$\begin{aligned} y(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{(2i+1)}} \Big[-f^{(2i)}(0^-)\cos(kx_0) + f^{(2i)}(-l/2)\cos\big[k(-l/2+x_0)\big] \\ & \qquad - f^{(2i)}(l/2)\cos\big[k(l/2+x_0)\big] + f^{(2i)}(0^+)\cos(kx_0) \Big] \\ & + \frac{(-1)^i}{k^{(2i+2)}} \Big[f^{(2i+1)}(0^-)\sin(kx_0) - f^{(2i+1)}(-l/2)\sin\big[k(-l/2+x_0)\big] \\ & \qquad + f^{(2i+1)}(l/2)\sin\big[k(l/2+x_0)\big] - f^{(2i+1)}(0^+)\sin(kx_0) \Big] \Bigg\} \end{aligned} \tag{2.75}$$

Assuming $f(x)$ is symmetric, and expanding the trigonometric terms, (2.75) simplifies to

$$y(k) = \sum_{i=0}^{\infty} \left\{ \frac{2(-1)^i}{k^{(2i+1)}} \left[f^{(2i)}(l/2) \sin(kl/2) \right] \sin(kx_0) + \frac{2(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \cos(kl/2) - f^{(2i+1)}(0^+) \right] \sin(kx_0) \right\} \tag{2.76}$$

The transfer function is then obtained by dividing (2.76) by $\sin(kx_0)$ to obtain

$$y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{2(-1)^i}{k^{2i+1}} f^{(2i)}(-l/2) \sin(kl/2) + \frac{2(-1)^i}{k^{2i+2}} \left[f^{(2i+1)}(l/2) \cos(kl/2) - f^{(2i+1)}(0^+) \right] \right\} \tag{2.77}$$

It can be seen that this is exactly the same as (2.46), thus showing that the transfer function of a weighting with a derivative discontinuity is independent of x_0, and therefore where it is placed relative to the point of symmetry in the strain field. In addition, this shows that the sensor transfer function is the same, no matter whether the strain field is assumed to be spatially sinusoidal or cosinusoidal, proving that the sine and cosine transforms of a spatially symmetric weighting function are the same.

2.4.4 Sensors Simply Truncated at Structure Boundaries

In distributing an array of sensors on a structure, sensors will inevitably have to be placed in such a way as to cause some portion of the sensor weighting to

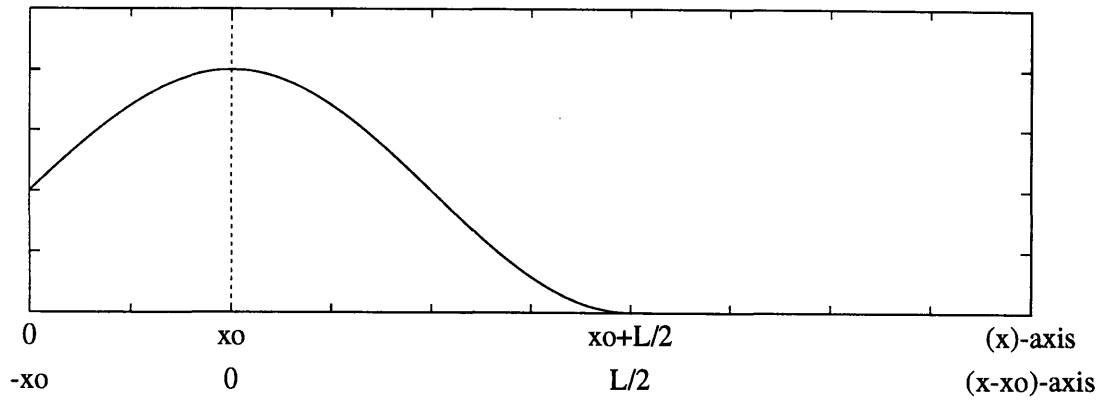

Figure 2.20: Sensor truncated at a boundary of the structure.

fall beyond a boundary of the structure. This portion must be truncated in the implementation of the sensor. The frequency characteristics of the transfer function of a sensor are changed dramatically if a portion of the sensor is truncated at a boundary of the structure. The output of a sensor mounted near a boundary of a structure, at a distance x_0 from the point of symmetry in a sinusoidal strain field is therefore investigated. It is again assumed that the strain field is of the form

$$\epsilon(x) = \sin(kx) \tag{2.78}$$

where k is the wave number. It is further assumed that the boundary of the structure is at $x = 0$. The output of a sensor with weighting function $f(x)$, centered at $x = x_0$ is then given by

$$y(k) = \int_0^{x_0+l/2} f(x - x_0)\sin(kx)\, dx \tag{2.79}$$

Figure 2.20 shows a schematic of a sensor truncated at a boundary of the structure. Such a truncated weighting must be scaled such that low spatial frequency signals

are measured correctly. This is accomplished by requiring that

$$\int_0^{l/2+x_0} f(x - x_0)\, dx = 1 \tag{2.80}$$

The upper horizontal axis system in Figure 2.20 is the axis system of the structure, where the boundary has been set at $x = 0$. The lower horizontal axis system is relative to the sensor, where $x = 0$ has been set at the center of the sensor. The distance between these two points is x_0. By repeatedly integrating (2.79) by parts, an infinite series solution for the output of a sensor mounted in the interior of a structure, away from its boundaries, centered at a distance x_0 from the point of symmetry in a sinusoidal strain field can be written as

$$\begin{aligned} y(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{(2i+1)}} \Big[-f^{(2i)}(l/2) \cos\big[k(l/2 + x_0)\big] + f^{(2i)}(-x_0) \Big] \\ & + \frac{(-1)^i}{k^{(2i+2)}} \Big[f^{(2i+1)}(l/2) \sin\big[k(l/2 + x_0)\big] \Big] \Bigg\} \end{aligned} \tag{2.81}$$

The derivation of this expression is shown in Section A.1 of Appendix A. Equation (2.81) is divided by $\sin(kx_0)$ in order to obtain the transfer function for a sensor truncated at a structure boundary in a sinusoidal strain field. The derivation given in Section A.1 of Appendix A yields

$$\begin{aligned} \mathcal{T}(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{(2i+1)}} \left[-f^{(2i)}(l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} - \sin(kl/2) \right) + f^{(2i)}(-x_0) \left(\frac{1}{\sin(kx_0)} \right) \right] \\ & + \frac{(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \left(\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right] \Bigg\} \end{aligned} \tag{2.82}$$

The term $f^{(2i)}(-x_0)$ is the value of the weighting and all its even derivatives at the boundary of the structure. Although for a smoothly tapering weighting, the

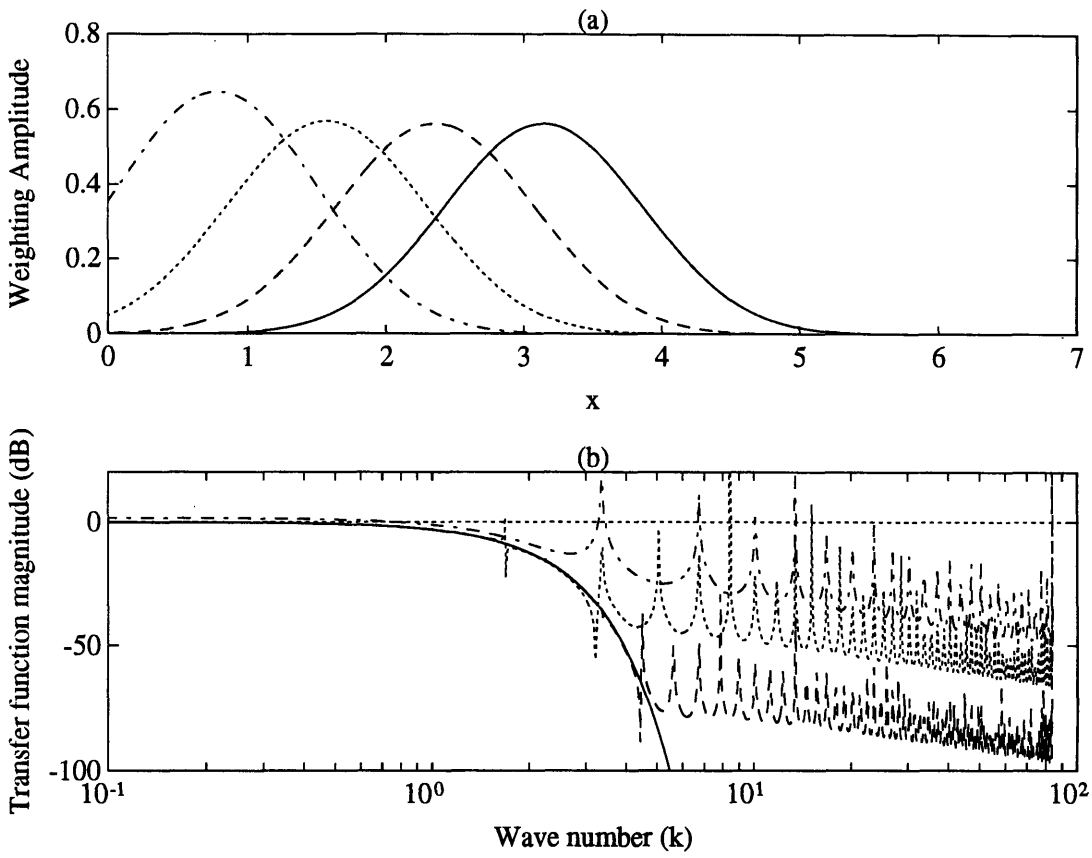

Figure 2.21: Transfer functions of Gauss-Hanning sensors as they are more and more truncated. Solid line: untruncated sensor; dashed line: eighth of sensor truncated; dotted line: quarter of sensor truncated; dot-dashed line: three eighths of sensor truncated.

$f^{(2i)}(\pm l/2)$ terms are generally zero for low values of i, the $f^{(2i)}(-x_0)$ terms are not. Therefore, the rolloff of the transfer function (2.82) is limited to $1/k$. The presence of the $\tan(kx_0)$ and $\sin(kx_0)$ terms in the denominator means that the transfer function $\mathcal{T}(k)$ is no longer bounded in amplitude, even when $f(x)$ is symmetric. This is simply because the sensor is no longer symmetric in its truncated state and can report a nonzero strain even when the strain at its center, $x = x_0$, is zero.

Figure 2.21 shows the effect of sensor truncation on the rolloff behavior of the sensor transfer function $\mathcal{T}(k)$. Figure 2.21(a) shows a series of four Gauss-Hanning weightings. The first sensor, shown in a solid line, is not truncated at all. The next

three weightings, shown in dashed, dotted and dot-dashed lines respectively, are more and more drastically truncated. Figure 2.21(b) shows the transfer functions for the untruncated and three truncated sensors. It is clear that as soon as the sensor is truncated by any small amount, the long-term rolloff rate becomes $1/k$. As more and more of the sensor is truncated, the average magnitude of $\mathcal{T}(k)$ beyond the rolloff point increases. Finally, it is clear that the transfer functions of the truncated sensors are no longer bounded in amplitude. This is because the sensor can report a nonzero strain measurement even when the strain is zero at the center of the sensor.

In an effort to satisfy Functional Requirement #9, a powerful truncation technique which improves the rolloff behavior of the truncated sensor is presented in the next section.

2.4.5 Sensors Folded at Structure Boundaries

In this section an improved sensor truncation technique is discussed. The frequency characteristics are very interesting when the portion of the weighting function beyond the structure boundary is folded over about the boundary and subtracted from the remaining interior portion of the sensor weighting. Figure 2.22 shows this procedure.

The solid line shows the weighting of an untruncated sensor that extends beyond the boundary of the structure which is assumed to be at $x = 0$. The dashed line shows the truncated portion of the sensor after being reflected about the boundary of the structure and then made negative. This part is then added to the untruncated part of the original sensor. The resulting truncated sensor weighting is shown by the dot-dashed line. The frequency characteristics of this modified sensor are investigated

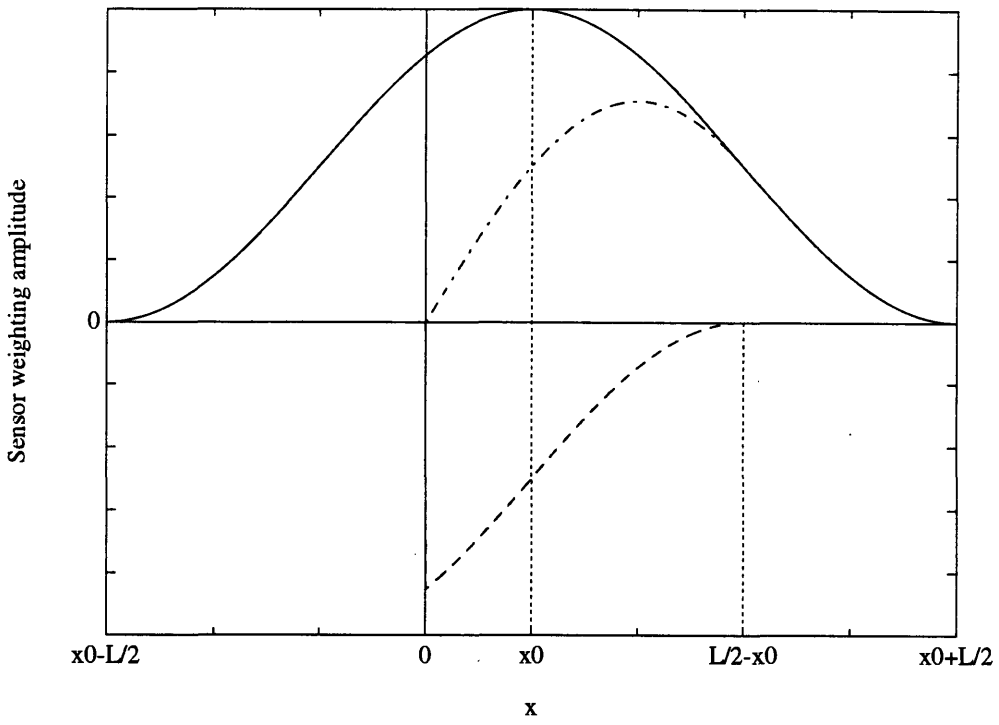

Figure 2.22: Folding over and subtracting the truncated portion of a sensor. Solid line: untruncated sensor; dashed line: truncated portion once folded and subtracted; dot-dashed line: resulting sensor with truncated portion folded and subtracted.

next. The output of the modified sensor is given by

$$y(k) = \int_0^{l/2-x_0} \left[f(x - x_0) - f(-x - x_0) \right] \sin(kx)\, dx + \int_{l/2-x_0}^{l/2+x_0} f(x - x_0) \sin(kx)\, dx \tag{2.83}$$

Note that we will require that

$$\int_{-l/2+x_0}^{l/2+x_0} f(x - x_0)\, dx = 1 \tag{2.84}$$

That is, the area of the *original*, untruncated sensor is unity. Note that no further scaling of $f(x)$ is performed. The weighting $f(-x - x_0)$ is the original weighting $f(x - x_0)$ reflected about the boundary of the structure at $x = 0$. The length of the

portion of the sensor that extends beyond the boundary of the sensor is $l/2 - x_0$. Equation (2.83) can be rewritten

$$y(k) = \int_0^{l/2+x_0} f(x - x_0)\, \sin(kx)\, dx - \int_0^{l/2-x_0} f(-x - x_0)\, \sin(kx)\, dx \quad (2.85)$$

$$y(k) = G(k) - H(k) \quad (2.86)$$

The first integral $G(k)$ in (2.85) was derived in the previous section and is shown in (2.81). The evaluation of the second integral $H(k)$ is shown in Section A.1 of Appendix A. By repeated integration by parts, $H(k)$ can be written as an infinite series in the following manner:

$$H(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \Big[-f^{(2i)}(-l/2) \cos\Big[k(l/2 - x_0)\Big] + f^{(2i)}(-x_0) \Big] \right.$$
$$\left. + \frac{(-1)^i}{k^{(2i+2)}} \Big[-f^{(2i+1)}(-l/2) \sin\Big[k(l/2 - x_0)\Big] \Big] \right\} \quad (2.87)$$

By subtracting this expression for $H(k)$ from (2.81), the output of the modified sensor as a function of frequency, Equation (2.86) can be found to be

$$y(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \Big[-f^{(2i)}(l/2) \cos\Big[k(l/2 + x_0)\Big] + f^{(2i)}(-l/2) \cos\Big[k(-l/2 + x_0)\Big] \Big] \right.$$
$$\left. + \frac{(-1)^i}{k^{(2i+2)}} \Big[f^{(2i+1)}(l/2) \sin\Big[k(l/2 + x_0)\Big] - f^{(2i+1)}(-l/2) \sin\Big[k(-l/2 + x_0)\Big] \Big] \right\} \quad (2.88)$$

It can be seen that this expression is the same as (2.69). Therefore, the transfer function for such a sensor will be the same as (2.71):

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \left[-f^{(2i)}(l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} - \sin(kl/2) \right) \right. \right.$$
$$\left. + f^{(2i)}(-l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} + \sin(kl/2) \right) \right]$$
$$+ \frac{(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \left(\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right.$$
$$\left. \left. - f^{(2i+1)}(-l/2) \left(-\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right] \right\} \tag{2.89}$$

In addition, the transfer function of an originally symmetric sensor which has been truncated by this folding approach is the same as (2.73):

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{2(-1)^i}{k^{(2i+1)}} f^{(2i)}(l/2) \sin(kl/2) + \frac{2(-1)^i}{k^{(2i+2)}} f^{(2i+1)}(l/2) \cos(kl/2) \right\} \tag{2.90}$$

This result shows that sensors truncated with this 'folding' technique enjoy the same properties as untruncated sensors provided the strain field is assumed to be sinusoidal. This is true regardless of whether the original weighting $f(x)$ is symmetric or not.

2.4.6 Summary

For a centered sensor whose center is placed at the point of symmetry in a cosinusoidal strain field, its transfer function is the same as the spatial Fourier transform, provided that $f(x)$ is symmetric; that is, if $f(x) = f(-x)$. In the limit as $x_0 \to 0$, the transfer function for a symmetric, centered sensor in a sinusoidal field is also equal

to the spatial Fourier transform. The rolloff rate of the sensor transfer function is governed by how smoothly $f(x)$ tapers to zero at the ends ($x = \pm l/2$). The more smoothly $f(x)$ goes to zero the more derivatives of $f(x)$ are zero at the ends, and the quicker the rolloff. Table 2.2 shows this behavior. Generally, as the weighting tapers more smoothly to zero, its length must necessarily increase in order to begin to roll off at the same spatial frequency. In addition, the transition band of the transfer function increases, and the side lobe structure is reduced in height.

For a noncentered sensor whose center is a distance x_0 from the point of symmetry in a sinusoidal strain field, similar properties hold. As the weighting is tapered more smoothly to zero, the rolloff rate increases in the same way as for a centered sensor. If $f(x)$ is not symmetric, the transfer function will not be bounded in amplitude. If $f(x)$ is symmetric, however, the transfer function is bounded in amplitude and becomes independent of x_0, and equal to the transfer function for a centered sensor.

When a sensor is truncated, its rolloff performance decreases drastically, in proportion to the fraction of the sensor weighting which is truncated. The transfer function is no longer bounded in amplitude because the truncated weighting is no longer symmetric. In addition, the rolloff rate decreases to $1/k$ no matter how smoothly the weighting tapers to zero at the ends because the value of the weighting at the structure boundary is nonzero in general.

A truncation technique that involves 'folding' the portion of the weighting that falls beyond the boundary was introduced to improve sensor rolloff performance in an effort to better satisfy Functional Requirements #2 and #9. In a sinusoidal strain field, the transfer function of untruncated sensors and sensors truncated with this technique are identical whether $f(x)$ is symmetric or not.

2.5 Transfer Functions for Exponential Strain

The mode shapes of structures are generally composed of sinusoidal and exponential-like functions. The properties of sensors in sinusoidal and cosinusoidal strain fields were analyzed above. In this section, the properties of sensors in an exponential strain fields are investigated.

The exponential strain fields are assumed to have the form e^{-kx}. The origin will refer to the point where the strain remains the same no matter what the spatial frequency is. For the strain field e^{-kx}, this point is simply $x = 0$, where the strain is unity for all k. It is important to note that k is not a spatial frequency as it is for the strain field $\sin(kx)$ because e^{-kx} is not oscillatory. Rather, k is a scale length that determines how quickly the strain field tapers toward zero for positive x. Since the strain field is not oscillatory, the properties of the transfer functions will not resemble those described for sinusoidal curvature. Further, since mode shapes are generally spatially oscillatory, it is of limited use to analyze the properties of the transfer function of sensors in a purely exponential strain field. The form of the exponential near-field terms in the mode shapes of structures should be considered when examining the following results.

2.5.1 Sensors Centered at Origin

The output of a sensor mounted on a structure, away from its boundaries, centered at the origin of an exponential strain field is investigated here. It is assumed that the strain field is of the form

$$\epsilon(x) = e^{-kx} \tag{2.91}$$

where k is the wave number. The output of a sensor with weighting function $f(x)$, centered at $x = 0$ is then given by

$$y(k) = \int_{-l/2}^{l/2} f(x)e^{-kx}\,dx \tag{2.92}$$

By repeatedly integrating this expression by parts, an infinite series solution for the output of a sensor mounted in the interior of a structure, away from its boundaries, at the origin of an exponential strain field can be written as

$$y(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}} \left[-f^{(i)}(l/2)e^{-kl/2} + f^{(i)}(-l/2)e^{kl/2}\right] \tag{2.93}$$

Since the strain at the center of the sensor is always unity, the transfer function for such a sensor is

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}} \left[-f^{(i)}(l/2)e^{-kl/2} + f^{(i)}(-l/2)e^{kl/2}\right] \tag{2.94}$$

The derivation of (2.93) and (2.94) is shown in Section A.2 of Appendix A. It can be seen that the transfer function has the general form of (2.40), except that the modulating functions of the $f^{(i)}(\pm l/2)$ terms are always exponentials, rather than sinusoidal functions. Due to the definition of the strain field in 2.91, it is clear that the magnitude of the field increases with wave number k for all $x < 0$. The sensor will sense this increase and consequently the output function $y(k)$ increases exponentially. In addition, since the strain at the center of the sensor is unity for all k, the transfer function of the sensor also increases exponentially. As $f(x)$ is made smoother at the ends, the $f^{(i)}(\pm l/2)$ terms are zero for higher and higher values of i, and the first nonzero term will be attenuated by a smaller and smaller $1/k^{i+1}$ term. However, $e^{kl/2}/k^{i+1} \to \infty$ as $k \to \infty$ for any value of i because all derivatives of the numerator $e^{kl/2}$ increase with k, while the $(i+2)$-nd derivative of k^{i+1} is zero.

2.5.2 Sensors Not Centered at Origin

Now the output of a sensor mounted in the interior of a structure, away from its boundaries, centered at a distance x_0 from the origin of an exponential strain field is investigated. It is assumed that the strain field is of the form

$$\epsilon(x) = e^{-kx} \tag{2.95}$$

where k is the wave number. The output of a sensor with weighting function $f(x)$, centered at $x = x_0$ is then given by

$$y(k) = \int_{-l/2+x_0}^{l/2+x_0} f(x - x_0)e^{-kx}\, dx \tag{2.96}$$

By repeated integration by parts of this expression, an infinite series solution for the output of a sensor mounted in the interior of a structure, away from its boundaries, at a distance x_0 from the origin of an exponential strain field can be written as

$$y(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}} \left[-f^{(i)}(l/2)e^{-kl/2} + f^{(i)}(-l/2)e^{kl/2} \right] e^{-kx_0} \tag{2.97}$$

This expression is the same as (2.93), the output of a sensor centered at the origin, attenuated by the factor e^{-kx_0}. This attenuation arises from the inherent attenuation in the assumed exponential strain field. It is clear that when $x_0 < l/2$, the output of the sensor increases exponentially by the factor $e^{k(l/2-x_0)}$ because the exponential strain field is increasing by that factor over part of the sensitive area of the sensor. When $x_0 = l/2$, the sensor output rolls off due to the $1/k^{i+1}$ factors. Finally, when $x_0 > l/2$, the output rolls off due not only to $1/k^{i+1}$ factors, but by the exponential factor $e^{k(l/2-x_0)}$.

In order to obtain the transfer function for a sensor centered at a distance x_0 from

the origin of the strain field, (2.97) must be divided by e^{-kx_0}, the strain at the center of the sensor. The transfer function obtained is:

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}} \left[-f^{(i)}(l/2)e^{-kl/2} + f^{(i)}(-l/2)e^{kl/2} \right] \tag{2.98}$$

This is the same expression as (2.94), as shown in Section A.2 of Appendix A, indicating that the transfer function of a sensor in an exponential strain field, away from the boundaries of the structure is constant, no matter where the sensor is centered with respect to the origin of the strain field. Note that the sensor does not have to be symmetric for this to be true, unlike the case for a sinusoidal strain field.

2.5.3 Sensors Simply Truncated at Structure Boundaries

Now the output of a sensor mounted near a boundary of a structure, at a distance x_0 from the origin of an exponential strain field is investigated. This case is extremely important as it simulates the behavior of a sensor truncated at the boundary of a structure with an evanescent exponential strain field. Examples of such strain fields include the mode shapes of clamped-free and free-free beams. For this problem, it is again assumed that the strain field is of the form

$$\epsilon(x) = e^{-kx} \tag{2.99}$$

where k is a parameter that depends on mode number. It is assumed that the boundary of a structure is at $x = 0$. The output of a sensor with weighting function $f(x)$, centered at $x = x_0$ is then given by

$$y(k) = \int_0^{l/2+x_0} f(x - x_0)e^{-kx}\, dx \tag{2.100}$$

Again, an infinite series solution for the output of a sensor mounted near a boundary of a structure, at a distance x_0 from the point of symmetry in an exponential strain field can be found by repeatedly integrating this expression by parts to obtain:

$$y(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}} \left[-f^{(i)}(l/2)e^{-k(l/2+x_0)} + f^{(i)}(-x_0) \right] \tag{2.101}$$

It is assumed that the truncated sensor is centered somewhere in the interior of the structure. This guarantees that $0 < x_0 < l/2$. Thus $(l/2 + x_0)$ is positive and the factor $e^{-k(l/2+x_0)}$ is an attenuating one. The first term in (2.101) will therefore roll off quicker than the $f^{(i)}(-x_0)$ term, which is the value of the spatial weighting at the boundary of the structure. It was stated that for the case of truncated sensors in sinusoidal strain fields that this value is, in general, nonzero. This therefore limits the rolloff rate of the sensor output to $1/k$. This behavior is very similar to the case of a sinusoidally varying strain field.

The transfer function for a sensor truncated at the boundary of the structure, in an exponential strain field is simply (2.101) divided by the strain at the sensor's center:

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}} \left[-f^{(i)}(l/2)e^{-kl/2} + f^{(i)}(-x_0)e^{kx_0} \right] \tag{2.102}$$

This expression suggests that while the first term still decreases in magnitude as k is increased, for any value of i, the second term instead increases as e^{kx_0}/k^{i+1}. This is simply because the strain at the center of the sensor (e^{-kx_0}) decreases more rapidly than the strain at the boundary of the structure, which is unity. The derivations of (2.101) and (2.102) are shown in Section A.2 of Appendix A.

2.5.4 Sensors Folded at Structure Boundaries

As for sinusoidally varying strain fields, and effort is made here to improve the truncation technique for sensors near a structural boundary, in an exponential strain

field. Figure 2.22 shows the procedure used to fold and flip the truncated part of a sensor. In this section, the effect of performing this operation when the strain field is exponential rather than sinusoidal is investigated. The output of the modified sensor is given by

$$y(k) = \int_0^{l/2-x_0} \Big[f(x-x_0) - f(-x-x_0)\Big]\, e^{-kx}\, dx + \int_{l/2-x_0}^{l/2+x_0} f(x-x_0)\, e^{-kx}\, dx \quad (2.103)$$

The weighting $f(-x-x_0)$ is again the original weighting $f(x-x_0)$ reflected about the boundary of the structure at $x = 0$. The length of the portion of the sensor that extends beyond the boundary of the sensor is $(l/2 - x_0)$. The above equation can be rewritten

$$y(k) = \int_0^{l/2+x_0} f(x-x_0)\, e^{-kx}\, dx - \int_0^{l/2-x_0} f(-x-x_0)\, e^{-kx}\, dx \quad (2.104)$$

$$y(k) = G(k) - H(k) \quad (2.105)$$

The first integral $G(k)$ in (2.104) was dealt with in the previous section and the result is shown in (2.101). The evaluation of the second integral $H(k)$ is shown in Section A.2 of Appendix A. By repeated integration by parts, $H(k)$ can be written as an infinite series in the following manner:

$$H(k) = \sum_{i=0}^{\infty} \frac{(-1)^i}{k^{i+1}} \Big[f^{(2i)}(-x_0) - f^{(2i)}(-l/2)e^{-k(l/2-x_0)}\Big] \quad (2.106)$$

Now $G(k)$ and $H(k)$ are rewritten in slightly different forms:

$$G(k) = \sum_{i=0}^{\infty} \Bigg\{ \frac{1}{k^{(2i+1)}} \Big[-f^{(2i)}(l/2)e^{-k(l/2+x_0)} + f^{(2i)}(-x_0)\Big] + \frac{1}{k^{(2i+2)}} \Big[-f^{(2i+1)}(l/2)e^{-k(l/2+x_0)} + f^{(2i+1)}(-x_0)\Big] \Bigg\} \quad (2.107)$$

$$H(k) = \sum_{i=0}^{\infty} \left\{ \frac{1}{k^{(2i+1)}} \left[f^{(2i)}(-x_0) - f^{(2i)}(-l/2)e^{-k(l/2-x_0)} \right] + \frac{1}{k^{(2i+2)}} \left[-f^{(2i+1)}(-x_0) + f^{(2i+1)}(-l/2)e^{-k(l/2-x_0)} \right] \right\} \quad (2.108)$$

By subtracting the expression for $H(k)$ from $G(k)$ the output of the modified sensor as a function of frequency can be found:

$$y(k) = \sum_{i=0}^{\infty} \left\{ \frac{1}{k^{(2i+1)}} \left[-f^{(2i)}(l/2)e^{-k(l/2+x_0)} + f^{(2i)}(-l/2)e^{-k(l/2-x_0)} \right] + \frac{1}{k^{(2i+2)}} \left[-f^{(2i+1)}(l/2)e^{-k(l/2+x_0)} - f^{(2i+1)}(-l/2)e^{-k(l/2-x_0)} + 2f^{(2i+1)}(-x_0) \right] \right\} \quad (2.109)$$

Noting that we have $l/2 > x_0 > 0$, we see that the sensor output (2.109) must roll off as k is increased, because all the exponential terms are attenuating ones. By dividing this expression by e^{-kx_0}, the transfer function of the sensor is found to be

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{1}{k^{(2i+1)}} \left[-f^{(2i)}(l/2)e^{-kl/2} + f^{(2i)}(-l/2)e^{-k(l/2-2x_0)} \right] + \frac{1}{k^{(2i+2)}} \left[-f^{(2i+1)}(l/2)e^{-kl/2} - f^{(2i+1)}(-l/2)e^{-k(l/2-2x_0)} + 2f^{(2i+1)}(-x_0)e^{kx_0} \right] \right\} \quad (2.110)$$

As with the simple truncation scheme, in the second half of the summation, we are left with the term $2f^{(2i+1)}(-x_0)e^{kx_0}/k^{(2i+2)}$, which increases exponentially as k is increased. However, this term depends on the value of the first spatial derivative of the sensor weighting at the boundary of the structure, not the value of the weighting itself. In many ways, this behavior is similar to the corresponding transfer function for a sinusoidal strain field. The derivation of (2.109) and (2.110) is shown in Section A.2 of Appendix A.

2.5.5 Summary

For a centered sensor, the rolloff rate of the output and transfer function is increased as the ends of the sensor are tapered more smoothly to zero because the weighting and its derivatives are made zero at the ends. However, note that both the output and transfer function actually increase exponentially as a function of k due to the $e^{kl/2}$ factor in the first term. This is due to the fact that the assumed strain field e^{-kx} is increasing in magnitude for negative x.

For a noncentered sensor, the same rolloff behavior as for the centered sensor holds. However, note that the output $y(k)$ will decrease when $x_0 > l/2$, that is, when the entire sensor is placed on the positive x axis. This is because the strain field is everywhere decreasing with k for $x > l/2$. The transfer function will still increase exponentially, because the strain at the center of the sensor decreases much more rapidly than the strain at the end of the sensor closest to the structure boundary.

For a truncated sensor, as for a sinusoidal strain field, the rolloff rate of the output $y(k)$ is reduced to at most $1/k$ no matter how large x_0 is because the value of the sensor weighting at the structure boundary is nonzero. For the transfer function, the exponential increase with k is made mode severe as x_0 is increased.

For a folded sensor, the rolloff rate of the sensor output is now only limited to $1/k^2$ because the term $f(-x_0)$ is made zero. Although the transfer function still increases exponentially, it does so more slowly than for a simply truncated sensor.

Chapter 3

Numerical Integration Schemes

In this chapter a selected set of numerical integration schemes are described. The schemes are used to estimate the global shape of a structure by spatially integrating the measurements made by an array of spatially averaging strain sensors mounted along the structure. Details of the derivation of the expressions presented are given in Appendix C.

3.1 Introduction

An individual sensor provides a single spatially filtered measurement of strain for a certain area of the structure. In order to estimate the global shape of the entire structure, an array of sensors must be used, and their individual outputs spatially integrated. Numerical integration schemes allow accurate estimation of deflection by spatially integrating strain measurements made by an array of sensors. To choose a difficult example, the problem of estimating the slope (first integral of curvature) and displacement (second integral of curvature) of a beam, using an array of spatially averaging strain sensors, will be investigated. In addition to the weighting discussed in the previous section, the placement and spacing of the sensors along the beam are now important and are typically determined by the integration scheme to be used.

Table 3.1: Important properties of selected numerical integration schemes.

Integration Scheme	Specific abscissas?	End points required?	Exactness	Error
Midpoint	No	No	Linear functions	$\mathcal{O}(1/n^2)$
Trapezoidal	No	Yes	Linear functions	$\frac{(b-a)^3}{24n^2} f''(\xi)$ $(a<\xi<b)$
Simpson's	No	Yes	Cubic functions	$-\frac{(b-a)^5}{180n^4} f^{(4)}(\xi)$ $(a<\xi<b)$
Cubic Spline	No	Yes	Constant functions	$\mathcal{O}(1/n^3)$
B-Spline	No	Yes	Constant functions	$\mathcal{O}(1/n^2)$
Gauss	Yes	No	Polynomials of degree $2n-1$	$\frac{2^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3} f^{(2n)}(\xi)$ $(-1<\xi<1)$
Radau	Yes	One end	Polynomials of degree $2n-1$	$\frac{2^{2n-1}n[(n-1)!]^4}{[(2n-1)!]^3} f^{(2n-1)}(\xi)$ $(-1<\xi<1)$
Chebyshev	No	Yes	Polynomials of degree $2n-1$	
Structural Shape Functions	No	No	Any combination of the set of shape functions	
Least Squares Structural Shape Functions	No	No	Any combination of the set of shape functions	

A variety of numerical integration methods exist which integrate a function known only at a discrete set of points. A restricted set of these was chosen for the purposes of investigating the performance of strain sensors as estimators of global shape, as shown in Table 3.1. As a test case in the next chapter, the various schemes will be applied to the task of estimating the tip slope and deflection of a cantilevered beam under transverse loading.

The second column of Table 3.1 indicates whether the integration scheme requires the sensors to be placed at specific locations. It is seen that the optimal Gauss and

Radau rules are the only ones which have stringent sensor placement requirements. Note that Simpson's rule is customarily defined for a set of equally spaced abscissas due to the simplicity of the resulting expression for the integral. However, it is possible to formulate the rule to accept abscissas which are arbitrarily placed. The third column of the table shows whether the integration scheme requires that the function to be integrated be known at the edges of the integration interval. This is often a difficult requirement for practical implementation. In the case of a beam-like structure, this requires knowledge of the curvature at the root and at the tip of the beam. The fourth column indicates the highest order polynomial for which the integration scheme will yield an exact integral. The parameter n is the number of points at which the function to be integrated is known. For this work, n is equal to the number of sensors. It is clear that the Gauss and Radau rules are the ones which yield the most accurate integrals of polynomials using few sensors. The fifth column shows the order or form of the remainder or error term that must be considered when using the scheme. The function $f^{(i)}(x)$ is the i-th derivative of the function to be integrated, and the point ξ is some point within the integration interval. Since this point is unknown, only an upper bound to the magnitude of the error can be found, by evaluating $\max_{a<\xi<b} f^{(i)}(\xi)$.

Examining the options in Table 3.1, it is obvious that there is no perfect solution. Only the simplest, least accurate Midpoint rule, and the rules fitting structural shape functions to sensor measurements do not require measurements at the endpoints and do not require specific abscissas. In general, as the degree to which the location of sensors is constrained (a practical problem), the degree of accuracy increases (a theoretical advantage). Therefore the trade between practicality and accuracy must be assessed and the absolute, rather than relative error examined.

The integration schemes presented in Table 3.1 will now be briefly described.

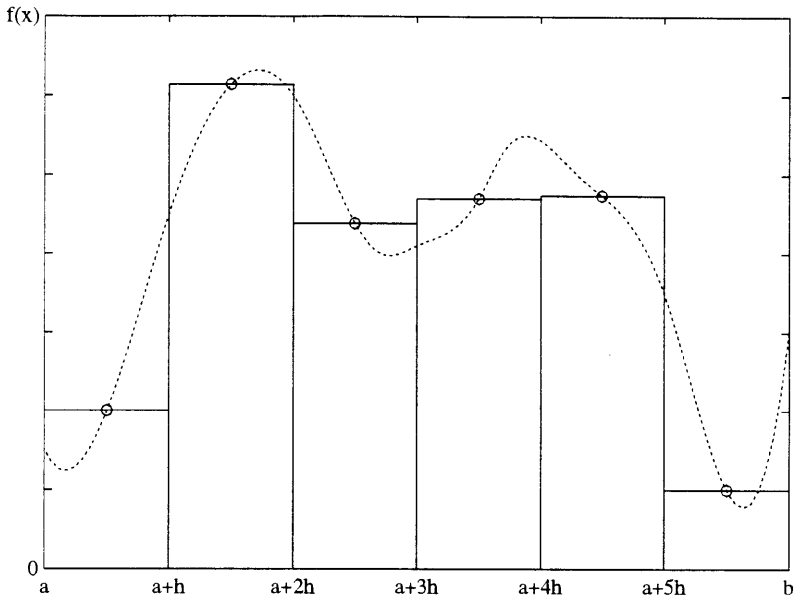

Figure 3.1: The area the midpoint rule calculates. Circles: points at which the argument function $f(x)$ is known; solid rectangles: areas that are summed to produce an estimate of the integral; dashed line: the argument function $f(x)$.

3.2 Integration Schemes

3.2.1 Midpoint Rule

The midpoint rule is a primitive integration rule given by a Riemann sum of the form

$$M_n(f) = h \sum_{k=0}^{n-1} f\left(a + \left(k + \frac{1}{2}\right) h\right) \tag{3.1}$$

where $h = (b-a)/n$. The interval of integration is $[a, b]$, and $f(x)$ is the function to be integrated [16]. The area the midpoint rule calculates is shown in Figure 3.1. The dashed line shows the argument function $f(x)$. The circles show the values $f(x_i)$ of the argument function that are known. The width of each rectangle is h, and their combined area is the area the midpoint rule calculates.

3.2.2 Trapezoidal Rule

The trapezoidal rule is also a primitive rule given by a Riemann sum of the form

$$T_n(f) = h\left[\frac{f(a)}{2} + f(a+h) + f(a+2h) + \cdots + f(a+(n-1)h) + \frac{f(b)}{2}\right] \quad (3.2)$$

where $h = (b-a)/n$. A derivation of the trapezoidal rule is shown in Section C.2 of Appendix C. The trapezoidal rule calculates the area shown in Figure 3.2. The dashed line shows the argument function $f(x)$. The circles show the values $f(x_i)$ of the argument function that are known. The width of each trapezoid is h, and their combined area is the area the trapezoidal rule calculates. It can be shown that the error (or remainder term) for the compound trapezoidal rule is

$$E_n = \frac{(b-a)^3}{24n^2} f''(\xi), \qquad a < \xi < b \quad (3.3)$$

when $f''(x)$ is bounded [16].

3.2.3 Simpson's Rule

Simpson's rule was first given by Cavalieri [11], and only later by James Gregory [26] and Thomas Simpson [51, 59]. The formula for Simpson's rule can be derived a number of different ways, and one such derivation is given in Section C.3 of Appendix C. The composite Simpson's rule calculates the area shown in Figure 3.3. The dashed line is the argument function $f(x)$. The circles show the values $f(x_i)$ of the argument function that are known. The solid parabolic segments are fitted to three successive points. The area under these segments is integrated analytically and calculated.

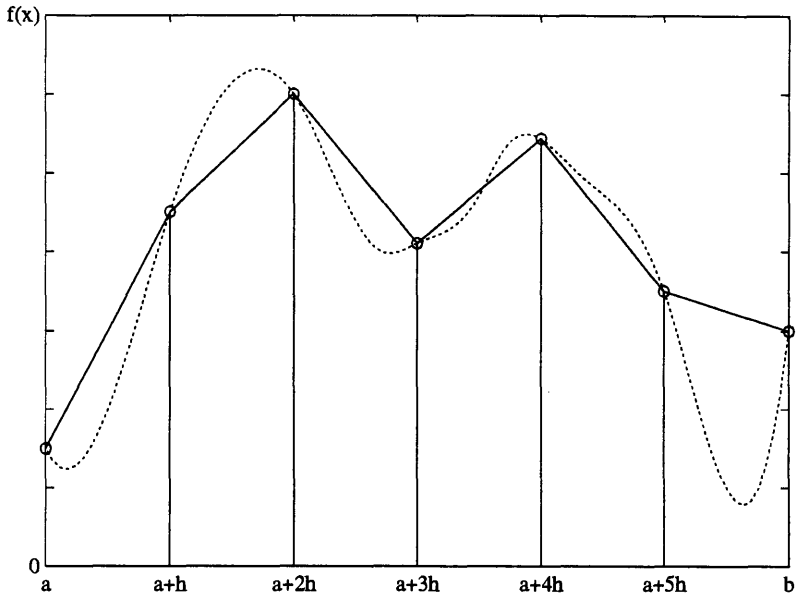

Figure 3.2: The area the trapezoidal rule calculates. Circles: points at which the argument function $f(x)$ is known; solid trapezoids: areas that are summed to produce an estimate of the integral; dashed line: the argument function $f(x)$.

The result is given by the composite form of Simpson's rule:

$$\int_a^b f(x)\,dx \approx \int_a^{a+2h} f(x)\,dx + \int_{a+2h}^{a+4h} f(x)\,dx + \int_{a+4h}^{b} f(x)\,dx \tag{3.4}$$

$$= \frac{2h}{6}\Big[f(a) + 4\,f(a+h) + f(a+2h)\Big]$$
$$+ \frac{2h}{6}\Big[f(a+2h) + 4\,f(a+3h) + f(a+4h)\Big]$$
$$+ \frac{2h}{6}\Big[f(a+4h) + 4\,f(a+5h) + f(b)\Big] \tag{3.5}$$

The error in using Simpson's rule is shown in Table 3.1, and shows that the integral converges as $1/n^4$.

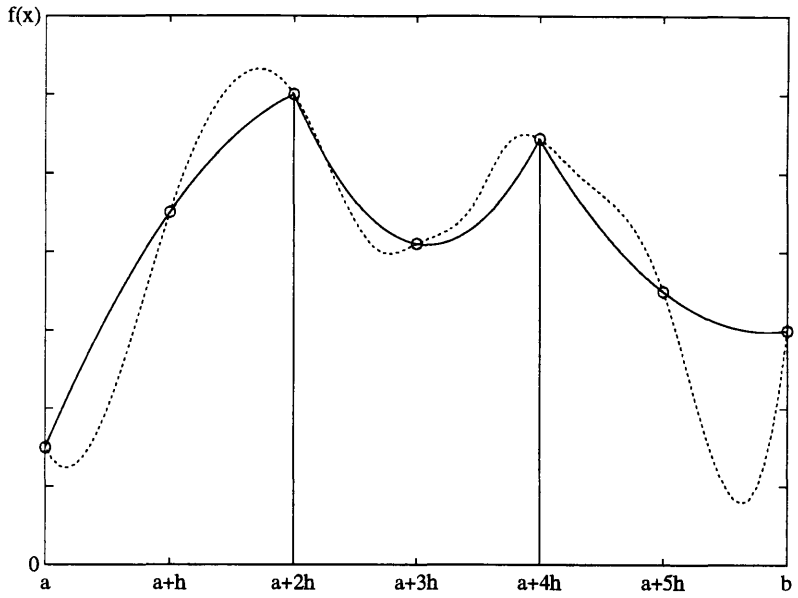

Figure 3.3: The area Simpson's rule calculates. Circles: points at which the argument function $f(x)$ is known; solid line: segments of parabolas interpolating the given points $f(x_i)$; dashed line: the argument function $f(x)$.

3.2.4 Cubic Interpolating Splines

A cubic interpolating spline can be passed through the discrete set of strain measurements and integrated. A cubic interpolating spline is a piecewise cubic parametric polynomial that passes through an arbitrary set of points. It does so while maintaining derivative and curvature continuity at the sample points while minimizing the overall integral of the curvature squared, $\int_a^b |w''(x)|^2\, dx$. This is precisely what a thin beam made to pass through a series of points does. A beam will change its shape so as to minimize its strain energy at all times, and this energy is proportional to the integral of the square of the curvature along the length of the beam. The analytical integral of a cubic interpolating spline has the form of that for the trapezoidal rule with an added third order correction. The convergence rate of the spline algorithm is therefore $1/n^3$.

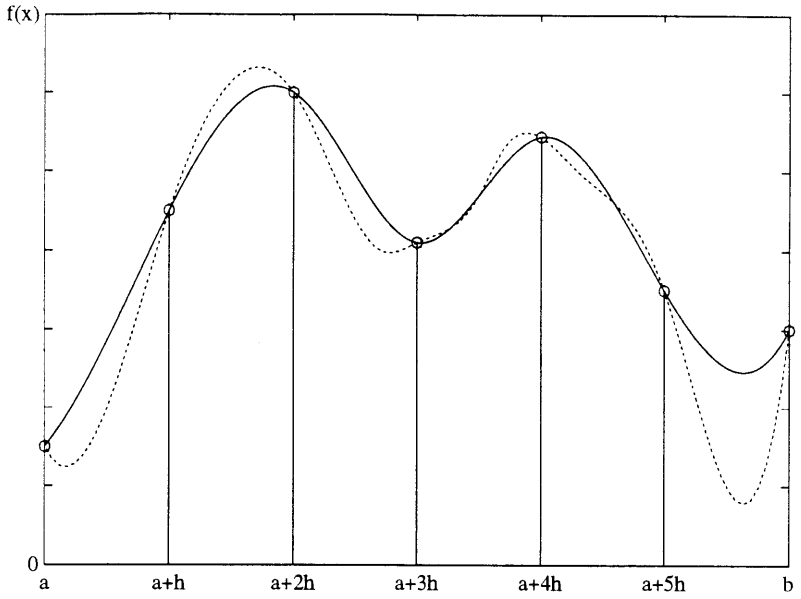

Figure 3.4: The area the Spline rule calculates. Circles: points at which the argument function $f(x)$ is known; solid line: the cubic spline interpolating the data points; dashed line: the argument function $f(x)$.

Figure 3.4 shows the area the spline rule calculates. The dashed line shows the argument function $f(x)$. The circles show the values $f(x_i)$ if the argument function that are known. The solid line shows the cubic spline that interpolates the data points. It is the area under this curve that is calculated by the spline rule.

The spline curve is composed of $n-1$ polynomial segments, where n is the number of points the spline passes through. The i-th polynomial segment stretches between (x_i, y_i) and (x_{i+1}, y_{i+1}) (two successive points). The curve is forced to pass through the control points and have slope and curvature continuity at the control points. The curvature is assumed to be zero at the ends. The x- and y-coordinate values of the curve are defined separately as parametric equations of an independent parameter t. The curve of the i-th segment is defined as $(X_i(t), Y_i(t))$. The x-coordinates of the curve $X_i(t)$ can be determined solely from the x-coordinates of the control points x_i and the y-coordinates of the curve $Y_i(t)$ can be determined solely from the

y-coordinates of the control points y_i. $X_i(t)$ and $Y_i(t)$ are defined as

$$X_i(t) = a_{1i}t^3 + b_{1i}t^2 + c_{1i}t + d_{1i}, \qquad (0 < t < 1) \tag{3.6}$$

$$Y_i(t) = a_{2i}t^3 + b_{2i}t^2 + c_{2i}t + d_{2i}, \qquad (0 < t < 1) \tag{3.7}$$

The process of finding the constants a_{1i}, a_{2i}, b_{1i}, etc., is shown in Section C.4 of Appendix C. Once the full curve is known, it can be integrated in order to find the area under it. This is more easily done numerically because $Y_i(t)$ is not a linear function of $X_i(t)$.

3.2.5 B-Splines

Cubic interpolating splines and other free-form curves defined using a discrete number of points have long been in use in the ship building industry [23]. However, a mathematical analysis of such curves was first attempted only a few decades ago by such mathematicians as Schoenberg, Bézier, and De Casteljau. A result of this research was the Bézier curve, defined simply by a control polygon consisting of a number of discrete control points. These curves use Bernstein polynomials and are of degree $n - 1$ if n points are specified, and start and end at the first and last points, respectively. Pierre Bézier used these curves at Renault to design car bodies. The problem with cubic interpolating splines, and Bézier curves, is that the number of computations required to calculate such a curve goes up faster than linearly with the number of control points. In addition, moving any control point changes the shape of the entire curve. B-splines, in contrast, are composed of short curve segments that can be of low order. In addition, moving a control point only affects part of the curve localized around the moved point.

B-splines, *basis splines*, or *basic splines* are composed of polynomial segments, but unlike cubic interpolating splines, they are not interpolatory and do not pass through

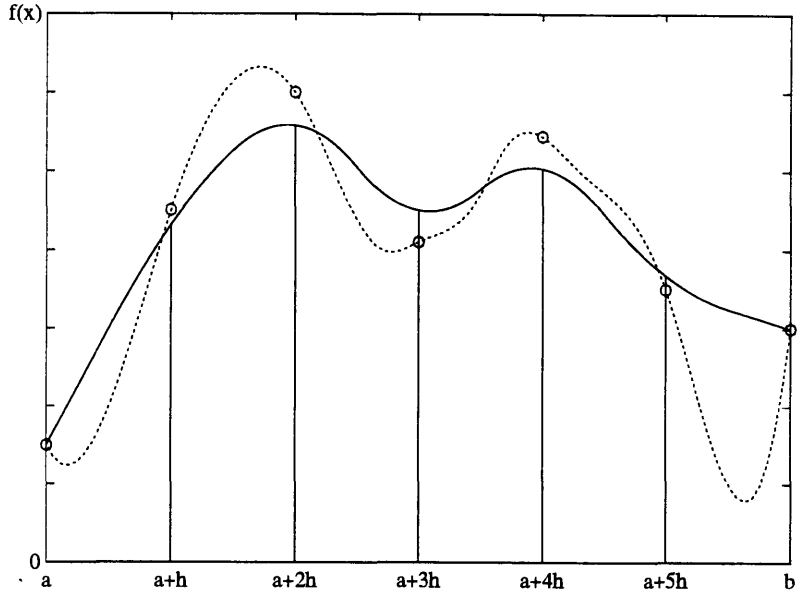

Figure 3.5: The area the B-Spline rule calculates. Circles: points at which the argument function $f(x)$ is known; solid line: the second-order B-spline resulting from the given data points $f(x_i)$; dashed line: the argument function $f(x)$.

the specified control points [7]. These control points become the vertices of the control polygon, which defines the shape of the B-spline. B-splines can be of various orders, from 1 to the one less than the number of vertices in the control polygon. B-splines of first order are linear functions which interpolate the data in a trapezoidal fashion. Second order B-splines are quadratic functions which yield a curve which is tangent to the midpoint of the line connecting two successive control points. Finally, B-splines of order $n-1$, where n is the number of vertices of the control polygon, are Bézier curves.

For the purposes of this work, only second order B-splines will be considered. Figure 3.5 gives an example of the area a second order B-spline would calculate. The dashed line is the argument function $f(x)$. The circles show the values $f(x_i)$ of the argument function that are known. The solid line shows the second order B-spline that only interpolates the first and last points. It is the area under this curve that

the B-spline rule calculates.

It can be seen from Figure 3.5 that the curve becomes tangent at the midpoint of the lines connecting the vertices of the control polygon. It is at these points that successive polynomials connect. These polynomials are parametrically expressed as a function of an independent variable t as follows:

$$X_i(t) = A_{1i}t^2 + B_{1i}t + C_{1i} \tag{3.8}$$

$$Y_i(t) = A_{2i}t^2 + B_{2i}t + C_{2i} \tag{3.9}$$

Typically t is assumed to be in the range $[0, 1]$ for each polynomial in order to simplify the mathematics, but any range can be used. The procedure for solving for the constants A_{1i}, A_{2i}, B_{1i}, etc., is outlined in Section C.5 of Appendix C. As with the cubic spline, the B-spline is integrated numerically to yield an approximation to the integral of the argument function $f(x)$.

3.2.6 Gauss Quadrature

The Gauss-Legendre quadrature rule uses optimal abscissa distributions to yield accurate approximations to integrals using about half as many sample points as simple Riemann sums such as the trapezoidal rule [30]. In the case of the Gauss rule, the abscissas at which the argument function is to be sampled are the zeros of the Legendre polynomials [24], and are therefore irrational numbers. Usually the weightings on the ordinates are also irrational. The Gauss quadrature rule takes the form

$$\int_{-1}^{1} f(x)\,dx = \sum_{i=1}^{n} w_i\, f(x_i) + E_n \tag{3.10}$$

The sampling points x_i are the n zeros of the Legendre polynomial of degree n, which all lie in the interval $[-1, 1]$. The argument function and the integration

interval $[a, b]$ can be linearly transformed to the interval $[-1, 1]$ by the change of variables

$$u = 2\left(\frac{x-a}{b-a}\right) - 1 \tag{3.11}$$

The Legendre polynomials have the form

$$P_n(x) = \frac{1}{2^n n!}\left[\frac{d^n}{dx^n}(x^2-1)^n\right] \tag{3.12}$$

The w_i, the weightings on the ordinates, are computed with any of the following formulae [32, 30, 35]:

$$w_i = \frac{2}{(n+1)P_{n+1}(x_i)P'_n(x_i)} = \frac{2}{nP_{n-1}(x_i)P'_n(x_i)} = \frac{2}{(1-x_i^2)[P'_n(x_i)]^2} \tag{3.13}$$

where $P'_n(x) = \frac{d}{dx}P_n(x)$. For the purposes of calculating values of the Legendre polynomials, it is useful to use the recursion formula [30]:

$$P_0(x) = 1 \tag{3.14}$$

$$P_1(x) = x \tag{3.15}$$

$$P_2(x) = \frac{1}{2}(3x^1 - 1) \tag{3.16}$$

$$P_{n+1}(x) = \frac{2n+1}{n+1}\, x\, P_n(x) - \frac{m}{m+1}P_{n-1}(x) \tag{3.17}$$

The error incurred by the use of the Gauss rule is

$$E_n = \int_{-1}^{1} f(x)\, dx - \sum_{i=1}^{n} w_i\, f(x_i) \tag{3.18}$$

and is given by [16, 30, 18, 35]:

$$E_n = \frac{2^{2n+1}\,(n!)^4}{(2n+1)\,[(2n)!]^3}\, f^{(2n)}(\xi), \qquad -1 < \xi < 1 \tag{3.19}$$

where $f^{(2n)}(x)$ is the $2n$-th derivative of $f(x)$ with respect to x, $\frac{d^{2n}}{dx^{2n}}f(x)$.

To integrate curvature to yield slope it is necessary to integrate twice. The numerical scheme must allow integration of the curvature over part of the interval in order to obtain a function for the slope. This slope function can then be integrated again to yield deflection. Unfortunately, the Gauss quadrature rule gives the integral for the entire region. Therefore only tip slope can be calculated using this rule. No deflection predictions can be made.

3.2.7 Radau Quadrature

Radau quadrature is a case of Gauss quadrature for which one of the abscissas can be placed anywhere in the integration interval $[-1, 1]$. For this work, the fixed abscissa will be at $x = -1$. Radau quadrature was included in this study because the curvature at the tip of a cantilevered beam with no moment loading can be assumed to be zero. Therefore the curvature measurement at the beam tip can be thought of as being at a preassigned abscissa, since the curvature is known there. The equation for the integral is modified and is given by [30]:

$$\int_{-1}^{1} f(x)\,dx = \frac{2}{(n+1)^2}f(-1) + \sum_{i=1}^{n} w_i\, f(x_i) + E_n \tag{3.20}$$

where n is the number of abscissas not preassigned. Since the abscissa at $x = -1$ is preassigned, there remain $2n-1$ variables that can be chosen freely. Thus, for (3.20) to yield an exact estimate of the integral, $f(x)$ must be a polynomial of degree $2n-2$ or less. The n abscissas located in the interior of the interval $[-1, 1]$ are now the zeros of the polynomials [30]:

$$\begin{aligned} \phi_n(x) &= P_n(x) + \frac{x-1}{n+1}P_n'(x) \\ &= \frac{P_n(x) + P_{n+1}(x)}{1+x} \end{aligned} \tag{3.21}$$

where the $P_i(x)$ are again the Legendre polynomials. The n weights w_i are given by the expression [30]:

$$w_i = \frac{1}{(n+1)^2} \frac{1-x_i}{[P_n(x_i)]^2} = \frac{1}{1-x_i} \frac{1}{[P_n'(x_i)]^2} \tag{3.22}$$

The error resulting from the use of the Radau quadrature scheme is given by [30]:

$$E_n = \frac{2^{2n+1}[n!]^4}{[(2n+1)!]^3} f^{(2n+1)}(\xi), \qquad -1 < \xi < 1 \tag{3.23}$$

As with Gauss Quadrature, Radau Quadrature does not yield a running integral of the numerical data being integrated. Rather, it gives the integral over the entire interval in one step, and therefore can only be used to find the slope of a beam given curvature measurements along it, and not the deflection.

3.2.8 Interpolation by Chebyshev Polynomials

The Chebyshev integration scheme consists of interpolating n data points by a sum of the first n Chebyshev polynomials. Figure 3.6 shows the area calculated by the Chebyshev scheme. The dashed line is the argument function $f(x)$. The circles are the points $f(x_i)$ at which the argument function is known. The solid line shows the sum of the seven (in this case) Chebyshev polynomials that interpolate the 7 given points. It is the area under this curve that the Chebyshev scheme computes. The Chebyshev polynomials have the form

$$T_n(x) = \cos(n \cos^{-1}(x)) \tag{3.24}$$

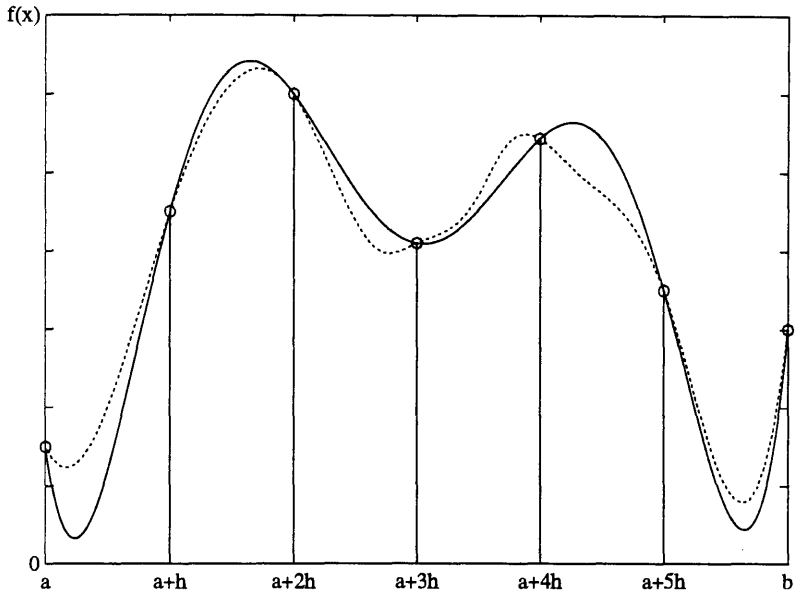

Figure 3.6: The area the Chebyshev rule calculates. Circles: points at which the argument function $f(x)$ is known; solid line: the sum of Chebyshev polynomials that interpolates the given points $f(x_i)$; dashed line: the argument function $f(x)$.

The first five Chebyshev polynomials are

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1 \end{aligned} \tag{3.25}$$

For the n given points at which $f(x)$ is known, a set of n simultaneous equations can be written:

$$f(x_i) = \sum_{j=1}^{n} p_j \, T_{j-1}(x_i), \qquad i = 1, 2, \ldots, n \tag{3.26}$$

It is the n weightings p_j we wish to find. The following matrix equation is obtained from (3.26):

$$\mathbf{f} = \mathbf{T}\mathbf{p} \tag{3.27}$$

where $\mathbf{T}_{ij} = T_{j-1}(x_i)$. The solution is immediate when all the x_i are distinct, and is given simply by

$$\mathbf{p} = \mathbf{T}^{-1}\mathbf{f} \tag{3.28}$$

Once the weightings p_i have been solved the following approximation for the function $f(x)$ is formed:

$$f(x) \approx \bar{f}(x) = \sum_{i=1}^{n} p_i \, T_i(x) \tag{3.29}$$

The approximate argument function $\bar{f}(x)$ is then integrated analytically over the interval $-1 < x < 1$. For a known set of abscissas x_i, the tip slope and deflection can then be expressed simply as a weighted sum of the n discrete argument function values $f_i = f(x_i)$.

3.3 Integration Rules as Global Shape Functions

Thus far in this chapter, eight integration schemes have been presented, which can be categorized as those which integrate by performing spatial interpolation over part of the domain (midpoint, trapezoidal, Simpson's and B-spline rules), and those which explicitly or implicitly fit polynomials to all the data in the domain, and then are integrated (spline, Gauss, Radau and Chebyshev rules). The latter category can be thought of as global shape functions, which have been fitted to the data. In this section, it will be shown by example that the rules which interpolate over part of the domain can also be interpreted as rules that fit global (but discontinuous) shape functions to the data.

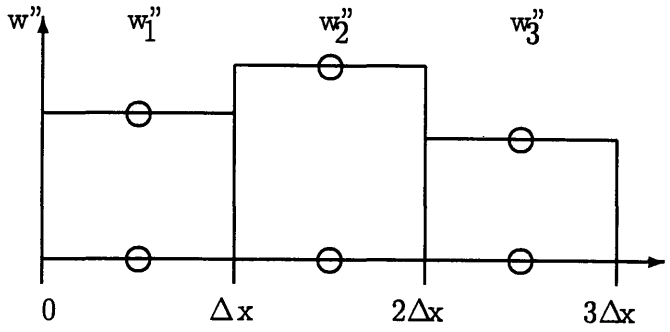

Figure 3.7: The midpoint rule applied to a beam with three sensors.

3.3.1 Midpoint Rule

Here it is shown how the midpoint rule can be represented by a global shape function fitting procedure. A global, although not spatially continuous shape function is formulated for each measurement point available in the integration interval. These shape functions are then fitted to the sensor measurements, and then integrated analytically.

It is assumed that there is an array of strain sensors distributed along the length of a structure. In the example presented here, it will be assumed that there are three strain sensors positioned at the center of three sections of length Δx, as shown in Figure 3.7. The three sensors measure the strains w_1'', w_2'' and w_3''. The midpoint rule assumes that the measured strains are constant over the entire region of length Δx centered around each sensor, then the slope of the structure at the tip can be found as:

$$w'_{\text{tip}} = \Delta x[w_1'' + w_2'' + w_3''] + w'_0 \tag{3.30}$$

where w'_o is the slope of the structure at its root.

Global shape functions can also be used to obtain this result. The shape functions are uniform over the interval Δx and zero elsewhere. In the case of obtaining a slope

estimate from curvature measurements, they are:

$$\phi_1''(x) = \begin{cases} \frac{1}{\Delta x^2} & 0 < x < \Delta x \\ 0 & \Delta x < x < 3\Delta x \end{cases} \tag{3.31}$$

$$\phi_2''(x) = \begin{cases} 0 & 0 < x < \Delta x \\ \frac{1}{\Delta x^2} & \Delta x < x < 2\Delta x \\ 0 & 2\Delta x < x < 3\Delta x \end{cases} \tag{3.32}$$

$$\phi_3''(x) = \begin{cases} 0 & 0 < x < 2\Delta x \\ \frac{1}{\Delta x^2} & 2\Delta x < x < 3\Delta x \end{cases} \tag{3.33}$$

These are integrated analytically to yield:

$$\phi_1'(x) = \begin{cases} \frac{x}{\Delta x^2} & 0 < x < \Delta x \\ \frac{1}{\Delta x} & \Delta x < x < 3\Delta x \end{cases} \tag{3.34}$$

$$\phi_2'(x) = \begin{cases} 0 & 0 < x < \Delta x \\ \frac{1}{\Delta x^2}(x - \Delta x) & \Delta x < x < 2\Delta x \\ \frac{1}{\Delta x} & 2\Delta x < x < 3\Delta x \end{cases} \tag{3.35}$$

$$\phi_3'(x) = \begin{cases} 0 & 0 < x < 2\Delta x \\ \frac{1}{\Delta x^2}(x - 2\Delta x) & 2\Delta x < x < 3\Delta x \end{cases} \tag{3.36}$$

The global shape functions and their integrals are shown in Figure 3.8. The measured strains reported by the sensors can be expressed as a linear combination of the global shape functions by using three generalized coordinates p_i:

$$\begin{bmatrix} w_1'' \\ w_2'' \\ w_3'' \end{bmatrix} = \begin{bmatrix} \phi_1''\left(\frac{\Delta x}{2}\right) & \phi_2''\left(\frac{\Delta x}{2}\right) & \phi_3''\left(\frac{\Delta x}{2}\right) \\ \phi_1''\left(\frac{3\Delta x}{2}\right) & \phi_2''\left(\frac{3\Delta x}{2}\right) & \phi_3''\left(\frac{3\Delta x}{2}\right) \\ \phi_1''\left(\frac{5\Delta x}{2}\right) & \phi_2''\left(\frac{5\Delta x}{2}\right) & \phi_3''\left(\frac{5\Delta x}{2}\right) \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{3.37}$$

which is of the same form as (3.27).

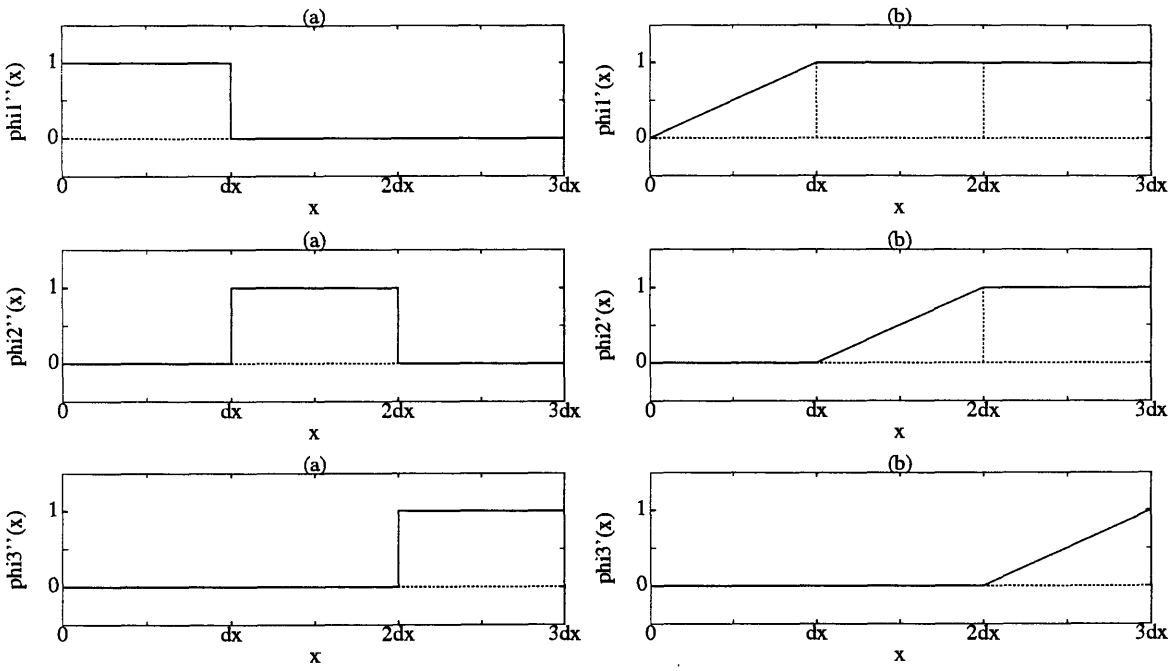

Figure 3.8: The global shape functions of the midpoint rule: (a) $\phi_i''(x)$; (b) $\phi_i'(x)$.

The generalized coordinates p_i can be found by matrix inversion:

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} \phi_1''\left(\frac{\Delta x}{2}\right) & \phi_2''\left(\frac{\Delta x}{2}\right) & \phi_3''\left(\frac{\Delta x}{2}\right) \\ \phi_1''\left(\frac{3\Delta x}{2}\right) & \phi_2''\left(\frac{3\Delta x}{2}\right) & \phi_3''\left(\frac{3\Delta x}{2}\right) \\ \phi_1''\left(\frac{5\Delta x}{2}\right) & \phi_2''\left(\frac{5\Delta x}{2}\right) & \phi_3''\left(\frac{5\Delta x}{2}\right) \end{bmatrix}^{-1} \begin{bmatrix} w_1'' \\ w_2'' \\ w_3'' \end{bmatrix} \tag{3.38}$$

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta x^2} & 0 & 0 \\ 0 & \frac{1}{\Delta x^2} & 0 \\ 0 & 0 & \frac{1}{\Delta x^2} \end{bmatrix}^{-1} \begin{bmatrix} w_1'' \\ w_2'' \\ w_3'' \end{bmatrix} \tag{3.39}$$

which is of the same form as (3.28).

The generalized coordinates can then be used to find the estimated slope of the structure at the sensor locations and at the tip of the structure:

$$\begin{bmatrix} w'_{\text{tip}} \end{bmatrix} = \begin{bmatrix} \phi'_1(3\Delta x) & \phi'_1(3\Delta x) & \phi'_1(3\Delta x) \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{3.40}$$

$$\begin{bmatrix} w'_{\text{tip}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta x} & \frac{1}{\Delta x} & \frac{1}{\Delta x} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{3.41}$$

Equation (3.39) can now be used to solve for the tip slope in terms of the measured curvatures:

$$\begin{bmatrix} w'_{\text{tip}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta x} & \frac{1}{\Delta x} & \frac{1}{\Delta x} \end{bmatrix} \begin{bmatrix} \frac{1}{\Delta x^2} & 0 & 0 \\ 0 & \frac{1}{\Delta x^2} & 0 \\ 0 & 0 & \frac{1}{\Delta x^2} \end{bmatrix}^{-1} \begin{bmatrix} w''_1 \\ w''_2 \\ w''_3 \end{bmatrix} \tag{3.42}$$

$$\begin{bmatrix} w'_{\text{tip}} \end{bmatrix} = \begin{bmatrix} \Delta x\, w''_1 + \Delta x\, w''_2 + \Delta x\, w''_3 \end{bmatrix} \tag{3.43}$$

After including the initial conditions at the root of the structure, we have:

$$w'_{\text{tip}} = \Delta x \left[w''_1 + w''_2 + w''_3 \right] + w'_0 \tag{3.44}$$

It can be seen that this expression corresponds exactly to (3.30). It has therefore been shown that the midpoint rule can be derived by fitting the sensor measurements to a set of simple global shape functions.

3.3.2 Trapezoidal Rule

Similar comparisons can be made between other rules and global shape functions.

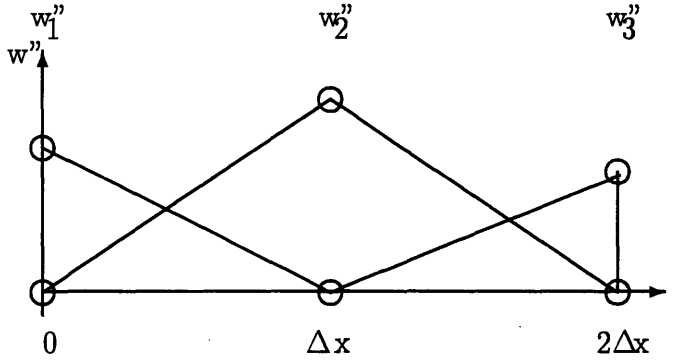

Figure 3.9: The trapezoidal rule applied to a beam with three sensors.

As a second example, the trapezoidal rule is investigated. It is assumed that three sensors are used: a sensor is placed at each end, and one is placed at the center of the structure, as shown in Figure 3.9. The strain profile assumed is also shown in Figure 3.9. The slope of the structure at the tip can be found by the trapezoidal rule:

$$w'_{\text{tip}} = \frac{1}{2}\Delta x[w''_1 + 2w''_2 + w''_3] + w'_0 \tag{3.45}$$

where w'_o is the slope of the structure at its root.

As with the midpoint rule, global shape functions are now used to obtain the same result. The shape functions used are triangular over the domain. For a slope estimate based on curvature measurements, they are:

$$\phi''_1(x) = \begin{cases} \frac{1}{\Delta x^2}\left(1 - \frac{x}{\Delta x}\right) & 0 < x < \Delta x \\ 0 & \Delta x < x < 2\Delta x \end{cases} \tag{3.46}$$

$$\phi''_2(x) = \begin{cases} \frac{1}{\Delta x^2}\left(\frac{x}{\Delta x}\right) & 0 < x < \Delta x \\ \frac{1}{\Delta x^2}\left(2 - \frac{x}{\Delta x}\right) & \Delta x < x < 2\Delta x \end{cases} \tag{3.47}$$

$$\phi''_3(x) = \begin{cases} 0 & 0 < x < \Delta x \\ \frac{1}{\Delta x^2}\left(\frac{x}{\Delta x} - 1\right) & \Delta x < x < 2\Delta x \end{cases} \tag{3.48}$$

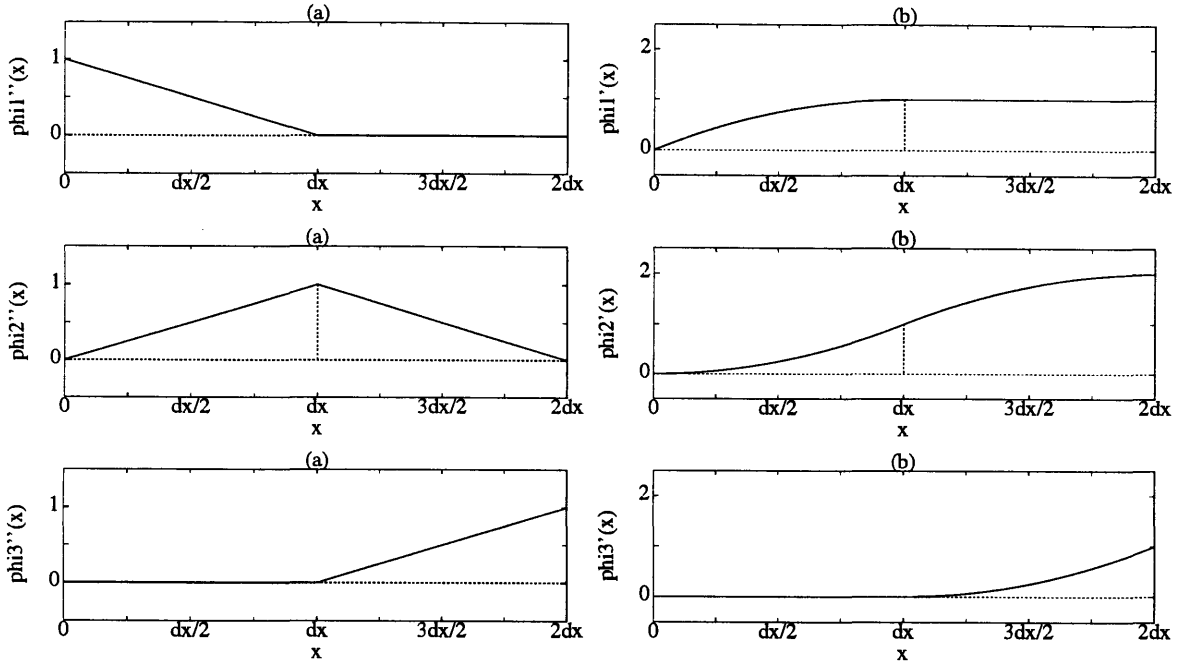

Figure 3.10: The global shape functions of the trapezoidal rule: (a) $\phi_i''(x)$; (b) $\phi_i'(x)$.

These are integrated analytically to yield:

$$\phi_1'(x) = \begin{cases} \frac{1}{\Delta x^2}\left(x - \frac{x^2}{2\Delta x}\right) & 0 < x < \Delta x \\ \frac{1}{2\Delta x} & \Delta x < x < 2\Delta x \end{cases} \tag{3.49}$$

$$\phi_2'(x) = \begin{cases} \frac{1}{\Delta x^2}\left(\frac{x^2}{2\Delta x}\right) & 0 < x < \Delta x \\ \frac{1}{\Delta x^2}\left(2x - \frac{x^2}{2\Delta x}\right) - \frac{1}{\Delta x} & \Delta x < x < 2\Delta x \end{cases} \tag{3.50}$$

$$\phi_3'(x) = \begin{cases} 0 & 0 < x < \Delta x \\ \frac{1}{\Delta x^2}\left(\frac{x^2}{2\Delta x} - x\right) + \frac{1}{2\Delta x} & \Delta x < x < 2\Delta x \end{cases} \tag{3.51}$$

The global shape functions and their integrals are shown in Figure 3.10. The measured strains reported by the sensors can be expressed as a linear combination of

the global shape functions by using three generalized coordinates p_i:

$$\begin{bmatrix} w_1'' \\ w_2'' \\ w_3'' \end{bmatrix} = \begin{bmatrix} \phi_1''(0) & \phi_2''(0) & \phi_3''(0) \\ \phi_1''(\Delta x) & \phi_2''(\Delta x) & \phi_3''(\Delta x) \\ \phi_1''(2\Delta x) & \phi_2''(2\Delta x) & \phi_3''(2\Delta x) \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{3.52}$$

The generalized coordinates p_i can be found by matrix inversion:

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} \phi_1''(0) & \phi_2''(0) & \phi_3''(0) \\ \phi_1''(\Delta x) & \phi_2''(\Delta x) & \phi_3''(\Delta x) \\ \phi_1''(2\Delta x) & \phi_2''(2\Delta x) & \phi_3''(2\Delta x) \end{bmatrix}^{-1} \begin{bmatrix} w_1'' \\ w_2'' \\ w_3'' \end{bmatrix} \tag{3.53}$$

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta x^2} & 0 & 0 \\ 0 & \frac{1}{\Delta x^2} & 0 \\ 0 & 0 & \frac{1}{\Delta x^2} \end{bmatrix}^{-1} \begin{bmatrix} w_1'' \\ w_2'' \\ w_3'' \end{bmatrix} \tag{3.54}$$

The generalized coordinates can then be used to find the estimated slope of the structure at the sensor locations (and the tip of the structure):

$$\begin{bmatrix} w_{\text{tip}}' \end{bmatrix} = \begin{bmatrix} \phi_1'(2\Delta x) & \phi_2'(2\Delta x) & \phi_3'(2\Delta x) \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{3.55}$$

$$\begin{bmatrix} w_{\text{tip}}' \end{bmatrix} = \begin{bmatrix} \frac{1}{2\Delta x} & \frac{1}{\Delta x} & \frac{1}{2\Delta x} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{3.56}$$

Equation (3.54) can now be used to solve for the tip slope in terms of the measured

strains:

$$\begin{bmatrix} w'_{\text{tip}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2\Delta x} & \frac{1}{\Delta x} & \frac{1}{2\Delta x} \end{bmatrix} \begin{bmatrix} \frac{1}{\Delta x^2} & 0 & 0 \\ 0 & \frac{1}{\Delta x^2} & 0 \\ 0 & 0 & \frac{1}{\Delta x^2} \end{bmatrix}^{-1} \begin{bmatrix} w''_1 \\ w''_2 \\ w''_3 \end{bmatrix} \tag{3.57}$$

$$\begin{bmatrix} w'_{\text{tip}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\Delta x\, w''_1 + \Delta x\, w''_2 + \frac{1}{2}\Delta x\, w''_3 \end{bmatrix} \tag{3.58}$$

After including the initial conditions at the root of the structure, we have:

$$w'_3 = \frac{1}{2}\Delta x\,[w''_1 + 2w''_2 + w''_3] + w'_0 \tag{3.59}$$

It can be seen that this expression corresponds exactly to (3.45). It has therefore been shown that the trapezoidal rule can be derived by fitting the sensor measurements to a set of simple global shape functions. Similar comparisons can be made for the Simpson and B-spline rules.

3.4 Structural Global Shape Functions

All the previously discussed integration rules can be reduced to a two step process: fitting the shape function to the data; and integrating the shape function. An open question remains: what other shape functions might be used in this process, which would lead to a more accurate or easily applicable integration?

An obvious set of possible shape functions are the set of structural global shape functions, some combination of the natural modes and static deformations due to actuator or disturbance input. Such shape functions can incorporate the boundary conditions appropriate to the structural geometry. Assuming an orthogonal set, we can write an estimate of the strain as a linear combination of the assumed global

shape functions $\phi_j''(x)$ of the structure by using n constants p_j:

$$\bar{w}''(x) = \sum_{j=1}^{n} p_j \, \phi_j''(x) \tag{3.60}$$

Let the n sensors report curvature measurements $w''(x_i)$. For point sensors, centered at $x = x_i$, we can write

$$w''(x_i) = \sum_{j=1}^{n} p_j \, \phi_j''(x_i), \qquad i = 1, \ldots, n \tag{3.61}$$

For sensors with a spatial weighting $f(x)$, we instead write

$$w''(x_i) = \sum_{j=1}^{n} p_j \int_0^L f_i(x) \, \phi_j''(x) \, dx, \qquad i = 1, \ldots, n \tag{3.62}$$

where $f_i(x)$ is the spatial weighting of the i-th sensor. Equations (3.61) and (3.62) can both be written in matrix form as:

$$\mathbf{w}'' = \Phi'' \, \mathbf{p} \tag{3.63}$$

where, for point sensors, we have:

$$\Phi_{ij}'' = \phi_j''(x_i) \tag{3.64}$$

and for spatially averaging sensors we have:

$$\Phi_{ij}'' = \int_0^L f_i(x) \, \phi_j''(x) \, dx \tag{3.65}$$

If the number of shape functions is equal to the number of sensors, then the shape coordinates p_j can be determined by inversion:

$$\mathbf{p} = [\Phi'']^{-1} \, \mathbf{w}'' \tag{3.66}$$

and the estimated strain $\bar{w}''$ from (3.60) can be integrated analytically to obtain the slope and displacement. This will be referred to as the Global Static Shape Function (GSSF) integration scheme when static shape functions are used as the $\phi_j''(x)$, and the Global Dynamic Shape Function (GDSF) integration scheme when the natural modes of the structure are used as the $\phi_j''(x)$.

This procedure parallels the process of fitting Chebyshev polynomials, discussed above in Section 3.2.8. Taking the points of interpolation within the interval to be the the zeros of the Chebyshev polynomials will produce the best fit to an arbitrary set of data. The advantage of the Chebyshev scheme is that no knowledge of the structural deformation modes is required. However, to accurately model boundary conditions, measurements are required at the structural boundaries. By comparison, the structural shape functions do not require end points, they incorporate the boundary condition and potentially nonhomogeneous load introduction information automatically, but require some (usually available) information to derive *a priori*.

Two global shape function integration schemes will be investigated. The first is the Global Static Shape Function integration scheme (GSSF), which uses as many global static shape functions as there are sensors. These static shape functions are the deflection shapes of the structure for the seven static loads shown in Figure 4.2. When more than seven sensors are used, and more than seven shape functions are required, the natural dynamic mode shapes of the structure are used. The second global shape function integration scheme is the Global Dynamic Shape Function integration scheme (GDSF), which uses as many global dynamic mode shapes of the structure as there are sensors.

A last alternative occurs when the number of assumed structural shape functions is smaller than the number of sensors. In this case, the estimated shape is still given

by (3.60), but the evaluation at a discrete location is, for point sensors:

$$w''(x_i) = \sum_{j=1}^{m} p_j \, \phi_j''(x_i), \qquad i = 1, \ldots n \quad (m < n) \tag{3.67}$$

and for spatially averaging sensors:

$$w''(x_i) = \sum_{j=1}^{m} p_j \int_0^L f_i(x) \, \phi_j''(x) \, dx, \qquad i = 1, \ldots n \quad (m < n) \tag{3.68}$$

For both these cases, the coordinates can be solved for by

$$\mathbf{p} = \left(\Phi^T \, \Phi\right)^{-1} \Phi \, \mathbf{w} \tag{3.69}$$

the solution to which can be substituted into (3.60) and integrated analytically to obtain slope and displacement information. Two integration schemes of this type are investigated. The first is the Least Squares Global Static Shape Function integration scheme using two shape functions with two or more sensors (LSGSSF-2). The two shape functions used are the deflection shapes of the structure for the first two static loading conditions shown in Figure 4.2. The second integration scheme is the Least Squares Global Static Shape Function integration scheme using four shape functions with four or more sensors (LSGSSF-4). The four shape functions used in this case are the deflection shapes for the first four loadings shown in Figure 4.2

Ten methods for integration of strain sensors have been presented. The errors introduced by these in estimation of static displacements will be presented in Chapter 4 and dynamic displacements in Chapter 5.

Chapter 4

Accuracy of Integration Schemes for Static Shape Estimation

In order to study the accuracy of the integration schemes described in Chapter 3, for the estimation of the static deformations of a structure, a series of transverse static loadings were applied to a clamped-free beam in a number of simulations. It was assumed that the beam had an array of spatially averaging strain sensors distributed along its length. The number of sensors as well as their spatial weighting was varied in order to determine the number and type of sensors required to obtain good displacement estimation performance. In order to calculate the displacement of the beam, the curvature measured by the array of sensors was spatially integrated to estimate the slope of the beam, and integrated once again to obtain an estimate of the transverse displacement. Gage factor and sensor position errors were introduced to model errors that would be observed in an experimental implementation. Finally, a simple experiment was carried out in an attempt to verify some of the results of the simulations.

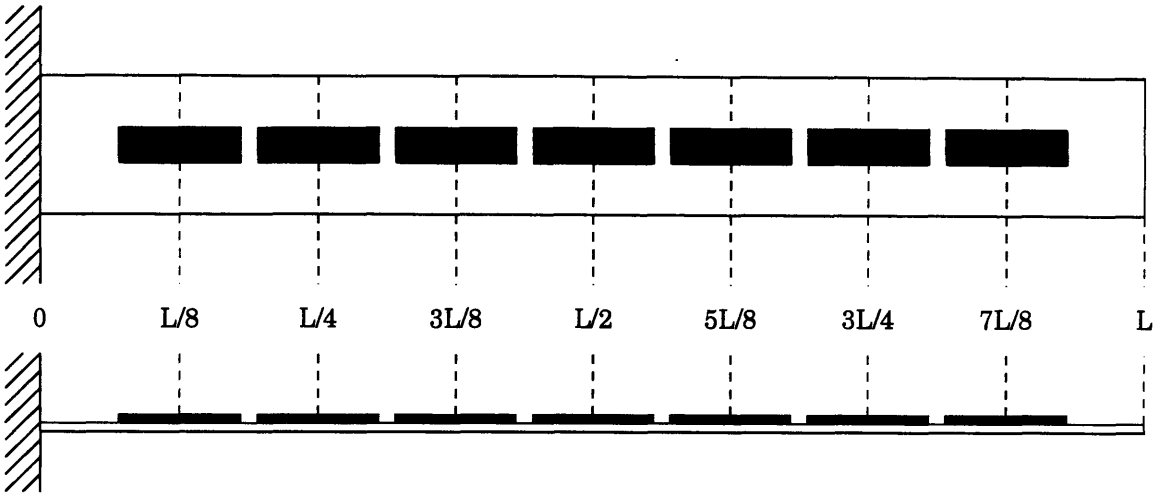

Figure 4.1: An array of 7 sensors mounted on a beam.

4.1 System Description

4.1.1 Beam Description

The beam was modeled as a uniform, clamped-free Bernoulli-Euler beam undergoing small transverse displacements. A sensor array consisting of a varying number of sensors was modeled. Different numbers of sensors were used to study the convergence performance of the integration schemes. For Simpson's rule, 3, 5, 7, 9 and 11 sensors were used to yield an even number of subdivisions of the integration interval. For all other rules, 2, 4, 6, 8 and 10 sensors were employed. As an example, Figure 4.1 shows how the sensors would be placed along the interior of the beam to form an array of 7 sensors.

The location of the center of the i-th sensor is then $x_i = iL/(n+1)$ where n is the number of sensors and L is the length of the beam. The single strain measurement each sensors reports is used to calculate the local beam curvature. The measured curvature for each of sensors is assigned to the center of the sensor, located at x_i.

All schemes except the Midpoint, Gauss, Radau and structural shape function

fitting rules require measurements at the end points of the integration interval, as shown in Table 3.1. The end points of the interval correspond to the root and tip of the beam. Since a complete sensor cannot be centered at the root of the beam, the root curvature was extrapolated by fitting a parabola through the curvatures measured by the first three sensors outboard of the root. This parabola was then extrapolated to the root, and the curvature estimated there. The tip curvature was simply assumed to be zero, as the beam force boundary conditions require.

Three types of sensors, of two different lengths were considered. In order to compare the effect of measuring curvature at a discrete point with finite area averaging sensors, point, rectangular and Bartlett (triangular) spatial weightings were employed. Due to the low spatial frequency of the static loadings, it is expected that Gauss-Hanning sensors will exhibit similar performance to Bartlett sensors. The point sensor is modeled as having a delta function weighting, and extracts the value of the curvature at the point where it is placed. The rectangular and Bartlett sensors are assumed to be long enough to extend over the entire subdivision of length $L/(n+1)$.

4.1.2 Loading Functions

In order to induce transverse displacements to be estimated using the sensor array, a set of seven representative loading functions was selected in order to model a generic quasistatic loading situation. These seven loadings are shown in Figure 4.2

The first two loading functions ((a) and (b)) are a point load at the tip, and a point load at the tip with another point load at the midspan, in the opposite direction. These loadings give rise to displacements expressed as third order polynomials. These loadings can be written as

$$p_a(x) = P\,\delta(x-L), \qquad 0 \le x \le L \tag{4.1}$$

$$p_b(x) = P\,\delta(x-L) - P\,\delta(x-L/2), \qquad 0 \le x \le L \tag{4.2}$$

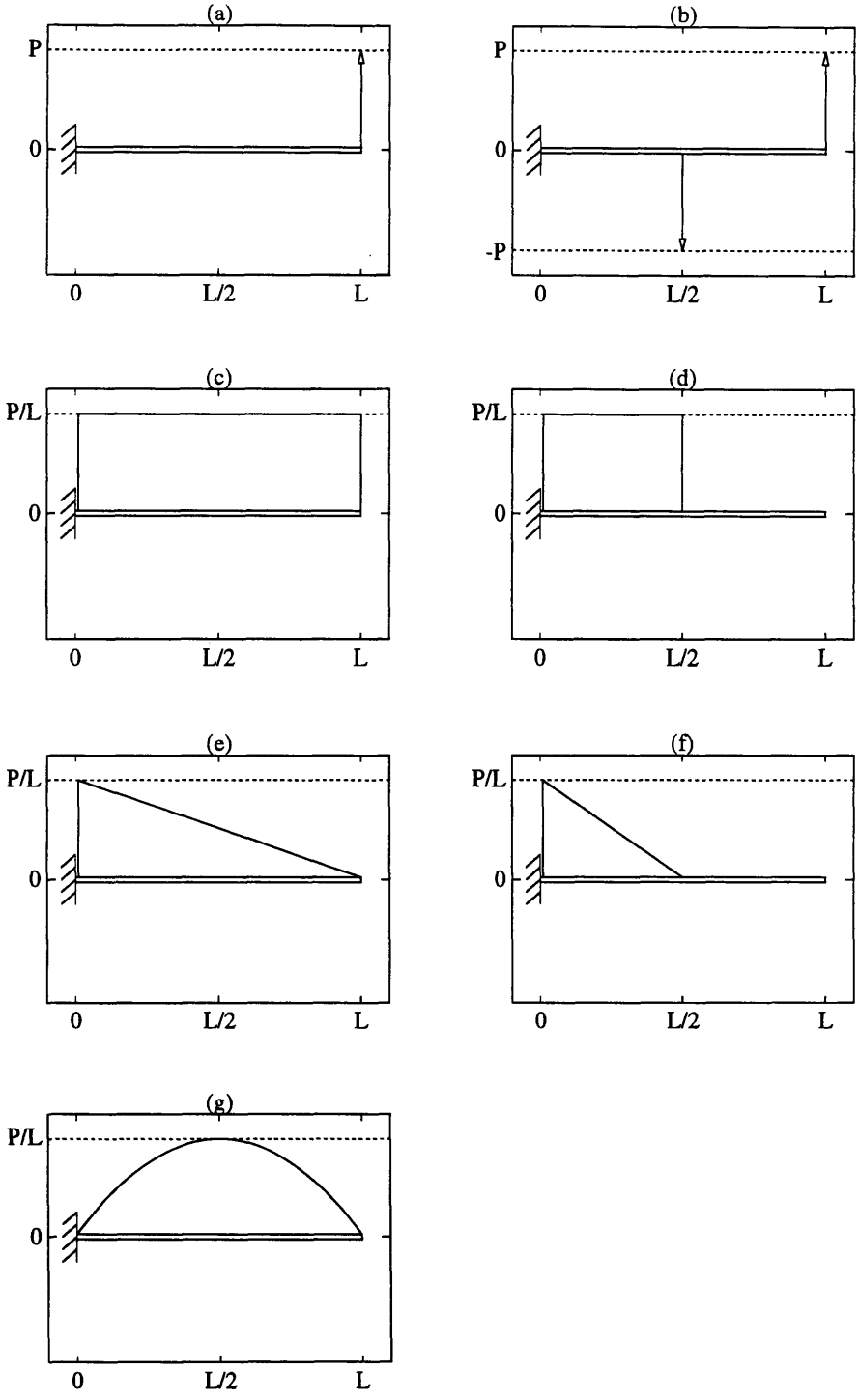

Figure 4.2: The set of 7 generic quasistatic beam loadings used to induce transverse displacements to be estimated by a sensor array mounted on a clamped-free beam.

The third and fourth loadings ((c) and (d)) are uniform distributed rectangular loadings acting over either the entire beam or over the inner half of the beam. These loadings give rise to displacements expressed as fourth order polynomials. The loadings are given as

$$p_c(x) = \frac{P}{L}, \qquad 0 \le x \le L \tag{4.3}$$

$$p_d(x) = \frac{P}{L}\left(u(x) - u\left(x - \frac{L}{2}\right)\right), \qquad 0 \le x \le L \tag{4.4}$$

where $u(x)$ is the unit step function

$$u(x) = \begin{cases} 1, & x \ge 0 \\ 0, & x < 0 \end{cases} \tag{4.5}$$

The fifth and sixth loadings ((e) and (f)) are distributed triangular loadings that taper off linearly, ending at zero force per unit length either at the tip or at the midspan of the beam. These loadings give rise to displacements expressed at fifth order polynomials. The loadings are given by

$$p_e(x) = \frac{P}{L}\left(1 - \frac{x}{L}\right), \qquad 0 \le x \le L \tag{4.6}$$

$$p_f(x) = \frac{P}{L}\left(1 - \frac{2x}{L}\right), \qquad 0 \le x \le L \tag{4.7}$$

The last loading (g) is a distributed symmetric parabolic loading with its maximum value and point of symmetry at the midspan. This loading gives rise to a displacement expressed as a sixth order polynomial. The loading is given by

$$p_f(x) = \frac{4P}{L}\left(\frac{x}{L}\right)\left(1 - \frac{x}{L}\right), \qquad 0 \le x \le L \tag{4.8}$$

The curvature and displacement for these seven loadings are given by:

$$\kappa_a(x) = \frac{P}{EI}(L-x) \tag{4.9}$$

$$w_a(x) = \frac{1}{6}\frac{P}{EI}(3L-x) \tag{4.10}$$

$$\kappa_b(x) = \begin{cases} \frac{1}{2}\frac{P}{EI}L, & 0 < x < L/2 \\ \frac{P}{EI}(L-x), & L/2 < x < L \end{cases} \tag{4.11}$$

$$w_b(x) = \begin{cases} \frac{1}{4}\frac{P}{EI}Lx^2, & 0 < x < L/2 \\ \frac{1}{48}\frac{P}{EI}(24Lx^2 - 8x^3 - 6L^2x + L^3), & L/2 < x < L \end{cases} \tag{4.12}$$

$$\kappa_c(x) = \frac{1}{2}\frac{P}{EI}(L-x)^2 \tag{4.13}$$

$$w_c(x) = \frac{1}{24}\frac{P}{EI}x^2(6L^2 - 4Lx + x^2) \tag{4.14}$$

$$\kappa_d(x) = \begin{cases} \frac{1}{2}\frac{P}{EI}(x - L/2)^2, & 0 < x < L/2 \\ 0, & L/2 < x < L \end{cases} \tag{4.15}$$

$$w_d(x) = \begin{cases} \frac{1}{48}\frac{P}{EI}x^2(2x^2 - 4Lx + 3L^2), & 0 < x < L/2 \\ \frac{1}{384}\frac{P}{EI}L^3(8x - L), & L/2 < x < L \end{cases} \tag{4.16}$$

$$\kappa_e(x) = \frac{1}{6}\frac{P}{EI}(L-x)^3 \tag{4.17}$$

$$w_e(x) = \frac{1}{120}\frac{P}{EI}x^2(5Lx^2 - 10L^2x + 10L^3 - x^3) \tag{4.18}$$

$$\kappa_f(x) = \begin{cases} \frac{1}{6}\frac{P}{EI}(x - L/2)^2(2L - x), & 0 < x < L/2 \\ 0, & L/2 < x < L \end{cases} \tag{4.19}$$

$$w_f(x) = \begin{cases} \frac{1}{240}\frac{P}{EI}x^2(-2x^3+10Lx^2-15L^2x+10L^3), & 0<x<L/2 \\ \frac{1}{1920}\frac{P}{EI}L^3(25x-3L), & L/2<x<L \end{cases} \quad (4.20)$$

$$\kappa_g(x) = \frac{1}{3L^2}\frac{P}{EI}(L-x)^3(L+x) \quad (4.21)$$

$$w_g(x) = \frac{1}{90L^2}\frac{P}{EI}x^2(-x^4+3Lx^3-10L^3x+15L^4) \quad (4.22)$$

4.2 Simulation of Experimental Errors

There are two potential sources of inaccuracy in a physical implementation where strain measurements are spatially integrated using a numerical integration scheme to estimate deflection. The first is the absolute error introduced by the numerical integration scheme itself, even when the outputs of the sensors are nominally perfect. The other source of inaccuracy in the estimated shape is due to inaccuracies in the measurements made by the sensors. In an effort to model the latter type of inaccuracy, the numerical simulation of the measurement and spatial integration process included simulated measurement uncertainties.

Such measurement uncertainties can be of two general types. The first type are uncertainties that directly affect the magnitude of the measured strain, through uncertainties or changes in the gain or gage factor of the sensors. The second type of uncertainties are those that affect the integrated measurement due to uncertainties in the effective location of the strain measurement. The modeling of these types of error is now described.

4.2.1 Gage Factor Uncertainties

Gage factor uncertainties are introduced when the gain of the sensor is not accurately known. Uncertainties in gage factor and distance of the sensor from the

centerline of a beam in bending affect the magnitude of the reported strain in a linear manner. Angular misalignment of the sensor with respect to the longitudinal axis of the structure affects the sensor output in a nonlinear manner.

Optimistic and more realistic uncertainty levels for this class of errors were considered and introduced in the numerical simulations. For the optimistic uncertainty level, gage factor errors of up to $\pm 1\%$ were used, while for the more realistic uncertainty level, gage factor errors of up to $\pm 5\%$ were used. A uniform random distribution of such uncertainties was assumed for each sensor. For example, in the case of $\pm 1\%$ gage factor uncertainty, each sensor was assigned a random gage factor in the interval $[0.99, 1.01]$ ($\pm 1\%$). The outputs of the sensors was then computed by multiplying the actual curvature each sensor should measure by the random gage factor.

4.2.2 Sensor Placement Uncertainties

Uncertainties in the effective location of the strain measurement occur when the longitudinal position of the sensor or its sensitive region is not accurately known. These uncertainties can be introduced as fabrication errors, or by having a complex sensitive grid, as in the case of a conventional strain gage. As with gage factor uncertainties, optimistic and more realistic uncertainty levels were defined. For the optimistic uncertainty level, sensor position uncertainties of up to $\pm 0.2\%$ of the length of the beam were used. For the more realistic uncertainty level, sensor position uncertainties of up to $\pm 1\%$ of the length of the beam were used. Again, a uniform random distribution of such uncertainties was assumed for each sensor. For example, in the case of the $\pm 0.2\%$ sensor position uncertainty, each sensor was 'moved' longitudinally from its original position by a random distance in the interval $[-0.002L, 0.002L]$, where L is the length of the beam. The outputs of the sensors were then computed for the new positions of the sensors.

4.2.3 Error Computation

To quantify the first source of inaccuracy (the absolute error due only to the integration scheme), an array of perfect sensors was simulated, and their outputs were numerically integrated to estimate the tip slope and deflection. The absolute error of the integration scheme is then the difference between the estimated tip deflection or slope, and the theoretically calculated tip deflection or slope. Then, a Monte Carlo scheme of hundreds of iterations was run for each of the two types of uncertainties at both the optimistic and realistic levels. This was done for each integration scheme and for each loading distribution. In this manner the standard deviation of the error in estimated tip slope and deflection could be computed.

Values of both tip slope and deflection could be generated for 5 numbers of sensors, 3 sensor types, 7 loadings and 6 integration rules, each for perfect sensors and four types and levels of uncertainty. This amounts to over 6,000 values of tip slope and displacement. In light of this, it was necessary to somehow reduce this wealth of information and present it in a more compact manner. Thus, for each integration scheme, sensor type and number of sensors, the absolute errors due only to the integration from the 7 individual loadings were combined as a root mean square error. In addition, two different standard deviations of the error in estimated tip slope and displacement were defined and computed. The first was representative of optimistic sensors, with gage factor uncertainties of up to $\pm 1\%$ *combined* with sensor position uncertainties of up to $\pm 0.2\%$ of the beam length. The second represented a more realistic situation with sensors having gage factor uncertainties of up to $\pm 5\%$ *combined* with sensor position uncertainties of up to $\pm 1\%$ of the beam length. Gage factor and position uncertainties were combined by computing the square root of the sum of the squares of the individual standard deviations, due only to gage factor uncertainty and only sensor position uncertainty [12, 8].

4.3 Performance of Integration Schemes

In the following sections, the performance of the integration schemes described in Chapter 4 in estimating the quasistatic slope and deflection of a clamped-free beam is presented. In Figures 4.3–4.26, the tip slope and tip deflection estimation error is plotted as a function of the number of sensors employed in the interior of the beam. The error is expressed as a percentage of the actual tip slope and tip deflection. Note that deflection estimates cannot be obtained using the Gauss and Radau rules.

The performance of point, rectangular and Bartlett sensors are shown for all integration schemes. The circles represent the absolute error incurred purely due to the numerical integration scheme, with no gage factor or sensor position uncertainties included. The two error bars above and below these points represent one standard deviation for the optimistic and realistic error levels, respectively. In all cases the smaller error bars indicate the optimistic uncertainty level, which will be referred to as σ_o, and the larger error bars the more realistic, and higher uncertainty level, which will be referred to as σ_r.

4.3.1 Midpoint Rule

Figures 4.3(a), 4.4(a) and 4.5(a) show tip slope estimation errors observed with the midpoint rule as the integration scheme, when using point, rectangular and Bartlett sensors, respectively. It can be seen that when using 2 sensors, the tip slope estimation errors are largest when using point sensors. Bartlett sensors yield slightly lower errors, and rectangular gages yield the lowest errors. As the number of sensors is increased, the error using the point sensors decreases rapidly, at about $1/n^3$. In fact, for 4 sensors, the point sensors give a lower tip slope estimation error (1.2%) than the rectangular or Bartlett sensors. The error when using the Bartlett sensors converges to zero more slowly, at about $1/n^2$. Finally, the rectangular sensors yield the slowest convergence rate: approximately $1/n$. Thus, although the point sensors

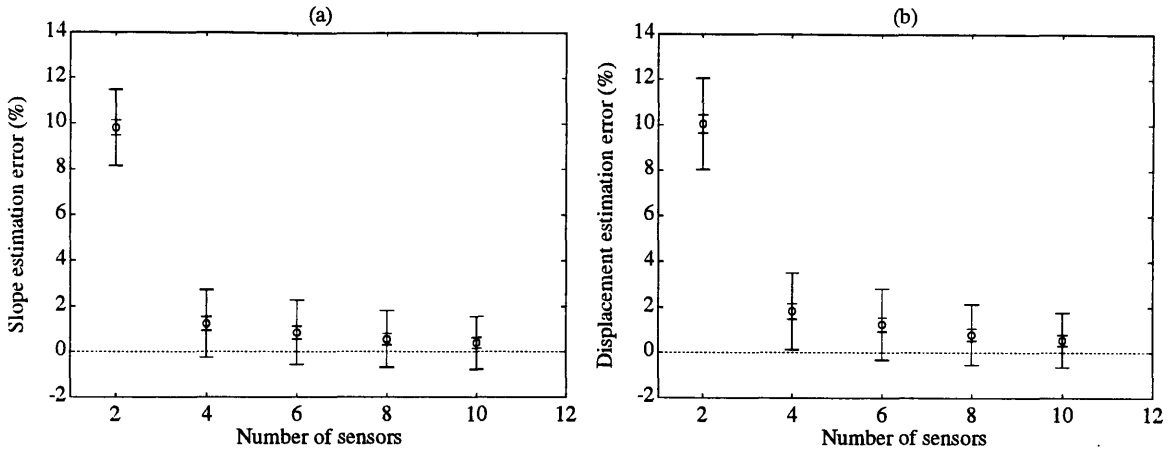

Figure 4.3: Accuracy of slope and displacement estimation using the Midpoint rule and point sensors; (a) slope estimation error; (b) displacement estimation error.

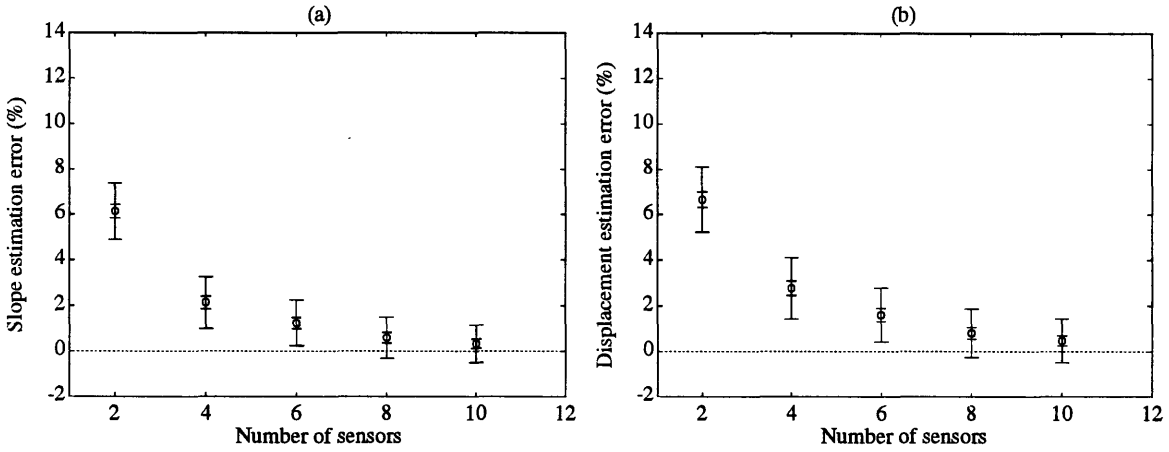

Figure 4.4: Accuracy of slope and displacement estimation using the Midpoint rule and rectangular sensors; (a) slope estimation error; (b) displacement estimation error.

give the highest errors for a small number of sensors, they make up for it with a faster convergence rate. For 10 sensors, the point and rectangular sensors give the lowest errors, at 0.38% and 0.33% respectively. The Bartlett sensors yield a larger tip slope deflection error of 0.47%.

For the standard deviation of the tip slope estimation error, the point sensors consistently give larger standard deviations than the rectangular and Bartlett sensors. It will be shown that this is the case no matter what integration scheme is being used.

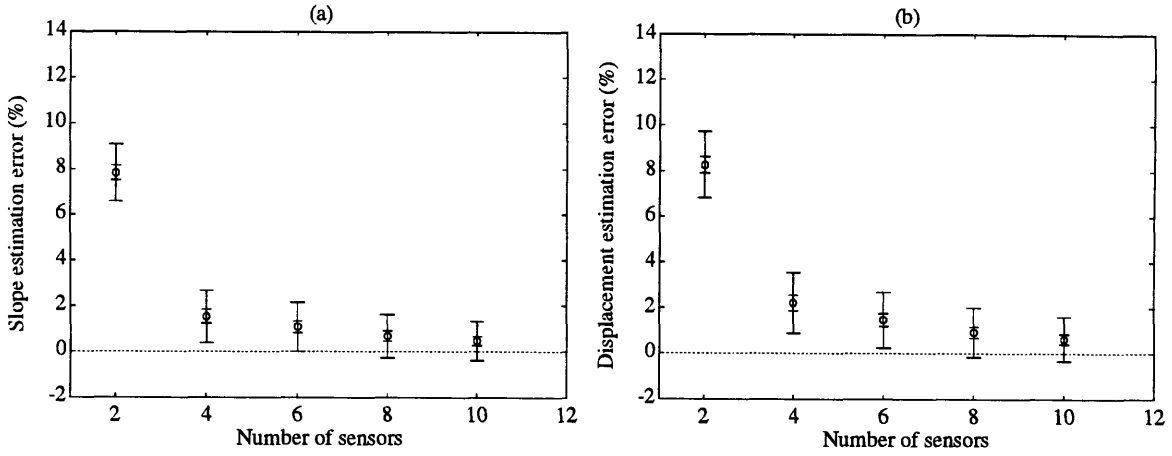

Figure 4.5: Accuracy of slope and displacement estimation using the Midpoint rule and Bartlett sensors; (a) slope estimation error; (b) displacement estimation error.

The strain-averaging properties of the rectangular and Bartlett sensors filter out the relatively high spatial frequency gage factor and sensor position uncertainties. It can be seen that for the point sensors, the absolute error in tip slope estimation is within σ_r of zero when using 4 or more sensors. With rectangular and Bartlett sensors, 8 or more sensors need to be used.

Figures 4.3(b), 4.4(b) and 4.5(b) show tip deflection estimation errors obtained with the midpoint rule as the integration scheme, when using point, rectangular and Bartlett sensors, respectively. For tip deflection estimation, the point sensors give the largest errors for 2 sensors, while rectangular sensors yield the highest errors for 4 and 6 sensors, and Bartlett sensors give the highest errors for 8 and 10 sensors. The smallest errors for 4, 6 and 8 sensors are obtained by using point sensors, whereas the lowest errors for 10 sensors are observed when using rectangular sensors. It can be seen that in general, the absolute error due only to the integration rule converges to zero relatively quickly, and is on the order of 1% for all three sensor types when 6 sensors are employed.

The standard deviation of the estimation error follows the same trend as for the

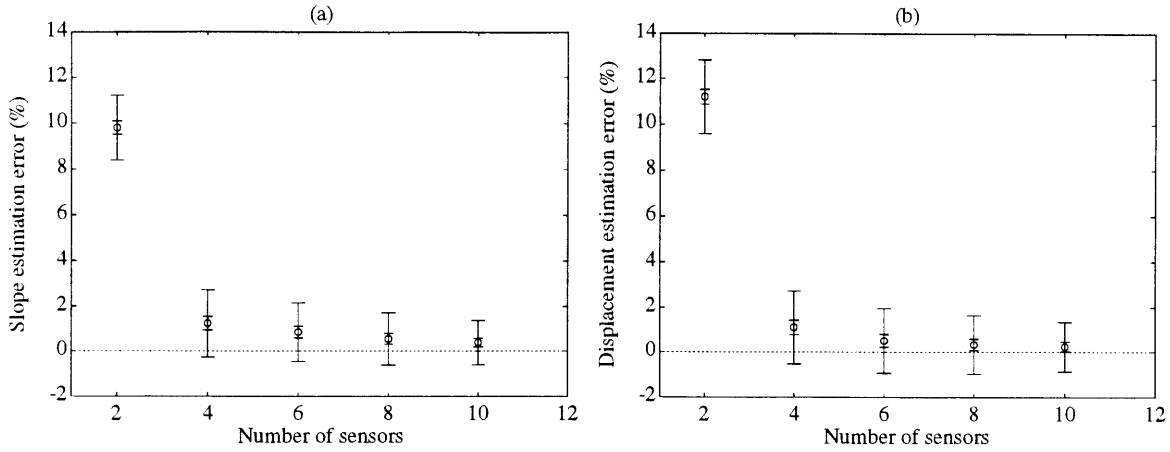

Figure 4.6: Accuracy of slope and displacement estimation using the trapezoidal rule and point sensors; (a) slope estimation error; (b) displacement estimation error.

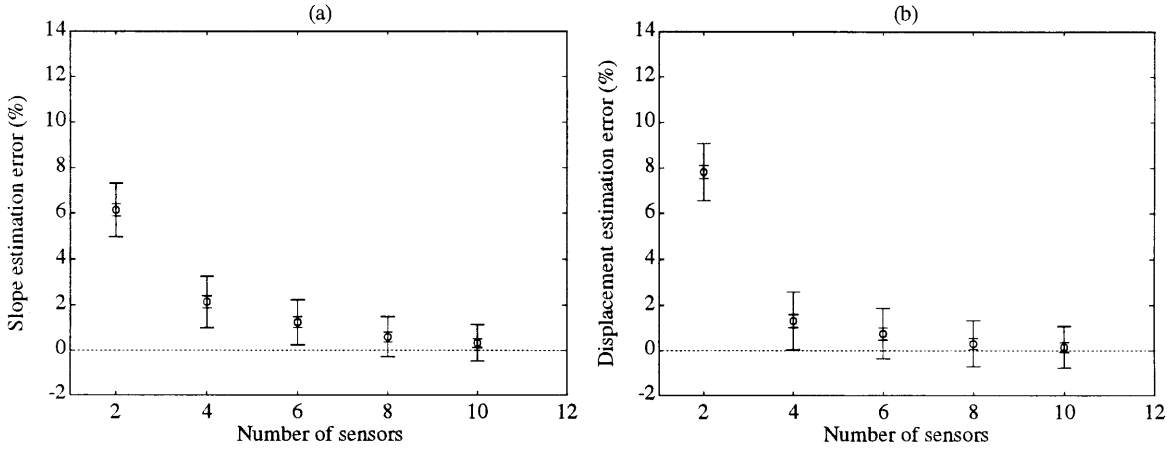

Figure 4.7: Accuracy of slope and displacement estimation using the trapezoidal rule and rectangular sensors; (a) slope estimation error; (b) displacement estimation error.

slope estimation problem: it is largest for the point sensors, and up to 0.5% smaller for the rectangular and Bartlett sensors. The standard deviation of the rectangular sensors is slightly smaller than that of the Bartlett sensors, for all but 4 sensors.

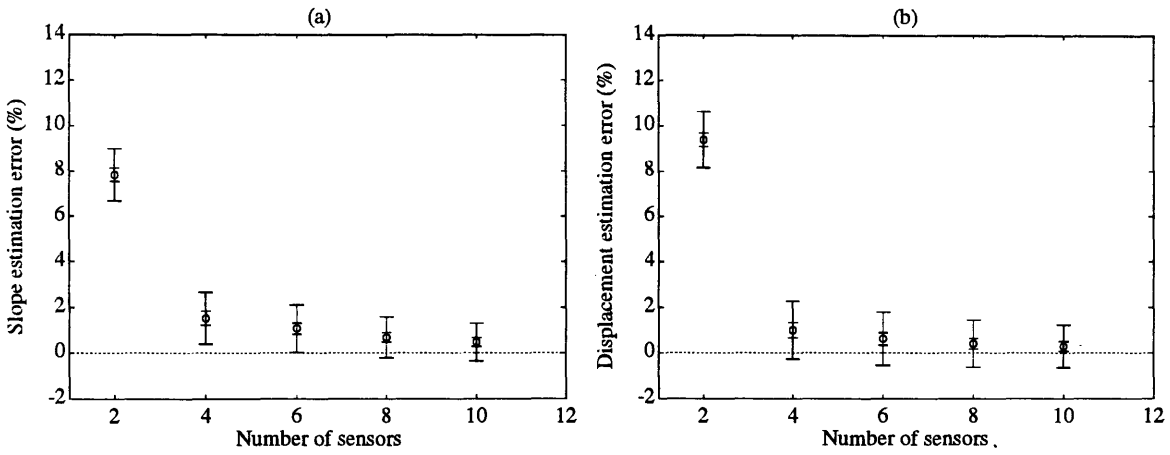

Figure 4.8: Accuracy of slope and displacement estimation using the trapezoidal rule and Bartlett sensors; (a) slope estimation error; (b) displacement estimation error.

4.3.2 Trapezoidal Rule

Figures 4.6(a), 4.7(a) and 4.8(a) show the tip slope estimation errors observed with the trapezoidal rule as the integration scheme, when using point, rectangular and Bartlett sensors, respectively. It can be seen that the best performance for 2 sensors is obtained when using rectangular sensors. Also, when using 10 sensors, the rectangular sensors yield the best results. However, for an intermediate number of sensors, the point sensors yield the best results. The error for the point sensors exhibits approximately $1/n^3$ rolloff, while the Bartlett sensors yield only $1/n^2$ rolloff, and the rectangular sensors only $1/n$ rolloff. It is seen that using point sensors, the tip slope estimation error is below 1% for 6 or more sensors. In contrast, 8 or more sensors are needed to achieve less than 1% error when using rectangular or Bartlett sensors.

For the standard deviation of the estimation error, it is larger for the point sensors, and smaller and approximately the same for the rectangular and Bartlett sensors. For the case of realistic uncertainties, the standard deviation of the error using point sensors is about 0.2% higher than when using rectangular or Bartlett sensors.

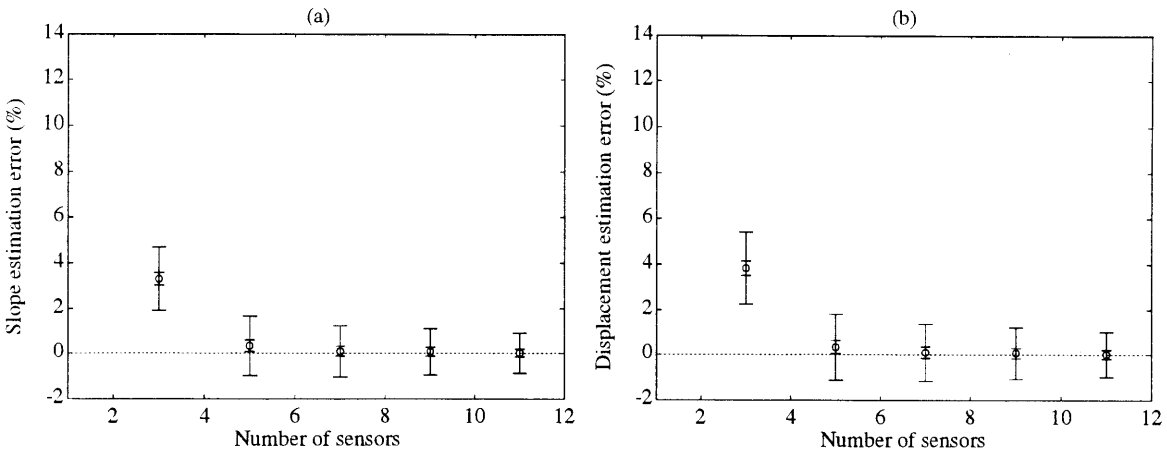

Figure 4.9: Accuracy of slope and displacement estimation using the Simpson's rule and point sensors; (a) slope estimation error; (b) displacement estimation error.

Figures 4.6(b), 4.7(b) and 4.8(b) show the tip deflection estimation errors obtained with the trapezoidal rule for point, rectangular and Bartlett sensors, respectively. As for slope estimation, for 2 sensors, the best performance and smallest estimation error is obtained with rectangular sensors. Again, for 10 sensors, rectangular sensors yield the lowest error. For an intermediate number of sensors along the structure, point sensors again yield better results than Bartlett or rectangular sensors. The deflection estimation error with point sensors exhibits approximately $1/n^4$ rolloff, while Bartlett sensors have $1/n^3$ rolloff and rectangular sensors $1/n^2$ rolloff. These rates are exactly $1/n$ more than for slope estimation using the trapezoidal rule.

As expected, the standard deviation of the estimation error is consistently smaller for rectangular and Bartlett sensors than for point sensors. However, this difference is less pronounced than in the case of slope estimation. This could be due to the filtering property of a second integration step used to obtain deflection estimates from slope estimates.

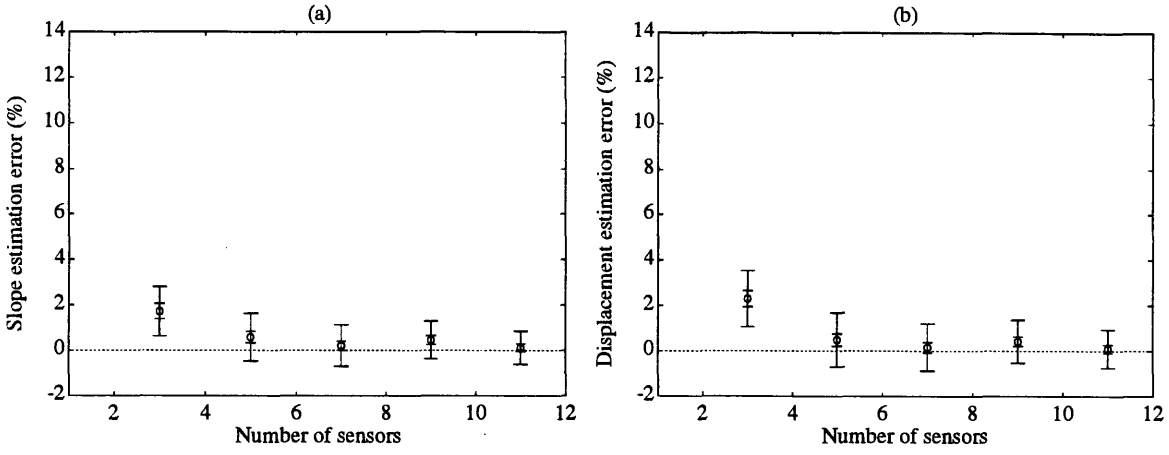

Figure 4.10: Accuracy of slope and displacement estimation using the Simpson's rule and rectangular sensors; (a) slope estimation error; (b) displacement estimation error.

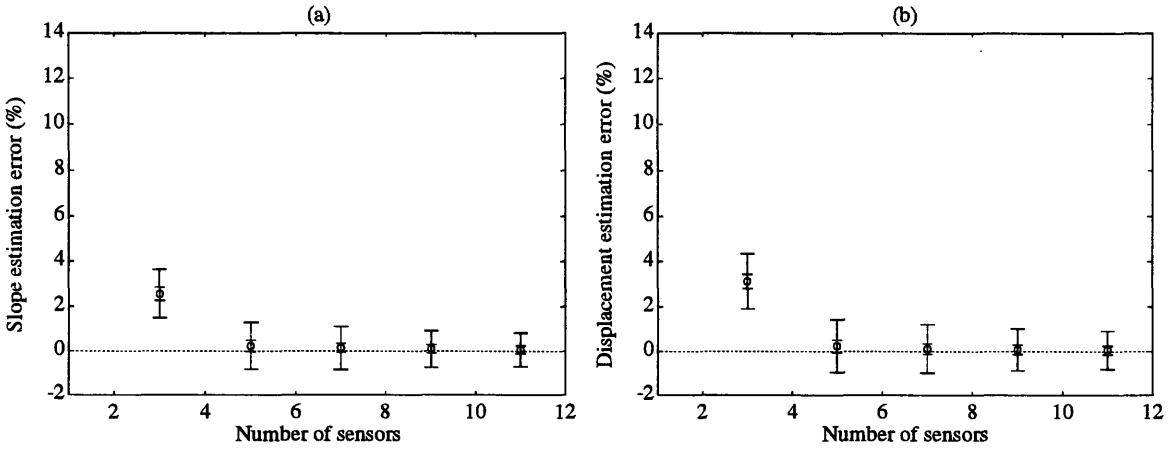

Figure 4.11: Accuracy of slope and displacement estimation using the Simpson's rule and Bartlett sensors; (a) slope estimation error; (b) displacement estimation error.

4.3.3 Simpson's Rule

Figures 4.9(a), 4.10(a) and 4.11(a) show the tip slope estimation errors exhibited by Simpson's rule, when using point, rectangular and Bartlett sensors, respectively. The best performance for few sensors is obtained with rectangular sensors, while point sensors yield the best results for 7 or more sensors. Note that to use Simpson's rule, an even number of subdivisions of the integration interval (the length of the structure

in this case) must be used. Therefore, an odd number of sensors is required. For 10 sensors, the error in tip slope estimation with point sensors is a fraction of the error obtained with rectangular and Bartlett sensors. All sensor types yield errors under 1% for 5 or more sensors, and for point sensors, the error is under 0.1% for 7 or more sensors. The point sensors seem to yield an estimation error rolloff of $1/n^4$, while the rectangular sensors yield approximately $1/n^2$ rolloff, and Bartlett sensors $1/n^5$ rolloff.

It can be seen that the standard deviation of the error is smallest for rectangular and Bartlett sensors due to their spatial filtering properties. This is more evident as the number of sensors is decreased as this in turn increases the length of the rectangular and Bartlett sensors in the array.

Figures 4.9(b), 4.10(b) and 4.11(b) show the tip deflection estimation errors observed with Simpson's rule as the integration rule for point, rectangular and Bartlett sensors, respectively. As with other integration rules, the rectangular sensors give the lowest errors for few sensors. The error with point sensors decreases more rapidly and yields the lowest errors for 7 or more sensors. For all sensor types, the error is below 0.5% for just 5 or more sensors. The standard deviation of the estimation error is greater than 0.5% for any number of sensors, and reaches 1% or more for few sensors. This indicates that while the performance of the sensors is excellent when no gage factor or sensor placement uncertainty is included, it is overwhelmed by the standard deviation of the estimation error when such uncertainties are included. It is clearly more beneficial to improve the quality of the sensor manufacturing process, as well as the accuracy of the measurement electronics in order to achieve an optimistic uncertainty level in the sensors. For 11 or more sensors, even the optimistic uncertainty level should be improved.

The standard deviation of the estimation error is higher for point sensors than for rectangular or Bartlett sensors. For the optimistic uncertainty level, however, this difference is not as marked, and is reversed in a few cases.

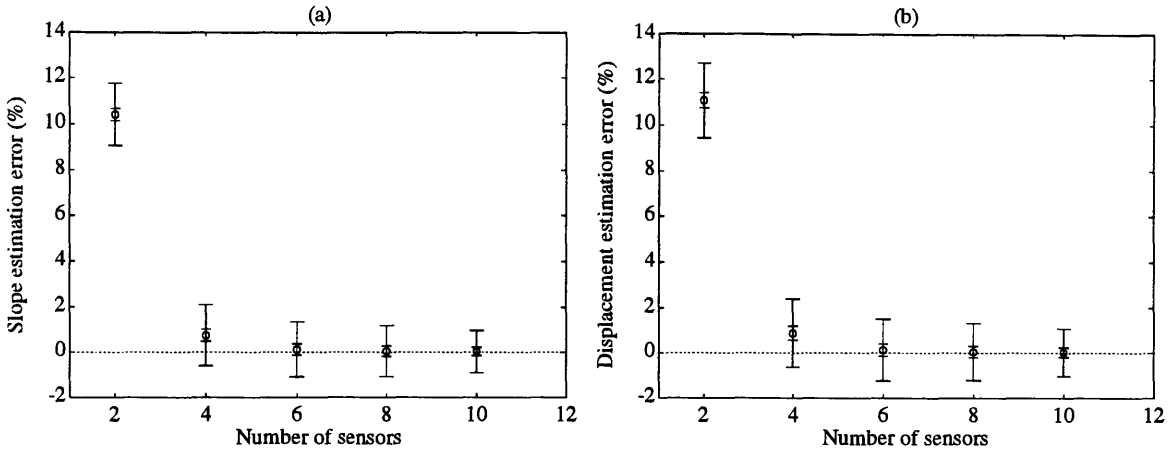

Figure 4.12: Accuracy of slope and displacement estimation using the Cubic Spline integration scheme and point sensors; (a) slope estimation error; (b) displacement estimation error.

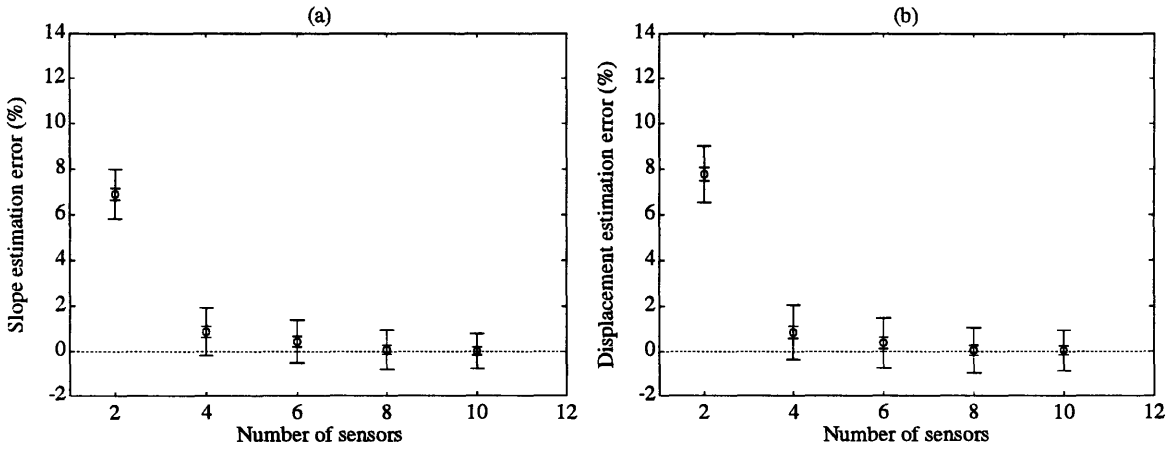

Figure 4.13: Accuracy of slope and displacement estimation using the Cubic Spline integration scheme and rectangular sensors; (a) slope estimation error; (b) displacement estimation error.

4.3.4 Cubic Spline Integration Scheme

Figures 4.12(a), 4.13(a) and 4.14(a) show the tip slope estimation errors exhibited by the cubic interpolating spline rule, when using point, rectangular and Bartlett sensors, respectively. As expected, the rectangular sensors yield the lowest errors when employing few sensors. For an intermediate number of sensors, the lowest errors are obtained with point sensors. However, for 10 sensors, it is actually the

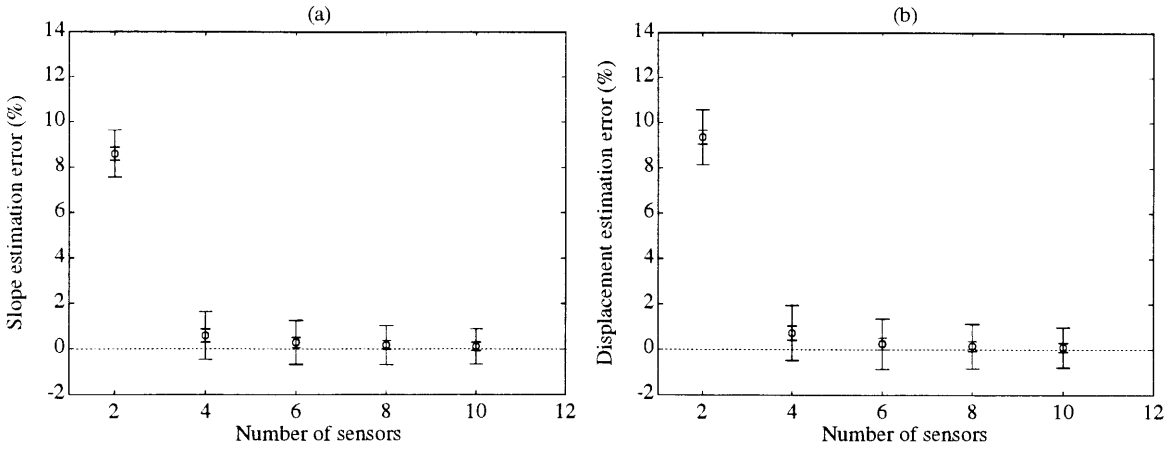

Figure 4.14: Accuracy of slope and displacement estimation using the Cubic Spline integration scheme and Bartlett sensors; (a) slope estimation error; (b) displacement estimation error.

rectangular sensors that yield the lowest error, under 0.023%. This is due to the very fast convergence of the scheme beyond 6 sensors. As the number of sensors is increased from 6 to 8, the error drops by a factor of 5, and drops by another factor of 4 as the number of sensors is increased to 10. The Bartlett sensors yield the best performance for 4 sensors, but the error does not converge to zero quickly as the number of sensors is increased. The errors are less than 1% for 4 sensors or more, however.

It is clear that very few sensors are required before the accuracy of the integration scheme is masked by the uncertainties present in a physical implementation. The realistic standard deviation of the estimation error, σ_r, is more than 0.5% for all three sensors types, for any number of sensors in the array. This standard deviation is significantly smaller in the strain-averaging sensors than in the point sensors.

Figures 4.12(b), 4.13(b) and 4.14(b) show the tip deflection estimation errors obtained with the spline rule, when using point, rectangular and Bartlett sensors, respectively. The same trend as for slope estimates does not hold: rectangular sensors yield the lowest errors for 2 sensors, but for 6 or more sensors, the point sensors exhibit

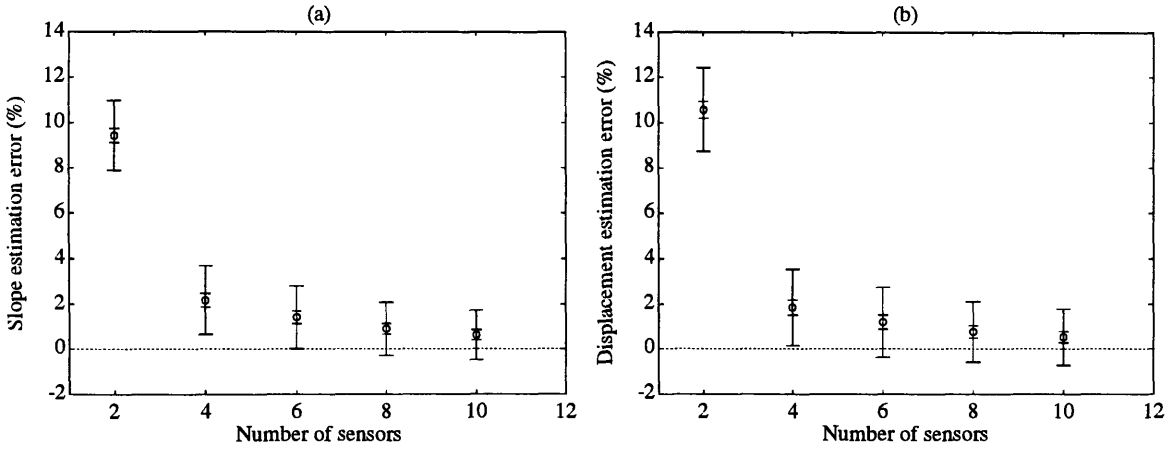

Figure 4.15: Accuracy of slope and displacement estimation using the B-Spline integration scheme and point sensors; (a) slope estimation error; (b) displacement estimation error.

the lowest errors. There is no rapid decrease in the tip deflection estimation error between 8 and 10 sensors, like there is between 6 and 8 sensors, where the error decreases by a factor of 6.5. As for slope estimation, the errors for all three sensor types are less than 1% for 2 or more sensors. The point and Bartlett sensors yield an improvement in estimation error of a factor of approximately 12.5 as the number of sensors is increased from 2 to 4. In the case of slope estimation, the corresponding improvement was by a factor of about 14.

The standard deviation of the estimation error is again larger for point sensors than for rectangular or Bartlett sensors, but this difference is not as large as for some of the other integration schemes.

4.3.5 B-Spline Integration Scheme

Figures 4.15(a), 4.16(a) and 4.17(a) show the tip slope estimation errors found by using the B-spline integration rule with point, rectangular and Bartlett sensors, respectively. It is immediately evident that while errors for few sensors are slightly lower when using the B-spline rule than when using the cubic spline scheme, the

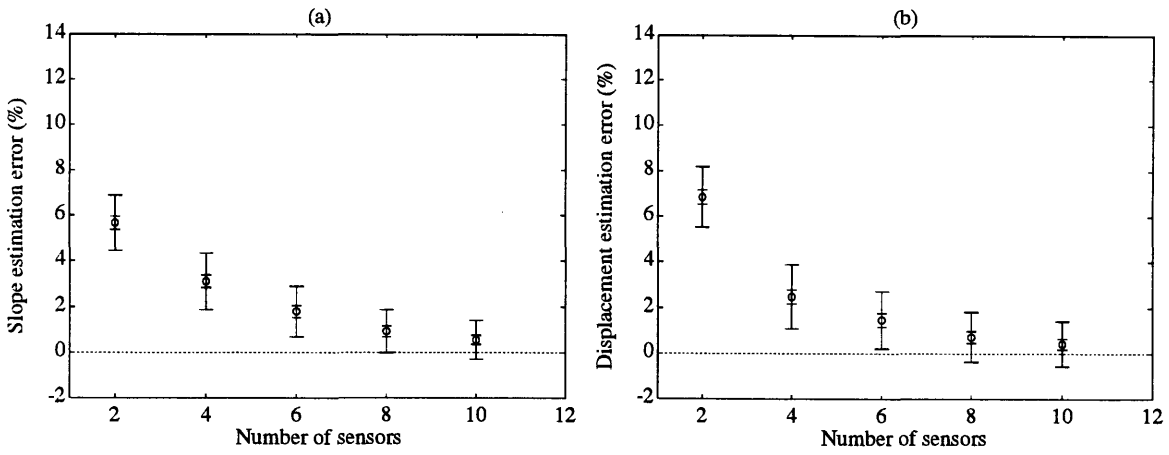

Figure 4.16: Accuracy of slope and displacement estimation using the B-Spline integration scheme and rectangular sensors; (a) slope estimation error; (b) displacement estimation error.

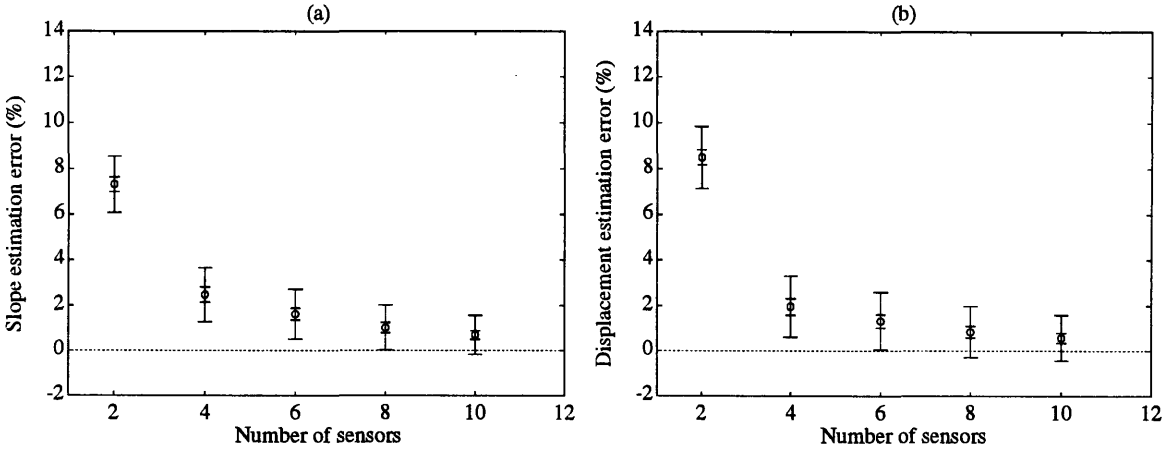

Figure 4.17: Accuracy of slope and displacement estimation using the B-Spline integration scheme and Bartlett sensors; (a) slope estimation error; (b) displacement estimation error.

convergence of the error for the B-spline rule is much slower. For example, 8 or more sensors are required to obtain errors smaller than 1%, while only 4 are required with the cubic spline rule. For 2 and 10 sensors, the lowest errors occur for rectangular sensors, and for 4, 6 and 8 sensors, the point sensors yield the lowest errors.

The lowest standard deviation of the estimation errors occur for rectangular and Bartlett sensors. The standard deviation of the point sensors is up to 0.3% larger

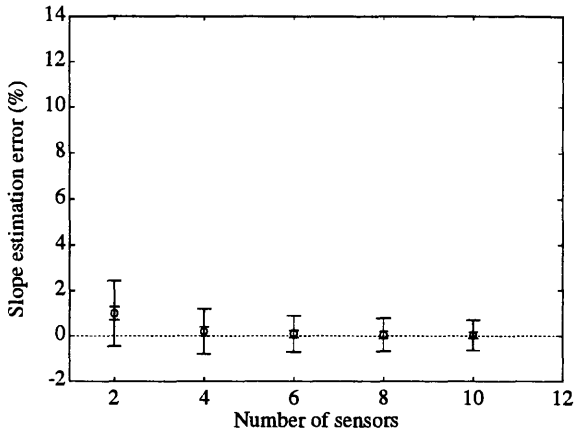

Figure 4.18: Accuracy of slope estimation using the Gauss integration scheme and point sensors.

than for rectangular or Bartlett sensors when employing only 2 or 4 sensors.

Figures 4.15(b), 4.16(b) and 4.17(b) show the tip deflection estimation errors that the B-spline integration scheme yields for point, rectangular and Bartlett sensors, respectively. The rectangular sensors give the lowest error for 2, 8 and 10 sensors. For 4 and 6 sensors, it is the point sensors that yield the best results. However, as with the slope estimates for the B-spline rule, the convergence of the error even for the rectangular sensors is very slow. For comparison, the error for 10 point sensors is 0.53% for the B-spline integration rule, while it is only 0.02% for Simpson's rule. This difference of a factor of 26 can be explained by noting that the B-spline curve does not interpolate its control points. The curve is merely tangent at the midpoint of the lines connecting two successive data points. This means that accurate representation of the actual strain field in the structure is, in general, only possible when then number of the sensors along the structure is large. In this case, it is more advantageous to use a more accurate integration scheme. The B-spline rule also produces standard deviations of the estimation error which are larger than for the cubic spline rule.

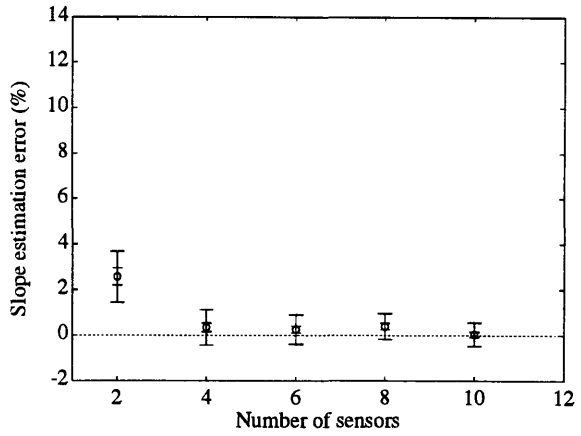

Figure 4.19: Accuracy of slope estimation using the Gauss integration scheme and rectangular sensors.

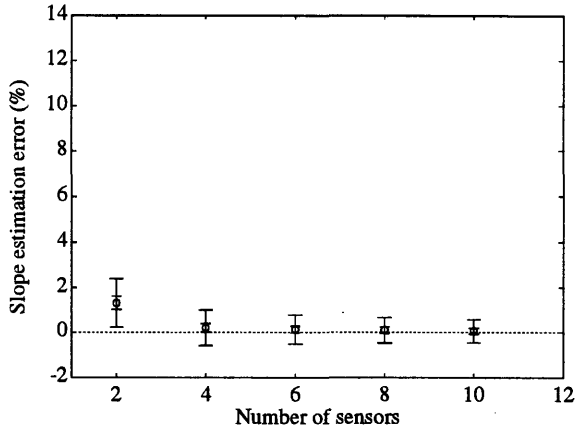

Figure 4.20: Accuracy of slope estimation using the Gauss integration scheme and Bartlett sensors.

4.3.6 Gauss Integration Scheme

Figures 4.18, 4.19 and 4.20 show the tip slope estimation errors observed with the Gaussian quadrature integration scheme when using point, rectangular and Bartlett sensors, respectively. It is interesting to note that the best performance is seen when using point sensors, for any number of sensors distributed along the structure. The Bartlett sensors yield the next lowest error, except for 10 sensors, where rectangular sensors provide the next lowest error. It is also evident that the general error levels

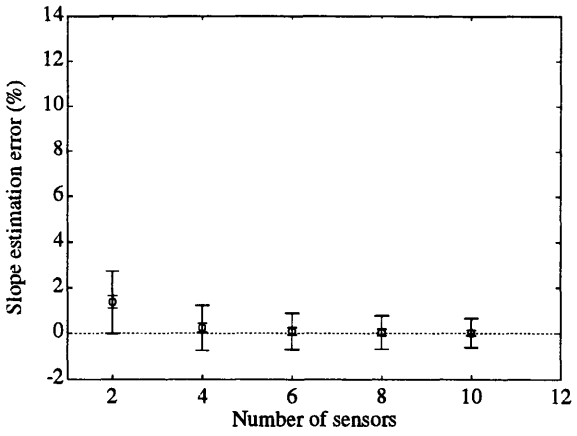

Figure 4.21: Accuracy of slope estimation using the Radau integration scheme and point sensors.

are very low compared to other integration schemes. This is especially evident for the point sensors, where the error for 2 or more sensors is below 1%. For point sensors, the absolute error due only to the integration rule is less than the standard deviation of the estimation error, for 2 or more sensors. For rectangular and Bartlett sensors, 4 or more sensors are required for this to be true. This shows that it is more beneficial to improve the accuracy of the sensors in order to lower the standard deviation of the estimation error, than to lower the absolute error introduced by the integration rule by increasing the number of sensors.

As for the other integration rules, the standard deviation of the estimation error is higher for point sensors than for rectangular and Bartlett sensors.

4.3.7 Radau Integration Scheme

Figures 4.21, 4.22 and 4.23 show the tip slope estimation errors exhibited by the Radau quadrature rule using point, rectangular and Bartlett sensors, respectively. For Radau quadrature, the lowest estimation errors are again obtained using point sensors. The next to lowest errors are obtained using Bartlett sensors. The absolute errors due only to the integration rule are approximately the same as for Gauss quadrature.

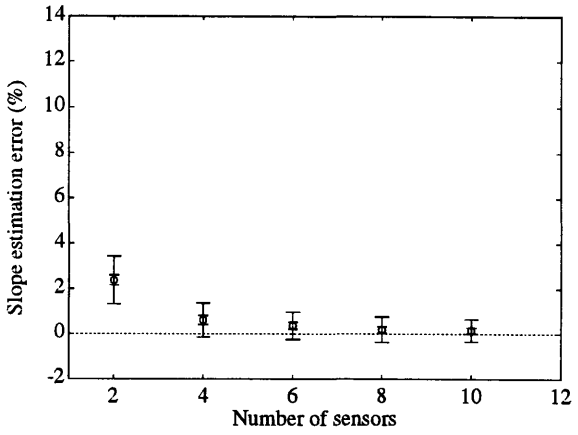

Figure 4.22: Accuracy of slope estimation using the Radau integration scheme and rectangular sensors.

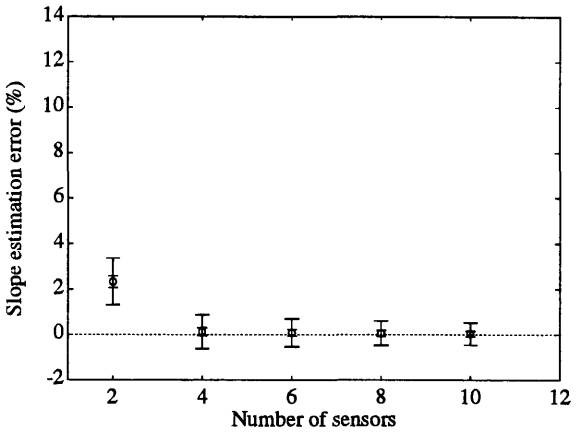

Figure 4.23: Accuracy of slope estimation using the Radau integration scheme and Bartlett sensors.

As with Gauss quadrature, the absolute error obtained with point sensors is smaller than the standard deviation of the estimation error for the realistic uncertainty level, even for just 2 sensors. Again, the accuracy of the individual sensors should be improved before using a large number of sensors in the sensor array. For rectangular and Bartlett sensors, this occurs for 4 or more sensors.

The standard deviation of the estimation error for the realistic uncertainty level is largest for point sensors, and approximately the same for rectangular and Bartlett

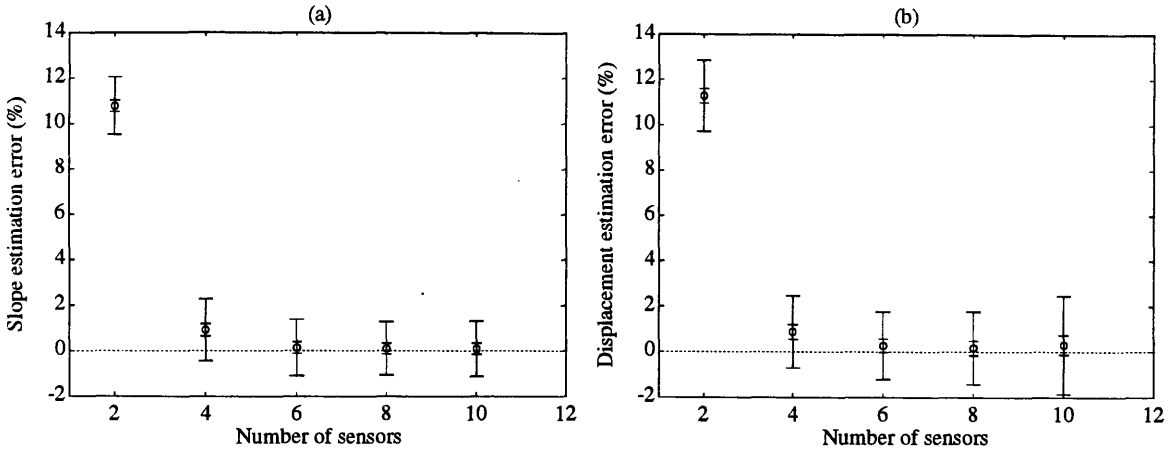

Figure 4.24: Accuracy of slope and displacement estimation using the Chebyshev integration scheme and point sensors; (a) slope estimation error; (b) displacement estimation error.

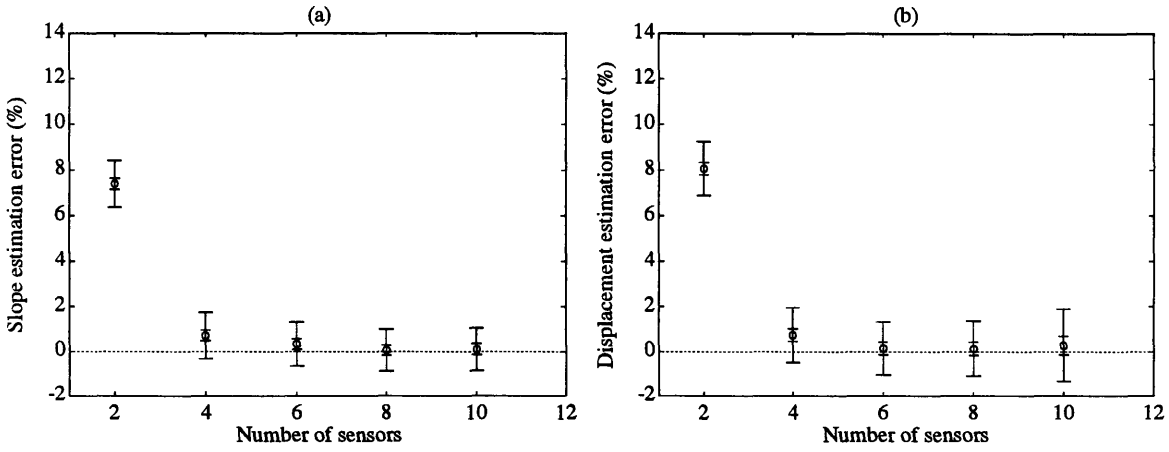

Figure 4.25: Accuracy of slope and displacement estimation using the Chebyshev integration scheme and rectangular sensors; (a) slope estimation error; (b) displacement estimation error.

sensors. However, the standard deviations for all three sensor types are smaller for the Radau quadrature scheme than they are for the Gauss quadrature rule.

4.3.8 Chebyshev Integration Scheme

Figures 4.24(a), 4.25(a) and 4.26(a) show the tip slope estimation errors exhibited by the Chebyshev interpolation integration rule when using point, rectangular and

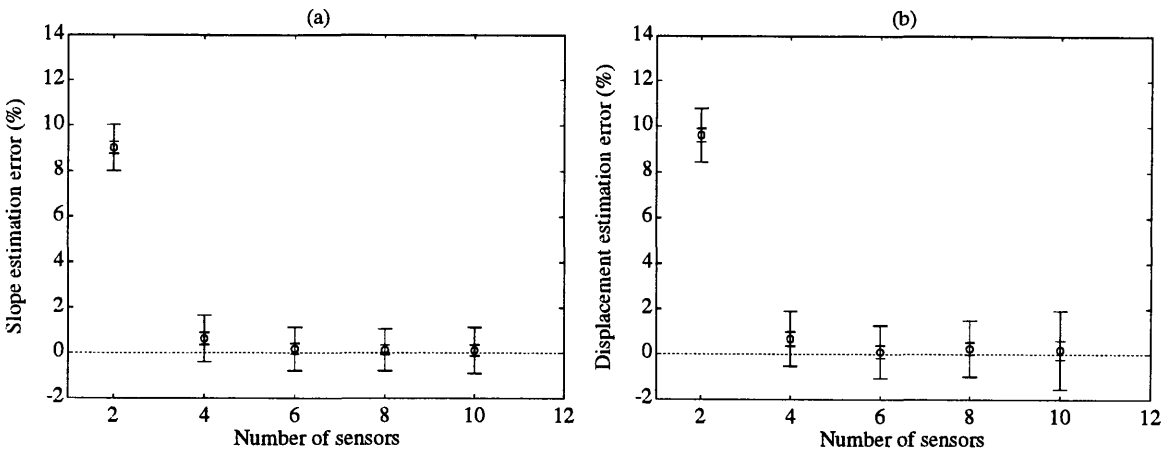

Figure 4.26: Accuracy of slope and displacement estimation using the Chebyshev integration scheme and Bartlett sensors; (a) slope estimation error; (b) displacement estimation error.

Bartlett sensors, respectively. For 2 sensors, rectangular sensors yield the lowest errors, while for intermediate and large numbers of sensors, point sensors give the lowest estimation errors. The estimation error is less than 1% for 4 sensors or more for all three sensor types. The estimation error is also less than the standard deviation of the error for realistic uncertainty levels. Thus, to improve the performance of a real system with 4 sensors, the accuracy of the sensors should be improved before additional sensors are added.

The standard deviation of the estimation error is smaller for the rectangular and Bartlett sensors than for the point sensors. Also, for all three sensor types, the standard deviation of the estimation error is larger for 10 sensors than for 8 sensors. This could be due to the fact that the interpolating polynomial is of such high order that its shape is very sensitive to small changes in the positions of the data points it interpolates.

Figures 4.24(b), 4.25(b) and 4.26(b) show the tip deflection estimation errors observed by the Chebyshev integration rule when using point, rectangular and Bartlett sensors, respectively. As for the problem of slope estimation, when employing 2 sen-

sors, the lowest errors are found to occur when using rectangular sensors. However, for nearly all other cases, it is the Bartlett sensors that give the lowest errors. Convergence for the integration scheme is fast, and the absolute error due only to the integration scheme is smaller than the standard deviation of the estimation error, σ_r, for 4 or more sensors. This standard deviation is higher for point sensors than for rectangular or Bartlett sensors. It is interesting to see the marked increase in the standard deviation of the estimation error as the number of sensors is increased from 8 to 10. As for the case of slope estimation, this is probably due to the high sensitivity of the high order interpolating polynomial to the position of the data point to be interpolated.

4.3.9 Global Static Shape Function Rule

Figures 4.27 and 4.28 show the tip deflection estimation errors observed when using the Global Static Shape Function integration rule with point, rectangular and Bartlett sensors. As with most of the other schemes, for 2 sensors, the lowest tip deflection estimation error, which is less than 1%, is observed using rectangular sensors. This error decreases rapidly as the number of sensors is increased, and is identically zero for 8 or more sensors. This is because the shape functions used correspond exactly to the 7 deflection shapes exhibited by the beam under application of the 7 static loadings shown in Figure 4.2. For 4 or more sensors, the next to lowest errors are obtained with Bartlett sensors, which exhibit errors which are higher than those for point sensors, but lower than those for rectangular sensors.

In general the standard deviation of the tip deflection estimation error is much lower than for all the integration schemes introduced earlier, for all three sensor types. This indicates that the Global Static Shape Function integration scheme is much more robust to uncertainties in sensor gage factor and placement. The standard deviation of the error generally decreases as more sensors are added, indicating that sensor

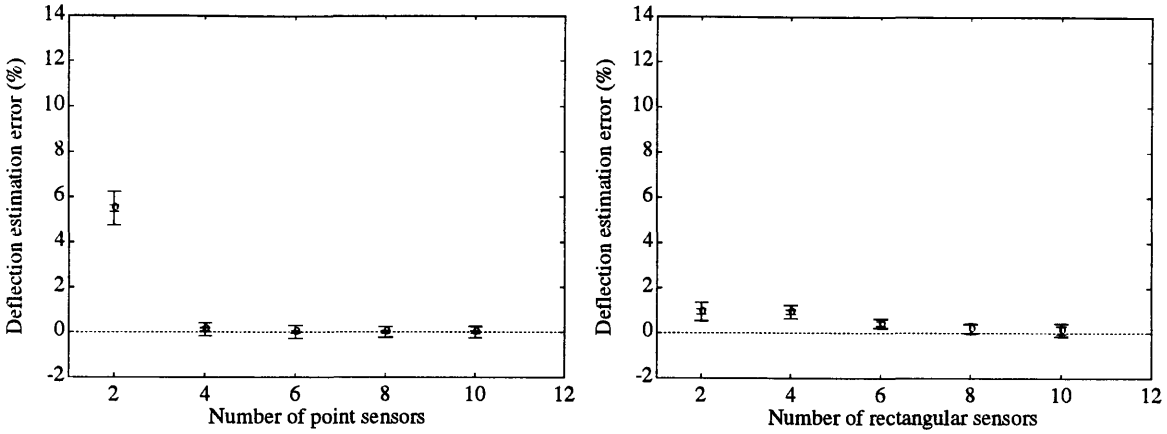

Figure 4.27: Accuracy of displacement estimation using the Global Static Shape Function integration scheme with point and rectangular sensors.

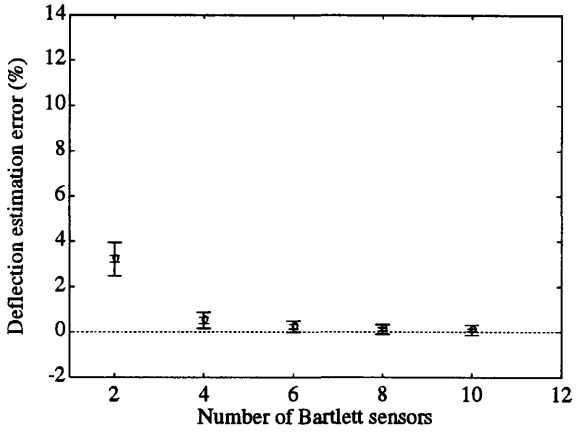

Figure 4.28: Accuracy of displacement estimation using the Global Static Shape Function integration scheme with Bartlett sensors.

uncertainties impact the displacement estimate less as more sensors, and consequently more shape functions, are used.

4.3.10 Global Dynamic Shape Function Rule

Figures 4.29 and 4.30 show the tip deflection estimation errors observed when using the Global Dynamic Shape Function integration rule with point, rectangular and Bartlett sensors. The general trends in the tip deflection estimation error are similar to those for the Global Static Shape Function integration rule. For 2 sensors,

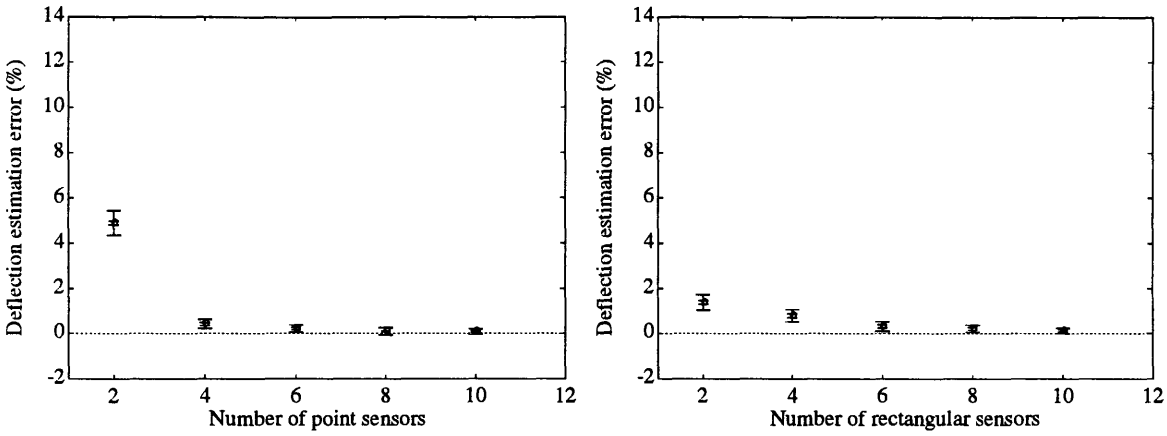

Figure 4.29: Accuracy of displacement estimation using the Global Dynamic Shape Function integration scheme with point and rectangular sensors.

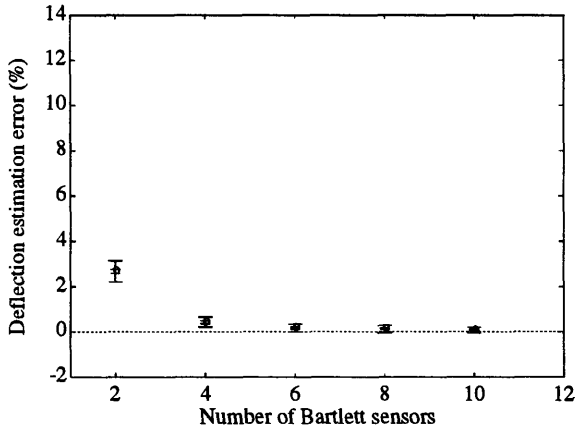

Figure 4.30: Accuracy of displacement estimation using the Global Dynamic Shape Function integration scheme with Bartlett sensors.

the lowest error can be obtained with rectangular sensors. This error, although larger than for the GSSF integration scheme, is less than 2%. For more than 2 sensors, the lowest estimation errors are obtained when using point sensors. The errors are very low, and show that even the dynamic beam mode shapes can be used to fit to sensor measurements obtained under static beam loading. In fact, the first dynamic clamped-free beam mode is very similar to the static beam deflection shape for a tip or distributed load. As with the GSSF scheme, the standard deviation of the tip displacement estimation error for the Global Dynamic Shape Function scheme is

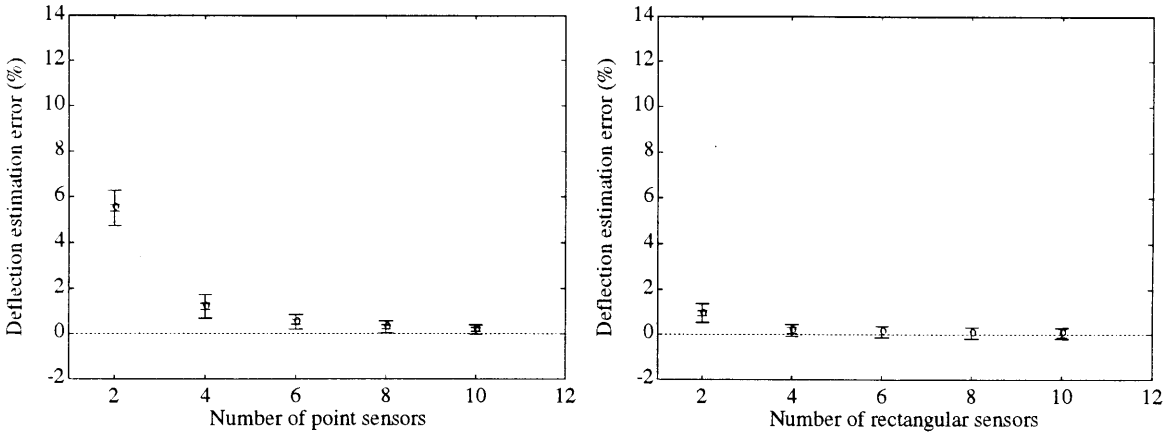

Figure 4.31: Accuracy of displacement estimation using the Least Squares Global Static Shape Function integration scheme with point and rectangular sensors, and two shape functions.

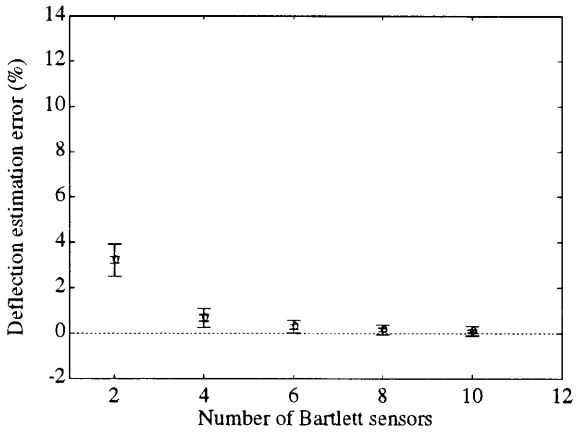

Figure 4.32: Accuracy of displacement estimation using the Least Squares Global Static Shape Function integration scheme with Bartlett sensors, and two shape functions.

much lower than for any of the other integration schemes. For 2 sensors, the lowest standard deviation in the error is obtained with rectangular sensors, while for more than 2 sensors, the standard deviations for the three sensor types differ little.

4.3.11 Least Squares Global Static Shape Function Rule

Figures 4.31 and 4.32 show the tip deflection estimation errors observed when using the Least Squares Global Static Shape Function integration rule with point, rectangular and Bartlett sensors, and two shape functions. For 2 sensors, the Least Squares Global Static Shape Function 2 scheme yields exactly the same tip displacement estimation error as the GSSF scheme since both fit the same two global static shape functions to the outputs of two identically placed outputs. However, for more than two sensors, the LSGSSF-2 scheme fits the same two shape functions to the outputs of the sensors in a least squares sense. Although this means that the scheme has limited flexibility in trying to fit the two shape functions to the sensor outputs, we find that it offers very good performance for 4 or more sensors. Rectangular sensors yield the lowest errors, which are less than 1% for 4 or more sensors. Point and Bartlett sensors also yield errors less than 1%, but require 6 or more sensors to do so. The LSGSSF-2 scheme also shows increased robustness to sensor uncertainties, compared with integration schemes that do not fit structural shape functions to sensor outputs.

Figures 4.33 and 4.34, show the tip deflection estimation errors observed when using the Least Squares Global Static Shape Function integration rule with point, rectangular and Bartlett sensors, and four shape functions. For 4 sensors, the Least Squares Global Static Shape Function 4 scheme yields exactly the same tip displacement estimation error as for the regular GSSF scheme. For point sensors, excellent performance is achieved with this scheme. Performance is slightly worse when rectangular or Bartlett sensors are used, but the estimation errors are less than 1% for any number of sensors. The standard deviations of the estimation errors are comparable to those obtained with the other structural shape function fitting schemes, and significantly smaller than for the other integration rules.

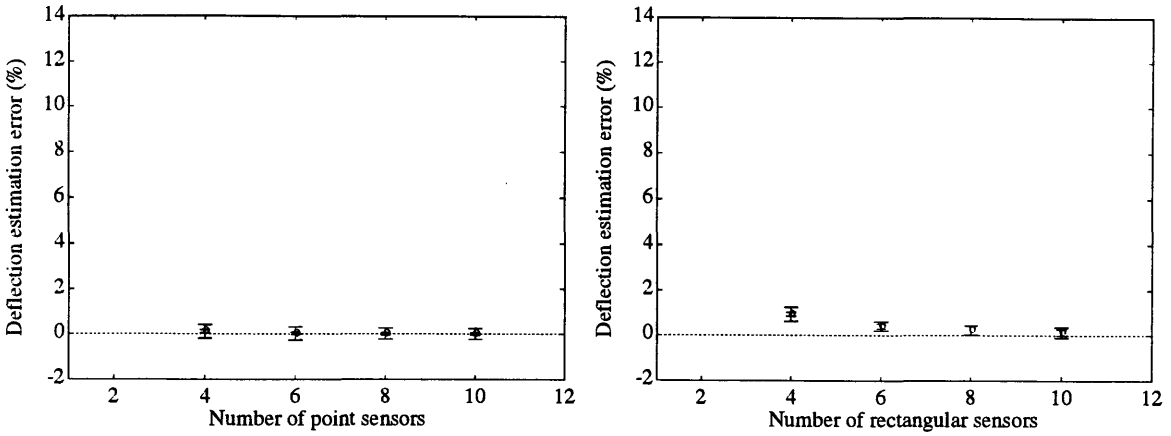

Figure 4.33: Accuracy of displacement estimation using the Least Squares Global Static Shape Function integration scheme with point and rectangular sensors, and four shape functions.

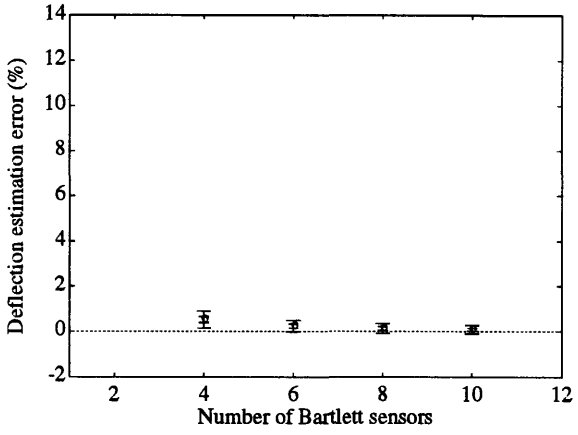

Figure 4.34: Accuracy of displacement estimation using the Least Squares Global Static Shape Function integration scheme with Bartlett sensors, and four shape functions.

4.4 Conclusions

In general, relatively few sensors are required to obtain good estimation performance with the selected set of integration schemes. It is to be noted, however, that the static shapes being estimated have long wavelengths compared to the length of the structure. All of the schemes have the property that the estimation error converges quite quickly towards zero. Table 4.1 shows the minimum number of sensors

Table 4.1: Minimum number of sensors required to insure that the absolute error due only to the integration rule error is less than 1%. Note that the Gauss and Radau rules cannot be used to obtain deflection estimates.

Integration Rule	Slope Estimation			Deflection Estimation		
	Point Sensors	Rectangular Sensors	Bartlett Sensors	Point Sensors	Rectangular Sensors	Bartlett Sensors
Midpoint	6	8	8	8	8	8
Trapezoidal	6	8	8	6	6	4
Simpson's	5	5	5	5	5	5
Spline	4	4	4	4	4	4
B-Spline	8	8	10	8	8	8
Gauss	2	4	4	N/A	N/A	N/A
Radau	4	4	4	N/A	N/A	N/A
Chebyshev	4	4	4	4	4	4
GSSF	-	-	-	4	2	4
GDSF	-	-	-	4	4	4
LSGSSF-2	-	-	-	4	2	4
LSGSSF-4	-	-	-	4	4	4

required for the absolute error due only to the integration schemes to decrease below 1%. It is seen that the GSSF and LSGSSF-2 schemes exhibit tip displacement estimation errors less than 1% using only 2 sensors. As more accurate schemes are used and their convergence rates increase, the minimum number of sensors required to make the absolute error due only to the integration scheme less than the standard deviation of the estimation error, decreases dramatically. This is seen in Table 4.2 for the standard deviation of the error due to realistic uncertainty levels, σ_r.

The Gauss and Radau schemes have good accuracy, but they both require exact

Table 4.2: Minimum number of sensors required to insure that the absolute error due only to the integration rule error less than one standard deviation due to realistic uncertainty levels. Note that the Gauss and Radau rules cannot be used to obtain deflection estimates.

Integration Rule	Slope Estimation			Deflection Estimation		
	Point Sensors	Rectangular Sensors	Bartlett Sensors	Point Sensors	Rectangular Sensors	Bartlett Sensors
Midpoint	4	8	8	6	8	8
Trapezoidal	6	8	8	4	6	6
Simpson's	5	5	5	5	5	5
Spline	4	4	4	4	4	4
B-Spline	8	8	10	6	8	8
Gauss	2	4	4	N/A	N/A	N/A
Radau	2	4	4	N/A	N/A	N/A
Chebyshev	4	4	4	4	4	4
GSSF	-	-	-	4	8	8
GDSF	-	-	-	8	10	8
LSGSSF-2	-	-	-	10	4	8
LSGSSF-4	-	-	-	4	10	6

placement of sensors, and cannot be used to obtain a second integral, as they do not provide running values of the integral. In addition, the extrapolation of these schemes to more complicated domains is problematic. Thus the Gauss and Radau rules are excluded from further consideration as practical integration schemes.

The second order B-spline rule in general has lower accuracy than the cubic interpolating spline. Since the only advantage it offers is that it is simpler to compute than the cubic spline, it will also be excluded from further consideration.

The trapezoidal rule is noticeably better than the midpoint rule on the second integration step, for displacement estimation, although they both offer similar performance for slope estimation. Therefore, we retain the trapezoidal rule as the favored of the two, but do not discard midpoint from further consideration.

Simpson's rule outperforms the trapezoidal rule in slope and deflection estimation. However, Simpson's rule has constraints on the number and placement of sensors which make generalization problematic. Thus Simpson's rule is also excluded from further consideration.

The global shape function fitting integration schemes offer excellent performance because they fit displacement shapes that correspond to actual beam deformation shapes, and thus incorporate the root and tip boundary conditions automatically. In addition, there are no sensor placement requirements, and root and tip measurements are not required.

The exclusion of some of the integration schemes leaves the trapezoidal, cubic interpolating spline, Chebyshev and global shape function fitting rules. For these remaining rules, only about 4 sensors are required for the absolute error due only to the integration scheme to fall below 1% for both slope and deflection estimation.

For the sensor uncertainty, it is seen that in general, it affects all the ordinary integration rules in approximately the same manner. For the global shape function schemes, however, the effect on the tip deflection estimation error is reduced significantly. In general there is a very slight decrease in the standard deviation of the estimation error due to optimistic and realistic uncertainty levels as the number of sensors is increased. However, there is no consistent or noticeable change in the standard deviation of the error from rule to rule.

For the trapezoidal, cubic spline and Chebyshev integration schemes, only about 4 sensors are required to guarantee that the estimation error due only to the integration

rule is comparable to the standard deviation of the estimation error due to realistic uncertainties in the sensors.

It can be seen that there is some sensitivity of the standard deviation of the estimation error to the spatial weighting of the sensors. The standard deviation of the error is in fact smaller when using the rectangular and Bartlett (triangular) sensors than when using the point sensors. The strain-averaging properties of the rectangular and Bartlett sensors filter out the relatively high spatial frequency gage factor and sensor position uncertainty effects.

To achieve the best accuracy, for small numbers of sensors (2), it is advisable to use rectangular sensors. For an intermediate number of sensors (4 or 6), the accuracy is approximately the same for all three sensor types. At higher numbers of sensors (8 or 10), the point sensors offer the best accuracy, although they suffer from higher standard deviations in the estimation error due to sensor uncertainties.

In summary, 4 sensors are in general enough to achieve good performance, with tip slope and deflection estimation errors well under 5%. In fact, the estimation errors for 4 sensors are actually under 1% for the cubic spline, Gauss, Radau, Chebyshev and all global shape function fitting schemes. Thus the estimation error quickly drops below σ_r, one standard deviation of the estimation error due to realistic uncertainty levels.

4.5 Beam Experiment

An experiment was conducted to experimentally correlate with the results of the simulations conducted and discussed above. Sets of distributed, surface mounted resistive foil strain gages were used to provide discrete curvature measurements along the length of two cantilevered aluminum beams. The performance of various numerical integration algorithms in predicting the tip displacement was investigated. In

addition, theoretical curvature values with simulated measurement uncertainties are calculated and used to estimate tip displacement and the standard deviation of the tip displacement error due to measurement uncertainties. It is then possible to see whether the experimental errors encountered fall within one standard deviation of the absolute estimation error due only to the integration rule as found by numerical simulation. The more realistic assumptions of 5% gage factor uncertainty and 1% gage position uncertainty were assumed. This corresponds to the more realistic uncertainty level described earlier. Midpoint, trapezoidal, Simpson, spline, and Chebyshev integration methods were used to process both the simulated measurements and experimental curvature data since they were identified in Section 4.4 above as the ones to continue to examine.

Two thin beams of 2024 aluminum were fabricated for these tests. The beams were 33.35 cm long (13.2 inches), 2.54 cm wide (1 inch), and 0.787 mm thick (0.031 inches). They were divided into five sections of equal length, and each had a strain gage mounted at the center of each section. Gages were mounted symmetrically on both sides of the beams. One beam used very short gages which simulated point measurements. These gages were WA series 500 GB resistive foil gages with a grid 15.75 mm long (0.62 inches). The other beam used much longer gages which simulated measurements made by a rectangular sensor. These gages were 20 CBW resistive foil gages with a grid 2.25 inches long.

The beams were clamped in a machined block, and in the first case a tip displacement was *applied* using a micrometer. A schematic for this test is shown in Figure 4.35 for the beam with short sensors. In the second case a tip displacement was again applied, but the midspan of the beam was constrained to have zero deflection. A schematic for this test is shown in Figure 4.36 for the beam with long sensors. Tip displacements of 7.62 mm (0.3 inches), 15.24 mm (0.6 inches), and 22.86 mm (0.9 inches) were applied to the beams for both configurations. Each gage on the top

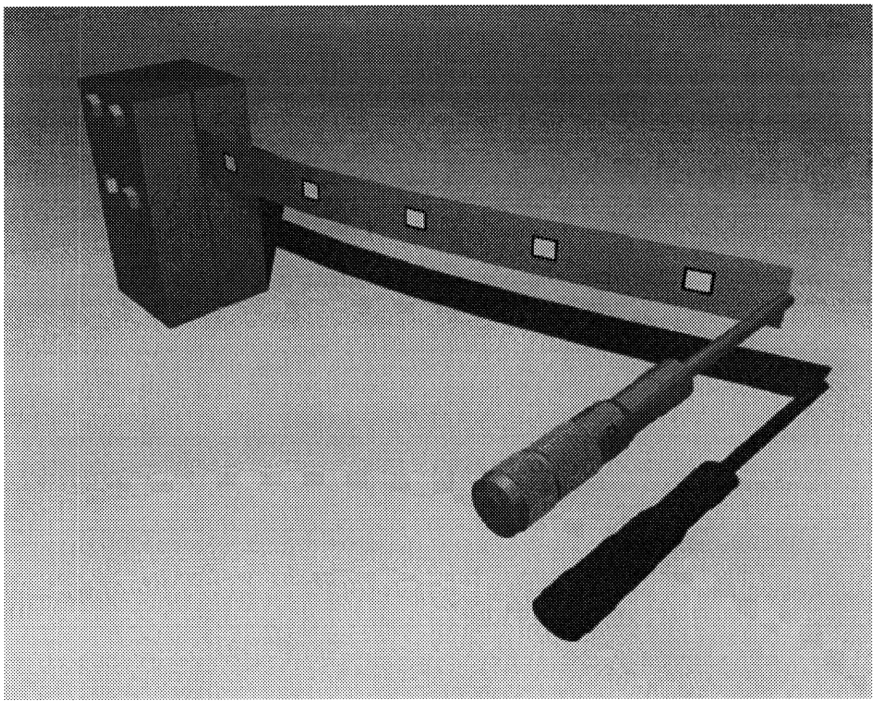

Figure 4.35: Beam with short gages, under applied tip deflection.

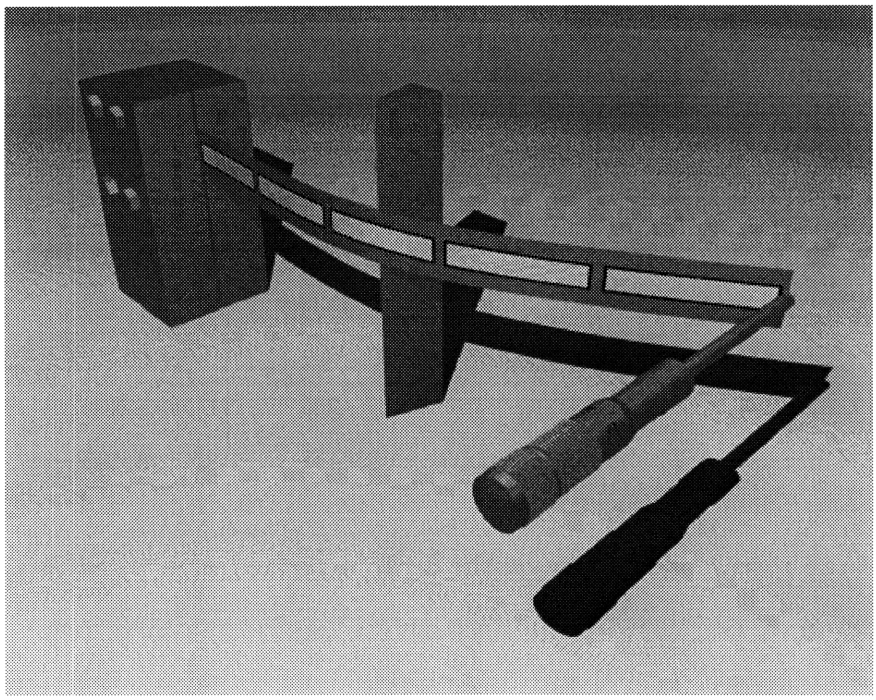

Figure 4.36: Beam with long gages, under applied tip deflection with midspan constraint.

Table 4.3: Experimental error, simulated absolute error and standard deviation of the error in tip displacement estimation for a cantilevered beam with point gages for an applied tip displacement.

Integration Scheme	Tip Displacement		
	Simulated Absolute Error	Simulated Standard Deviation	Experimental Error
Midpoint	1.44%	2.06%	3.08%
Trapezoidal	0.84%	1.82%	2.46%
Simpson	0.00%	1.76%	1.65%
Spline	0.01%	1.87%	1.66%
Chebyshev	0.01%	1.82%	1.67%

Table 4.4: Experimental error, simulated absolute error and standard deviation of the error in tip displacement estimation for a cantilevered beam with point gages for an applied tip displacement with midspan constraint.

Integration Scheme	Tip Displacement with midspan constraint		
	Simulated Absolute Error	Simulated Standard Deviation	Experimental Error
Midpoint	-8.37%	3.74%	-10.45%
Trapezoidal	-0.49%	2.83%	-1.99%
Simpson	-10.84%	3.10%	-12.29%
Spline	-5.07%	2.63%	-6.89%
Chebyshev	-3.93%	2.84%	-6.49%

surface and the corresponding one on the bottom surface were wired in a half bridge configuration.

The experimental outputs of the strain gages yielded measurements of curvature along the length of the beam. This data was integrated twice numerically in order to

Table 4.5: Experimental error, simulated absolute error and standard deviation of the error in tip displacement estimation for a cantilevered beam with rectangular gages for an applied tip displacement.

Integration Scheme	Tip Displacement		
	Simulated Absolute Error	Simulated Standard Deviation	Experimental Error
Midpoint	-0.08%	2.40%	-3.42%
Trapezoidal	0.76%	2.68%	-0.21%
Simpson	-0.03%	2.57%	-1.30%
Spline	-0.01%	2.54%	-1.38%
Chebyshev	-0.02%	2.74%	-1.66%

Table 4.6: Experimental error, simulated absolute error and standard deviation of the error in tip displacement estimation for a cantilevered beam with rectangular gages for an applied tip displacement with midspan constraint.

Integration Scheme	Tip Displacement with midspan constraint		
	Simulated Absolute Error	Simulated Standard Deviation	Experimental Error
Midpoint	-25.81%	3.85%	-29.51%
Trapezoidal	5.88%	3.56%	19.15%
Simpson	0.76%	3.18%	9.14%
Spline	2.49%	2.98%	10.68%
Chebyshev	-3.38%	4.00%	13.24%

estimate the tip displacement. The *experimental error* is defined as the error between this estimate and the actual applied tip displacement.

The theoretical outputs of the gages were calculated by knowing the applied tip displacement and the beam configuration. Simulated gage factor uncertainties of up to $\pm 5\%$ and gage placement uncertainties of up to $\pm 1\%$ of the length of the beam were used. The errors due to gage factor and gage position uncertainties are combined with the theoretical outputs of the gages. These were integrated numerically and the simulated tip displacement was calculated. The *simulated error* is defined as the error between the simulated experimental tip displacement and the actual tip displacement.

Results comparing displacement prediction errors of integrating the experimental curvature data to the error bounds simulated theoretically are shown for the beam with point gages in Table 4.3 for the simple cantilevered beam, and in Table 4.4 for the cantilevered beam with its midspan constrained to have zero deflection. Tables 4.5 and 4.6 show the corresponding information for the beam with rectangular gages.

The experimental errors are compared with the simulated error bounds to see whether the experimental errors actually encountered fell within one standard deviation of the modeled absolute error. It can be seen from Table 4.3, that for point gages and tip displacement, this is the case for all integration schemes. In Table 4.4, we see that for point gages and tip displacement with midspan constrained, the experimental errors are again bounded by the simulated errors.

For the rectangular gages and tip displacement, we see in Table 4.5 that the simulated errors bound the experimental ones for all but the Midpoint rule. For rectangular gages and tip displacement of the beam with its midspan constrained, we see from Table 4.6 that the experimental errors are bounded by the simulated ones for the Midpoint rule only. The simulated errors badly underestimate the experimentally observed errors for the other four rules.

Since the beams are very thin (0.787 mm), it is very difficult to accurately estimate the distance between the elastic axis of the beam and the foil of the gage. In fact, an error of one thousandth of an inch in distance from centerline generates an error of approximately 6.5% in measured curvature. Errors of this magnitude are easy to introduce because of the uncertainty concerning the height of the adhesive layer between the surface of the beam and the gages. Additional errors can be introduced due to uncertainty about the distance between the bottom surface of the gage and the location of the foil.

In conclusion, it has been shown that the observed experimental tip deflection estimation errors are in most cases within one standard deviation of the absolute estimation error due only to the integration rule used. This absolute estimation error was calculated by integrating the theoretical outputs of the sensors and introducing gage factor and sensor position uncertainties. These uncertainties model those which are likely to be present in any physical implementation.

Chapter 5

Estimation of the Dynamic Mode Shapes of Beams

In this chapter, the performance of an array of spatially averaging strain sensors in estimating the dynamic mode shapes of a pinned-pinned and a clamped-free beam is investigated. This is accomplished by simulating the theoretical strain measurements made by the sensors. In all the simulations it is assumed that there are 9 sensors equally spaced in the interior of the beam, such that the i-th sensor is centered around the point $iL/10$ where L is the length of the beam. Additional sensors near the root are employed for the clamped-free beam in order to perform more accurate root curvature estimation.

5.1 Simulation Results for a Pinned-Pinned Beam

The performance of a sensor array mounted on a pinned-pinned beam with purely sinusoidal mode shapes is investigated first. A typical sensor array is shown in Figure 5.1. It is assumed that there are 9 equally spaced sensors in the array, and that boundary sensors are truncated by the folding method described in Chapter 2. It was

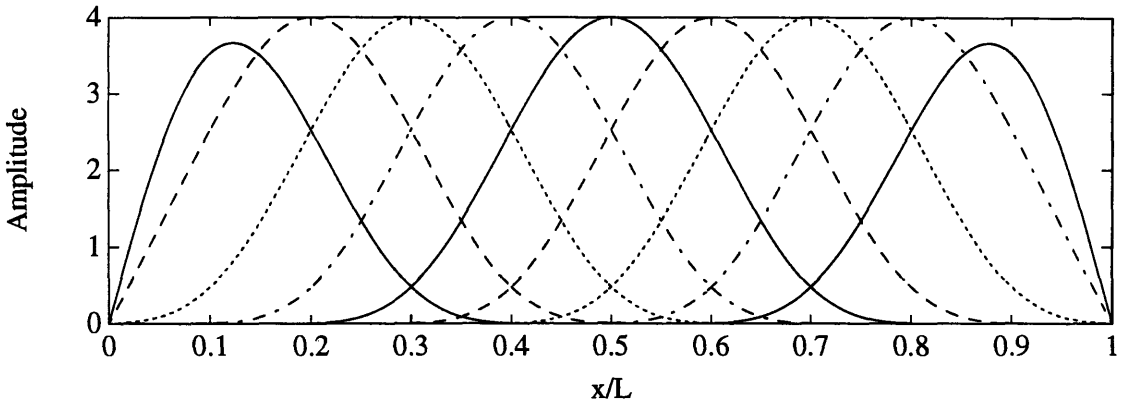

Figure 5.1: A typical array of 9 sensors mounted on a pinned-pinned beam.

shown that sensors truncated in this fashion retain the properties of the corresponding untruncated sensor because the strain field for any mode of a pinned-pinned beam is purely sinusoidal and the point of symmetry of the strain field is at the root of the beam. The lengths of the sensors are all equal and set such that they begin to roll off at the spatial frequency corresponding to the third dynamic mode of the beam, except the Gauss-Hanning sensor, whose length was set such that it began to roll off at the fourth mode.

The displacement, slope and curvature fields for the dynamic modes of a pinned-pinned beam are given by:

$$w(x) = \frac{L}{m\pi} \sin \frac{m\pi x}{L} \tag{5.1}$$

$$\frac{dw(x)}{dx} = \sin \frac{m\pi x}{L} \tag{5.2}$$

$$\kappa(x) = \frac{m\pi}{L} \sin \frac{m\pi x}{L} \tag{5.3}$$

where m is the mode number, L is the beam length, $w(x)$ is the deflection, $dw(x)/dx$ is the slope, and $\kappa(x)$ is the curvature of the beam. The slope is normalized to be unity at the root of the beam for any mode number m.

The trapezoidal, Simpson, cubic spline and Chebyshev integration schemes were

used to integrate curvature measurements made by the sensors in order to estimate the tip deflection, which should be zero for all modes. It is assumed that the curvature is zero at the root and tip of the beam. The initial slope of unity at the root of the beam is supplied to each integration scheme as an initial condition. The *error* between the estimated tip deflection and the actual displacement (zero) is found. This error is given by:

$$\mathcal{E} = \frac{\tilde{w}_{\text{tip}}}{L} \tag{5.4}$$

where $\tilde{w}_{\text{tip}}$ is the estimated tip deflection. This error should be as close to zero as possible for low mode numbers indicating that accurate shape estimation is being performed. As the mode number is increased, however, this error should approach 100%. This is because as the mode number of the beam is increased, the outputs of the sensors should approach zero because their outputs should be rolling off with the increasing spatial frequency. Thus each integration scheme will only integrate the unity initial slope condition given at the root of the beam, and it will obtain $\tilde{w}_{\text{tip}} = L$. The effective *observability* of the sensor array is defined as:

$$\mathcal{O} = 1 - \mathcal{E} = 1 - \frac{\tilde{w}_{\text{tip}}}{L} \tag{5.5}$$

The observability should be unity for low mode numbers m, and approach zero as the mode number is increased, due to the rolloff exhibited by the sensors.

5.1.1 Point Sensors

Figure 5.2 shows the tip deflection estimation error as a percentage of the length of the beam for an array of point sensors, using the trapezoidal, Simpson, cubic spline and Chebyshev integration schemes to integrate the sensor measurements to obtain an estimate of the tip displacement. Figure 5.3 shows the observability as defined in (5.5) for the case of trapezoidal integration.

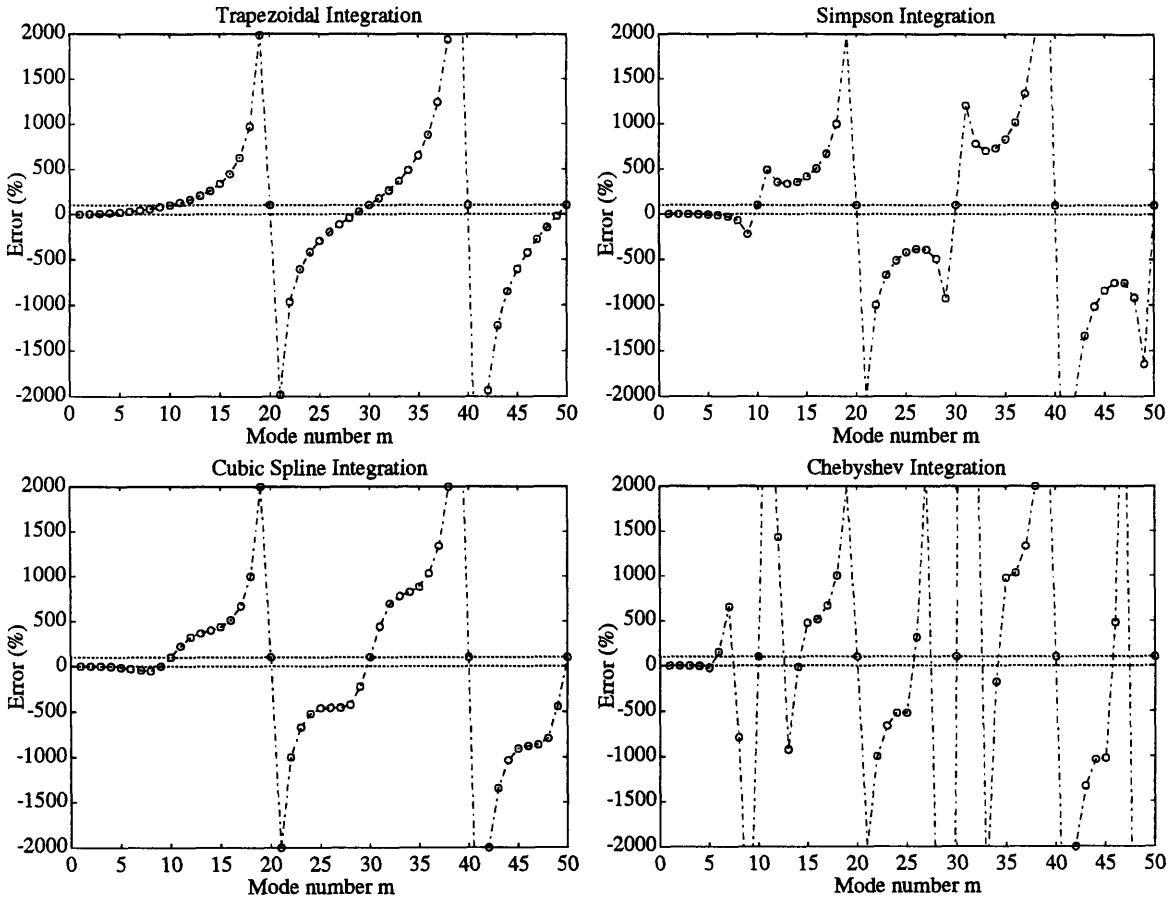

Figure 5.2: Tip deflection estimation error for pinned-pinned beam using point sensors with trapezoidal, Simpson, spline or Chebyshev integration.

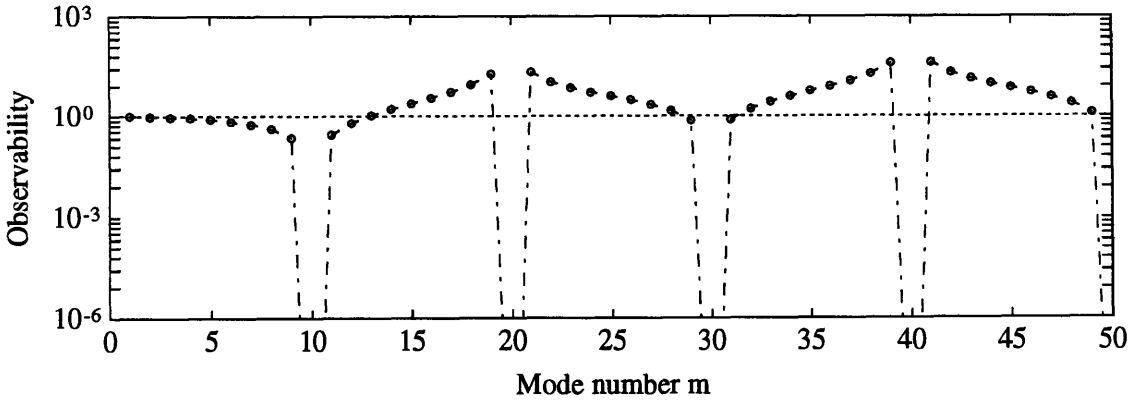

Figure 5.3: Sensor array observability for pinned-pinned beam using point sensors with trapezoidal integration.

For trapezoidal integration, for low mode numbers, the tip deflection estimation error is very close to zero, for all the other integration schemes. This behavior satisfies Functional Requirement #4 by accurately observing the low order dynamic mode shapes. The observability is correspondingly near unity for low mode numbers. As the mode number is increased, the error approaches 100% and the observability decreases. When the mode number is increased to 10, spatial aliasing of the sensors occurs, and they each exhibit zero output. Thus the observability drops to zero, and the error is 100%. As the mode number is increased beyond this aliasing frequency, the error increases dramatically to approximately 2000% at mode 19. At mode 20, twice the aliasing frequency, aliasing again occurs, and all sensor outputs are zero. This behavior is exhibited at every mode number which is an integer multiple of the aliasing mode number.

For Simpson's rule, the low order modes are again estimated accurately. As the mode number is increased, however, the estimation error begins to decrease. For mode 9, the error is less than -200%, and for the aliasing frequency of mode 10, it is exactly 100% because all sensor outputs are identically zero. Beyond the local maximum of 490% in the error at mode 11, the error decreases slightly, reaching a local minimum at mode 13. The error then rises steeply, reaching about 2000% at mode 19, as with trapezoidal integration. For higher mode numbers, the error envelope increases, and the swings between positive and negative errors that occur at multiples of twice the aliasing frequency become more severe.

For the cubic spline integration scheme, the low mode number behavior seen with trapezoidal and Simpson integration is detected. For larger errors, the behavior is similar to Simpson's rule, except that the sharp swings at modes 9 and 11, and modes 29 and 31 have been smoothed. The error envelope is practically identical to the trapezoidal and Simpson schemes.

For the Chebyshev scheme, we see good estimation for modes 1 through 5, with

the error falling to about -25% at mode 5. As the mode number is increased to 7, the error increases to about 650%. For even higher modes, the error swings wildly about zero, clearly yielding the most unstable output of the four integration schemes investigated.

In comparing the four integration schemes, it is clear that the most stable error behavior is obtained with the trapezoidal scheme. It can also be seen that the long-term behavior of the observability envelope shows an increase in magnitude which is linear with mode number. Equation (5.3) shows that the maximum curvature also increases linearly with mode number m. It is therefore clear that the outputs of the individual sensors are not decreasing as the spatial frequency of the strain field is increased. This behavior violates Functional Requirements #5 and #6.

5.1.2 Rectangular Sensors

Figure 5.4 shows the tip deflection estimation error for an array of rectangular spatially filtering sensors, using the trapezoidal, Simpson, cubic spline and Chebyshev integration schemes. Figure 5.5 shows the observability of the sensor array for the trapezoidal scheme only.

For the trapezoidal integration scheme, as in the case of point sensors, the error is close to zero for low mode numbers, but increases more rapidly, reaching 100% at only mode 5. Beyond this mode number, the error overshoots 100% slightly, but returns to approximately 100% by mode 10. This is shown in the plot of observability, as it decreases to about 10^{-3} at mode 10. In contrast to the point sensors, the observability does not drop to zero at this aliasing frequency because the sensors are reporting a filtered strain measurement from a wider area of the structure. However, unstable behavior can be seen in the error at modes 21, 39 and 40, where the magnitude of the error is very large.

For Simpson's rule, we see behavior similar to that obtained with the trapezoidal

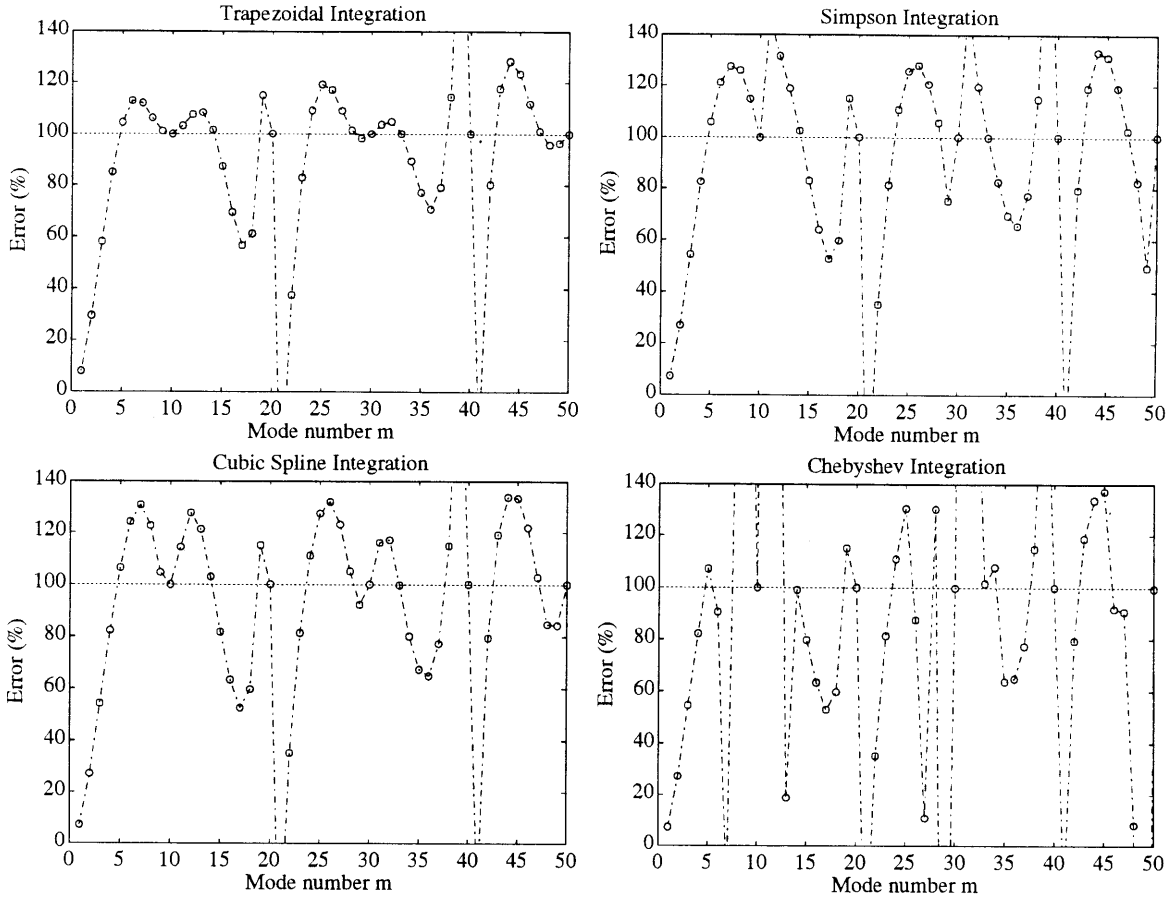

Figure 5.4: Tip deflection estimation error for pinned-pinned beam using rectangular sensors with trapezoidal, Simpson, spline or Chebyshev integration.

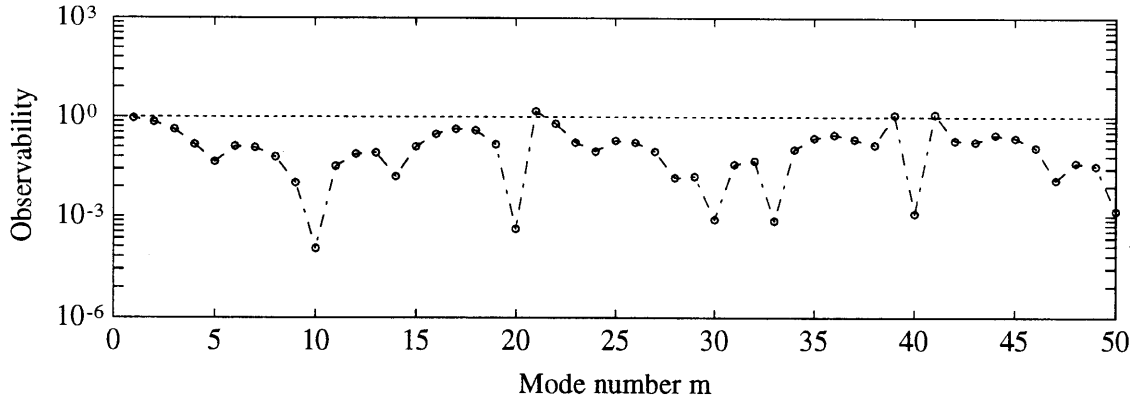

Figure 5.5: Sensor array observability for pinned-pinned beam using rectangular sensors with trapezoidal integration.

error, but many swings above and below the 100% line are magnified by up to a factor of 2. It is evident that aliasing again occurs for mode numbers that are a multiple of 10, and that performance is not acceptable.

The error obtained with the cubic spline integration scheme shows behavior similar to that obtained with trapezoidal and Simpson integration. The swings in the error are larger in magnitude than for trapezoidal integration, but less than those exhibited by Simpson's rule.

For Chebyshev integration, we see unstable behavior similar to that obtained with point sensors. The swings in the error are much larger than for the other three integration schemes.

In comparing the four integration schemes, it is again clear that the most stable error behavior is obtained with the trapezoidal scheme. It can also be seen that the long-term behavior of the observability envelope for shows that the magnitude remains approximately constant with mode number. The trend of the error is clearly not monotonic towards 100% for any integration scheme, and so Functional Requirement #6 is violated. In addition, Functional Requirement #5 is also violated because the observability does not roll off as the mode number is increased. However, the results are an improvement over the performance of an array of point sensors.

5.1.3 Bartlett Sensors

The array of Bartlett (triangular) sensors used is shown in Figure 5.6. Figure 5.7 shows the error in tip deflection estimation for this sensor array, with trapezoidal, Simpson, cubic spline and Chebyshev integration schemes used to estimate beam displacement. Figure 5.8 shows the observability of the array in the case of trapezoidal integration.

In the case of trapezoidal integration, as with point and rectangular sensors, the

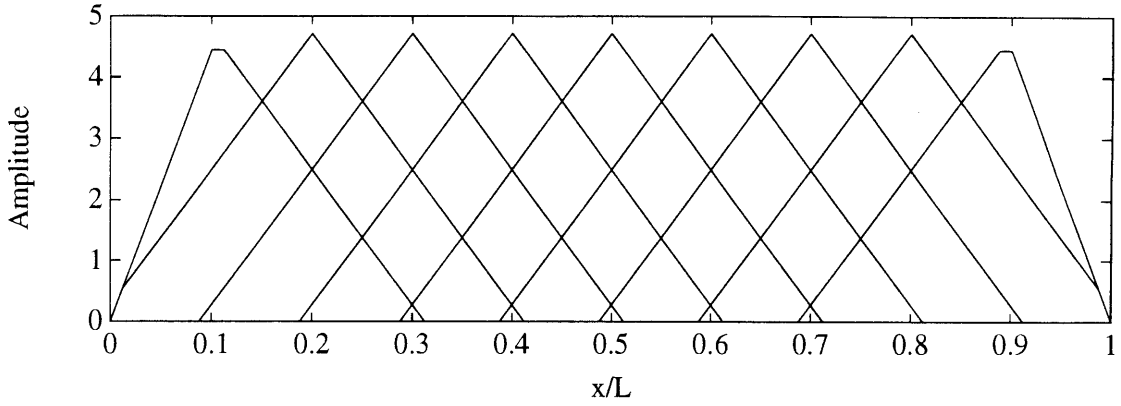

Figure 5.6: Sensor weightings for Bartlett sensors mounted on a pinned-pinned beam.

error is small for low mode numbers. It then increases for larger mode numbers, and reaches 100% at mode number 10. However, this increase in error is significantly slower than for rectangular sensors and trapezoidal integration. This is shown in the behavior of the observability, which is seen to drop off to below 10^{-6} at mode 10. Beyond mode 10, the error remains reasonably close to 100%, exhibiting smooth variations except at mode numbers 20 and 40.

For Simpson's rule, we see very similar behavior to that obtained with trapezoidal integration. For mode numbers up to 10 behavior is nearly identical. For larger mode numbers, the deviations of the error above and below the 100% line are slightly larger for Simpson's rule. In addition, the slope of the error is now discontinuous at modes 10 and 30.

For the cubic spline integration scheme, the increase in error to 100% between mode 1 and mode 10 is slightly less rapid than for the trapezoidal and Simpson schemes. Beyond mode 10, the smooth behavior of error found using the trapezoidal rule has been retained, but the swings above and below 100% are similar in magnitude to those found when using Simpson's rule.

For the Chebyshev scheme, we again see highly non-smooth variations in the tip

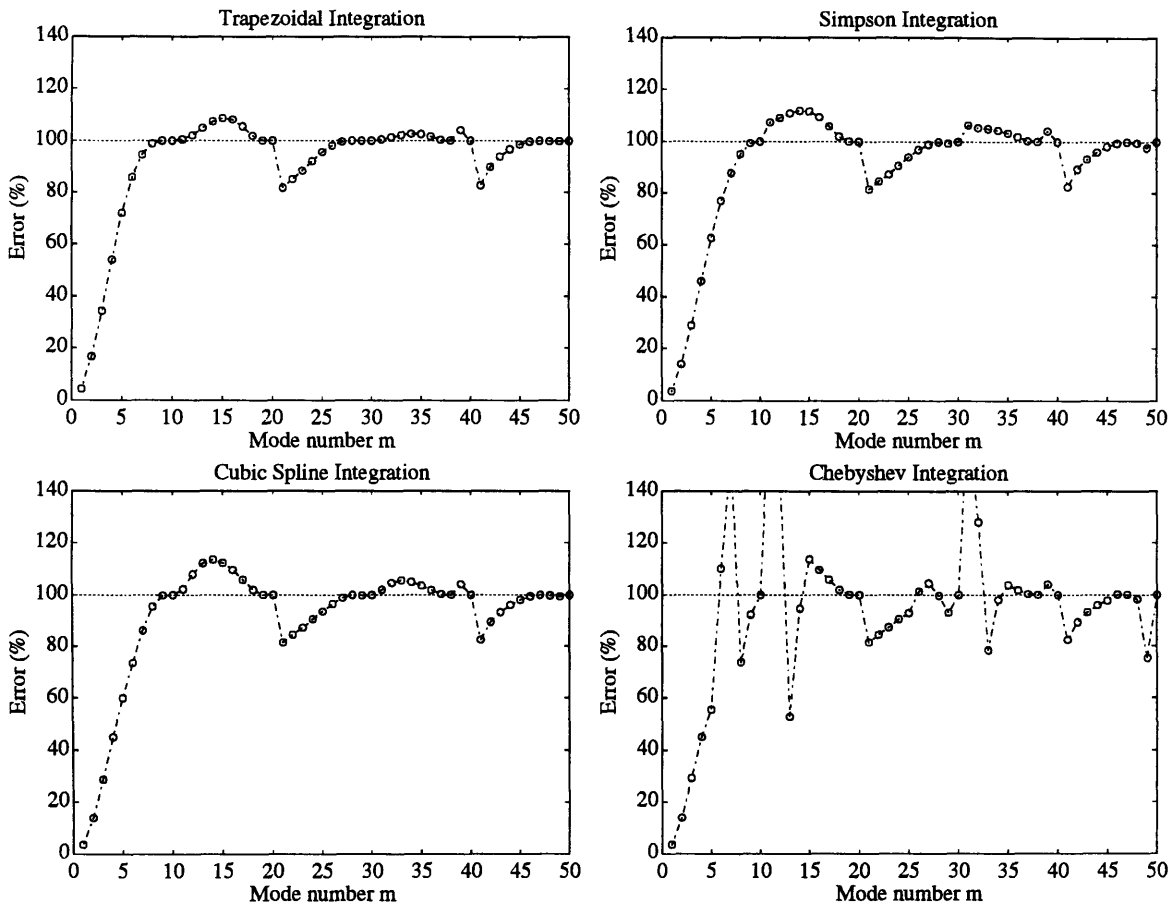

Figure 5.7: Tip deflection estimation error for pinned-pinned beam using Bartlett sensors with trapezoidal, Simpson, spline or Chebyshev integration.

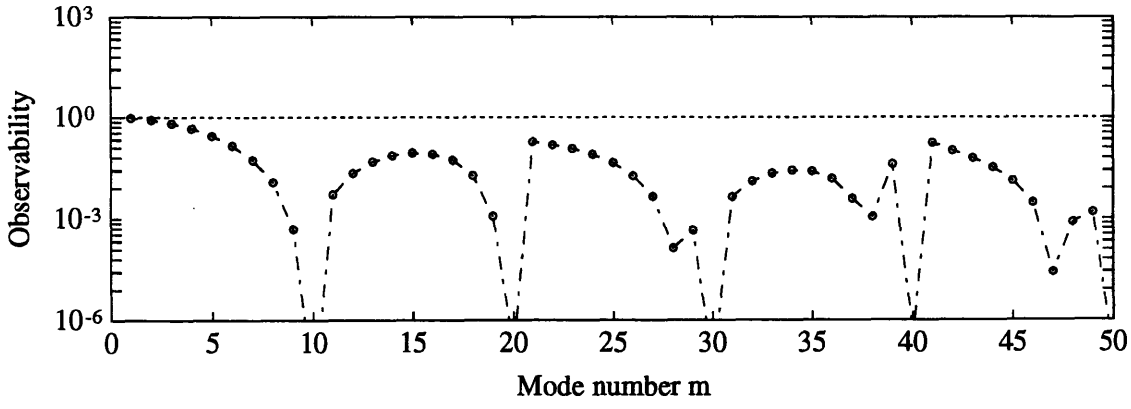

Figure 5.8: Sensor array observability for pinned-pinned beam using Bartlett sensors with trapezoidal integration.

estimation error. In fact, the monotonic increase in the error between mode 1 and mode 10 that is seen in the other three integration schemes is not present for the Chebyshev scheme. In addition, for higher mode numbers, the swings in the error are of much greater magnitude than for the other integration schemes.

The observability for the trapezoidal scheme contains local minima at all the multiples of the aliasing frequency, i.e. modes 10, 20 , 30, etc. In general, the observability is seen to decrease approximately linearly with mode number. Since the maximum value of the curvature increases linearly with mode number m, the observability actually rolls off at $1/k^2$ with respect to the curvature. This is to be expected, since the transfer function of a Bartlett sensor rolls off at $1/k^2$, as shown in Chapter 2.

It is clear that the most stable error behavior is obtained with the trapezoidal scheme for Bartlett sensors. The long term behavior of the observability envelope for the trapezoidal rule shows that the magnitude decreases approximately linearly with mode number. Although the observability is seen to possess a decreasing trend with mode number, satisfying Functional Requirement #5, the nonmonotonic trend in the error violates Functional Requirement #6.

5.1.4 Gauss-Hanning Sensors

The array of Gauss-Hanning sensors used is shown in Figure 5.9. For this array, the error in tip deflection estimation, as a function of percentage of the length of the beam, is shown in Figure 5.10, for the trapezoidal, Simpson, cubic spline and Chebyshev integration schemes. In the case of trapezoidal integration, the observability for the same array is shown in Figure 5.11.

Not surprisingly, the error behavior for mode numbers between 1 and 10 for the trapezoidal rule is much like that obtained using Bartlett sensors. The error starts near zero, satisfying Functional Requirement #4, and reaches 100% by mode 10. The

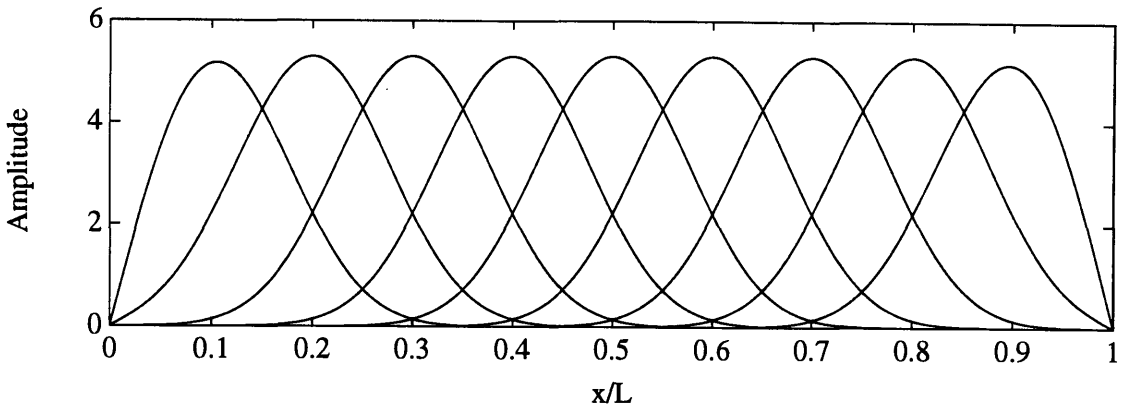

Figure 5.9: Sensor weightings for Gauss-Hanning sensors mounted on a pinned-pinned beam.

overshoot of the error beyond mode 10 is very small, and can be decreased by increasing the length of the sensors, and thus rolling the sensors off sooner. No other significant variations in the error are visible for higher mode numbers. Thus Functional Requirement #6 is satisfied, as the error approaches 100% virtually monotonically.

For Simpson's rule, we see a monotonic increase in the error until mode 8, where a local maximum is seen. The error for mode 9 is slightly lower, and then jumps to 100% for the tenth mode. A significant overshoot of more than 10% is then seen for mode 11. For higher modes, the error is seen to decrease monotonically towards 100% from above.

In the case of the cubic spline integration scheme, we see a relatively linear increase in the tip deflection estimation error, which reaches 100% at mode 10, as expected. Beyond this point, however, we see an overshoot of approximately 3.7%. For higher mode numbers, the error decreases monotonically, approaching 100% from above, as with Simpson's rule.

For Chebyshev integration, error behavior is acceptable until mode 6, where a sudden swing to about 120% is observed. At mode 7 the error is 240%, and at mode 8 a downward swing to −45% is observed. The error swings once more between

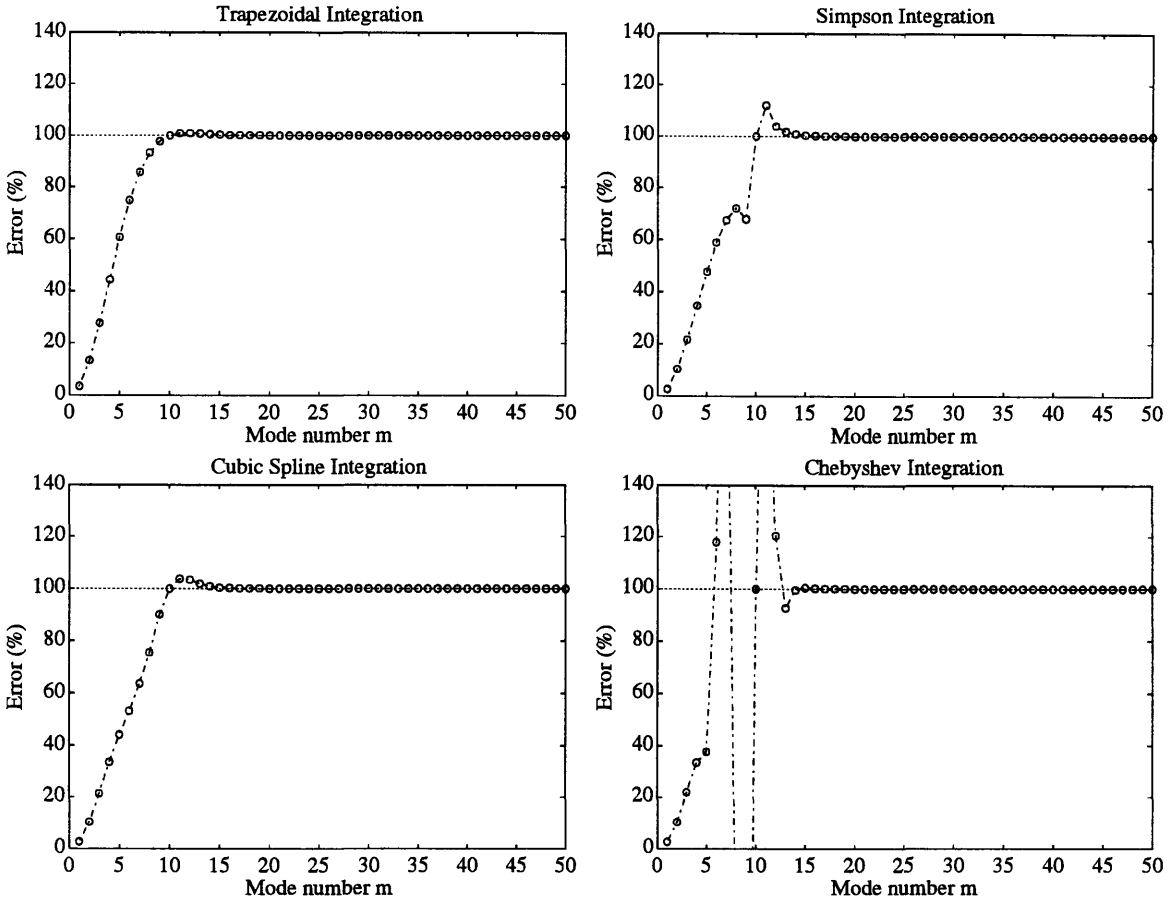

Figure 5.10: Tip deflection estimation error for pinned-pinned beam using Gauss-Hanning sensors with trapezoidal, Simpson, spline or Chebyshev integration.

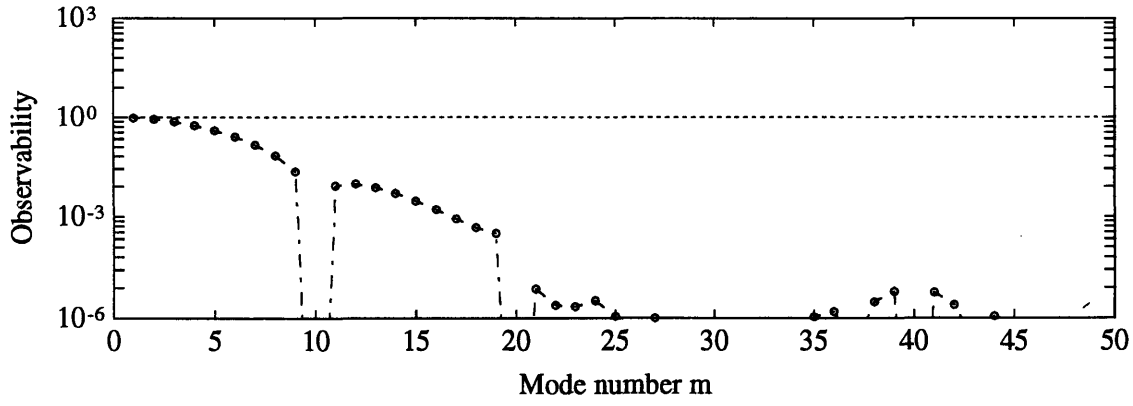

Figure 5.11: Sensor array observability for pinned-pinned beam using Gauss-Hanning sensors with trapezoidal integration.

modes 9 and 11, going through 100% at mode 10. The behavior in the error is clearly unacceptable and violates Functional Requirement #6.

The observability starts near unity at low mode numbers, and decreases more and more rapidly as the mode number approaches 10. For the aliasing frequencies of mode 10 and 20, the observability drops far below 10^{-6}. The envelope of the observability decreases steadily for mode numbers above 20, falling below 10^{-6} by mode 45, and satisfying Functional Requirement #5.

It is clear that the most stable error behavior is obtained with the trapezoidal scheme for Gauss-Hanning sensors. The long term behavior of the observability envelope for shows that the magnitude decreases quickly with increasing mode number. This decreasing trend in the error satisfies Functional Requirement #6, and the monotonic increase in the error satisfies Functional Requirement #5.

5.1.5 Summary

In general it is seen that the only integration scheme to yield consistently the most acceptable trends in the tip displacement estimation error is the trapezoidal integration scheme. This is especially evident when nearing the first aliasing frequency associated with mode number 10. For low order modes, the change in the outputs of the sensors along the length of the beam is relatively smooth. Therefore, the estimate of the tip deflection will be very similar no matter which integration scheme is used. For mode numbers slightly below or above the aliasing mode number, however, the outputs of the sensors are seen to alternate in sign in a sawtooth fashion. Clearly, the trapezoidal integration scheme is most apt to deal with such high frequency oscillations in the sensor measurements. Simpson's rule will attempt to fit parabolas between three successive points and typically overestimate the area. In the cubic spline rule, a spline will be fit to the spatially oscillating sensor outputs, and the area estimate will be highly inaccurate. Finally, the Chebyshev scheme will attempt to

fit a single polynomial to all the sensor outputs. Typically this is a highly unstable process for polynomials of high order. Such unstable behavior is clearly visible in the change in tip error for modes 9, 10 and 11, for all sensor types.

As more complex sensor weightings are used, and their rolloff rates increase, it is seen that the error behavior becomes more controlled. Although this is true for all integration schemes, it is only in the case of Gauss-Hanning sensors and trapezoidal integration that a good tip deflection estimation error trend is obtained. This combination of integration scheme and sensor weighting yields good results because the rolloff of the sensors is very high and the trapezoidal integration scheme integrates the spatially oscillating sensor measurements found near the aliasing mode numbers (which are multiples of 10) in a stable fashion.

5.2 Simulation Results for a Clamped-Free Beam

The performance of a sensor array in shape estimation of the dynamic modes of a clamped-free beam is investigated here. The exponential components of the clamped-free mode shapes at the root and tip of such a beam makes tip deflection estimation a much harder problem than for a pinned-pinned beam with purely spatially periodic mode shapes. Specifically, good knowledge of the strain at the root is required for good estimation of the tip deflection for low order modes. However, for higher order modes, measurements of the root strain should roll off as quickly as measurements of strain in the interior of the beam.

The deflection, slope and curvature of the dynamic modes of a clamped-free beam are given by:

$$w(x) = A_m\Big[C_1(\sin\beta_m x - \sinh\beta_m x) + C_2(\cos\beta_m x - \cosh\beta_m x)\Big] \quad (5.6)$$

$$\frac{dw(x)}{dx} = \beta_m A_m\Big[C_1(\cos\beta_m x - \cosh\beta_m x) + C_2(-\sin\beta_m x - \sinh\beta_m x)\Big] \quad (5.7)$$

$$\kappa(x) = \beta_m^2 A_m\Big[C_1(-\sin\beta_m x - \sinh\beta_m x) + C_2(-\cos\beta_m x - \cosh\beta_m x)\Big] \quad (5.8)$$

where

$$C_1 = \sin\beta_m L - \sinh\beta_m L \quad (5.9)$$

$$C_2 = \cos\beta_m L + \cosh\beta_m L \quad (5.10)$$

and where the constant A_m is set such that the tip displacement is unity: $w_{\text{tip}} = w(L) = 1$. The spatial frequencies β_m are found from numerical solutions to the transcendental equation

$$\cos\beta L \ \cosh\beta L = -1 \quad (5.11)$$

which are

$$\begin{aligned} \beta_1 &= 1.87510 \\ \beta_2 &= 4.69409 \\ \beta_3 &= 7.85476 \\ \beta_n &\approx \frac{2n-1}{2}\pi \qquad (n > 3) \end{aligned} \quad (5.12)$$

A variety of integration schemes are used to integrate the curvatures reported by the array of sensors, in order to estimate the beam deflection, which is normalized to be unity. The *error* between the estimated tip deflection and the actual tip

displacement (unity) is found:

$$\mathcal{E} = 1 - \frac{\tilde{w}_{\text{tip}}}{L} \tag{5.13}$$

where $\tilde{w}_{\text{tip}}$ is the estimated tip deflection. This error should be as close to zero as possible for low mode numbers. As the mode number is increased, this error should approach 100%. This is because as the mode number of the beam is increased, the outputs of the sensors should approach zero. The effective *observability* of the sensor array is defined as:

$$\mathcal{O} = 1 - \mathcal{E} = 1 - \left(1 - \frac{\tilde{w}_{\text{tip}}}{L}\right) \tag{5.14}$$

$$\mathcal{O} = \frac{\tilde{w}_{\text{tip}}}{L} \tag{5.15}$$

The observability should be unity for low mode numbers m, and approach zero as the mode number is increased, due to the rolloff exhibited by the sensors.

Since Bartlett sensors are simple to implement and yield $1/k^2$ rolloff, shape estimation simulations for an array of Bartlett sensors were conducted. In addition, simulations were performed using arrays of Hanning-Hanning and Gauss-Hanning sensors, since their transfer functions exhibit high rolloff rates.

5.2.1 Bartlett Sensors

The array of Bartlett (triangular) sensors used is shown in Figure 5.12. The lengths of the sensors have been set such that the -3 dB point of their transfer functions correspond to the spatial frequency of the fourth mode of the beam. A half sensor is centered at the root, and nine sensors are equally spaced in the interior of the beam. The sensors at $x/L = 0.1$ and $x/L = 0.9$ are simply truncated at the root and tip, respectively. All sensor weightings $f_i(x)$ are scaled such that $\int_0^L f_i(x)\,dx = 1$. The output of the sensor centered at the root is taken to be the root curvature, and the curvature at the beam tip is assumed to be zero. Figure 5.13 shows the error in

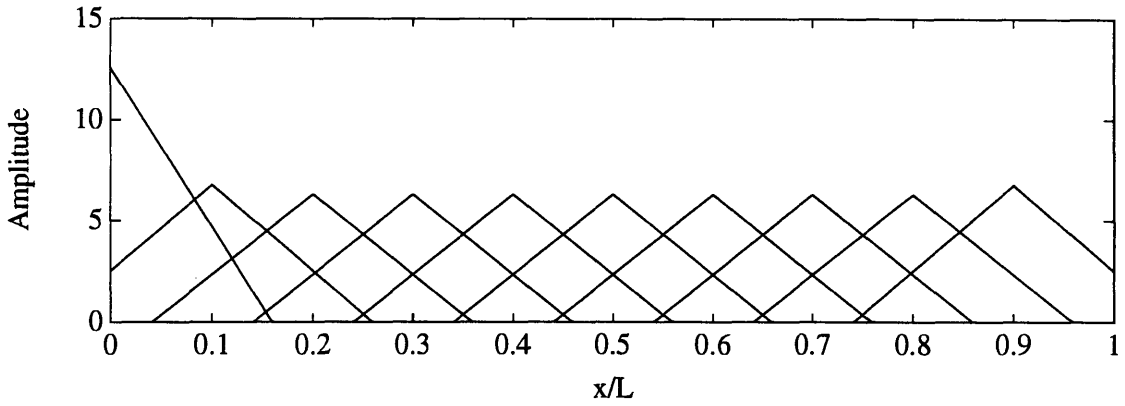

Figure 5.12: Sensor weightings for Bartlett sensors mounted on a clamped-free beam.

tip deflection estimation for this sensor array using the trapezoidal integration scheme to estimate beam displacement. Figure 5.14 shows the observability of the array.

It can be seen from the error plot that the tip deflection estimation error is reasonably small for modes 1–3. For these modes the trapezoidal scheme manages to approximate the theoretical deflection, even though the measured root curvature is slightly underestimated. By mode 4, the measured root curvature is only 43% of the theoretical curvature. This causes the tip deflection estimation error to drop to nearly -90%. The alternating error pattern seen for higher modes is due to the fact that the root curvature is of the same sign as the tip deflection for odd modes, but of the opposite sign for even modes. Thus, since the integration scheme is most sensitive to the root curvature, the estimated tip deflection tends to alternate sign in accordance with the measured root curvature. Figure 5.15(a) shows the measured and theoretical curvature for the fifth dynamic mode of the beam. It is evident that the interior sensors are beginning to roll off. However, the root curvature measurement is underestimated by the half sensor centered at the root. The resulting effect is visible in Figure 5.15(b), which shows the theoretical and estimated beam deflection for the fifth mode. The estimated deflection has the same characteristics as the theoretical deflection with an added linear negative drift, caused by the low measurement of the

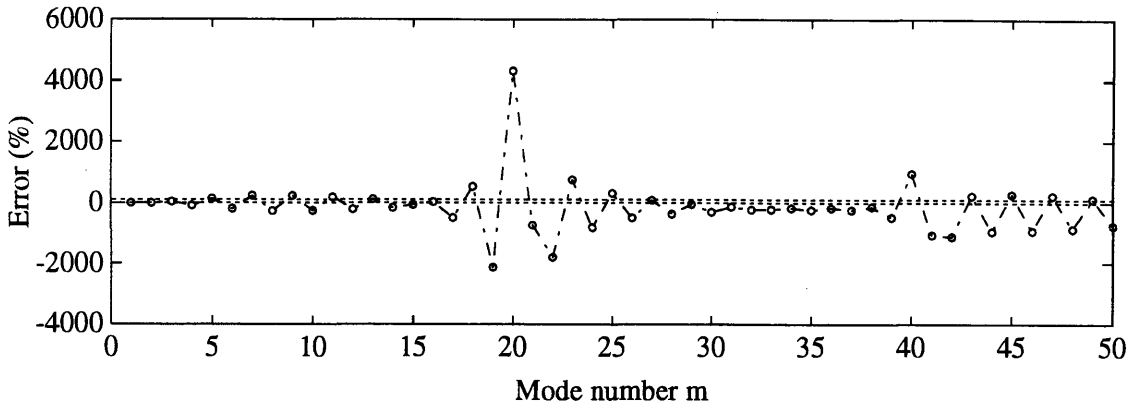

Figure 5.13: Tip deflection estimation error for clamped-free beam using Bartlett sensors with trapezoidal integration.

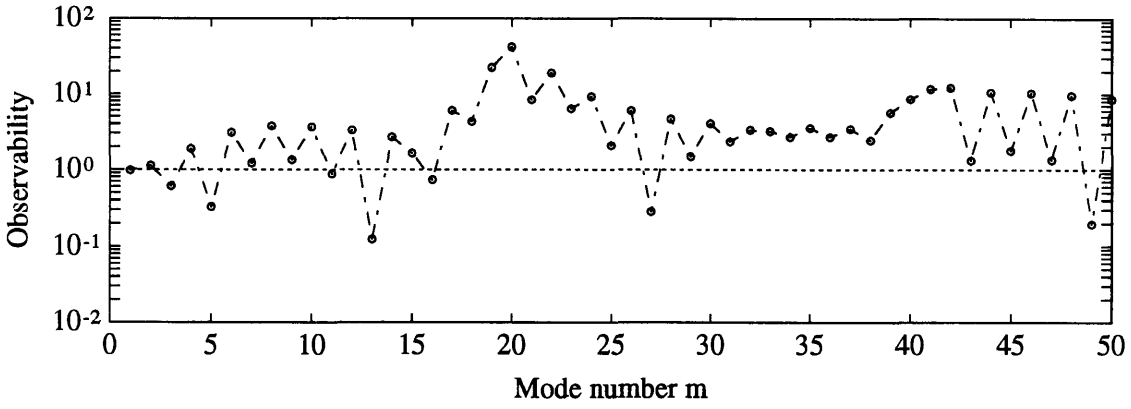

Figure 5.14: Sensor array observability for clamped-free beam using Bartlett sensors with trapezoidal integration.

root curvature.

For modes 5–15, root curvature underestimation is responsible for most of the error. Added negative drifts are seen for odd modes, and positive drifts for even modes. We find that the interior sensors have effectively rolled off by mode 15. However, for modes 10–15, we find that the rolloff for the sensor at $x/L = 0.9$ lags behind the rolloff of even the root sensor, and its contribution significantly impacts the estimated tip deflection. For modes 19–22, the wavelength of the curvature is

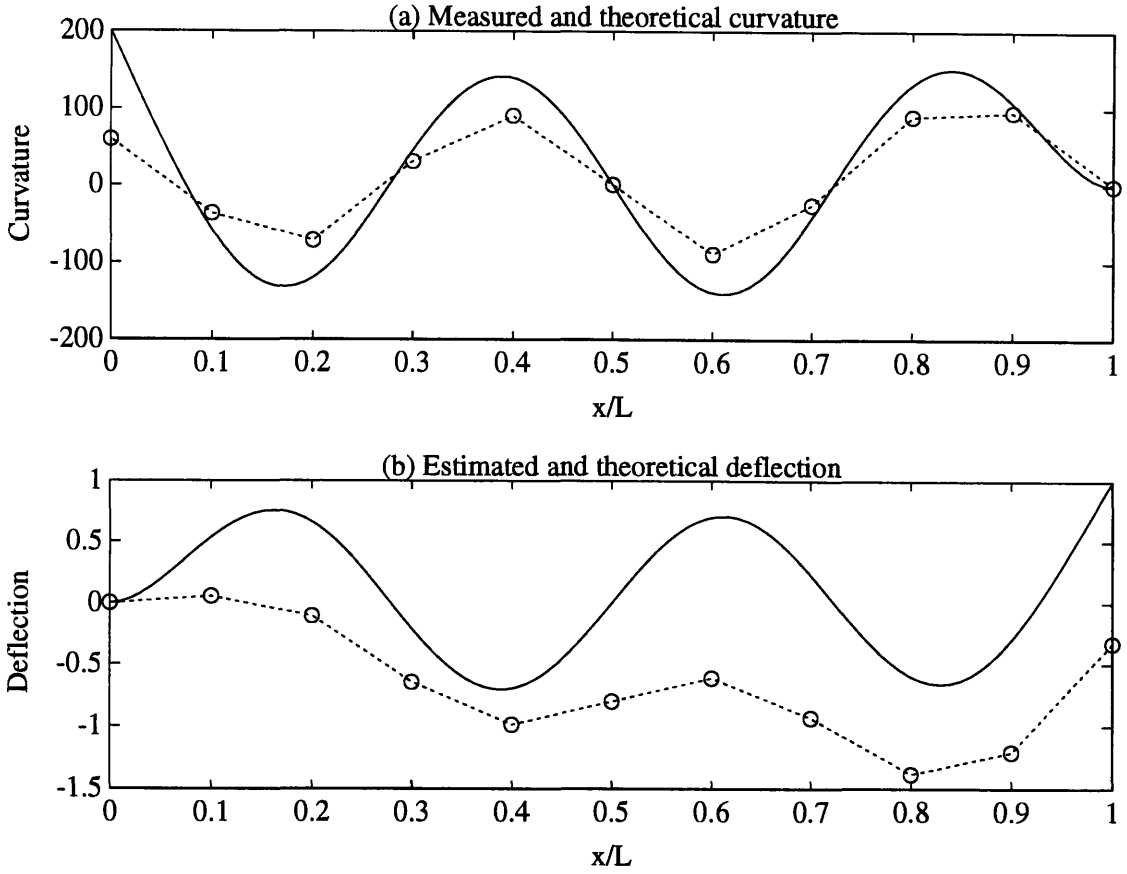

Figure 5.15: Measured and theoretical curvature, and estimated and theoretical displacement for the fifth dynamic mode of a clamped-free beam using Bartlett sensors with trapezoidal integration.

close to the distance between the sensors, and spatial aliasing occurs. The result is that the average curvature measured will no longer be close to zero because most or all the sensors report measurements of the same sign. These measurements accumulate in the numerical integration algorithm, and the estimated deflection will deviate more and more strongly from the theoretical deflection as the beam tip is approached.

It was shown in Chapter 2 that in a sinusoidal strain field, the transfer function of a complete Bartlett sensor rolls off at $1/k^2$, and for a simply truncated sensor, at $1/k$. The sensors centered at $x/L = 0$, $x/L = 0.1$ and $x/L = 0.9$ are simply truncated, and should therefore roll off at $1/k$, while the interior ones will roll off

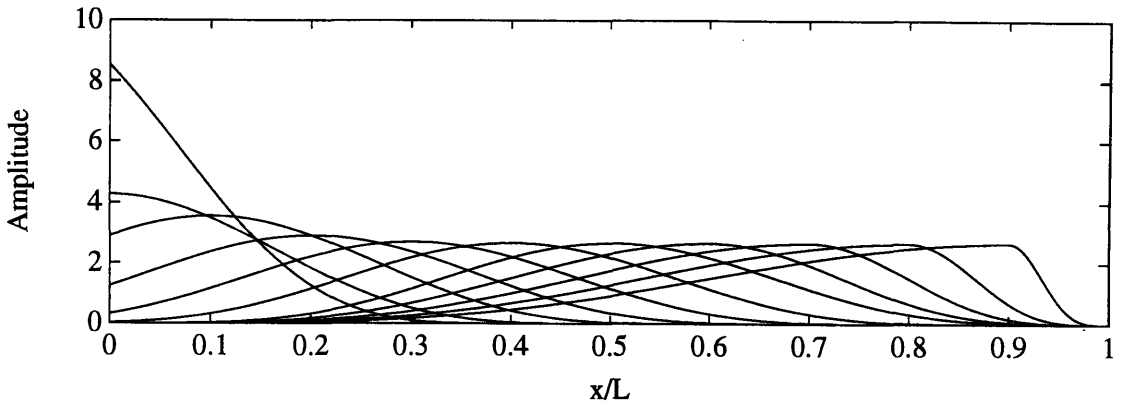

Figure 5.16: Sensor weightings for Hanning-Hanning sensors mounted on a clamped-free beam.

at $1/k^2$. In spite of this, no distinctive rolloff trend is visible in Figure 5.14, which shows the observability of the sensor array. However, there is a decrease of a factor of four in the tip displacement estimation error at the two aliasing mode numbers 20 and 40. This indicates that the sensor array observability is bounded, and rolls off with mode number. However, this rolloff is clearly not quick enough to satisfy Functional Requirement #5, and the nonmonotonic trend in the error violates Functional Requirement #6.

5.2.2 Hanning-Hanning Sensors

The array of Hanning-Hanning sensors used is shown in Figure 5.16. The lengths of the sensors have been set such that the -3 dB point of their transfer functions correspond to the spatial frequency of the second mode of the beam. From Chapter 2, we find that l, the length of the sensor weighting is equal to L, the length of the beam.

A half sensor is centered at the root, and nine sensors are equally spaced in the interior of the beam. The output of an additional sensor centered at $x/L = -0.1$ is used to estimate the root curvature. The tip curvature is again assumed to be zero. Any part of a sensor extending beyond the root of the beam is simply truncated at the

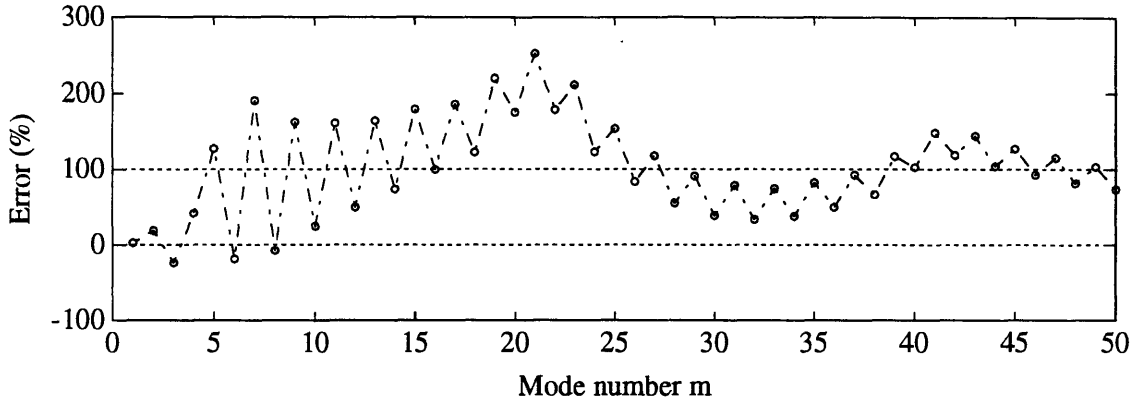

Figure 5.17: Tip deflection estimation error for clamped-free beam using Hanning-Hanning sensors with trapezoidal integration.

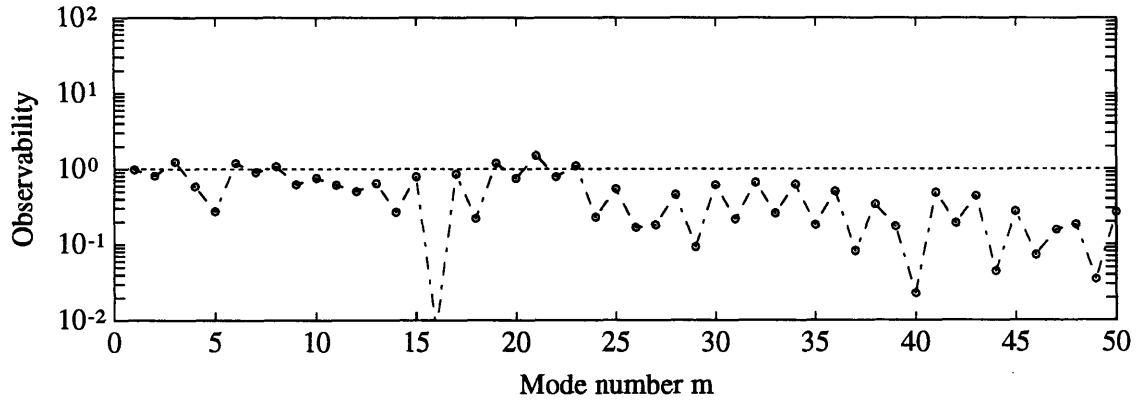

Figure 5.18: Sensor array observability for clamped-free beam using Hanning-Hanning sensors with trapezoidal integration.

root. In contrast, any sensor which would normally extend beyond the tip is modified as follows: the outboard half is shortened such that the smoothly tapering end is at the tip, while the inboard half is lengthened to restore the sensor's original length. This is done in an attempt to preserve the rolloff properties of the Hanning-Hanning sensors.

The outputs of all the sensors, except the one used to estimate the root curvature, are assigned at their centroids when spatial integration is performed. The centroid of

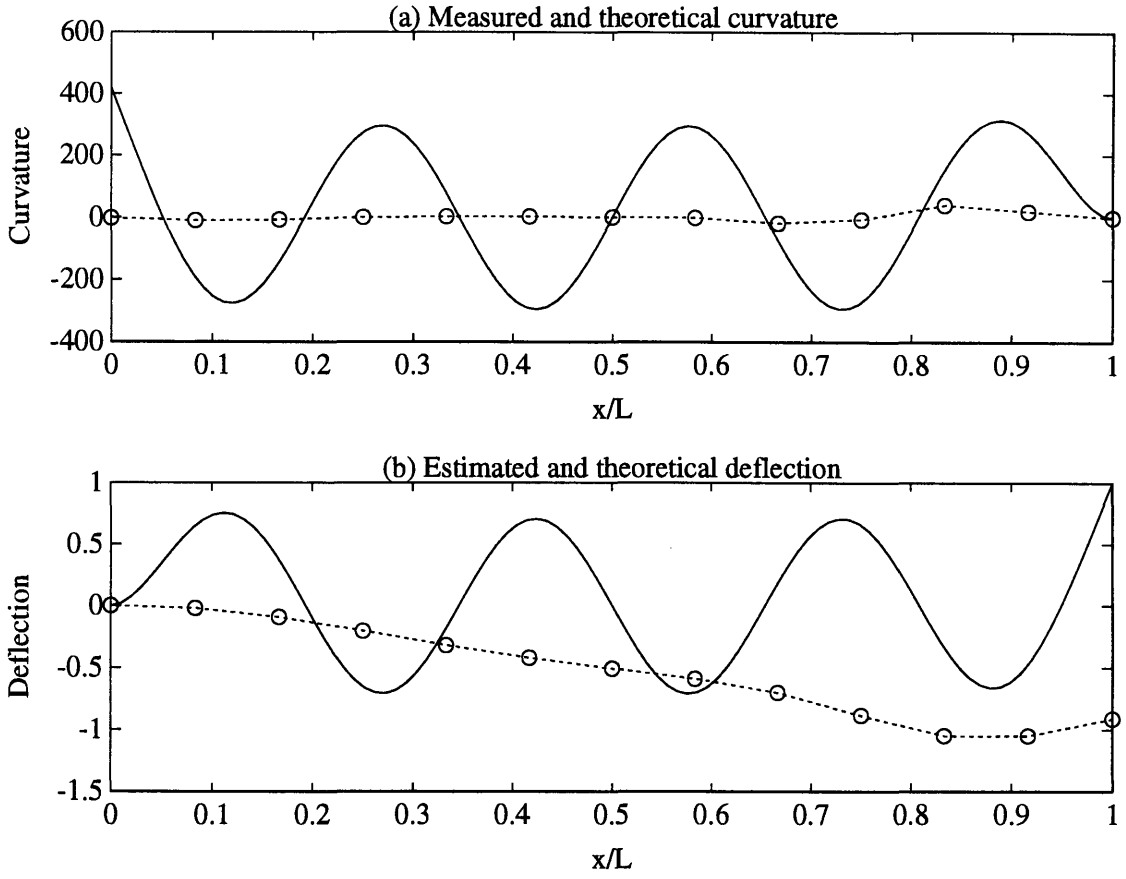

Figure 5.19: Measured and theoretical curvature, and estimated and theoretical displacement for the seventh dynamic mode of a clamped-free beam using Hanning-Hanning sensors with trapezoidal integration.

a sensor weighting is given by:

$$x_{\text{centroid}} = \frac{\int_0^L x\, f(x)\, dx}{\int_0^L f(x)\, dx} \tag{5.16}$$

Figure 5.13 shows the error in tip deflection estimation for the sensor array using the trapezoidal integration scheme to estimate beam displacement. Figure 5.14 shows the observability of the array.

It is seen that the tip displacement estimation error is within $\pm 25\%$ of the theoret-

ical value for modes 2 and 3, and within 2.5% for the first mode. Beyond this point, however, underestimated root curvature estimates cause the deflection estimate to drift in accordance with the sign of the root curvature. This causes the alternating pattern in the error. Figure 5.19 shows this situation clearly, in that the estimated tip deflection is opposite in sign to the theoretical tip displacement. This situation deteriorates as the mode number is increased further.

It can be seen in Figure 5.19 that the magnitude of the curvature measured by the last two sensors is unusually high, compared to the outputs of the remaining sensors. This is due to poor rolloff of these sensors. The effect is to cause the deflection to begin to swing upward towards the end of the beam. As the mode number is increased further, it is found that the two outer sensors begin to register even higher measurements. This forces the deflection estimate to swing upwards towards the tip of the beam, and causes the increasing trend of the error between modes 5 and 22. Between mode 22 and 32, the last two sensors measure negative curvature, forcing the deflection to swing downward near the tip, and causing the error to begin to drift downward, toward 0%. It is evident that this effect is responsible for most of the tip deflection estimation error. The drift in estimated deflection resulting from improperly estimated root curvature is only responsible for the high frequency jitter in the error. As the mode number is increased from 30 to 50, we see that the low frequency oscillations are decreasing in magnitude and centered about 100%. This means that as the mode number is increased, the observability of the sensors to the beam mode shapes rolls off. Thus the observability $\mathcal{O}$, should approach zero.

Figure 5.18 shows the rolloff trend in the observability of the sensor array as the mode number is increased. This rolloff is approximately $1/k$, but is limited by the simply truncated sensors at the root. This is due to the fact that it is impossible to construct sensors that: (a) accurately measure curvature relatively close to the boundary of the beam; (b) begin to roll off at a low order mode (c) and roll off

quickly with frequency. This is because accurate measurement of curvature near the beam boundary requires the sensor to be centered in the vicinity of the beam boundary. To ensure that the sensor begins to roll off at a low order mode, it is necessary to make the length of the sensor a significant fraction of the length of the beam. It will therefore be forced to extend beyond the beam boundary, and will therefore require spatial truncation. This spatial truncation guarantees that the resulting sensor's transfer function for a spatially sinusoidal strain field will not roll off quickly in frequency. The rolloff exhibited by the observability is clearly not quick enough to satisfy Functional Requirement #5 , and the nonmonotonic trend in the error violates Functional Requirement #6.

5.2.3 Gauss-Hanning Sensors

In Chapter 2, Gauss-Hanning sensors were shown to exhibit excellent rolloff characteristics. In this section, results using three different arrays of Gauss-Hanning sensors are presented. These arrays will be referred to as the first, second and third iterations.

First Iteration

The array of first iteration Gauss-Hanning sensors used is shown in Figure 5.20. The lengths of the sensors have been set such that the -3 dB points of their transfer functions occur at the spatial frequency corresponding to the fourth mode of the beam. As with the Bartlett sensors, a sensor is centered at the root, and is used to estimate the root curvature. Nine more sensors are equally spaced in the interior of the beam. Sensors extending beyond the root are simply truncated, and sensors extending beyond the beam tip have the outboard half of their weighting shortened such that the end of the sensor weighting falls at the beam tip. Their inboard half is left unchanged. The curvatures measured by the interior sensors are treated as if they were point measurements, and are assigned at the centers of the sensors for the

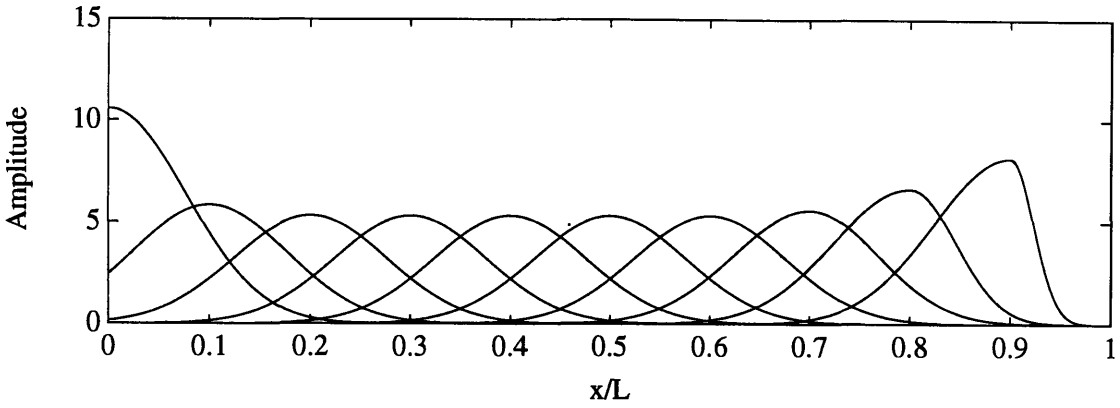

Figure 5.20: Sensor weightings for first iteration Gauss-Hanning sensors mounted on a clamped-free beam.

purposes of spatial integration. A Least Squares Global Dynamic Shape Function integration scheme with 4 shape functions is used to estimate beam deflection.

Figure 5.21 shows the error in tip deflection estimation for the sensor array, using the LSGDSF-4 integration scheme. The error is only 1.5% for the first mode, but drops to -26% for the second mode. The high frequency alternation in error is a result of the alternation of the sign of the root curvature when the tip displacement is normalized to be unity. The sharply increasing trend in error between mode 16 and mode 20 is due to abnormally large measurements of negative sign made by the last two sensors. This causes the integration scheme to create a deflection estimate which swings down sharply near the beam tip. The decrease in error between mode 20 and mode 30 is caused by the same phenomenon, when the last two sensors measure high curvature of positive sign. It can be seen that this low frequency oscillation in error is damped, and that the steady-state error will be 100%, indicating a zero observability state is eventually achieved. Figure 5.22 shows the observability for the array of sensors. It shows that no rolloff trend is visible until beyond mode 40. Clearly, this rolloff is not quick enough to satisfy Function Requirement #5. In addition, Functional Requirement #4 is not satisfied because good estimation of several low

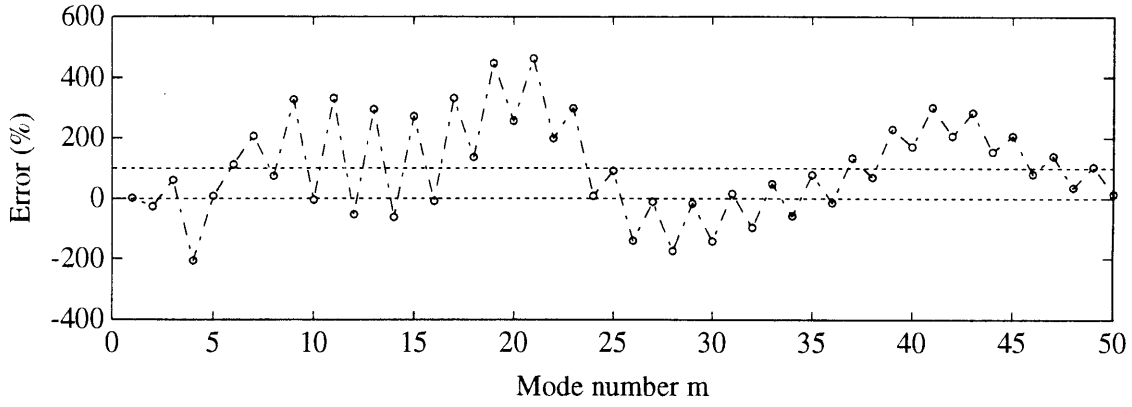

Figure 5.21: Tip deflection estimation error for clamped-free beam using first iteration Gauss-Hanning sensors with LSGDSF-4 integration.

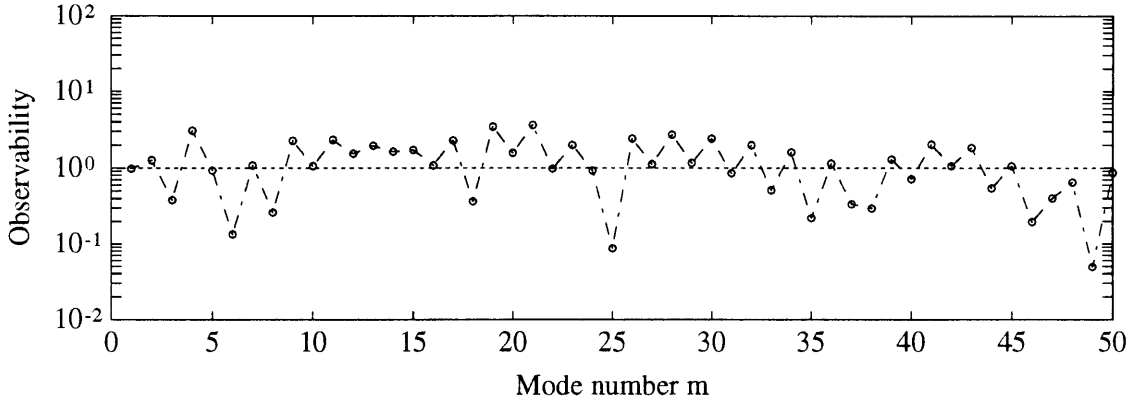

Figure 5.22: Sensor array observability for clamped-free beam using first iteration Gauss-Hanning sensors with LSGDSF-4 integration.

order modes is not guaranteed. Finally, Functional Requirement #6 is not satisfied because the decrease in observability is not monotonic.

Second Iteration

The array of second iteration Gauss-Hanning sensors used is shown in Figure 5.23. The lengths of the sensors have been set such that the -3 dB points of their transfer functions occur at the spatial frequency corresponding to the fourth mode of the

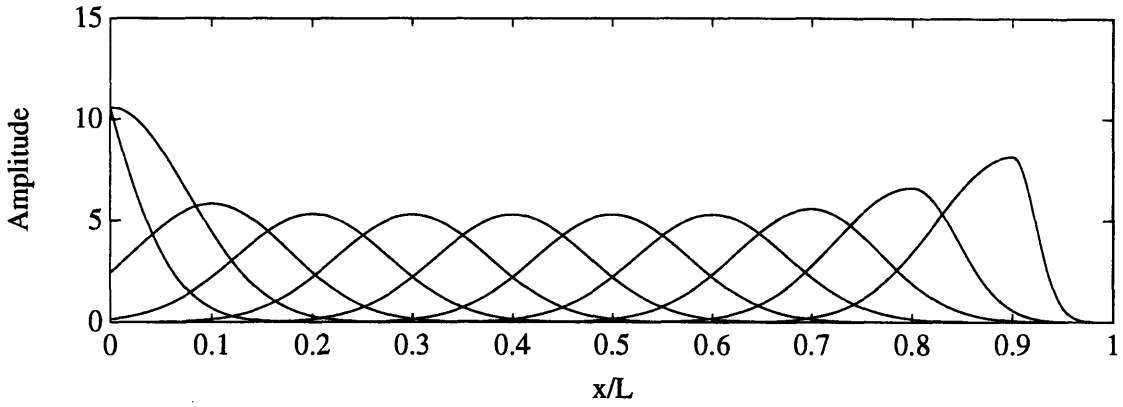

Figure 5.23: Sensor weightings for second iteration Gauss-Hanning sensors mounted on a clamped-free beam.

beam. A sensor is centered at the root and nine more sensors are equally spaced in the interior of the beam. A sensor centered at $x/L = -0.1$ is used to estimate the root curvature. Sensors extending beyond the root are simply truncated, and sensors extending beyond the beam tip have the outboard half of their weighting shortened such that the end of the sensor weighting falls at the beam tip. Their inboard half is left unchanged. The curvatures measured by the interior sensors are used as if they were point measurements, and are assigned at the centroids of the sensors. The centroid location was calculated using Equation (5.16). All sensors except the one used to estimate root curvature are scaled such that $\int_0^L f(x)\,dx = 1$. The weighting of the root-measuring sensor was reduced in an attempt to reduce the high frequency oscillation in the error due to the oscillation in sign of the root curvature with respect to tip displacement. A Least Squares Global Dynamic Shape Function integration scheme with 6 shape functions is used to estimate beam deflection.

Figure 5.24 shows the tip displacement estimation error for the sensor weightings shown in Figure 5.23. Low order mode estimation is accurate for the first four modes, where the tip displacement estimation error is under 15%. It can be seen that beyond mode 14, the error is nearly completely smooth. Between mode 4 and mode 14,

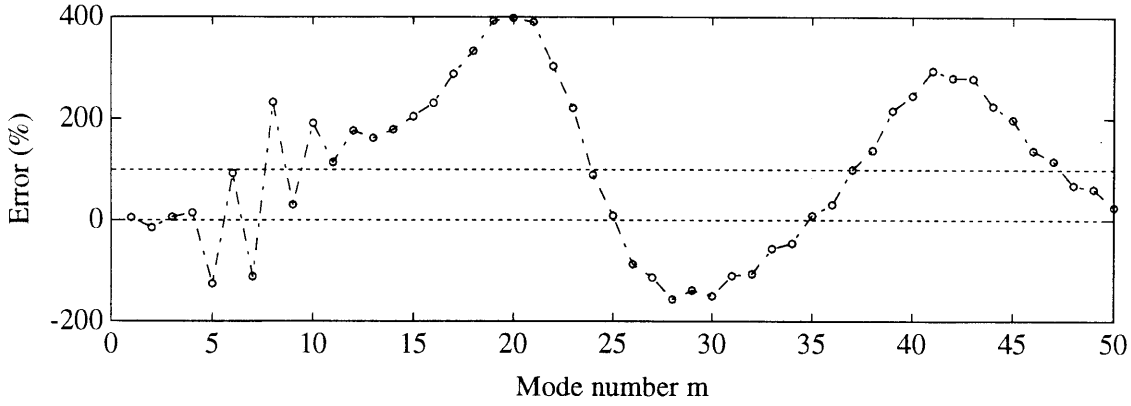

Figure 5.24: Tip deflection estimation error for clamped-free beam using second iteration Gauss-Hanning sensors with LSGDSF-6 integration.

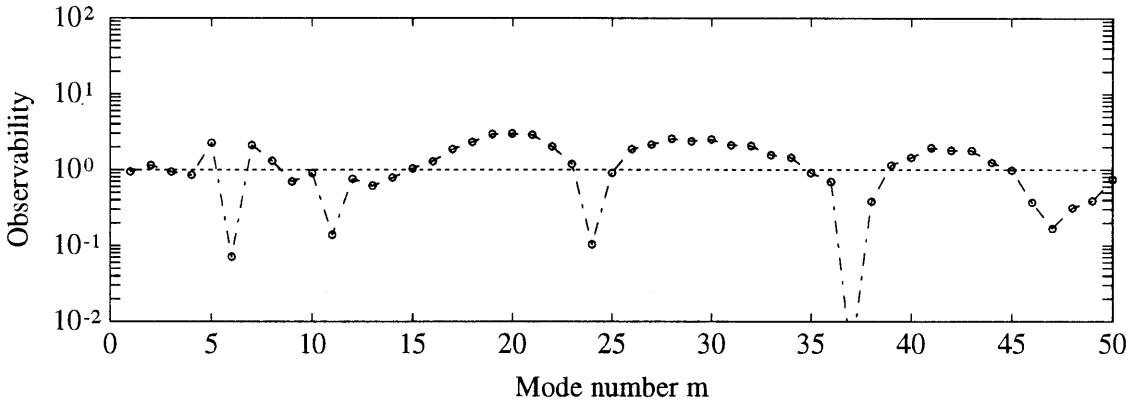

Figure 5.25: Sensor array observability for clamped-free beam using second iteration Gauss-Hanning sensors with LSGDSF-6 integration.

however, high frequency oscillations in the error are still observed. The low frequency deviations of the error are still present, because they are caused by abnormally high measurements of curvature reported by the last two sensors. Attempts to reduce these oscillations for high modes reduces the estimation accuracy for low order modes. As these deviations in the error dampen out for higher modes, it can be seen that the error will reach a steady-state value of 100%, indicating that very high modes are unobservable to the sensor array. Figure 5.25 shows this observability. Functional

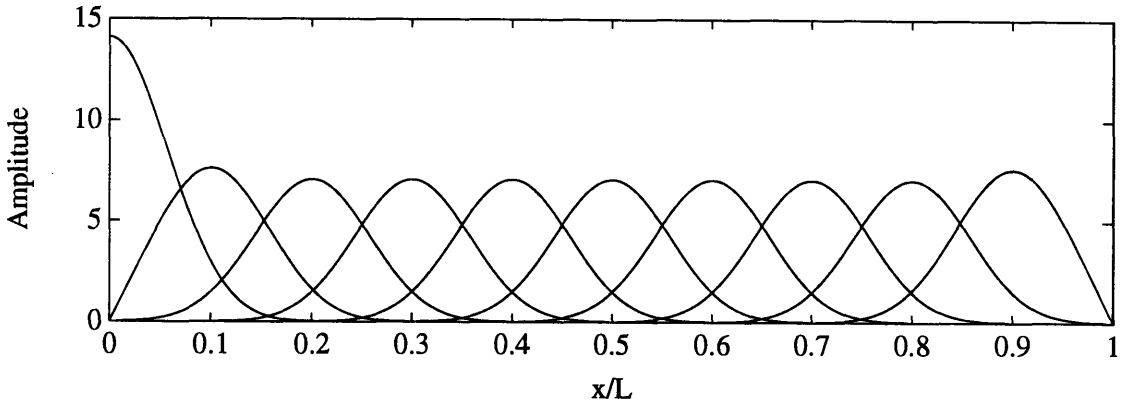

Figure 5.26: Sensor weightings for third iteration Gauss-Hanning sensors mounted on a clamped-free beam.

Requirement #5 is violated because the observability does not roll off quickly with frequency.

Third Iteration

The array of third iteration Gauss-Hanning sensors used in shown in Figure 5.26. The sensor layout is exactly the same as for the second iteration array, except that the sensor at $x/L = -0.1$ has been removed. In addition, the sensor measurements are not assigned to a specific point. Rather, the Global Dynamic Shape Function integration scheme using as many shape functions as sensors (10) is employed in beam displacement estimation. Equation (3.62) is used to form the system of equations that is used to solve for the 10 shape coordinates p_j.

Figure 5.27 shows the tip displacement estimation error for the third iteration Gauss-Hanning sensor array. It shows that for the first ten modes, perfect tip displacement estimation performance is accomplished. This is due to the fact that ten sensors and ten shape functions are employed, and because for the first ten modes, the system of equations (3.62) extracts only the shape function corresponding to the measured mode. As soon as displacement estimation of mode 11 is attempted, large

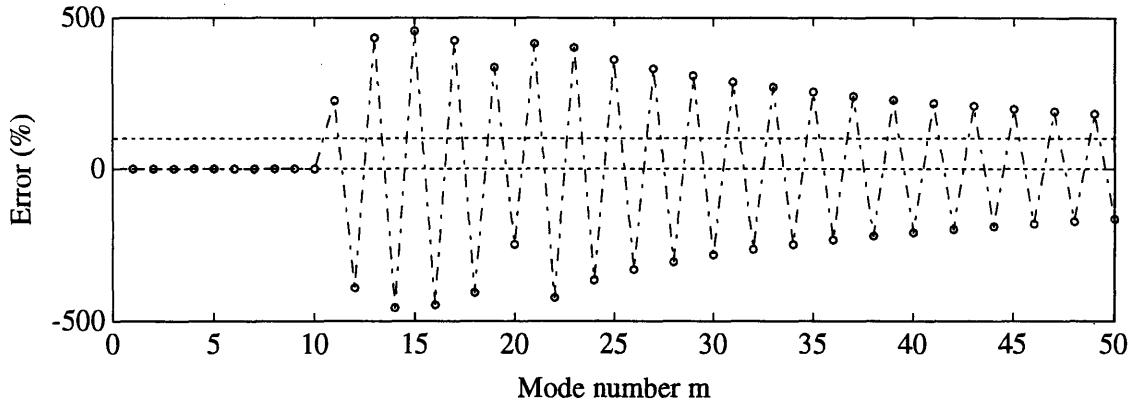

Figure 5.27: Tip deflection estimation error for clamped-free beam using third iteration Gauss-Hanning sensors with GDSF integration.

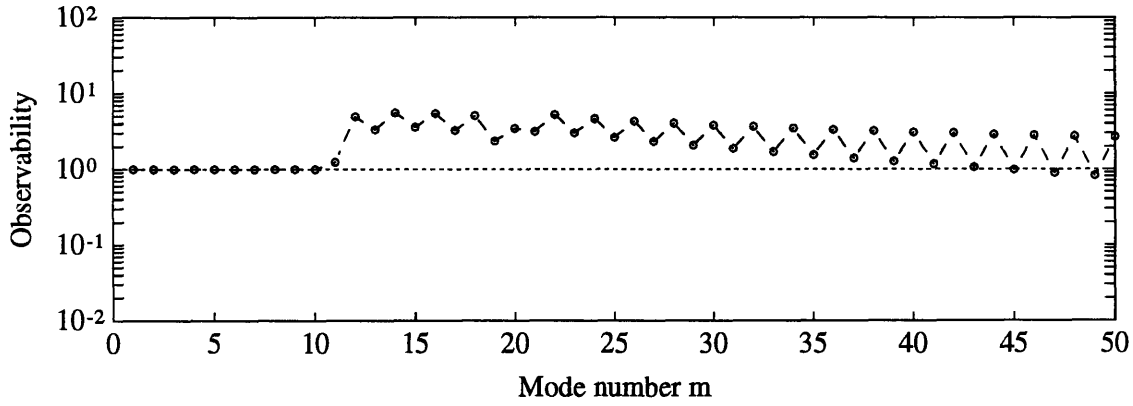

Figure 5.28: Sensor array observability for clamped-free beam using third iteration Gauss-Hanning sensors with GDSF integration.

errors are immediately observed. Figure 5.29(a) shows the measured and theoretical curvature for this case. Since mode 11 is of a higher spatial frequency than any of the ten shape functions available to the integration scheme, the fitting procedure is hard-pressed to produce a convincing fit to the measurements. Figure 5.29(b) shows the theoretical and estimated displacement for mode 11. It is seen that underestimation of the root curvature introduces a linear drift in the estimated deflection, and accounts for most of the resulting error.

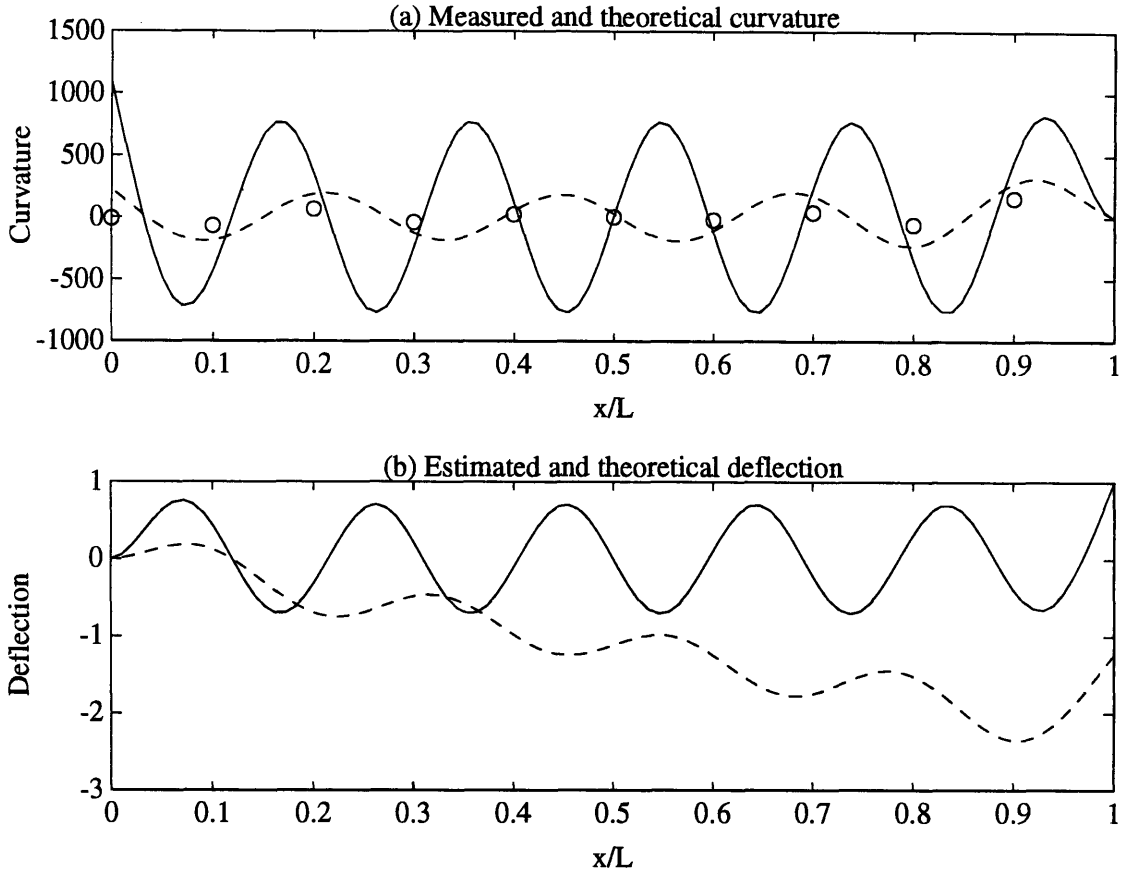

Figure 5.29: Measured and theoretical curvature, and estimated and theoretical displacement for the eleventh dynamic mode of a clamped-free beam using third iteration Gauss-Hanning sensors withGDSF integration.

Tip displacement estimation errors reach a maximum at mode 15. As the mode number is increased, an exponential decrease in the error is visible. However, high frequency oscillations are still present, and are caused by the alternation of the sign of the root curvature. Reducing these oscillations would require reducing the observability of the root curvature, which is required for good estimation performance for low order modes. Figure 5.28 shows the observability of the sensor array to the dynamic mode shapes of the beam. Although limited rolloff is exhibited by the observability, it is clearly not enough to satisfy Functional Requirement #5. In addition, the nonmonotonic nature of the error violates Functional Requirement #6.

5.2.4 Summary

It is seen that the worst tip deflection estimation errors are obtained using Bartlett sensors with the trapezoidal integration scheme. Since the trapezoidal rule was shown to yield the most well-behaved error trends for the case of estimation of the dynamic mode shapes of a pinned-pinned beam, the poor tip estimation performance must be a result of the sensor weighting used. In Chapter 2 it was shown that the rate of rolloff of the transfer function for a Bartlett sensor in a sinusoidal strain field is $1/k^2$. The curvature for the mode shapes of a clamped-free beam go as k^2 when the displacement is held constant. It is therefore necessary for the transfer functions of the sensor weightings to roll off at $1/k^3$ or faster, in order to assure a rolloff in the observability of the sensor array. It was seen in Figure 5.14 that the observability for an array of Bartlett sensors does not seem to roll off as the mode number is increased.

In Chapter 2 it was found that the rolloff rate of the transfer function of the Hanning-Hanning sensor is $1/k^5$. This should guarantee a rolloff rate of $1/k^3$ in the observability shown in Figure 5.18. It is evident that such rolloff is not observed. It was also shown in Chapter 2 how the truncation of sensors at the boundaries of structures reduces their rolloff rate to $1/k$ for simple truncation, and $1/k^2$ for 'folded' truncation. Thus, even though the interior and untruncated sensors provide fast rolloff, the truncated ones do not. This effect is exacerbated by the fact that the tip displacement is often most sensitive to the measurements reported by the sensors closest to the root.

The high frequency jitter seen in the error is a result of root curvature estimates which do not roll off with frequency quickly enough. As mentioned earlier, the sign of the root curvature alternates with respect to the tip displacement. Since integration schemes integrate from the root out to the tip of the beam, tip displacement estimates are most sensitive to the root curvature. When the interior sensors roll off for high frequency modes, most of the tip displacement estimate is generated by the measured

root curvature. For high frequency modes, root curvature measurements do not roll off quickly enough because they are typically made by truncated sensors which have reduced rolloff. Accurate root curvature measurements are necessary in order to accurately resolve low frequency mode shapes.

The low frequency oscillations in the error are caused by poor rolloff of the last few sensors. This was shown in plots of estimated displacement for specific modes. As with root curvature, measurements of curvature near the beam tip are necessary for accurate estimation of the shape of low frequency modes. However, such measurements are made by sensors close enough to the beam boundary to require spatial truncation. The rolloff of the sensors is therefore significantly reduced, and a significant oscillation in tip displacement estimation error is typically observed.

The results for the shape estimation of the dynamic mode shapes of a clamped-free beam contrast with the excellent results obtained for the pinned-pinned beam. The presence of evanescent exponential strain fields at the beam boundaries, and the difficulty in obtaining quick rolloff of root curvature observability make it impossible to obtain results which satisfy the functional requirements. It is clear that no combination of sensor weighting, truncation method and integration rule produced results which satisfy the functional requirements.

Chapter 6

Conclusions

6.1 Shaped Sensors

It was shown in Chapter 2 how traditional temporal sampling window design techniques can be used to design the spatial weighting shapes of strain averaging sensors. Since the mode shapes of beam-like structures generally contain spatially sinusoidal and exponential components, the performance of sensors is analyzed for spatially sinusoidal and exponential strain fields.

The spatial filtering performance of a shaped sensor is contained in its transfer function. The transfer function of such a sensor describes the behavior of the sensor output for strain fields of different spatial frequencies. The transfer function is defined as the sensor output divided by the point value of the field at the center of the sensor. The transfer function for *any* sensor of finite length rolls off at least as fast as $1/k$, whereas the transfer function for a point sensor does not roll off at all.

It was shown that the smoother the sensor weighting is at its ends, and therefore the more derivatives are zero at these points, the higher the eventual rolloff rate of the sensor is, and the longer the sensor needs to be. It is seen that the spatially infinite weightings offer the best performance due to their infinite smoothness at the

ends, and infinite length.

Among the finite weightings considered, the Bartlett sensor is the simplest and can be manufactured easily. It satisfies Functional Requirements #1–3 and #6–8, although it only exhibits second order rolloff. The best performance for a spatially finite sensor is obtained with the Gauss-Hanning weighting. The weighting is positive everywhere, satisfying Functional Requirement #8, and finite, satisfying Functional Requirement #7. The transfer function rolls off quickly, satisfying Functional Requirement #2 while accurately sensing low frequency modes. Although the transfer function of the sensor is negative in some regions, violating Functional Requirement #3, the magnitude of the transfer function at these points is less than -120 dB.

At the boundaries of a structure, the sensor weightings must be truncated in some fashion. It was shown that simple truncation at the structure boundary leads to a change in rolloff rate, reducing it to -20 dB/decade. This is because the taper of the weighting at the structure boundary is no longer smooth. In addition, the sensor weighting is no longer symmetric, and the transfer function will no longer be bounded. This is because the sensor can report nonzero strain even though the strain at its center is zero.

To obtain better rolloff performance from sensors that must be truncated at the boundary of a structure, the truncated outboard portion should be "folded over" the boundary and subtracted from the remaining inboard weighting portion, as described in Section 2.4.5. This guarantees that the sensor weighting drops to zero at the boundary, thus allowing the sensor transfer function to roll off at -40 dB/decade. In addition, for a spatially sinusoidal strain field, the performance of a sensor truncated in such a fashion is the same as for an untruncated sensor.

For strain fields which are spatially exponential, we are no longer dealing with a strain field that is spatially periodic. Therefore, traditional ideas of rolloff and a

transfer function which maps a periodic input to a periodic output are no longer valid. However, some parallels can still be drawn.

6.2 Numerical Integration Schemes

As mentioned in Chapter 3, there is no perfect integration scheme. Generally, in order to increase accuracy, more constraints must be imposed on the location of the sample points, and more information about the function to be integrated is required.

Of the numerical integration schemes studied, only the midpoint and structural shape function fitting schemes do not require end point data, and do not have prescribed abscissas (points where the function to be integrated must be known). The structural shape function fitting schemes in general offer much better performance than the midpoint rule. For static integration purposes, structural shape function fitting schemes are therefore recommended as good candidate integration rules when structural shape functions of the structure are available.

For other integration rules that do not make use of structural information, as the locations of the sensors are more and more constrained, the accuracy of the integration scheme typically increases. In addition, for all rules, the accuracy generally increases as structural displacement information is used over a larger area of the structure when producing a sensor measurement. This is seen in the high level of accuracy exhibited by the cubic spline, Chebyshev, Gauss, Radau and structural shape function fitting schemes.

6.3 Accuracy of Numerical Integration Schemes

It was shown in Chapter 4 that two or three sensors are typically not enough to guarantee good tip deflection estimation performance for a statically loaded clamped-free beam. It is possible, however, to obtain a tip deflection estimation error under 1%

by using 2 rectangular sensors and the Global Static Shape Function integration scheme. When four or five sensors are used, however, tip deflection estimation errors are below 2% for all but the midpoint and B-spline integration schemes. In addition, and many of the more accurate integration schemes yield errors under 1%. This shows that the convergence rate is generally quite quick.

It is shown that the trapezoidal rule performs well and is very simple to implement. The integration rules that fit structural shape functions to the sensor measurements, either exactly (the GSSF and GDSF schemes), or in a least squares sense (the LSGSSF-n and LSGDSF-n schemes), offer a number of advantages over the trapezoidal rule. Measurements can be made anywhere in the interval, and end point measurements are not required. In addition, the boundary conditions the structure is subject to are incorporated automatically, as are the possible deformation patterns it could exhibit. Finally, all integration is done *a priori*, and only a curve fitting algorithm is required.

The standard deviation of the tip deflection estimation error due to uncertainties in the sensor gain or placement are approximately the same for all the integration rules that do not fit structural shape functions. For the structural shape function fitting algorithms, however, the standard deviation of the error is found to be significantly smaller. In addition, it is seen that for all integration schemes, the standard deviation found using rectangular or Bartlett sensors is slightly smaller than that exhibited by point sensors. This is due to the inherent filtering properties of rectangular and Bartlett sensors.

As the number of sensors is increased, the absolute error due only to the integration scheme decreases with a convergence rate that depends on the accuracy of the scheme. At some point the standard deviation of the error will exceed the integration scheme error. How quickly this point is reached is an important consideration in the selection of an integration rule and sensor weighting for static shape estimation.

6.4 Estimation of the Dynamic Mode Shapes of Beams

From the simulation results for a pinned-pinned beam it is clear that the error trend is well-behaved when the sensors used in the array have transfer functions for sinusoidal strain fields that roll off quickly. Point sensors do not offer any rolloff, and it can be seen that the observability increases steadily with frequency. Rectangular sensors only offer enough rolloff to keep the observability bounded, whereas Bartlett sensors allow the observability to roll off. Only by using an array of Gauss-Hanning sensors is it possible to obtain an observability which rolls off quickly with frequency. The 'folding' truncation of the Gauss-Hanning sensors at the boundaries of the beam is critical to the performance of the array. Without this truncation, the rolloff of the sensors would be reduced, and the monotonic error trend would not be guaranteed.

The simulations for the pinned-pinned beam also demonstrate that out of the four integration schemes selected, the trapezoidal rule generates the most well-behaved tip displacement estimation errors. This is because the measured curvature for high frequency modes appears as a sawtooth pattern along the beam. An array of Gauss-Hanning sensors in combination with the trapezoidal rule allows accurate estimation of the first few mode shapes of a pinned-pinned beam, and exhibits quick rolloff in the observability of higher modes by a monotonic increase in error from 0% to 100%. The sensor system accurately resolves the beam shape for low frequency modes, and thus satisfies Functional Requirement #4. This means that Functional Requirement #1 is also satisfied because each sensor is accurately measuring curvature for such modes. The observability of each sensor and the system as a whole rolls off quickly in frequency, satisfying Functional Requirements #2 and #5. As discussed in Chapter 2, Functional Requirement #3 is not strictly satisfied, because the transfer function of the Gauss-Hanning sensor contains negative regions. However, due to the fact that over -140 dB of rolloff occurs before the first negative region of the transfer functions,

uncertainties inherent in any physical implementation will introduce more significant errors. Functional Requirement #6 is satisfied because the behavior of the error is virtually monotonic. Finally, Functional Requirements #7, #8 and #9 are satisfied by the design of the Gauss-Hanning sensor array.

As mentioned in Chapter 5, it seems impossible to construct an array of sensors for the shape estimation of a clamped-free beam that satisfy the functional requirements. For low frequency modes, accurate measurements of curvature close to the root and tip of the beam are required. For high frequency modes, such measurements should roll off quickly with frequency. Since boundary curvature measurements must begin to roll off at relatively low frequency, a sensor whose length is a significant fraction of the length of the beam is required. Long sensors placed near the boundaries will require spatial truncation, and such spatial truncation is detrimental to the rolloff characteristics of the untruncated sensor.

Although displacement estimation performance does not appear to be satisfactory, it must be noted that the set of functional requirements introduced in Chapter 1 are very demanding. Since beam curvature goes as k^2 relative to the displacement, it becomes increasingly difficult to roll off the sensor array observability for high frequency modes. Although the error may appear to be significant, it is relative to the tip displacement at that frequency. In a physical implementation, the beam curvature is likely to be constant as the mode number is increased, because it is limited by the system energy. Tip displacement will therefore drop off as $1/k^2$, and displacements for high frequency mode shapes will be dwarfed by those for low frequency modes. Similarly, tip displacement estimation errors for high frequency modes will seem less significant when they are combined with tip displacement estimation errors for low frequency modes.

References

[1] ABRAMOWITZ M., AND I. STEGUN, 1972. *Handbook of Mathematical Functions*, Washington, DC: U.S. Government Printing Office.

[2] AHLBERG, J. H., E. N. NILSON, AND J. L. WALSH, 1967. *The Theory of Splines and Their Applications*, New York, NY: Academic Press, Inc.

[3] ANDERSSON, M. S., AND E. F. CRAWLEY, 1991. "Discrete Distributed Strain Sensing of Intelligent Structures," *Proceedings of the Second Annual U.S./Japan Conference on Adaptive Structures*, Nagoya, Japan, November 12–14, 1991.

[4] ANDERSSON, M. S., AND E. F. CRAWLEY, 1992. "Discrete Shaped Strain Sensors for Intelligent Structures," AIAA paper No. 92-2046, *Proceedings of the 33rd AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, Dallas, TX, April 13–15, 1992.

[5] ASHBY, M. F. AND D. R. H. JONES, 1980. *Engineering Materials 1*, New York, NY: Pergamon Press.

[6] BALAS, M. J., 1985. "Optimal Quasi-Static Shape Control for Large Aerospace Antennae", *Journal of Optimization Theory and Applications*, Vol. 46, No. 2, June 1985, pp. 153–170.

[7] BARTELS, RICHARD H., JOHN C. BEATTY, AND BRIAN A. BARSKY, 1987. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*, Los Altos, California: Morgan Kaufman Publishers, Inc.

[8] BEERS, YARDLEY, 1957. *Introduction to the Theory of Error*, Reading, MA: Addison-Wesley Publishing, Inc.

[9] BODE, HENDRIK WADE, 1945. *Network analysis and feedback amplifier design*, New York, NY: Van Nostrand.

[10] DE BOOR, CARL, 1978. *A Practical Guide to Splines*, Applied Mathematical Sciences Volume 27, New York, NY: Springer-Verlag.

[11] CAVALIERI, 1639. *Centuria di Varii Problemi*, Bologna, Italy, p. 446.

[12] CLIFFORD, A. A., 1973. *Multivariate Error Analysis*, New York, NY: John Wiley & Sons, Inc.

[13] COLLINS, S. A., C. E. PADILLA, E. SCHMITZ, R. J. NOTESTINE, M. RAMEY AND A. H. VON FLOTOW, 1990. "Design, Manufacture, and Application to Space Robotics of Distributed Piezoelectric Film Sensors," *Proceedings of the 31st AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, Long Beach, CA, April 2–4, 1990, pp. 1899–1906.

[14] CRAWLEY, E. F., AND E. H. ANDERSON, 1990. "Detailed Modeling of Piezoelectric Actuators of Beams," *Journal of Intelligent Material Systems and Structures*, Vol. 1, No. 1, January 1990, pp. 4–25.

[15] CRAWLEY, E. F., AND J. DE LUIS, 1987. "Use of Piezoelectric Actuators as Elements of Intelligent Structures", *AIAA Journal*, Vol. 25, No. 10, AIAA Paper 86-0878, October 1987, pp. 1373–1385.

[16] DAVIS, PHILIP J., AND PHILIP RABINOWITZ, 1984. *Methods of Numerical Integration*, Orlando, FL: Academic Press, Inc.

[17] EDWARDS, C. H., AND DAVID E. PENNEY, 1985. *Elementary Differential Equations with Applications*, Englewood Cliffs, NJ: Prentice-Hall, Inc.

[18] ENGELS, HERMANN, 1980. *Numerical Quadrature and Cubature*, New York, NY: Academic Press, Inc.

[19] EUBANK, RANDALL L., 1988. *Spline Smoothing and Nonparametric Regression*, New York, NY: M. Decker.

[20] EVANS, D. J. (Ed.), 1974. *Software for Numerical Mathematics*, New York, NY: Academic Press, Inc.

[21] DEFATTA, DAVID J., JOSEPH G. LUCAS, AND WILLIAM S. HODGKISS, 1988. *Digital Signal Processing: A System Design Approach*, New York, NY: John Wiley & Sons, Inc.

[22] FLETCHER, F., J. C. P. MILLER AND L. ROSENHEAD, 1946. *An Index of Mathematical Tables*, New York, NY: McGraw-Hill.

[23] GADANGI, ASHOK K., 1989. Non Uniform Rational B-Splines in Solid Modeling, *Cadence*, Vol. 4, No. 2, February, 1989, pp. 68–72.

[24] GAUSS, K. F., "Methodus nova integralium valores per approx. inveniendi", *Gött. Comm.*, III. (1814)=*Werke*, **3**, p. 163.

[25] GHIZZETTI, A. AND A. OSSICINI, 1970. *Quadrature Formulae*, New York, NY: Academic Press, Inc.

[26] GREGORY, JAMES, 1668. *Exercit. Geom.*, London, England.

[27] HAGOOD, N. W., E. F. CRAWLEY, J. DE LUIS AND E. H. ANDERSON, 1989. "Development of Integrated Components for Control of Intelligent Structures,", *Proceedings, Smart Materials, Structures, and Mathematical Issues*, (Rogers, C. A., ed.), 1989, pp. 80–104.

[28] HALL S. R., E. F. CRAWLEY, J. HOW AND B. WARD, 1991. "A Hierarchic Control Architecture for Intelligent Structures," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 14, No. 3, May–June 1991.

[29] HARTREE, D. R., 1958. *Numerical Analysis*, 2nd Ed., Oxford, England: Clarendon Press.

[30] HILDEBRAND, F. B., 1956. *Introduction to Numerical Analysis*, New York, NY: McGraw-Hill, Inc.

[31] HOLLAND, A. S. B., 1979. *The General Problem of Approximation and Spline Functions*, Huntington, NY: R. E. Krieger Publishing Company.

[32] JOHNSTON, R. L., 1982. *Numerical Methods: A Software Approach*, New York, NY: John Wiley & Sons, Inc.

[33] KARLIN, S., ET. AL., 1976. *Studies in Spline Functions and Approximation Theory*, New York, NY: Academic Press, Inc.

[34] KASHIWASE, TOSHIO, MASAKI TABATA AND KAZUO TSUCHIYA, 1991. "Shape Control of Flexible Structures", *Journal of Intelligent Material Systems and Structures*, Vol. 2, January 1991, pp. 110–125.

[35] KELLY, LOUIS G., 1967. *Handbook of Numerical Methods and Applications*, Reading, MA: Addison-Wesley Publishing Company, Inc.

[36] KRONROD, A. S., 1965. *Nodes and Weights of Quadrature Formulas*, New York, NY: Consultants Bureau.

[37] KUNT, MURAT, 1986. *Digital Signal Processing*, Norwood, MA: Artech House, Inc.

[38] LEE, C. -K., W. -W. CHIANG AND T. C. O'SULLIVAN, 1989. "Piezoelectric Modal Sensors and Actuators Achieving Critical Damping on a Cantilever Plate," *Proceedings of the* 30^{th} *AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, Mobile, AL, April 3–5, 1989, pp. 2018–2026.

[39] LEE, C. -K. AND F. C. MOON, June 1990. "Modal Sensors/Actuators," *Journal of Applied Mechanics*, Vol. 57, No. 2.

[40] DE LUIS, J., AND E. F. CRAWLEY, 1990. "Experimental Results of Active Control on a Prototype Intelligent Structure," Proceedings of the 31^{st} AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Long Beach, CA, April 2–4, 1990, pp. 2340–2350.

[41] MEIROVITCH, L. AND H. BARUH, 1985. "The Implementation of Modal Filters for Control of Structures", *AIAA Journal of Guidance, Control and Dynamics*, Vol. 8, No. 6, November–December 1985, pp. 707–716.

[42] MILLER, D. W., S. A. COLLINS, AND S. P. PELTZMAN, 1990. "Development of Spatially Convolving Sensors for Structural Control Applications," *Proceedings of the 31^{st} AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, Long Beach, CA, April 2–4, 1990, pp. 2283–2297.

[43] NURNBERGER, GUNTHER, 1989. *Approximation by Spline Functions*, Berlin, Germany: Springer-Verlag.

[44] OPPENHEIM, ALAN V., AND RONALD W . SCHAFER, 1975. *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, Inc.

[45] PINES, D. AND A. H. VON FLOTOW, 1990. "Active Control of Bending Wave Propagation at Acoustic Frequencies," *Journal of Sound and Vibration*, Vol. 143, No. 3, November 8, 1990, pp. 391–412.

[46] PRESS, WILLIAM H., BRIAN P. FLANNERY, SEUL A. TEUKOLSKY, AND WILLIAM T. VETTERLING, 1986. *Numerical Recipes*, New York, NY: Cambridge University Press.

[47] ROBERTS, RICHARD A., AND CLIFFORD T. MULLIS, 1987. *Digital Signal Processing*, Reading, MA: Addison-Wesley Publishing Company.

[48] SARD, ARTHUR, 1971. *A Book of Splines*, New York, NY: John Wiley & Sons, Inc.

[49] SCHUMAKER, LARRY L., 1981. *Spline Functions: Basic Theory*, New York, NY: John Wiley & Sons, Inc.

[50] SIEBERT, WILLIAM McC, 1986. *Circuits, Signals and Systems*, Cambridge, MA: The MIT Press.

[51] SIMPSON, THOMAS, 1743. *Math. Dissertations*, London, England, p. 109.

[52] SPATH, HELMUTH, 1974. *Spline Algorithms for Curves and Surfaces*, Winnipeg, Canada: Utilitas Mathematica Publishing.

[53] STRANG, GILBERT, 1986. *Introduction to Applied Mathematics*, Wellesley, MA: Wellesley-Cambridge Press.

[54] STROUD, A. H., 1961. *A Bibliography on Approximate Integration*, in *Mathematics of Computation*, **15**,: pp. 52–80.

[55] STROUD, A. H., AND DON SECREST, 1966. *Gaussian Quadrature Formulas*, Englewood Cliffs, NJ: Prentice-Hall.

[56] WADA, B. K., J. L. FANSON, AND E. F. CRAWLEY, 1990. "Adaptive Structures", *Journal of Intelligent Material Systems and Structures*, Vol. 1, April 1990, pp. 157–174.

[57] WAHBA, GRACE, 1990. *Spline Modes for Observational Data*, Philadelphia, PA: Society for Industrial and Applied Mathematics.

[58] WARKENTIN, D. J. AND E. F. CRAWLEY, 1991. "Embedded Electronics for Intelligent Structures," AIAA paper No. 91-1084, *Proceedings of the* 32^{nd} *AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, Baltimore, MD, April 8–10, 1991.

[59] WHITTAKER, E. W. AND G. ROBINSON, 1944. *The Calculus of Observations: A Treatise on Numerical Mathematics*, 4th Ed., London, England: Blackie & Son Ltd.

[60] YARNELL, J., 1951. *Resistance Strain Gages*, London, England: Electrical Engineering.

Appendix A

Derivation of Sensor Output and Transfer Functions

In this appendix, the output and transfer functions for spatially averaging strain sensors are derived in detail. These were discussed in Chapter 2. The behavior of sensors in strain fields that are spatially sinusoidal and spatially exponential are considered. Sensors centered at the point of symmetry or origin of the strain field are first investigated. The sensors are then assumed to be at a known distance x_0 from the point of symmetry or origin of the strain field. Derivations are then given for both nontruncated sensors and sensors that must be partially truncated at the boundaries of a structure.

A.1 Sinusoidal Curvature

In this section the output and transfer functions for sensors that are centered, noncentered, simply truncated and truncated by the 'folding' technique are derived when the strain field is spatially sinusoidal or cosinusoidal. In the following analysis, the exponential strain field is assumed to have the form $\sin(kx)$ or $\cos(kx)$. The point of symmetry will refer to the point where the strain remains the same no matter what the spatial frequency is. For these strain fields, this point is simply $x = 0$, where the

strain is zero or unity for all k.

A.1.1 Centered Sensors

Assuming that the spatial weighting function $f(x)$ of the spatially averaging sensor is continuous, has first derivative continuity, and is nonzero over the region $-l/2 < x < l/2$, its output can be found from

$$y(k) = \int_{-l/2}^{l/2} f(x)\ \cos(kx)\, dx \tag{A.1}$$

Integration by parts of this expression yields

$$\begin{aligned} y(k) &= \frac{1}{k} f(x)\sin(kx)\Big|_{-l/2}^{l/2} \\ &- \int_{-l/2}^{l/2} \frac{1}{k} f'(x)\sin(kx)\, dx \end{aligned} \tag{A.2}$$

$$\begin{aligned} y(k) &= \frac{1}{k}\Big[f(l/2)\sin(kl/2) - f(-l/2)\sin(-kl/2)\Big] \\ &- \int_{-l/2}^{l/2} \frac{1}{k} f'(x)\sin(kx)\, dx \end{aligned} \tag{A.3}$$

$$\begin{aligned} y(k) &= \frac{1}{k}\Big[f(l/2) + f(-l/2)\Big]\sin(kl/2) \\ &- \int_{-l/2}^{l/2} \frac{1}{k} f'(x)\sin(kx)\, dx \end{aligned} \tag{A.4}$$

Integration by parts three more times yields

$$\begin{aligned} y(k) &= \frac{1}{k}\Big[f(l/2)+f(-l/2)\Big]\sin(kl/2) \\ &+ \frac{1}{k^2}\Big[f'(l/2)-f'(-l/2)\Big]\cos(kl/2) \\ &+ \frac{1}{k^3}\Big[-f''(l/2)-f''(-l/2)\Big]\sin(kl/2) \\ &+ \frac{1}{k^4}\Big[-f'''(l/2)+f'''(-l/2)\Big]\cos(kl/2) \\ &+ \int_{-l/2}^{l/2}\frac{1}{k^4}f''''(x)\cos(kx)\,dx \end{aligned} \tag{A.5}$$

A pattern can be identified in the resulting expression, and an infinite series solution for the output and transfer function of a sensor mounted in the interior of the structure (away from its boundaries), centered at the point of symmetry in a cosinusoidal strain field can be written as

$$\begin{aligned} y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{2i+1}}\Big[f^{(2i)}(l/2)+f^{(2i)}(-l/2)\Big]\sin(kl/2) \\ & + \frac{(-1)^i}{k^{2i+2}}\Big[f^{(2i+1)}(l/2)-f^{(2i+1)}(-l/2)\Big]\cos(kl/2)\Bigg\} \end{aligned} \tag{A.6}$$

where $f^{(n)}(x)$ represents the n-th derivative of the function $f(x)$, $d^n f/dx^n$.

Typically, the weighting function $f(x)$ is symmetric. In this case, we have

$$f(x) = f(-x) \tag{A.7}$$

$$f'(x) = -f'(-x) \tag{A.8}$$

$$f''(x) = f''(-x) \tag{A.9}$$

$$f'''(x) = -f'''(-x) \tag{A.10}$$

Or, in general,

$$f^{(2i)}(x) = f^{(2i)}(-x) \tag{A.11}$$

$$f^{(2i+1)}(x) = -f^{(2i+1)}(-x) \tag{A.12}$$

By using Equations (A.11) and (A.12), Equation (A.6) can be simplified to

$$y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{2(-1)^i}{k^{2i+1}} f^{(2i)}(l/2)\sin(kl/2) + \frac{2(-1)^i}{k^{2i+2}} f^{(2i+1)}(l/2)\cos(kl/2) \right\} \tag{A.13}$$

Since the weighting function $f(x)$ must have a continuous derivative for (A.6) to be valid, the discontinuity in the Bartlett and Bartlett-Bartlett weightings must be taken into account. For example, for a weighting function with a discontinuity at $x = 0$, the integral (A.1) becomes

$$y(k) = \int_{-l/2}^{0^-} f(x)\ \cos(kx)\, dx + \int_{0^+}^{l/2} f(x)\ \cos(kx)\, dx \tag{A.14}$$

Equation (A.6) is therefore modified, and the result is

$$\begin{aligned} y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{2i+1}} \left[f^{(2i)}(-l/2)\sin(kl/2) \right] \\ & + \frac{(-1)^i}{k^{2i+2}} \left[-f^{(2i+1)}(-l/2)\cos(kl/2) + f^{(2i+1)}(0^-) \right] \\ & + \frac{(-1)^i}{k^{2i+1}} \left[f^{(2i)}(l/2)\sin(kl/2) \right] \\ & + \frac{(-1)^i}{k^{2i+2}} \left[-f^{(2i+1)}(0^+) + f^{(2i+1)}(l/2)\cos(kl/2) \right] \Bigg\} \end{aligned} \tag{A.15}$$

where $x = 0^-$ is a point just before the discontinuity, and $x = 0^+$ is a point just after

the discontinuity. By grouping similar terms, this expression further simplifies to

$$y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{2i+1}} \Big[f^{(2i)}(-l/2) + f^{(2i)}(l/2) \Big] \sin(kl/2) + \frac{(-1)^i}{k^{2i+2}} \Big[f^{(2i+1)}(0^-) - f^{(2i+1)}(0^+) + \Big[-f^{(2i+1)}(-l/2) + f^{(2i+1)}(l/2) \Big] \cos(kl/2) \Big] \right\} \tag{A.16}$$

If it is assumed that $f(x)$ is symmetric, (A.16) simplifies to

$$y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{2(-1)^i}{k^{2i+1}} f^{(2i)}(-l/2) \sin(kl/2) + \frac{2(-1)^i}{k^{2i+2}} \Big[f^{(2i+1)}(l/2) \cos(kl/2) - f^{(2i+1)}(0^+) \Big] \right\} \tag{A.17}$$

A.1.2 Noncentered Sensors

Here the output of a sensor mounted in the interior of a structure (away from its boundaries), centered at a distance x_0 from the point of symmetry in a sinusoidal strain field is investigated. It is assumed that the strain field is of the form

$$\epsilon(x) = \sin(kx) \tag{A.18}$$

where k is a parameter that depends on mode number. The output of a sensor with weighting function $f(x)$, centered at $x = x_0$ is then given by

$$y(k) = \int_{-l/2+x_0}^{l/2+x_0} f(x - x_0)\ \sin(kx)\, dx \tag{A.19}$$

Integration by parts of this expression yields

$$\begin{aligned} y(k) &= -\frac{1}{k} f(x-x_0)\cos(kx)\Big|_{-l/2+x_0}^{l/2+x_0} \\ &+ \int_{-l/2+x_0}^{l/2+x_0} \frac{1}{k} f'(x-x_0)\cos(kx)\,dx \qquad \text{(A.20)} \\ y(k) &= \frac{1}{k}\Big[-f(l/2)\cos\Big[k(l/2+x_0)\Big] + f(-l/2)\cos\Big[k(-l/2+x_0)\Big]\Big] \\ &+ \int_{-l/2+x_0}^{l/2+x_0} \frac{1}{k} f'(x-x_0)\cos(kx)\,dx \qquad \text{(A.21)} \end{aligned}$$

Integration by parts three more times yields

$$\begin{aligned} y(k) &= \frac{1}{k}\Big[-f(l/2)\cos\Big[k(-l/2+x_0)\Big] + f(-l/2)\cos\Big[k(l/2+x_0)\Big]\Big] \\ &+ \frac{1}{k^2}\Big[f'(l/2)\sin\Big[k(-l/2+x_0)\Big] - f'(-l/2)\sin\Big[k(l/2+x_0)\Big]\Big] \\ &+ \frac{1}{k^3}\Big[f''(l/2)\cos\Big[k(-l/2+x_0)\Big] - f''(-l/2)\cos\Big[k(l/2+x_0)\Big]\Big] \\ &+ \frac{1}{k^4}\Big[-f'''(l/2)\sin\Big[k(-l/2+x_0)\Big] + f'''(-l/2)\sin\Big[k(l/2+x_0)\Big]\Big] \\ &+ \int_{-l/2+x_0}^{l/2+x_0} \frac{1}{k^4} f''''(x-x_0)\sin(kx)\,dx \qquad \text{(A.22)} \end{aligned}$$

A pattern can be clearly seen in the resulting expression, and an infinite series solution for the output of a sensor mounted in the interior of a structure (away from its boundaries), centered at a distance x_0 from the point of symmetry in a sinusoidal strain field can be written as

$$\begin{aligned} y(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{(2i+1)}}\Big[-f^{(2i)}(l/2)\cos\Big[k(l/2+x_0)\Big] + f^{(2i)}(-l/2)\cos\Big[k(-l/2+x_0)\Big]\Big] \\ & + \frac{(-1)^i}{k^{(2i+2)}}\Big[f^{(2i+1)}(l/2)\sin\Big[k(l/2+x_0)\Big] - f^{(2i+1)}(-l/2)\sin\Big[k(-l/2+x_0)\Big]\Big]\Bigg\} \end{aligned} \qquad \text{(A.23)}$$

To obtain the transfer function from the strain at the center of the sensor to its

output, (A.23) must be divided by the strain at the center of the sensor, $\sin(kx_0)$:

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \left[\frac{-f^{(2i)}(l/2)\cos\left[k(l/2+x_0)\right] + f^{(2i)}(-l/2)\cos\left[k(-l/2+x_0)\right]}{\sin(kx_0)} \right] + \frac{(-1)^i}{k^{(2i+2)}} \left[\frac{f^{(2i+1)}(l/2)\sin\left[k(l/2+x_0)\right] - f^{(2i+1)}(-l/2)\sin\left[k(-l/2+x_0)\right]}{\sin(kx_0)} \right] \right\} \tag{A.24}$$

This expression can be simplified by expanding the $\sin[k(x_0 \pm l/2)]$ and $\cos[k(x_0 \pm l/2)]$ terms. The four identities

$$\sin(\theta+\phi) = \sin\theta\ \cos\phi + \cos\theta\ \sin\phi \tag{A.25}$$

$$\sin(\theta-\phi) = \sin\theta\ \cos\phi - \cos\theta\ \sin\phi \tag{A.26}$$

$$\cos(\theta+\phi) = \cos\theta\ \cos\phi - \sin\theta\ \sin\phi \tag{A.27}$$

$$\cos(\theta-\phi) = \cos\theta\ \cos\phi + \sin\theta\ \sin\phi \tag{A.28}$$

are used, and (A.24) simplifies to

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \left[-f^{(2i)}(l/2) \left(\frac{\cos(kx_0)\cos(kl/2) - \sin(kx_0)\sin(kl/2)}{\sin(kx_0)} \right) + f^{(2i)}(-l/2) \left(\frac{\cos(kx_0)\cos(kl/2) + \sin(kx_0)\sin(kl/2)}{\sin(kx_0)} \right) \right] + \frac{(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \left(\frac{\sin(kx_0)\cos(kl/2) + \cos(kx_0)\sin(kl/2)}{\sin(kx_0)} \right) - f^{(2i+1)}(-l/2) \left(\frac{\sin(kx_0)\cos(kl/2) + \cos(kx_0)\sin(kl/2)}{\sin(kx_0)} \right) \right] \right\} \tag{A.29}$$

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \left[-f^{(2i)}(l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} - \sin(kl/2) \right) \right. \right.$$
$$\left. + f^{(2i)}(-l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} + \sin(kl/2) \right) \right]$$
$$+ \frac{(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \left(\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right.$$
$$\left. \left. - f^{(2i+1)}(-l/2) \left(-\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right] \right\} \tag{A.30}$$

It is now shown that if the sensor if symmetric, (A.30) simplifies to (A.13), showing that the transfer function between the strain at the center or the sensor and its output is independent of where the sensor is centered and what x_0 is. The identities in (A.11) and (A.12) are used and (A.30) simplifies to:

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \left[-f^{(2i)}(l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} - \sin(kl/2) \right) \right. \right.$$
$$\left. + f^{(2i)}(l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} + \sin(kl/2) \right) \right]$$
$$+ \frac{(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \left(\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right.$$
$$\left. \left. + f^{(2i+1)}(l/2) \left(-\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right] \right\} \tag{A.31}$$

which immediately simplifies to

$$y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{2(-1)^i}{k^{2i+1}} f^{(2i)}(l/2) \sin(kl/2) + \frac{2(-1)^i}{k^{2i+2}} f^{(2i+1)}(l/2) \cos(kl/2) \right\} \tag{A.32}$$

which is exactly the same as (A.13). Note that although the expressions (A.13)

and A.32 the same, the analysis for centered sensors was done assuming a cosinusoidal strain field, and the analysis for noncentered sensors was done assume a strain field of sinusoidal form. It therefore does not make a difference which of the two strain field forms is assumed. Since the above analysis assumes that $f(x)$ has no derivative discontinuities, we now show that the same property holds for a weighting with a discontinuity at its center, such as a Bartlett weighting. To begin, the integral (A.19) is modified:

$$y(k) = \int_{-l/2+x_0}^{x_0^-} f(x - x_0)\sin(kx)\,dx + \int_{x_0^+}^{l/2+x_0} f(x - x_0)\sin(kx)\,dx \qquad \text{(A.33)}$$

where x_0^- is a point just before the discontinuity in $f(x)$, and x_0^+ is a point just after it. (A.33) is integrated repeatedly to obtain:

$$\begin{aligned} y(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{(2i+1)}} \Big[-f^{(2i)}(0^-)\cos(kx_0) + f^{(2i)}(-l/2)\cos\big[k(-l/2+x_0)\big] \\ & \qquad -f^{(2i)}(l/2)\cos\big[k(l/2+x_0)\big] + f^{(2i)}(0^+)\cos(kx_0) \Big] \\ & + \frac{(-1)^i}{k^{(2i+2)}} \Big[f^{(2i+1)}(0^-)\sin(kx_0) - f^{(2i+1)}(-l/2)\sin\big[k(-l/2+x_0)\big] \\ & \qquad +f^{(2i+1)}(l/2)\sin\big[k(l/2+x_0)\big] - f^{(2i+1)}(0^+)\sin(kx_0) \Big] \Bigg\} \end{aligned} \qquad \text{(A.34)}$$

Since $f(x)$ is symmetric, this simplifies to

$$y(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \left[f^{(2i)}(l/2) \Big(\cos\big[k(-l/2 + x_0)\big] - \cos\big[k(l/2 + x_0)\big] \Big) \right] \right.$$
$$+ \frac{(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \Big(\sin\big[k(-l/2 + x_0)\big] + \sin\big[k(l/2 + x_0)\big] \Big) \right.$$
$$\left. \left. -2f^{(2i+1)}(0^+) \sin(kx_0) \right] \right\} \tag{A.35}$$

By expanding the trigonometric terms, we are left with

$$y(k) = \sum_{i=0}^{\infty} \left\{ \frac{2(-1)^i}{k^{(2i+1)}} \left[f^{(2i)}(l/2) \sin(kl/2) \right] \sin(kx_0) \right.$$
$$\left. + \frac{2(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \cos(kl/2) - f^{(2i+1)}(0^+) \right] \sin(kx_0) \right\} \tag{A.36}$$

To obtain the transfer function, we simply divide the output (A.36) by the strain at the center of the sensor, $\sin(kx_0)$, to obtain

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{2(-1)^i}{k^{2i+1}} f^{(2i)}(-l/2) \sin(kl/2) \right.$$
$$\left. + \frac{2(-1)^i}{k^{2i+2}} \left[f^{(2i+1)}(l/2) \cos(kl/2) - f^{(2i+1)}(0^+) \right] \right\} \tag{A.37}$$

which is clearly the same as (A.17), thus showing that the transfer function of a weighting with a derivative discontinuity is independent of x_0, and therefore where it is placed relative to the point of symmetry in the strain field.

A.1.3 Sensors Simply Truncated at Structure Boundaries

Here the output of a sensor mounted near a boundary of a structure, at a distance x_0 from the point of symmetry in a sinusoidal strain field is investigated. It is again assumed that the strain field is of the form

$$\epsilon(x) = \sin(kx) \tag{A.38}$$

where k is a parameter that depends on mode number. It is assumed that the boundary of a structure is at $x = 0$. The output of a sensor with weighting function $f(x)$, centered at $x = x_0$ is then given by

$$y(k) = \int_0^{l/2+x_0} f(x - x_0)\sin(kx)\, dx \tag{A.39}$$

Integration by parts of this expression yields

$$\begin{aligned} y(k) &= -\frac{1}{k} f(x - x_0)\cos(kx)\Big|_0^{l/2+x_0} \\ &+ \int_0^{l/2+x_0} \frac{1}{k} f'(x - x_0)\cos(kx)\, dx \end{aligned} \tag{A.40}$$

$$\begin{aligned} y(k) &= \frac{1}{k}\Big[-f(l/2)\cos\Big[k(l/2 + x_0)\Big] + f(-x_0)\Big] \\ &+ \int_0^{l/2+x_0} \frac{1}{k} f'(x - x_0)\cos(kx)\, dx \end{aligned} \tag{A.41}$$

Integration by parts three more times yields

$$\begin{aligned} y(k) &= \frac{1}{k}\Big[-f(l/2)\cos\Big[k(l/2+x_0)\Big] + f(-x_0)\Big] \\ &+ \frac{1}{k^2}\Big[f'(l/2)\sin\Big[k(l/2+x_0)\Big]\Big] \\ &+ \frac{1}{k^3}\Big[f''(l/2)\cos\Big[k(l/2+x_0)\Big] - f''(-x_0)\Big] \\ &+ \frac{1}{k^4}\Big[-f'''(l/2)\sin\Big[k(l/2+x_0)\Big]\Big] \\ &+ \int_0^{l/2+x_0} \frac{1}{k^4} f''''(x-x_0)\sin(kx)\,dx \end{aligned} \tag{A.42}$$

A pattern can clearly be seen in the resulting expression, and an infinite series solution for the output of a sensor mounted in the interior of a structure (away from its boundaries), centered at a distance x_0 from the point of symmetry in a sinusoidal strain field can be written as

$$\begin{aligned} y(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{(2i+1)}}\Big[-f^{(2i)}(l/2)\cos\Big[k(l/2+x_0)\Big] + f^{(2i)}(-x_0)\Big] \\ & + \frac{(-1)^i}{k^{(2i+2)}}\Big[f^{(2i+1)}(l/2)\sin\Big[k(l/2+x_0)\Big]\Big]\Bigg\} \end{aligned} \tag{A.43}$$

To obtain the transfer function from the strain at the center of the sensor to its output, (A.43) must be divided by the strain at the center of the sensor, $\sin(kx_0)$:

$$\begin{aligned} \mathcal{T}(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{(-1)^i}{k^{(2i+1)}}\left[\frac{-f^{(2i)}(l/2)\cos\Big[k(-l/2+x_0)\Big] + f^{(2i)}(-x_0)}{\sin(kx_0)}\right] \\ & + \frac{(-1)^i}{k^{(2i+2)}}\left[\frac{f^{(2i+1)}(l/2)\sin\Big[k(-l/2+x_0)\Big]}{\sin(kx_0)}\right]\Bigg\} \end{aligned} \tag{A.44}$$

This expression can be simplified by expanding the $\sin[k(x_0 \pm l/2)]$ and $\cos[k(x_0 \pm$

$l/2)]$ terms using Equations (A.25) and (A.28). Then (A.44) simplifies to

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \left[-f^{(2i)}(l/2) \left(\frac{\cos(kx_0)\cos(kl/2) - \sin(kx_0)\sin(kl/2)}{\sin(kx_0)} \right) + f^{(2i)}(-x_0) \left(\frac{1}{\sin(kx_0)} \right) \right] + \frac{(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \left(\frac{\sin(kx_0)\cos(kl/2) + \cos(kx_0)\sin(kl/2)}{\sin(kx_0)} \right) \right] \right\} \tag{A.45}$$

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \left[-f^{(2i)}(l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} - \sin(kl/2) \right) + f^{(2i)}(-x_0) \left(\frac{1}{\sin(kx_0)} \right) \right] + \frac{(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \left(\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right] \right\} \tag{A.46}$$

A.1.4 Sensors Folded at Structure Boundaries

In Figure 2.22 a procedure used to fold and flip the truncated part of a sensor was shown. In this section, the effect of performing this operation when the strain field is sinusoidal is investigated. As discussed in Chapter 2, the output of the modified sensor is given by

$$y(k) = \int_0^{l/2-x_0} \left[f(x - x_0) - f(-x - x_0) \right] \sin(kx)\, dx + \int_{l/2-x_0}^{l/2+x_0} f(x - x_0) \sin(kx)\, dx \tag{A.47}$$

The weighting $f(-x - x_0)$ is the normal weighting $f(x - x_0)$ reflected about the boundary of the structure at $x = 0$. The length of the portion of the sensor that extends beyond the boundary of the sensor is $l/2 - x_0$. The above equation can be

rewritten

$$y(k) = \int_0^{l/2+x_0} f(x-x_0)\sin(kx)\,dx - \int_0^{l/2-x_0} f(-x-x_0)\sin(kx)\,dx \quad \text{(A.48)}$$

$$y(k) = G(k) - H(k) \quad \text{(A.49)}$$

The first integral, $G(k)$, in (A.48) was dealt with in the previous section and the result is shown in (A.43). The second integral, $H(k)$, is integrated by parts to obtain

$$\begin{aligned} H(k) &= -\frac{1}{k} f(-x-x_0)\cos(kx)\Big|_0^{l/2-x_0} \\ &- \int_0^{l/2-x_0} \frac{1}{k} f'(-x-x_0)\cos(kx)\,dx \end{aligned} \quad \text{(A.50)}$$

$$\begin{aligned} H(k) &= \frac{1}{k}\Big[-f(-l/2)\cos\Big[k(l/2-x_0)\Big] + f(-x_0)\Big] \\ &- \int_0^{l/2-x_0} \frac{1}{k} f'(-x-x_0)\cos(kx)\,dx \end{aligned} \quad \text{(A.51)}$$

Integration by parts three more times yields

$$\begin{aligned} H(k) &= \frac{1}{k}\Big[-f(-l/2)\cos\Big[k(l/2-x_0)\Big] + f(-x_0)\Big] \\ &+ \frac{1}{k^2}\Big[-f'(-l/2)\sin\Big[k(l/2-x_0)\Big]\Big] \\ &+ \frac{1}{k^3}\Big[f''(-l/2)\cos\Big[k(l/2-x_0)\Big] - f''(-x_0)\Big] \\ &+ \frac{1}{k^4}\Big[f'''(-l/2)\sin\Big[k(l/2-x_0)\Big]\Big] \\ &+ \int_0^{l/2-x_0} \frac{1}{k^4} f''''(-x-x_0)\sin(kx)\,dx \end{aligned} \quad \text{(A.52)}$$

A pattern emerges from this expression, and $H(k)$ can be written as an infinite series in the following manner:

$$H(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \Big[-f^{(2i)}(-l/2)\cos\Big[k(l/2 - x_0)\Big] + f^{(2i)}(-x_0) \Big] \right.$$
$$\left. + \frac{(-1)^i}{k^{(2i+2)}} \Big[-f^{(2i+1)}(-l/2)\sin\Big[k(l/2 - x_0)\Big] \Big] \right\} \tag{A.53}$$

By subtracting this expression for $H(k)$ from $G(k)$, which is the output $y(k)$ in (A.43) the output of the modified sensor as a function of frequency can be found.

$$y(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \Big[-f^{(2i)}(l/2)\cos\Big[k(l/2 + x_0)\Big] + f^{(2i)}(-x_0) \right.$$
$$\left. + f^{(2i)}(-l/2)\cos\Big[k(l/2 - x_0)\Big] - f^{(2i)}(-x_0) \Big] \right.$$
$$\left. + \frac{(-1)^i}{k^{(2i+2)}} \Big[f^{(2i+1)}(l/2)\sin\Big[k(l/2 + x_0)\Big] \right.$$
$$\left. + f^{(2i+1)}(-l/2)\sin\Big[k(l/2 - x_0)\Big] \Big] \right\} \tag{A.54}$$

$$y(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \Big[-f^{(2i)}(l/2)\cos\Big[k(l/2 + x_0)\Big] + f^{(2i)}(-l/2)\cos\Big[k(l/2 - x_0)\Big] \Big] \right.$$
$$\left. + \frac{(-1)^i}{k^{(2i+2)}} \Big[f^{(2i+1)}(l/2)\sin\Big[k(l/2 + x_0)\Big] + f^{(2i+1)}(-l/2)\sin\Big[k(l/2 - x_0)\Big] \Big] \right\} \tag{A.55}$$

By setting

$$\cos\Big[k(l/2 - x_0)\Big] = \cos\Big[k(-l/2 + x_0)\Big] \tag{A.56}$$
$$\sin\Big[k(l/2 - x_0)\Big] = -\sin\Big[k(-l/2 + x_0)\Big] \tag{A.57}$$

Equation (A.55) becomes

$$y(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \Big[-f^{(2i)}(l/2)\cos\Big[k(l/2+x_0)\Big] + f^{(2i)}(-l/2)\cos\Big[k(-l/2+x_0)\Big] \Big] \right.$$
$$\left. + \frac{(-1)^i}{k^{(2i+2)}} \Big[f^{(2i+1)}(l/2)\sin\Big[k(l/2+x_0)\Big] - f^{(2i+1)}(-l/2)\sin\Big[k(-l/2+x_0)\Big] \Big] \right\} \tag{A.58}$$

It can be seen that this expression is the same as (A.23). Therefore, the transfer function of the sensor will be the same as (A.30):

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{(-1)^i}{k^{(2i+1)}} \left[-f^{(2i)}(l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} - \sin(kl/2) \right) \right. \right.$$
$$\left. + f^{(2i)}(-l/2) \left(\frac{\cos(kl/2)}{\tan(kx_0)} + \sin(kl/2) \right) \right]$$
$$+ \frac{(-1)^i}{k^{(2i+2)}} \left[f^{(2i+1)}(l/2) \left(\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right.$$
$$\left. \left. - f^{(2i+1)}(-l/2) \left(-\frac{\sin(kl/2)}{\tan(kx_0)} + \cos(kl/2) \right) \right] \right\} \tag{A.59}$$

If $f(x)$ is symmetric, A.59 simplifies to A.13 and A.32:

$$y(k) = \mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{2(-1)^i}{k^{2i+1}} f^{(2i)}(l/2)\sin(kl/2) + \frac{2(-1)^i}{k^{2i+2}} f^{(2i+1)}(l/2)\cos(kl/2) \right\} \tag{A.60}$$

A.2 Exponential Curvature

In this section the output and transfer functions for sensors that are centered, noncentered, simply truncated and truncated by the 'folding' technique are derived

when the strain field is spatially sinusoidal or cosinusoidal. In the following analysis, the exponential strain field is assumed to have the form e^{-kx}. The origin will refer to the point where the strain remains the same no matter what the spatial frequency is. For the strain field e^{-kx}, this point is simply $x = 0$, where the strain is unity for all k.

A.2.1 Centered Sensors

Here the output of a sensor mounted in the interior of a structure (away from its boundaries), centered at the origin of an exponential strain field is investigated. It is assumed that the strain field is of the form

$$\epsilon(x) = e^{-kx} \tag{A.61}$$

where k is a parameter that depends on mode number. The output of a sensor with weighting function $f(x)$, centered at $x = 0$ is then given by

$$y(k) = \int_{-l/2}^{l/2} f(x)e^{-kx}\,dx \tag{A.62}$$

Integration by parts of this expression yields

$$\begin{aligned} y(k) &= -\frac{1}{k}f(x)e^{-kx}\Big|_{-l/2}^{l/2} \\ &+ \int_{-l/2}^{l/2} \frac{1}{k}f'(x)e^{-kx}\,dx \end{aligned} \tag{A.63}$$

$$\begin{aligned} y(k) &= \frac{1}{k}\left[-f(l/2)e^{-kl/2} + f(-l/2)e^{kl/2}\right] \\ &+ \int_{-l/2}^{l/2} \frac{1}{k}f'(x)e^{-kx}\,dx \end{aligned} \tag{A.64}$$

Integration by parts three more times yields

$$\begin{aligned} y(k) &= \frac{1}{k}\left[-f(l/2)e^{-kl/2} + f(-l/2)e^{kl/2}\right] \\ &+ \frac{1}{k^2}\left[-f'(l/2)e^{-kl/2} + f'(-l/2)e^{kl/2}\right] \\ &+ \frac{1}{k^3}\left[-f''(l/2)e^{-kl/2} + f''(-l/2)e^{kl/2}\right] \\ &+ \frac{1}{k^4}\left[-f'''(l/2)e^{-kl/2} + f'''(-l/2)e^{kl/2}\right] \\ &+ \int_{-l/2}^{l/2} \frac{1}{k^4} f''''(x)e^{-kx}\, dx \end{aligned} \tag{A.65}$$

A pattern can clearly be seen in this expression, and at this point an infinite series solution for the output of a sensor mounted in the interior of a structure (away from its boundaries), at a distance x_0 from the origin of an exponential strain field can be written as

$$y(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}}\left[-f^{(i)}(l/2)e^{-kl/2} + f^{(i)}(-l/2)e^{kl/2}\right] \tag{A.66}$$

To obtain the transfer function for this sensor, (A.66) is divided by the strain at the center of the sensor, which is simply unity. Therefore

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}}\left[-f^{(i)}(l/2)e^{-kl/2} + f^{(i)}(-l/2)e^{kl/2}\right] \tag{A.67}$$

A.2.2 Noncentered Sensors

Here the output of a sensor mounted in the interior of a structure (away from its boundaries), centered at a distance x_0 from the origin of an exponential strain field is investigated. It is assumed that the strain field is of the form

$$\epsilon(x) = e^{-kx} \tag{A.68}$$

where k is a parameter that depends on mode number. The output of a sensor with weighting function $f(x)$, centered at $x = x_0$ is then given by

$$y(k) = \int_{-l/2+x_0}^{l/2+x_0} f(x - x_0)e^{-kx}\, dx \tag{A.69}$$

Integration by parts of this expression yields

$$\begin{aligned} y(k) &= -\frac{1}{k} f(x - x_0)e^{-kx}\Big|_{-l/2+x_0}^{l/2+x_0} \\ &+ \int_{-l/2+x_0}^{l/2+x_0} \frac{1}{k} f'(x - x_0)e^{-kx}\, dx \end{aligned} \tag{A.70}$$

$$\begin{aligned} y(k) &= \frac{1}{k}\left[-f(l/2)e^{-k(l/2+x_0)} + f(-l/2)e^{-k(-l/2+x_0)}\right] \\ &+ \int_{-l/2+x_0}^{l/2+x_0} \frac{1}{k} f'(x - x_0)e^{-kx}\, dx \end{aligned} \tag{A.71}$$

Integration by parts three more times yields

$$\begin{aligned} y(k) &= \frac{1}{k}\left[-f(l/2)e^{-k(l/2+x_0)} + f(-l/2)e^{-k(-l/2+x_0)}\right] \\ &+ \frac{1}{k^2}\left[-f'(l/2)e^{-k(l/2+x_0)} + f'(-l/2)e^{-k(-l/2+x_0)}\right] \\ &+ \frac{1}{k^3}\left[-f''(l/2)e^{-k(l/2+x_0)} + f''(-l/2)e^{-k(-l/2+x_0)}\right] \\ &+ \frac{1}{k^4}\left[-f'''(l/2)e^{-k(l/2+x_0)} + f'''(-l/2)e^{-k(-l/2+x_0)}\right] \\ &+ \int_{-l/2+x_0}^{l/2+x_0} \frac{1}{k^4} f''''(x - x_0)e^{-kx}\, dx \end{aligned} \tag{A.72}$$

A pattern can clearly be seen in this expression, and an infinite series solution for the output of a sensor mounted in the interior of a structure (away from its boundaries), at a distance x_0 from the origin of an exponential strain field can be

written as

$$y(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}} \left[-f^{(i)}(l/2)e^{-k(l/2+x_0)} + f^{(i)}(-l/2)e^{-k(-l/2+x_0)} \right] \tag{A.73}$$

To obtain the transfer function for this sensor, (A.73) is divided by e^{-kx_0}, the strain at the center of the sensor. Equation (A.73) is first rewritten as

$$y(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}} \left[-f^{(i)}(l/2)e^{-kl/2} + f^{(i)}(-l/2)e^{kl/2} \right] e^{-kx_0} \tag{A.74}$$

By dividing this expression by e^{-kx_0}, it is seen that the transfer function is

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}} \left[-f^{(i)}(l/2)e^{-kl/2} + f^{(i)}(-l/2)e^{kl/2} \right] \tag{A.75}$$

Note that this is precisely the same transfer function as for sensors which are centered at the origin of the exponential strain field. The transfer function of a sensor in an exponential field is then independent of the distance of the sensor from the origin.

A.2.3 Sensors Simply Truncated at Structure Boundaries

Here the output of a sensor mounted near a boundary of a structure, at a distance x_0 from the origin of an exponential strain field is investigated. It is again assumed that the strain field is of the form

$$\epsilon(x) = e^{-kx} \tag{A.76}$$

where k is a parameter that depends on mode number. It is assumed that the boundary of a structure is at $x = 0$. The output of a sensor with weighting function $f(x)$,

centered at $x = x_0$ is then given by

$$y(k) = \int_0^{l/2+x_0} f(x - x_0)e^{-kx}\, dx \tag{A.77}$$

Integration by parts of this expression yields

$$\begin{aligned} y(k) &= -\frac{1}{k} f(x - x_0)e^{-kx}\Big|_0^{l/2+x_0} \\ &+ \int_0^{l/2+x_0} \frac{1}{k} f'(x - x_0)e^{-kx}\, dx \end{aligned} \tag{A.78}$$

$$\begin{aligned} y(k) &= \frac{1}{k}\left[-f(l/2)e^{-k(l/2+x_0)} + f(-x_0)\right] \\ &+ \int_0^{l/2+x_0} \frac{1}{k} f'(x - x_0)e^{-kx}\, dx \end{aligned} \tag{A.79}$$

Integration by parts three more times yields

$$\begin{aligned} y(k) &= \frac{1}{k}\left[-f(l/2)e^{-k(l/2+x_0)} + f(-x_0)\right] \\ &+ \frac{1}{k^2}\left[-f'(l/2)e^{-k(l/2+x_0)} + f'(-x_0)\right] \\ &+ \frac{1}{k^3}\left[-f''(l/2)e^{-k(l/2+x_0)} + f''(-x_0)\right] \\ &+ \frac{1}{k^4}\left[-f'''(l/2)e^{-k(l/2+x_0)} + f'''(-x_0)\right] \\ &+ \int_0^{l/2+x_0} \frac{1}{k^4} f''''(x - x_0)e^{-kx}\, dx \end{aligned} \tag{A.80}$$

A pattern can clearly be seen in this expression, and an infinite series solution for the output of a sensor mounted near a boundary of a structure, at a distance x_0 from the origin of an exponential strain field can be written as

$$y(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}}\left[-f^{(i)}(l/2)e^{-k(l/2+x_0)} + f^{(i)}(-x_0)\right] \tag{A.81}$$

To obtain the transfer function of the sensor, (A.81) is divided by e^{-kx_0}, the strain

at the center of the sensor:

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \frac{1}{k^{i+1}} \left[-f^{(i)}(l/2)e^{-kl/2} + f^{(i)}(-x_0)e^{kx_0} \right] \tag{A.82}$$

A.2.4 Sensors Folded at Structure Boundaries

In Figure 2.22 a procedure used to fold and flip the truncated part of a sensor was shown. In this section, the effect of performing this operation when the strain field is exponential rather than sinusoidal is investigated. As discussed in Chapter 2, the output of the modified sensor is given by

$$y(k) = \int_0^{l/2-x_0} \left[f(x-x_0) - f(-x-x_0) \right] e^{-kx}\, dx + \int_{l/2-x_0}^{l/2+x_0} f(x-x_0)\, e^{-kx}\, dx \tag{A.83}$$

The weighting $f(-x-x_0)$ is the normal weighting $f(x-x_0)$ reflected about the boundary of the structure at $x = 0$. The length of the portion of the sensor that extends beyond the boundary of the sensor is $l/2 - x_0$. The above equation can be rewritten

$$y(k) = \int_0^{l/2+x_0} f(x-x_0)\, e^{-kx}\, dx - \int_0^{l/2-x_0} f(-x-x_0)\, e^{-kx}\, dx \tag{A.84}$$

$$y(k) = G(k) - H(k) \tag{A.85}$$

The first integral, $G(k)$, in (A.84) was dealt with in the previous section and the result is shown in (A.81). The second integral, $H(k)$, is integrated by parts to yield

$$\begin{aligned} H(k) &= -\frac{1}{k} f(-x-x_0)e^{-kx}\Big|_0^{l/2-x_0} \\ &- \int_0^{l/2-x_0} \frac{1}{k} f'(-x-x_0)e^{-kx}\,dx \qquad \text{(A.86)} \\ H(k) &= \frac{1}{k}\Big[f(-x_0) - f(-l/2)e^{-k(l/2-x_0)}\Big] \\ &- \int_0^{l/2-x_0} \frac{1}{k} f'(-x-x_0)e^{-kx}\,dx \qquad \text{(A.87)} \end{aligned}$$

Integration by parts three more times yields

$$\begin{aligned} H(k) &= \frac{1}{k}\Big[f(-x_0) - f(-l/2)e^{-k(l/2-x_0)}\Big] \\ &= \frac{1}{k^2}\Big[-f'(-x_0) + f''(-l/2)e^{-k(l/2-x_0)}\Big] \\ &= \frac{1}{k^3}\Big[f''(-x_0) - f''(-l/2)e^{-k(l/2-x_0)}\Big] \\ &= \frac{1}{k^4}\Big[-f'''(-x_0) + f'''(-l/2)e^{-k(l/2-x_0)}\Big] \\ &+ \int_0^{l/2-x_0} \frac{1}{k^4} f''''(-x-x_0)e^{-kx}\,dx \qquad \text{(A.88)} \end{aligned}$$

A pattern emerges from this expression, and $H(k)$ can be written as an infinite series in the following manner:

$$H(k) = \sum_{i=0}^{\infty} \frac{(-1)^i}{k^{i+1}}\Big[f^{(2i)}(-x_0) - f^{(2i)}(-l/2)e^{-k(l/2-x_0)}\Big] \qquad \text{(A.89)}$$

Now $G(k)$ and $H(k)$ are rewritten in slightly different forms:

$$\begin{aligned} G(k) = \sum_{i=0}^{\infty} \Bigg\{ & \frac{1}{k^{(2i+1)}}\Big[-f^{(2i)}(l/2)e^{-k(l/2+x_0)} + f^{(2i)}(-x_0)\Big] \\ & + \frac{1}{k^{(2i+2)}}\Big[-f^{(2i+1)}(l/2)e^{-k(l/2+x_0)} + f^{(2i+1)}(-x_0)\Big]\Bigg\} \qquad \text{(A.90)} \end{aligned}$$

$$H(k) = \sum_{i=0}^{\infty} \left\{ \frac{1}{k^{(2i+1)}} \left[f^{(2i)}(-x_0) - f^{(2i)}(-l/2)e^{-k(l/2-x_0)} \right] + \frac{1}{k^{(2i+2)}} \left[-f^{(2i+1)}(-x_0) + f^{(2i+1)}(-l/2)e^{-k(l/2-x_0)} \right] \right\} \quad \text{(A.91)}$$

By subtracting the expression for $H(k)$ from $G(k)$, the output of the modified sensor as a function of k can be found:

$$y(k) = \sum_{i=0}^{\infty} \left\{ \frac{1}{k^{(2i+1)}} \left[-f^{(2i)}(l/2)e^{-k(l/2+x_0)} + f^{(2i)}(-x_0) \right. \right.$$
$$\left. f^{(2i)}(-x_0) - f^{(2i)}(-l/2)e^{-k(l/2-x_0)} \right]$$
$$+ \frac{1}{k^{(2i+2)}} \left[-f^{(2i+1)}(l/2)e^{-k(l/2+x_0)} + f^{(2i+1)}(-x_0) \right.$$
$$\left. \left. -f^{(2i+1)}(-x_0) + f^{(2i+1)}(-l/2)e^{-k(l/2-x_0)} \right] \right\} \quad \text{(A.92)}$$

$$y(k) = \sum_{i=0}^{\infty} \left\{ \frac{1}{k^{(2i+1)}} \left[-f^{(2i)}(l/2)e^{-k(l/2+x_0)} + f^{(2i)}(-l/2)e^{-k(l/2-x_0)} \right] \right.$$
$$\left. + \frac{1}{k^{(2i+2)}} \left[-f^{(2i+1)}(l/2)e^{-k(l/2+x_0)} - f^{(2i+1)}(-l/2)e^{-k(l/2-x_0)} + 2f^{(2i+1)}(-x_0) \right] \right\} \quad \text{(A.93)}$$

This expression is divided by e^{-kx_0}, the strain at the center of the sensor, in order to find the transfer function:

$$\mathcal{T}(k) = \sum_{i=0}^{\infty} \left\{ \frac{1}{k^{(2i+1)}} \left[-f^{(2i)}(l/2)e^{-kl/2} + f^{(2i)}(-l/2)e^{-k(l/2-2x_0)} \right] \right.$$
$$\left. + \frac{1}{k^{(2i+2)}} \left[-f^{(2i+1)}(l/2)e^{-kl/2} - f^{(2i+1)}(-l/2)e^{-k(l/2-2x_0)} + 2f^{(2i+1)}(-x_0)e^{kx_0} \right] \right\} \quad \text{(A.94)}$$

Appendix B

The Transfer Function of the Gauss-Hanning Weighting

As discussed in Section 2.3.3 of Chapter 2, the Gauss-Hanning weighting is produced by truncating a Gaussian weighting by multiplying it by a Hanning weighting. The transfer function of the Gauss-Hanning weighting is given here:

$$\begin{aligned} \mathcal{T}(k) &= \frac{\sqrt{2\pi \ln 2}\, M(\alpha)}{8\alpha} \exp\left(-\frac{\ln 2 k^2}{2\alpha^2} - \frac{k'}{\alpha}\right) \\ &\times \left\{ \operatorname{erf}\left(\frac{\pi\alpha + j(k'+\alpha')}{\alpha''}\right) - \operatorname{erf}\left(\frac{-\pi\alpha + j(k'+\alpha')}{\alpha''}\right) \right. \\ &\quad + \exp\left(\frac{2k'}{\alpha}\right)\left[\operatorname{erf}\left(\frac{\pi\alpha + j(k'-\alpha')}{\alpha''}\right) - \operatorname{erf}\left(\frac{-\pi\alpha + j(k'-\alpha')}{\alpha''}\right)\right] \\ &\quad \left. +4\exp\left(\frac{k'}{\alpha}\right)\left[\operatorname{erf}\left(\frac{\pi\alpha + jk')}{\alpha''}\right) - \operatorname{erf}\left(\frac{-\pi\alpha + jk')}{\alpha''}\right)\right]\right\} \end{aligned} \tag{B.1}$$

where $j = \sqrt{-1}$ and

$$k' = \sqrt{2}\ln 2\, k \tag{B.2}$$

$$\alpha' = 2\ln 2\, \alpha \tag{B.3}$$

$$\alpha'' = 2\sqrt{\ln 2}\,\alpha \tag{B.4}$$

The error function $\mathrm{erf}(z)$ is given by

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \tag{B.5}$$

and the parameter $M(\alpha)$ is given by

$$M(\alpha) = \frac{4\alpha}{\sqrt{2\pi \ln 2}} \left[\frac{1}{\mathrm{erf}\left(\frac{\pi + j\,2\ln 2}{2ln2}\right) - \mathrm{erf}\left(\frac{-\pi + j\,2\ln 2}{2ln2}\right) + 4\,\mathrm{erf}\left(\frac{\pi}{2\sqrt{\ln 2}}\right)} \right] \tag{B.6}$$

$$M(\alpha) \approx 0.33056\,\alpha \tag{B.7}$$

Appendix C

Derivation of Numerical Integration Schemes

C.1 Elementary Quadrature Formulae on a Finite Interval

We can construct a general formula to numerically estimate the value of any integral of the form

$$\int_a^b g(x)\, f(x)\, dx \tag{C.1}$$

where $f(x)$ is the argument function and $g(x)$ is an arbitrary weighting function [25]. In this work $g(x)$ is typically assumed to be unity over the interval $[a, b]$. We call $L[a, b]$ the class of functions which are Lebesgue-integrable (summable) in $[a, b]$, and $AC^k[a, b]$ the class of functions whose k-th derivative is absolutely continuous in $[a, b]$. It is assumed that the weighting function $g(x)$ and the argument function $f(x)$ satisfy the hypotheses

$$g(x) \in L[a, b], \tag{C.2}$$

$$f(x) \in AC^{m-1}[a, b]. \tag{C.3}$$

Within the region $[a, b]$ we fix some number $n \geq 1$ of points x_1, x_2,..., x_n where the function $f(x)$ will be evaluated. These points are called nodes. These nodes have the property that

$$x_0 = a \leq x_1 < x_2 < \cdots < x_n \leq b = x_{n+1}. \tag{C.4}$$

A linear differential operator E of order m is now introduced, of the form

$$E = \sum_{k=0}^{m} a_k(x) \frac{d^{m-k}}{dx^{m-k}} \tag{C.5}$$

along with its adjoint operator, which has the form

$$E^* = \sum_{k=0}^{m} (-1)^{m-k} a_k(x) \frac{d^{m-k}}{dx^{m-k}}. \tag{C.6}$$

The order m of these operators is the same as in (C.3). We then have an elementary quadrature formula for the evaluation of the general integral (C.1):

$$\int_a^b g(x) f(x) = \sum_{h=0}^{m-1} \sum_{i=1}^{n} A_{hi} f^{(h)}(x_i) + R[f(x)] \tag{C.7}$$

where the constant coefficients A_{hi} are independent of $f(x)$ and $R[f(x)]$ is a remainder term accounting for the error inherent in the numerical approximation. This remainder term is zero when $f(x)$ is a solution to the differential equation

$$E[f(x)] = 0. \tag{C.8}$$

In many practical applications, especially when dealing with numerical data rather than a specified analytic function, $R[f(x)]$ cannot be determined and the integral is

approximated by

$$\int_a^b g(x)f(x)\,dx = \sum_{h=0}^{m-1}\sum_{i=1}^{n} A_{hi} f^{(h)}(x_i) \tag{C.9}$$

where $f^{(h)}(x)$ denotes the h-th derivative of $f(x)$. $R[f(x)]$ is then used as an approximation of the error incurred in using the integration scheme. An example of a differential operator E which yields exact results for any polynomial of degree $m-1$ is

$$E = \frac{d^m}{dx^m}. \tag{C.10}$$

Another operator, one which is exact for trigonometric polynomials of order ν is

$$E = \frac{d}{dx}\prod_{k=1}^{\nu}\left(\frac{d^2}{dx^2} + k^2\right) \tag{C.11}$$

where $m = 2\nu + 1$. Now, we consider the non-homogeneous differential equation

$$E^*[\varphi(x)] = g(x) \tag{C.12}$$

where E^* is the adjoint operator, defined in (C.6). We then find $n-1$ solutions of the form

$$\varphi_1(x),\ \varphi_2(x), \ldots, \varphi_{n-1}(x) \in AC^{m-1}[a,b] \tag{C.13}$$

Two more solutions, $\varphi_0(x)$ and $\varphi_n(x)$, are determined by the initial conditions at the points a and b. It can be shown that when deriving a numerical integration formula of *closed type*, i.e. when $x_1 = a$ and $x_n = b$, it is not necessary to consider the two boundary solutions $\varphi_0(x)$ and $\varphi_n(x)$, and they can be assumed to be zero [25]. Further, it can be shown that the coefficients A_{hi} defined in (C.9) can be determined from the following expression:

$$A_{hi} = \left\{E^*_{m-h-1}[\varphi_i(x) - \varphi_{i-1}(x)]\right\}_{x=x_i}, \begin{cases} h = 0, 1, \ldots, m-1 \\ i = 1, 2, \ldots, n \end{cases} \tag{C.14}$$

In addition, the remainder term $R[f(x)]$ is expressed as

$$\begin{aligned} R[f(x)] &= \sum_{i=0}^{n} \int_{x_i}^{x_i+1} \varphi_i(x) E[f(x)]\, dx, \\ &= \int_a^b \Phi(x) E[f(x)]\, dx. \end{aligned} \tag{C.15}$$

It can be seen from (C.14) that the coefficients A_{hi} are indeed independent of the argument function $f(x)$. It can also be immediately seen from (C.15) that when $f(x)$ satisfies the differential equation given in (C.8), that the remainder $R[f(x)]$ will be zero. In these cases, the integration formula is exact.

In summary, the following scheme is used to derive elementary quadrature formulae of the type given in (C.7):

1. The number of nodes n which will be used is selected;
2. The order m and form of the operator E is selected. This choice may be influenced by any knowledge of the argument function $f(x)$ which will be integrated;
3. The linear differential equation $E^*[\varphi(x)] = g(x)$ is considered and the solutions $\varphi_0(x)$ and $\varphi_n(x)$ are found, if necessary (when $x_0 \neq a$ and $x_n \neq b$, respectively);
4. The other $n-1$ arbitrary solutions $\varphi_1(x)$, $\varphi_2(x)$,..., $\varphi_{n-1}(x)$ of the same differential equation $E^*[\varphi(x)] = g(x)$ are then found;
5. The constant coefficients A_{hi} are determined from (C.14);
6. Finally, the remainder term $R[f(x)]$ is determined from (C.15) when $f(x)$ does not satisfy the differential equation (C.8).

For convenience, the three fundamental formulae for deriving elementary quadrature formulae are summarized here:

$$\int_a^b g(x) f(x)\, dx = \sum_{h=0}^{m-1} \sum_{i=1}^{n} A_{hi} f^{(h)}(x_i) + R[f(x)], \tag{C.16}$$

$$A_{hi} = \left[E^*_{m-h-1}\left(\varphi_i(x) - \varphi_{i-1}(x)\right)\right]_{x=x_i}, \tag{C.17}$$

$$R[f(x)] = \sum_{i=0}^{n} \int_{x_i}^{x_{i+1}} \varphi_i(x) E[f(x)]\, dx = \int_a^b \Phi(x) E[f(x)]\, dx. \tag{C.18}$$

C.2 Trapezoidal Rule

Here we derive the trapezoidal formula using the techniques given in Section C.1. To start, we have $n = 2$. Since the formula will be exact for polynomials or order 1 (linear polynomials), we have $m = 2$. We therefore select [25]:

$$E = \frac{d^2}{dx^2}. \tag{C.19}$$

The n (2) nodes to be used are $x_1 = a$ and $x_2 = b$. Since the formula is of *closed type*, the boundary solutions $\varphi_0(x)$ and $\varphi_2(x)$ can be ignored. Clearly only a single solution, $\varphi_1(x)$ needs to be chosen. It must satisfy the nonhomogeneous differential equation

$$\begin{aligned} E^*[\varphi_1(x)] &= g(x), \\ E^*[\varphi_1(x)] &= 1. \end{aligned} \tag{C.20}$$

The adjoint operator here is equal to $\frac{d^2}{dx^2}$ and we have

$$\frac{d^2}{dx^2}\varphi_1(x) = 1. \tag{C.21}$$

By integrating (C.21) twice , we obtain

$$\varphi_1(x) = \frac{x^2}{2} + C_1 x + C_2 \tag{C.22}$$

where C_1 and C_2 are integration constants which are determined later. At this point, the integration formula is [25]:

$$\int_a^b f(x)\,dx = \sum_{h=0}^{1}\sum_{i=1}^{2} A_{hi} f^{(h)}(x_i) + R[f(x)] \tag{C.23}$$

There are always $(n-1)m$ arbitrary parameters, and in this case there are two, C_1 and C_2. These are determined by applying some condition to the constants A_{hi}. These constants are:

$$\begin{aligned}
A_{01} &= E_1^*[\varphi_1(x) - \varphi_0(x)]_{x=x_i} \\
&= -\frac{d}{dx}[\varphi_1(x)]_{x=x_1} \\
&= -\frac{d}{dx}\left[\frac{x^2}{2} + C_1 x + C_2\right]_{x=a} \\
&= [-x - C_1]_{x=a} \\
&= -a - C_1 \qquad \text{(C.24)} \\
A_{02} &= E_1^*[\varphi_2(x) - \varphi_1(x)]_{x=b} \\
&= -\frac{d}{dx}\left[-\frac{x^2}{2} - C_1 x - C_2\right]_{x=b} \\
&= b + C_1 \qquad \text{(C.25)} \\
A_{11} &= E_0^*[\varphi_1(x) - \varphi_0(x)]_{x=a} \\
&= \left[\frac{x^2}{2} + C_1 x + C_2\right]_{x=a} \\
&= \frac{a^2}{2} + C_1 a + C_2 \qquad \text{(C.26)} \\
A_{12} &= E_0^*[\varphi_2(x) - \varphi_1(x)]_{x=b} \\
&= \left[-\frac{x^2}{2} - C_1 x - C_2\right]_{x=b} \\
&= -\frac{b^2}{2} - C_1 b - C_2. \qquad \text{(C.27)}
\end{aligned}$$

Using these results we can write the approximate expression for the integral:

$$\begin{aligned}\int_a^b f(x)\,dx &\approx A_{01}f(a) + A_{02}f(b) + A_{11}f'(a) + A_{12}f'(b) \\ &= (-a - C_1)f(a) + (b + C_1)f(b) \\ &+ \left(\frac{a^2}{2} + C_1 a + C_2\right) f'(a) + \left(-\frac{b^2}{2} - C_1 b - C_2\right) f'(b) \quad \text{(C.28)}\end{aligned}$$

Since values of the first derivative $f'(x)$ are not available, A_{11} and A_{12} must be zero. This gives a way to solve for the arbitrary constants C_1 and C_2:

$$A_{11} = \frac{a^2}{2} + C_1 a + C_2 = 0, \quad \text{(C.29)}$$

$$A_{12} = -\frac{b^2}{2} - C_1 b - C_2 = 0. \quad \text{(C.30)}$$

Solving for C_1 and C_2 from these two equations yields

$$C_1 = -\frac{a+b}{2}, \quad \text{(C.31)}$$

$$C_2 = \frac{ab}{2}. \quad \text{(C.32)}$$

Using this result we find

$$A_{01} = A_{02} = \frac{b-a}{2}, \quad \text{(C.33)}$$

$$A_{11} = A_{12} = 0. \quad \text{(C.34)}$$

Thus (C.26) now becomes [25]

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}[f(a) + f(b)] \quad \text{(C.35)}$$

and $\varphi_1(x)$ is

$$\begin{aligned}\varphi_1(x) &= \frac{x^2}{2} - \frac{a+b}{2}x + \frac{ab}{2} \\ &= \frac{1}{2}(x-a)(x-b) \end{aligned} \tag{C.36}$$

We are now in a position to calculate the remainder $R[f(x)]$ for the trapezoidal integration formula. Using (C.15) and (C.36), we find $R[f(x)]$:

$$\begin{aligned} R[f(x)] &= \int_a^b \Phi(x)E[f(x)]\,dx, \\ &= \int_a^b \varphi_1(x)\frac{d^2}{dx^2}f(x)\,dx, \\ &= \int_a^b \frac{1}{2}(x-a)(x-b)f''(x)\,dx. \end{aligned} \tag{C.37}$$

If we assume that $f''(x)$ is continuous in $[a,b]$, and since $\varphi_1(x)$ is continuous and does not change sign in $[a,b]$, we can write, using the mean value theorem [25]:

$$\begin{aligned} R[f(x)] &= f''(\xi)\int_a^b \varphi_1(x)\,dx, \\ &= f''(\xi)\int_a^b \frac{1}{2}(x-a)(x-b)\,dx, \\ &= -\frac{(b-a)^3}{12}f''(\xi). \end{aligned} \tag{C.38}$$

where $a < \xi < b$. The value of ξ is not known, but since it lies within $[a,b]$ it is possible to calculate the maximum possible value of the remainder $R[f(x)]$ by choosing ξ such that $f''(\xi)$ is maximized. It is clear from (C.38) that the error increases dramatically as the spacing $(b-a)$ is increased. Note from (C.35) that this assumes that only one trapezoid is fitted between the limits a and b. That is, $h = b - a$ in this case. If the compound trapezoidal rule is employed, and as more trapezoids are used, the error will decrease as $1/N^2$ where N is the number of subdivisions of the interval [46, 30, 16].

C.3 Simpson's Rule

Here Simpson's rule will be derived using the three fundamental formulae, (C.16), (C.17) and (C.18). Since the formula uses values of $f(x)$ at three points, $n = 3$. The rule can also be derived using $m = 4$ since it is exact for cubic polynomials [25, 16]. However, it is simpler to derive using $m = 3$. Therefore, we have

$$E = \frac{d^3}{dx^3}, \tag{C.39}$$

and

$$E_0^* = 1,\ E_1^* = -\frac{d}{dx},\ E_2^* = \frac{d^2}{dx^3},\ E^* = E_3^* = -\frac{d^3}{dx^3} \tag{C.40}$$

The non-homogeneous differential equation (C.12), where $g(x) = 1$ must be satisfied:

$$\begin{aligned} E^*[\varphi(x)] &= 1, \\ -\frac{d^3}{dx^3}[\varphi(x)] &= 1, \\ -\varphi'''(x) &= 1. \end{aligned} \tag{C.41}$$

Since $n = 3$ (rather than 2 as in the trapezoidal rule), we will have two arbitrary solutions to the differential equation (C.41), $\varphi_1(x)$ and $\varphi_2(x)$:

$$\varphi_1(x) = \frac{x^3}{6} + \frac{C_{11}x^2}{2} + C_{12}x + C_{13}, \tag{C.42}$$

$$\varphi_2(x) = \frac{x^3}{6} + \frac{C_{21}x^2}{2} + C_{22}x + C_{23}. \tag{C.43}$$

By using (C.42) and (C.43) in (C.17), we obtain

$$A_{01} = -a + C_{11} \tag{C.44}$$

$$A_{02} = C_{21} - C_{11} \tag{C.45}$$

$$A_{03} = b - C_{21} \tag{C.46}$$

$$A_{11} = \frac{a^2}{2} - C_{11}a - C_{12} \tag{C.47}$$

$$A_{12} = (C_{11} - C21)x_2 + (C_{12} - C_{22}) \tag{C.48}$$

$$A_{13} = -\frac{b^2}{2} + C_{21}b + C_{22} \tag{C.49}$$

$$A_{21} = -\frac{a^3}{6} + \frac{C_{11}a^2}{2} + C_{12}a + C_{13} \tag{C.50}$$

$$A_{22} = \frac{(C_{21} - C_{11})x_2^2}{2} + (C_{22} - C_{12})x_2 + (C_{23} - C_{13}) \tag{C.51}$$

$$A_{23} = \frac{b^3}{6} - \frac{C_{21}b^2}{2} - C_{22}b - C_{23} \tag{C.52}$$

where it is assumed that $x_1 = a$ and $x_3 = b$. Typically it is assumed that $x_2 = \frac{1}{2}(a+b)$. To maintain generality, x_2 has not yet been specified. The expression for the integral, (C.16) now becomes

$$\begin{aligned} \int_a^b f(x) \approx\ & A_{01}f(a) + A_{02}f(x_2) + A_{03}f(b) \\ +\ & A_{11}f'(a) + A_{12}f'(x_2) + A_{13}f'(b) \\ +\ & A_{21}f''(a) + A_{22}f''(x_2) + A_{23}f''(b). \end{aligned} \tag{C.53}$$

Again, assuming we cannot evaluate values of $f'(x)$ or $f''(x)$, we must specify that the six constants A_{11}, A_{12}, A_{13}, A_{21}, A_{22} and A_{23} be zero. This requires solving the six simultaneous equations:

$$\frac{a^2}{2} - C_{11}a - C_{12} = 0 \tag{C.54}$$

$$(C_{11} - C_{21})x_2 + (C_{12} - C_{22}) = 0 \tag{C.55}$$

$$-\frac{b^2}{2} + C_{21}b + C_{22} = 0 \tag{C.56}$$

$$-\frac{a^3}{6} + \frac{C_{11}a^2}{2} + C_{12}a + C_{13} = 0 \tag{C.57}$$

$$\frac{(C_{21} - C_{11})x_2^2}{2} + (C_{22} - C_{12})x_2 + (C_{23} - C_{13}) = 0 \tag{C.58}$$

$$\frac{b^3}{6} - \frac{C_{21}b^2}{2} - C_{22}b - C_{23} = 0 \tag{C.59}$$

Solving these yields:

$$C_{11} = \frac{3x_2(a+b) - 4a^2 - ab - b^2}{6(x_2 - a)} \tag{C.60}$$

$$C_{12} = \frac{a(a^2 + ab + b^2 - 3bx_2)}{6(x_2 - a)} \tag{C.61}$$

$$C_{13} = \frac{a^2(x2(3b - a) - b^2 - ab)}{12(x_2 - a)} \tag{C.62}$$

$$C_{21} = \frac{3x_2(a+b) - a^2 - ab - 4b^2}{6(x_2 - b)} \tag{C.63}$$

$$C_{22} = \frac{b(a^2 + ab + b^2 - 3ax_2)}{6(x_2 - b)} \tag{C.64}$$

$$C_{23} = \frac{b^2(x2(3a - b) - a^2 - ab)}{12(x_2 - b)} \tag{C.65}$$

Using (C.60)–(C.65) we find

$$A_{01} = \frac{10a^2 + ab + b^2 - 3x_2(3a+b)}{6(x_2 - a)} \tag{C.66}$$

$$A_{02} = \frac{(b-a)^3}{6(x_2 - a)(x_2 - b)} \tag{C.67}$$

$$A_{03} = -\frac{a^2 + ab + 10b^2 - 3x_2(a+3b)}{6(x_2 - b)} \tag{C.68}$$

For convenience, it is typically assumed that $x_2 = \frac{(a+b)}{2}$. In this case, (C.66)–(C.68) simplify to

$$A_{01} = A_{03} = \frac{1}{6}(b-a) \tag{C.69}$$

$$A_{02} = \frac{4}{6}(b-a) \tag{C.70}$$

$$\tag{C.71}$$

leading to Simpson's formula [25, 16, 30]:

$$\int_a^b f(x)\,dx \approx \frac{(b-a)}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right). \tag{C.72}$$

Now the remainder term $R[f(x)]$ is computed. We have

$$R[f(x)] = \int_a^b \Phi(x) E[f(x)]\,dx$$

$$R[f(x)] = \int_a^b \Phi(x)(f''')\,dx \tag{C.73}$$

where

$$\Phi(x) = \begin{cases} \dfrac{(x-a)^2(a+b-2x)}{12}, & a \le x \le \dfrac{(a+b)}{2} \\ \dfrac{(x-b)^2(a+b-2x)}{12}, & \dfrac{(a+b)}{2} \le x \le b \end{cases} \tag{C.74}$$

so that

$$R[f(x)] = \int_a^{\frac{a+b}{2}} \frac{(x-a)^2(a+b-2x)}{12} f''''(x)\,dx + \int_{\frac{a+b}{2}}^b \frac{(x-b)^2(a+b-2x)}{12} f''''(x)\,dx \quad \text{(C.75)}$$

It can be shown that when $f''''(x)$ is a constant, i.e. when $f(x)$ is a cubic polynomial, the remainder $R[f(x)]$ always goes to zero, showing that Simpson's formula exactly integrates polynomials of order 3 or lower [30, 16]. This is because $\varphi_1(x)$ and $\varphi_2(x)$ are antisymmetric about the point $x_2 = \frac{a+b}{2}$. Mathematically, this means $\varphi_1(x) + \varphi_2(a+b-x) = 0$.

C.4 Cubic Interpolating Splines

A cubic interpolating spline can be passed through a discrete set of measurements and then integrated. A cubic interpolating spline is a piecewise cubic parametric polynomial that passes through an arbitrary set of points. It does so while maintaining derivative and curvature continuity at the sample points while minimizing the overall integral of the curvature squared. The spline curve is composed of $n-1$ polynomial segments, where n is the number of points the spline must pass through [52, 2]. The i-th polynomial segment stretches between (x_i, y_i) and (x_{i+1}, y_{i+1}) (two successive points). The x- and y-coordinate values of the curve are defined separately as parametric equations of an independent parameter t [52]. The curve of the i-th segment is defined as $(X_i(t), Y_i(t))$. The x-coordinates of the curve $X_i(t)$ can be determined solely from the x-coordinates of the control points x_i and the y-coordinates of the curve $Y_i(t)$ can be determined solely from the y-coordinates of the control points y_i [46, 52]. Since the solution procedure is the same for both $X_i(t)$ and $Y_i(t)$, only

$Y_i(t)$ will be dealt with here. $Y_i(t)$ is defined as:

$$Y_i(t) = a_i t^3 + b_i t^2 + c_i t + d_i, \quad (0 < t < 1) \tag{C.76}$$

If a spline is to be passed through n points, $n-1$ polynomials will be required to solve for all $Y_i(t)$ completely. Therefore, $4(n-1)$ constants need to be found. However, it suffices to find the derivatives $\frac{dY_i}{dt}$ or $Y_i'(t)$ of the spline at each control point y_i [46, 52]. Knowing these derivatives and the y_i, it is possible using the four Hermite cubic polynomials to find $Y_i(t)$. These four Hermite polynomials are:

$$H_1(t) = (2t+1)(t-1)^2 \tag{C.77}$$

$$H_2(t) = t^2(3-2t) \tag{C.78}$$

$$H_3(t) = t(t-1)^2 \tag{C.79}$$

$$H_4(t) = t^2(t-1) \tag{C.80}$$

These polynomials are shown in Figure C.1. Knowing the amplitude y_i and the derivative D_i at the start of the i-th curve segment, and the amplitude y_{i+1} and derivative D_{i+1} at the end of it, $Y_i(t)$ for that segment would be expressed as

$$Y_i(t) = H_1(t)y_i + H_2(t)y_{i+1} + H_3(t)D_i + H_4(t)D_{i+1} \tag{C.81}$$

Since there are n control points, there are n derivatives D_i that need to be found. Since we have

$$Y_i'(t=1) = Y_{i+1}'(t=0) = D_{i+1}, \quad (i = 1, 2, \ldots, n-1) \tag{C.82}$$

derivative continuity for both $X_i(t)$ and $Y_i(t)$ across control points is guaranteed. This means that we also have slope continuity of the curve, $\frac{dX_i}{dY_i}$. The way to obtain n

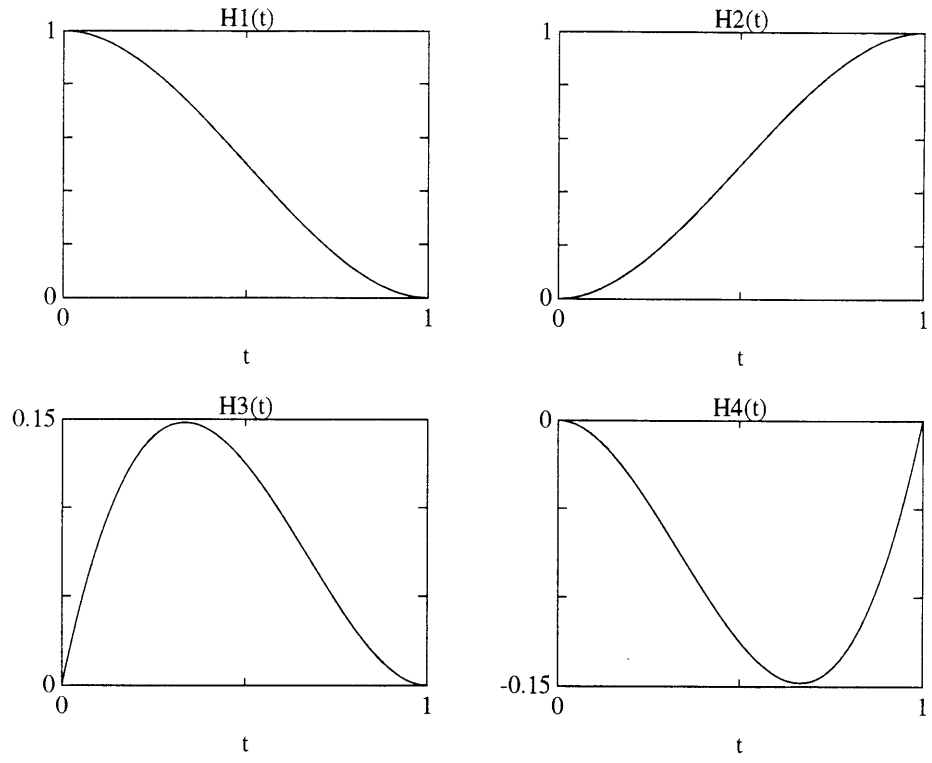

Figure C.1: The four Hermite polynomials: $H_1(t)$ is unity at $t = 0$, $H_2(t)$ is unity at $t = 1$, $H_3(t)$ has unity slope at $t = 0$ and $H_4(t)$ has unity slope at $t = 1$.

equations to solve for the D_i is to enforce *curvature* continuity at the interior control points. This means that:

$$Y_i''(t = 1) = Y_{i+1}''(t = 0), \quad (i = 1, 2, \ldots, n - 2) \tag{C.83}$$

This yields $n - 2$ equations. Two more equations are needed in order to solve for all the D_i. These are generally found by enforcing some condition on the spline at the two ends. For a *natural* cubic spline, the curvature is made to go to zero at the free ends [46]. Thus we have:

$$Y_1''(t = 0) = 0 \tag{C.84}$$

$$Y_{n-1}''(t = 1) = 0 \tag{C.85}$$

Another possibility is to actually specify the slopes D_1 and D_{n-1}, at the start and end. A further possibility is to construct a *closed* spline by requiring that $Y_1''(t = 0) = Y_{n-1}''(t = 1)$, and by making $D_1 = D_{n-1}$.

Using the fact that $b_i = 0$ from (C.76), we can derive expressions for the amplitude and derivative of $Y_i(t)$ at the start and end points:

$$Y_i(t = 0) = y_i = d_i \tag{C.86}$$

$$Y_i(t = 1) = y_{i+1} = a_i + b_i + c_i + d_i \tag{C.87}$$

$$Y_i'(t = 0) = D_i = c_i \tag{C.88}$$

$$Y_i'(t = 1) = D_{i+1} = 3a_i + 2b_i + c_i \tag{C.89}$$

We can solve (C.86)–(C.89) for the curve segment constants a_i, b_i, c_i and d_i:

$$a_i = 2(y_i - y_{i+1}) + D_i + D_{i+1} \tag{C.90}$$

$$b_i = 3(y_{i+1} - y_i) - 2D_i - D_{i+1} \tag{C.91}$$

$$c_i = D_i \tag{C.92}$$

$$d_i = y_i \tag{C.93}$$

Now we require that the curvature at the very start of the spline be zero. Thus

$$Y_1''(t = 0) = 2b_i = 0 \tag{C.94}$$

Using (C.91), we find

$$2D_i + D_{i+1} = 3(y_{i+1} - y_i) \tag{C.95}$$

For the $n-2$ interior points, we have

$$\begin{aligned} Y_i''(t=1) &= Y_{i+1}''(t=0), \quad (i=1,2,\ldots,n-2) \\ 6a_i + 2b_i &= 2b_{i+1} \qquad \text{(C.96)} \\ 3a_i + b_i &= b_{i+1} \qquad \text{(C.97)} \end{aligned}$$

Now using (C.90) and (C.91), we find

$$\begin{aligned} [6(y_i - y_{i+1}) + 3D_i + 3D_{i+1}] &+ [3(y_{i+1} - y_i) - 2D_i - D_{i+1}] \qquad \text{(C.98)} \\ &= [3(y_{i+2} - y_{i+1}) - 2D_{i+1} - D_{i+2}] \qquad \text{(C.99)} \\ D_i + 4D_{i+1} + D_{i+2} &= 3(y_{i+2} - y_i), \quad (i=1,2,\ldots,n-2) \qquad \text{(C.100)} \end{aligned}$$

At the far end, we have the zero curvature condition

$$Y_{n-1}''(t=1) = 6a_{n-1} + 2b_{n-1} = 0 \qquad \text{(C.101)}$$

Again using (C.90) and (C.91), we have

$$\begin{aligned} 3a_{n-1} + b_{n-1} &= 0 \\ 6(y_{n-1} - y_n) + D_{n-1} + D_n &= 3(y_n - y_{n-1}) - 2D_{n-1} - D_n \\ D_{n-1} + 2D_n &= 3(y_n - y_{n-1}) \qquad \text{(C.102)} \end{aligned}$$

This can be conveniently put in a simple matrix form [46, 52, 53]

$$\begin{bmatrix} 2 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ & & & \ddots & & & \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 2 \end{bmatrix} \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ \vdots \\ D_{n-1} \\ D_n \end{bmatrix} = \begin{bmatrix} 3(y_2 - y_1) \\ 3(y_3 - y_1) \\ 3(y_4 - y_2) \\ 3(y_5 - y_3) \\ \vdots \\ 3(y_n - y_{n-2}) \\ 3(y_n - y_{n-1}) \end{bmatrix} \tag{C.103}$$

The matrix is of a familiar form, and is positive definite. Thus it has full rank, and simple Gaussian elimination and back substitution can be used to efficiently solve for the D_i [46, 53]. Once the D_i are known, (C.81) is used on each interval to plot the curve segment. Simplifying (C.81) or using (C.90)–(C.93) in (C.76) yields

$$\begin{aligned} Y_i(t) &= [2(y_i - y_{i+1}) + D_i + D_{i+1}]t^3 \\ &+ [3(y_{i+1} - y_i) - 2D_i - D_{i+1}]t^2 \\ &+ D_i t + y_i \end{aligned} \tag{C.104}$$

A similar procedure is used to solve for the x-coordinates of the curve segments, $X_i(t)$.

C.5 B-Splines

B-splines, *basis splines*, or *basic splines* are composed of polynomial segments, but unlike cubic interpolating splines, they are not interpolatory and do not pass through the specified control points [7]. B-splines can be of various orders, from 1 to the one less than the number of vertices in the control polygon. B-splines of first order are linear functions which interpolate the data in a trapezoidal fashion. Second order B-

splines are quadratic functions which yield a curve which is tangent to the midpoint of the line connecting two successive control points. Finally, B-splines of order $n-1$, where n is the number of vertices of the control polygon, are Bézier curves. Some of the properties of B-splines are given here [23]:

- The convex hull formed from the vertices of the control polygon will always contain the B-spline curve. The convex hull is the closed convex polygon formed by stretching a loop of elastic around all the control vertices.
- A B-spline of order m is differentiable $m-1$ times when there are no repeated control vertices. The number of times the curve is differentiable goes down by one to $m-2$ when a double vertex exists, and down to $m-p$ when a single vertex is repeated p times. This means that for a B-spline of second order, there will be a slope discontinuity at a repeated vertex.
- B-spline curves are transformable by linear translation and rotation operations. This means that the curve computed after a translation and rotation of its control polygon yields the same curve as the same translation and rotation applied to the original curve.
- Moving a control vertex only affects n curve segments, when the order of the curve is $n-1$ and the control polygon is composed of n vertices.
- Any infinitely long line cannot intersect the B-spline curve without also intersecting the control polygon.

It can be seen from Figure 3.5 in Chapter 4 that the curve becomes tangent at the midpoint of the lines connecting the vertices of the control polygon. It is at these points that successive polynomials connect.

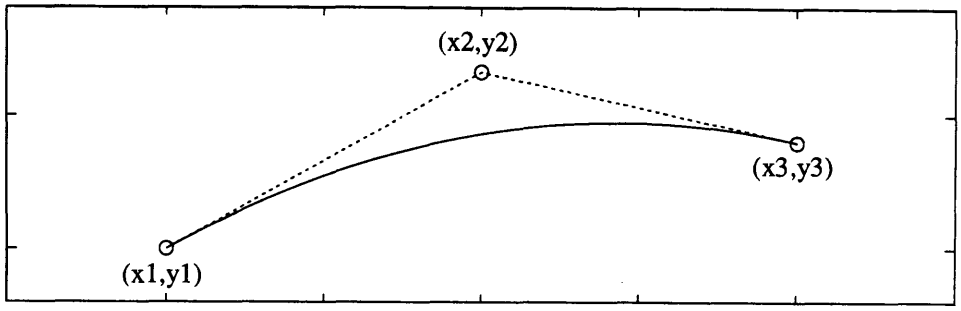

Figure C.2: A very simple third order B-spline.

These polynomials are parametrically expressed as a function of an independent variable t as follows;

$$X_i(t) = A_{1i}t^2 + B_{1i}t + C_{1i} \quad \text{(C.105)}$$

$$Y_i(t) = A_{2i}t^2 + B_{2i}t + C_{2i} \quad \text{(C.106)}$$

Typically t is assumed to be in the range $[0, 1]$ for each polynomial in order to simplify the mathematics, but any range can be used. In order to solve for the six unknown constants, six conditions must be specified so as to uniquely define the curve. We start by examining the simplest case: a B-spline formed of a single polynomial. An example of such a spline is shown in Figure C.2. This spline is defined by a single curve:

$$X(t) = A_1t^2 + B_1t + C_1 \quad \text{(C.107)}$$

$$Y(t) = A_2t^2 + B_2t + C_2 \quad \text{(C.108)}$$

We will find the six coefficients as a function of the location of the three vertices shown in Figure C.2, x_1, y_1, x_2, y_2, x_3, and y_3. The six equations are:

$$X(t=0) = C_1 = x_1 \tag{C.109}$$

$$Y(t=0) = C_2 = y_1 \tag{C.110}$$

$$X(t=1) = A_1 + B_1 + C_1 = x_3 \tag{C.111}$$

$$Y(t=1) = A_2 + B_2 + C_2 = y_3 \tag{C.112}$$

$$\frac{Y'}{X'}(t=0) = \frac{B_2}{B_1} = \frac{y_2 - y_1}{x_2 - x_1} \tag{C.113}$$

$$\frac{Y'}{X'}(t=1) = \frac{2A_2 + B_2}{2A_1 + B_1} = \frac{y_3 - y_2}{x_3 - x_2} \tag{C.114}$$

The solution to these equations is

$$\begin{aligned} A_1 &= (x_3 - x_2) - (x_2 - x_1) & & & A_2 &= (y_3 - y_2) - (y_2 - y_1) \\ B_1 &= 2(x_2 - x_1) & \text{and} & & B_2 &= 2(y_2 - y_1) \\ C_1 &= x_1 & & & C_2 &= y_1 \end{aligned} \tag{C.115}$$

and thus the polynomial is defined as

$$X(t) = (x_1 - 2x_2 + x_3)t^2 + 2(x_2 - x_1)t + x_1 \tag{C.116}$$

$$Y(t) = (y_1 - 2y_2 + y_3)t^2 + 2(y_2 - y_1)t + y_1 \tag{C.117}$$

This result is now used to write the polynomials for a B-spline defined by a polygon having n vertices:

$$\left.\begin{aligned} X_i(t) &= \left(x_i - 2x_{i+1} + \frac{x_{i+1}+x_{i+2}}{2}\right)t^2 + 2(x_{i+1}-x_i)t + x_i \\ Y_i(t) &= \left(y_i - 2y_{i+1} + \frac{y_{i+1}+y_{i+2}}{2}\right)t^2 + 2(y_{i+1}-y_i)t + y_i \end{aligned}\right\}(i=1) \tag{C.118}$$

$$\left.\begin{aligned} X_i(t) &= \left(\frac{x_i+x_{i+1}}{2} - 2x_{i+1} + \frac{x_{i+1}+x_{i+2}}{2}\right)t^2 + 2(x_{i+1}-x_i)t + \frac{x_i+x_{i+1}}{2} \\ Y_i(t) &= \left(\frac{y_i+y_{i+1}}{2} - 2y_{i+1} + \frac{y_{i+1}+y_{i+2}}{2}\right)t^2 + 2(y_{i+1}-y_i)t + \frac{y_i+y_{i+1}}{2} \end{aligned}\right\} (i=2,3,\ldots,n-3) \tag{C.119}$$

$$\left.\begin{aligned} X_i(t) &= \left(\frac{x_{i-2}+x_{i-1}}{2} - 2x_{i-1} + x_i\right)t^2 + 2(x_{i+1}-x_i)t + x_i \\ Y_i(t) &= \left(\frac{y_{i-2}+y_{i-1}}{2} - 2y_{i-1} + y_i\right)t^2 + 2(y_{i+1}-y_i)t + y_i \end{aligned}\right\}(i=n-2) \tag{C.120}$$

C.6 Gauss Quadrature

The Gauss Quadrature rule uses optimal abscissa distributions to yield accurate approximations to integrals using about half as many sample points as simple Riemann sums such as the trapezoidal rule [30]. Such optimal spacings generally yield abscissas that are specified by irrational numbers [36]. Usually the weightings on the ordinates are also irrational [36].

We assume we have a weighting function $w(x)$ defined over the integration interval $[a, b]$. Then the inner product of two functions $f(x)$ and $g(x)$ over the interval $[a, b]$

with respect to the weighting function $w(x)$ is given by [16]:

$$(f,g) = \int_a^b w(x)\, f(x)\, g(x)\, dx \tag{C.121}$$

When $f(x)$ and $g(x)$ are orthogonal to each other over the interval $[a,b]$ with respect to the weighting $w(x)$, we have

$$(f,g) = \int_a^b w(x)\, f(x)\, g(x)\, dx = 0 \tag{C.122}$$

The functions $f(x)$ and $g(x)$ can be replaced by two polynomials out of a sequence of polynomials that are all orthogonal to each other. Such a sequence can be found for a given weighting function $w(x)$. This sequence is defined as $p_0(x)$, $p_1(x)$, ..., and the n-th polynomial $p_n(x)$ is of degree n. Thus we have

$$(p_m, p_n) = \int_a^b w(x)\, p_m(x)\, p_n(x)\, dx = 0, \qquad m \neq n \tag{C.123}$$

A set of orthonormal polynomials $p_0^*(x)$, $p_1^*(x)$, ..., can be produced by multiplying each $p_i^*(x)$ by a special constant [16]. This set of orthonormal polynomials has the property that

$$(p_m^*, p_n^*) = \int_a^b w(x)\, p_m^*(x) p_n^*(x)\, dx = \delta_{mn} = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases} \tag{C.124}$$

Common sets of orthogonal polynomials are the Legendre, Chebyshev (first and second kind), Gegenbauer and Jacobi polynomials [16]. The zeros of real orthogonal polynomials are real, simple, and all lie in the interior of the interval $[a,b]$. This is a key property that is used in the derivation of the Gauss quadrature rules. In addition, the orthonormal polynomials $p_i^*(x)$ satisfy a three-term recurrence relation

of the form

$$p_i^*(x) = (a_i x + b_i)p_{i-1}^*(x) - c_i p_{i-2}^*(x), \quad i = 1, 2, 3, \ldots \tag{C.125}$$

where a_i, $c_i \neq 0$, $p_{-1}^*(x) = 0$, and $p_0^* = \sqrt{\int_a^b w(x)\, dx}$.

Now, given a set of n nodes x_1, x_2, ..., x_n located in the interior of $[a, b]$, a sequence of weightings w_1, w_2, ..., w_n can be found such that the integral

$$\int_a^b w(x)\, f(x)\, dx \approx \sum_{i=1}^{n} w_i\, f(x_i) \tag{C.126}$$

will be exact when $f(x)$ is a polynomial of degree $n - 1$ or less [18, 16]. However, if the nodes and weights can both be chosen freely, there are $2n$ choices, and there is a combination of them that allows the rule to be exact for polynomials of degree $2n - 1$ or less. To accomplish this, the nodes need to be chosen as the zeros of the set of polynomials $p_n^*(x)$ that are orthonormal to the weighting function $w(x)$ on the interval $[a, b]$. Thus we have that the x_i are the zeros of $p_n^*(x)$:

$$a < x_1 < x_2 < \cdots < x_n < b \tag{C.127}$$

A sequence of weights w_i can be found such that

$$\int_a^b w(x)\, f(x)\, dx = \sum_{i=1}^{n} w_i\, f(x_i) \tag{C.128}$$

if $f(x)$ is a polynomial of degree $2n - 1$ or less. In fact, the weights w_i are positive and can be found from [16]:

$$w_i = -\frac{k_{n+1}}{k_n} \frac{1}{p_{n+1}^*\, p_n^{*\prime}(x_i)} \tag{C.129}$$

where k_i is the leading term of the i-th orthonormal polynomial $p_i^*(x)$:

$$p_i^*(x) = k_i x^i + \cdots, \quad k_i > 0 \tag{C.130}$$

and k_i is taken to be positive.

For the Gauss-Legendre quadrature rule, the weighting function $w(x)$ is simply taken to be unity, and the rule has the form

$$\int_{-1}^{1} f(x)\, dx = \sum_{i=1}^{n} w_i\, f(x_i) + E_n \tag{C.131}$$

The orthogonal polynomials $p_i(x)$ are taken to be the Legendre polynomials. These polynomials have the form

$$P_n(x) = \frac{1}{2^n n!}\left[\frac{d^n}{dx^n}(x^2-1)^n\right]. \tag{C.132}$$

The sampling points x_i are the n zeros of the Legendre polynomial of degree n, which all lie in the interval $[-1,1]$ [24]. The argument function and the integration interval $[a,b]$ can be linearly transformed to the interval $[-1,1]$ by the change of variables

$$u = 2\left(\frac{x-a}{b-a}\right) - 1 \tag{C.133}$$

The w_i, the weightings on the ordinates, are computed with any of the following formulae [32, 30, 35]:

$$w_i = \frac{2}{(n+1)P_{n+1}(x_i)P_n'(x_i)} = \frac{2}{nP_{n-1}(x_i)P_n'(x_i)} = \frac{2}{(1-x_i^2)[P_n'(x_i)]^2} \tag{C.134}$$

where $P_n'(x) = \frac{d}{dx}P_n(x)$. For the purposes of calculating values of the Legendre

polynomials, it is useful to use the recursion formula [30]:

$$P_0(x) = 1 \tag{C.135}$$

$$P_1(x) = x \tag{C.136}$$

$$P_2(x) = \frac{1}{2}(3x^1 - 1) \tag{C.137}$$

$$P_{n+1}(x) = \frac{2n+1}{n+1}\, x\, P_n(x) - \frac{m}{m+1} P_{n-1}(x) \tag{C.138}$$

The error incurred by the use of the Gauss rule is

$$E_n = \int_{-1}^{1} f(x)\, dx - \sum_{i=1}^{n} w_i\, f(x_i) \tag{C.139}$$

and is given by [16, 30, 18, 35]:

$$E_n = \frac{2^{2n+1}\,(n!)^4}{(2n+1)\,[(2n)!]^3}\, f^{(2n)}(\xi), \quad -1 < \xi < 1 \tag{C.140}$$

where $f^{(2n)}(x)$ is the $2n$-th derivative of $f(x)$.

C.7 Interpolation by Chebyshev Polynomials

As an alternative to the method presented in Chapter 4, the interpolating polynomial for $p(x)$ that passes through the points $f_i = f(x_i)$ can be found from the Lagrange interpolation formula [16]. We begin by assuming that the abscissas x_i are distinct and ordered and in some interval $[a, b]$:

$$a \leq x_1 < x_2 < \cdots < x_n \leq b \tag{C.141}$$

We have n data points $f_i = f(x_i)$ that need to be interpolated. This can be accomplished by a unique polynomial $p_{n-1}(x)$ of degree $n-1$ that has the property

$$p_{n-1}(x_i) = f_i \tag{C.142}$$

The polynomial p_{n-1} can be written in the following way. We define n fundamental polynomials $l_i(x)$ of degree $n-1$ that are given by

$$l_i(x) = \frac{(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}, \quad i = 1, 2, \ldots, n \tag{C.143}$$

The $l_i(x)$ can also be written in the form

$$l_i(x) = \frac{\pi(x)}{(x-x_i)\,\pi'(x_i)} \tag{C.144}$$

where

$$\pi(x) = (x-x_1)(x-x_2)\cdots(x-x_n) \tag{C.145}$$

The polynomials $l_i(x)$ have the property that

$$l_i(x_j) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & j \neq j \end{cases} \tag{C.146}$$

We can now write the interpolating polynomial $p_{n-1}(x)$ in terms of the fundamental polynomials $l_i(x)$ as

$$p_{n-1}(x) = \sum_{i=1}^{n} f(x_i)\, l_i(x) \tag{C.147}$$

The desired integral can then be written as

$$\int_a^b p_{n-1}(x)\, dx = \sum_{i=1}^{n} f(x_i) \int_a^b l_i(x)\, dx \tag{C.148}$$

Appendix D

Results of Static Shape Estimation

D.1 Midpoint Rule

Table D.1: Results of static shape estimation using the midpoint rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	9.8137	0.3324	1.6632	10.0406	0.4017	2.0096
4	1.2318	0.2965	1.4833	1.8177	0.3360	1.6807
6	0.8515	0.2829	1.4155	1.2408	0.3129	1.5652
8	0.5517	0.2500	1.2503	0.7955	0.2677	1.3391
10	0.3825	0.2323	1.1620	0.5484	0.2418	1.2098

Table D.2: Results of static shape estimation using the midpoint rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	6.1463	0.3029	1.2463	6.6683	0.3434	1.4390
4	2.1281	0.2697	1.1329	2.7653	0.3098	1.3361
6	1.2381	0.2472	0.9987	1.6096	0.2829	1.1770
8	0.5899	0.2257	0.9027	0.8073	0.2548	1.0591
10	0.3304	0.1967	0.8206	0.4751	0.2183	0.9578

Table D.3: Results of static shape estimation using the midpoint rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	7.8422	0.3241	1.2493	8.2755	0.3599	1.4472
4	1.5248	0.3078	1.1466	2.2073	0.3474	1.3282
6	1.0696	0.2509	1.0619	1.4715	0.2864	1.2212
8	0.6822	0.2149	0.9468	0.9299	0.2441	1.0813
10	0.4688	0.1901	0.8511	0.6360	0.2160	0.9688

D.2 Trapezoidal Rule

Table D.4: Results of static shape estimation using the trapezoidal rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	9.8137	0.2838	1.4209	11.2107	0.3220	1.6111
4	1.2318	0.2986	1.4935	1.0984	0.3261	1.6312
6	0.8515	0.2609	1.3054	0.5072	0.2880	1.4405
8	0.5517	0.2318	1.1596	0.3354	0.2605	1.3031
10	0.3825	0.1961	0.9811	0.2364	0.2202	1.1017

Table D.5: Results of static shape estimation using the trapezoidal rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	6.1463	0.2755	1.1696	7.8229	0.2876	1.2579
4	2.1281	0.2602	1.1296	1.3096	0.2829	1.2670
6	1.2381	0.2376	0.9908	0.7440	0.2648	1.1162
8	0.5899	0.2185	0.8917	0.3045	0.2449	1.0203
10	0.3304	0.1940	0.8086	0.1548	0.2171	0.9208

Table D.6: Results of static shape estimation using the trapezoidal rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	7.8422	0.2995	1.1539	9.3881	0.3011	1.2364
4	1.5248	0.3095	1.1355	0.9956	0.3296	1.2639
6	1.0696	0.2437	1.0319	0.6093	0.2692	1.1648
8	0.6822	0.2077	0.8994	0.4049	0.2326	1.0339
10	0.4688	0.1903	0.8197	0.2847	0.2132	0.9318

D.3 Simpson's Rule

Table D.7: Results of static shape estimation using Simpson's rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
3	3.2982	0.2779	1.3907	3.8291	0.3150	1.5758
5	0.3345	0.2631	1.3161	0.3349	0.2935	1.4683
7	0.0936	0.2269	1.1355	0.1034	0.2547	1.2742
9	0.0734	0.2051	1.0257	0.0650	0.2300	1.1505
11	0.0185	0.1784	0.8927	0.0204	0.2013	1.0071

Table D.8: Results of static shape estimation using Simpson's rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
3	1.7098	0.3371	1.0810	2.2955	0.3565	1.2284
5	0.5773	0.2503	1.0492	0.4902	0.2785	1.1955
7	0.2070	0.2005	0.9084	0.1638	0.2268	1.0424
9	0.4509	0.1903	0.8192	0.4237	0.2104	0.9454
11	0.1221	0.1704	0.7272	0.1015	0.1880	0.8406

Table D.9: Results of static shape estimation using Simpson's rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
3	2.5446	0.2931	1.0743	3.1120	0.3155	1.2224
5	0.2360	0.2506	1.0354	0.2401	0.2789	1.1749
7	0.1439	0.2040	0.9502	0.1191	0.2291	1.0817
9	0.0979	0.1864	0.8094	0.0825	0.2096	0.9374
11	0.0781	0.1675	0.7487	0.0655	0.1885	0.8566

D.4 Cubic Interpolating Spline Rule

Table D.10: Results of static shape estimation using the Cubic Spline rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	10.4176	0.2706	1.3546	11.0719	0.3246	1.6243
4	0.7399	0.2710	1.3555	0.8839	0.3027	1.5142
6	0.1212	0.2445	1.2227	0.1293	0.2756	1.3786
8	0.0500	0.2263	1.1320	0.0519	0.2546	1.2735
10	0.0280	0.1872	0.9362	0.0286	0.2123	1.0619

Table D.11: Results of static shape estimation using the Cubic Spline rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	6.9190	0.2510	1.0806	7.7763	0.2811	1.2270
4	0.8780	0.2399	1.0499	0.8536	0.2704	1.2011
6	0.4446	0.2237	0.9483	0.3839	0.2548	1.0956
8	0.0884	0.1958	0.8557	0.0586	0.2219	0.9852
10	0.0224	0.1761	0.7624	0.0613	0.2011	0.8843

Table D.12: Results of static shape estimation using the Cubic Spline rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	8.5834	0.2874	1.0379	9.3482	0.3117	1.2103
4	0.5879	0.2834	1.0488	0.7376	0.3114	1.2044
6	0.2854	0.2268	0.9568	0.2576	0.2573	1.1088
8	0.1742	0.1957	0.8560	0.1559	0.2220	0.9824
10	0.1165	0.1809	0.7700	0.1041	0.2039	0.8885

D.5 B-Spline Rule

Table D.13: Results of static shape estimation using the B-Spline rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	9.4405	0.3100	1.5518	10.5734	0.3704	1.8532
4	2.1695	0.3044	1.5227	1.8412	0.3392	1.6967
6	1.4134	0.2780	1.3905	1.1908	0.3116	1.5587
8	0.8988	0.2369	1.1848	0.7657	0.2698	1.3496
10	0.6197	0.2212	1.1067	0.5299	0.2506	1.2539

Table D.14: Results of static shape estimation using the B-Spline rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	5.6711	0.2919	1.2253	6.8473	0.3137	1.3371
4	3.0959	0.2721	1.2290	2.4807	0.3017	1.3904
6	1.7865	0.2601	1.0960	1.4708	0.2945	1.2493
8	0.9243	0.2310	0.9435	0.7318	0.2625	1.0826
10	0.5587	0.2058	0.8452	0.4297	0.2334	0.9847

Table D.15: Results of static shape estimation using the B-Spline rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	7.3124	0.3203	1.2238	8.4894	0.3295	1.3562
4	2.4633	0.3283	1.1831	1.9574	0.3580	1.3439
6	1.6170	0.2605	1.0992	1.3307	0.2926	1.2654
8	1.0197	0.2266	0.9867	0.8498	0.2564	1.1341
10	0.6992	0.1956	0.8720	0.5852	0.2219	1.0068

D.6 Gauss Rule

Table D.16: Results of static shape estimation using the Gauss rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	0.9874	0.2864	1.4334	-	-	-
4	0.2042	0.1976	0.9885	-	-	-
6	0.0949	0.1589	0.7950	-	-	-
8	0.0549	0.1456	0.7286	-	-	-
10	0.0358	0.1328	0.6640	-	-	-

Table D.17: Results of static shape estimation using the Gauss rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	2.5721	0.3772	1.1163	-	-	-
4	0.3461	0.1887	0.7723	-	-	-
6	0.2602	0.1429	0.6472	-	-	-
8	0.3962	0.1345	0.5671	-	-	-
10	0.0492	0.1146	0.5137	-	-	-

Table D.18: Results of static shape estimation using the Gauss rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	1.3107	0.2926	1.0817	-	-	-
4	0.2072	0.1829	0.7893	-	-	-
6	0.1289	0.1513	0.6439	-	-	-
8	0.0960	0.1362	0.5717	-	-	-
10	0.0598	0.1244	0.5113	-	-	-

D.7 Radau Rule

Table D.19: Results of static shape estimation using the Radau rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	1.3632	0.2732	1.3666	-	-	-
4	0.2398	0.1975	0.9878	-	-	-
6	0.0846	0.1574	0.7875	-	-	-
8	0.0400	0.1437	0.7190	-	-	-
10	0.0223	0.1265	0.6328	-	-	-

Table D.20: Results of static shape estimation using the Radau rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	2.3704	0.2244	1.0476	-	-	-
4	0.6173	0.2062	0.7517	-	-	-
6	0.3709	0.1597	0.6119	-	-	-
8	0.1936	0.1456	0.5656	-	-	-
10	0.1600	0.1350	0.5005	-	-	-

Table D.21: Results of static shape estimation using the Radau rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	2.3241	0.2529	1.0251	-	-	-
4	0.1330	0.1785	0.7478	-	-	-
6	0.0883	0.1443	0.6143	-	-	-
8	0.0642	0.1306	0.5409	-	-	-
10	0.0383	0.1198	0.4979	-	-	-

D.8 Chebyshev Rule

Table D.22: Results of static shape estimation using the Chebyshev rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	10.7948	0.2537	1.2701	11.2871	0.3110	1.5564
4	0.9159	0.2726	1.3636	0.8671	0.3166	1.5835
6	0.1382	0.2481	1.2412	0.2761	0.2966	1.4839
8	0.1006	0.2353	1.1774	0.1710	0.3190	1.5959
10	0.0822	0.2435	1.2180	0.2985	0.4305	2.1532

Table D.23: Results of static shape estimation using the Chebyshev rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	7.3894	0.2430	1.0207	8.0600	0.2725	1.1718
4	0.7108	0.2405	1.0202	0.7428	0.2794	1.2080
6	0.3336	0.2321	0.9667	0.1555	0.2830	1.1652
8	0.0714	0.2228	0.9188	0.1412	0.2926	1.2151
10	0.1124	0.2448	0.9322	0.2800	0.4068	1.6098

Table D.24: Results of static shape estimation using the Chebyshev rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	9.0180	0.2700	1.0156	9.6050	0.2973	1.1848
4	0.6355	0.2769	1.0297	0.6832	0.3151	1.2077
6	0.1785	0.2363	0.9562	0.1114	0.2825	1.1650
8	0.1404	0.2098	0.9252	0.2524	0.2777	1.2433
10	0.1186	0.2410	1.0195	0.1881	0.4113	1.7304

D.9 Global Static Shape Function Rule

Table D.25: Results of static shape estimation using the GSSF rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	-	-	-	5.5045	0.1449	0.7417
4	-	-	-	0.1155	0.0731	0.2886
6	-	-	-	0.0062	0.0494	0.2857
8	-	-	-	0.0000	0.0498	0.2354
10	-	-	-	0.0000	0.0521	0.2491

Table D.26: Results of static shape estimation using the GSSF rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	-	-	-	0.9432	0.1137	0.4070
4	-	-	-	0.9282	0.0899	0.2973
6	-	-	-	0.4130	0.1094	0.2032
8	-	-	-	0.1874	0.2422	0.2056
10	-	-	-	0.1313	0.1803	0.2861

Table D.27: Results of static shape estimation using the GSSF rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	-	-	-	3.2060	0.1431	0.7385
4	-	-	-	0.5176	0.1448	0.3601
6	-	-	-	0.2359	0.1072	0.2553
8	-	-	-	0.1139	0.0713	0.2194
10	-	-	-	0.0862	0.0545	0.2302

D.10 Global Dynamic Shape Function Rule

Table D.28: Results of static shape estimation using the GDSF rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	-	-	-	4.8842	0.1543	0.7420
4	-	-	-	0.4163	0.0936	0.2886
6	-	-	-	0.2150	0.0733	0.2782
8	-	-	-	0.0919	0.0477	0.2056
10	-	-	-	0.0628	0.0418	0.1969

Table D.29: Results of static shape estimation using the GDSF rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	-	-	-	1.3698	0.0925	0.3436
4	-	-	-	0.7950	0.0659	0.3050
6	-	-	-	0.3133	0.1572	0.2182
8	-	-	-	0.2137	0.2287	0.2025
10	-	-	-	0.1279	0.1638	0.2186

Table D.30: Results of static shape estimation using the GDSF rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	-	-	-	2.6726	0.1295	0.6482
4	-	-	-	0.4282	0.0870	0.3014
6	-	-	-	0.1756	0.0544	0.2422
8	-	-	-	0.1265	0.0550	0.1997
10	-	-	-	0.0763	0.0352	0.1977

D.11 Least Squares Global Static Shape Function Rule With 2 Sensors

Table D.31: Results of static shape estimation using the LSGSSF-2 rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	-	-	-	5.5045	0.1455	0.7743
4	-	-	-	1.1976	0.1369	0.5211
6	-	-	-	0.5241	0.1062	0.3234
8	-	-	-	0.2933	0.0931	0.2639
10	-	-	-	0.1873	0.0821	0.2089

Table D.32: Results of static shape estimation using the LSGSSF-2 rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	-	-	-	0.9432	0.1089	0.4127
4	-	-	-	0.1811	0.1384	0.2598
6	-	-	-	0.1266	0.2392	0.2479
8	-	-	-	0.0738	0.2433	0.2503
10	-	-	-	0.0635	0.1922	0.2462

Table D.33: Results of static shape estimation using the LSGSSF-2 rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
2	-	-	-	3.2060	0.1431	0.7082
4	-	-	-	0.6748	0.1388	0.4104
6	-	-	-	0.2948	0.1071	0.2845
8	-	-	-	0.1655	0.0737	0.2151
10	-	-	-	0.1052	0.0530	0.2079

D.12 Least Squares Global Static Shape Function Rule With 4 Sensors

Table D.34: Results of static shape estimation using the LSGSSF-4 rule and point sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
4	-	-	-	0.1155	0.0716	0.3038
6	-	-	-	0.0230	0.0493	0.2902
8	-	-	-	0.0115	0.0437	0.2432
10	-	-	-	0.0071	0.0455	0.2341

Table D.35: Results of static shape estimation using the LSGSSF-4 rule and rectangular sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
4	-	-	-	0.9282	0.0878	0.3162
6	-	-	-	0.4029	0.1122	0.2004
8	-	-	-	0.2286	0.1947	0.2003
10	-	-	-	0.1370	0.1391	0.2309

Table D.36: Results of static shape estimation using the LSGSSF-4 rule and Bartlett sensors.

Number Gages	Slope Estimation			Deflection Estimation		
	Absolute Error (%)	σ_o (%)	σ_r (%)	Absolute Error (%)	σ_o (%)	σ_r (%)
4	-	-	-	0.5176	0.1345	0.3759
6	-	-	-	0.2386	0.0949	0.2556
8	-	-	-	0.1339	0.0839	0.2135
10	-	-	-	0.0862	0.0564	0.1815

Appendix E

Source Code

E.1 Midpoint Integration: midpoint.m

```
function [integral,tip] = midpoint(x,y,initial)
%MIDPOINT
%
% Midpoint integration.
%
% MIDPOINT computes the running integral of the curve (x,y) using
%       the midpoint rule.  It is assumed the interval of integration
%       begins at x=0.
%
%       integral: Running integral of the curve (x,y).
%       tip:      Value of the integral at the end of the interval.
%       x:        Horizontal abscissae of curve.
%       y:        Vertical values of curve.
%       initial:  Initial value of integral at the start of the interval.

%-------------------------------------------------------------------------
%
%       Author: Mark S. Andersson.
%       Date:   January 10, 1992.
%       Copyright (c) 1992
%-------------------------------------------------------------------------

if nargin < 3,
  initial = 0;
end

[mx,nx] = size(x);
[my,ny] = size(y);

if((mx ~= 1) & (nx ~= 1))
error('The x vector must be a column or row vector.')
```

```
end

if((my ~= 1) & (ny ~= 1))
error('The y vector must be a column or row vector.')
end

% Check that x and y are the same size:

x = x(:)';
y = y(:)';

if (length(x) ~= length(y))
        error('x and y must be the same size.');
end

% Initialize the integral vector:

integral = 0*ones(mx,1);

% Set the initial condition at the start of the interval:

integral(1,1)=initial + (x(1,1)-0)*y(1);

% Sum up the areas of the rectangular areas:

for i = 2:mx,
integral(i,1) = integral(i-1,1) + ...
    ((x(i,1)+x(i-1,1))/2 - x(i-1,1)) * y(i-1) + ...
    (x(i,1) - (x(i,1)+x(i-1,1))/2) * y(i);
end

% Find the tip deflection:

tip = integral(mx,1) + (x(mx,1) - (x(mx,1)+x(mx-1,1))/2)*y(mx);
```

E.2 Trapezoidal Integration: trapezoidal.m

```
function [integral] = trapezoidal(x,y,initial)
%TRAPEZOIDAL
%
%       Trapezoidal integration.
%
%       TRAPEZOIDAL computes the running integral of the curve (x,y)
%       using the Trapezoidal rule:
%
%       integral = trapezoidal(x,y,initial)
%
%       integral: Running integral of the curve (x,y).
%       x:        Horizontal abscissae of curve.
%       y:        Vertical values of curve.
%       initial:  Initial value of integral at the start of the interval.
```

```
%-------------------------------------------------------------------------
%
%       Author: Mark S. Andersson.
%       Date:   February 7, 1991.
%       Copyright (c) 1991
%-------------------------------------------------------------------------

if nargin < 3,
  initial = 0;
end

[mx,nx] = size(x);
[my,ny] = size(y);

if((mx ~= 1) & (nx ~= 1))
error('The x vector must be a column or row vector.')
end

if((my ~= 1) & (ny ~= 1))
error('The y vector must be a column or row vector.')
end

% Check that x and y are the same size:

x = x(:)';
y = y(:)';

if (length(x) ~= length(y))
error('x and y must be the same size.');
end

% Initialize the integral vector:

integral = zeros(length(x),1);

% Set the initial condition at the start of the interval:

integral(1) = initial;

% Sum up the areas of the trapezoids:

for i = 2:mx,
integral(i,1)=integral(i-1,1)+(x(i,1)-x(i-1,1))*(y(i) + y(i-1))/2;
end
```

E.3 Simpson's Rule Integration: simpson.m

```
function [integral] = simpson(x,y,initial)
%SIMPSON
%
%       Simpson integration.
%
```

```
%       SIMPSON computes the running integral of the curve (x,y)
%       using Simpson's rule:
%
%       integral = simpson(x,y,initial)
%
%       integral: Running integral of the curve (x,y).
%       x:        Horizontal abscissae of curve.
%       y:        Vertical values of curve.
%       initial:  Initial value of integral at the start of the interval.

%-------------------------------------------------------------------------
%
%       Author: Mark S. Andersson.
%       Date:   January 10, 1992.
%       Copyright (c) 1992
%-------------------------------------------------------------------------

if nargin < 3,
  initial = 3;
end

[mx,nx] = size(x);
[my,ny] = size(y);

if((mx ~= 1) & (nx ~= 1))
error('The x vector must be a column or row vector.')
end

if((my ~= 1) & (ny ~= 1))
error('The y vector must be a column or row vector.')
end

% Check that x and y are the same size:

x = x(:)';
y = y(:)';

if (length(x) ~= length(y))
        error('x and y must be the same size.');
end

if (mod(length(x),2) ~= 1)
error('x and y must contain an odd number of entries.');
end

% Initialize the integral vector:

integral = 0*ones(mx,1);

% Set the initial condition at the start of the interval:

integral(1) = initial;
```

```
% Sum up the areas under each of the parabolic segments:

for i = 1:2:mx-2,
  % Solve for the parabolic segment.
  %
  % Governing equations:
  %
  %   a * x1^2 + b * x1 + c = y1
  %   a * x2^2 + b * x2 + c = y2
  %   a * x3^2 + b * x3 + c = y3
  %
  % Construct matrix and compute coefficients a, b, c:
  %
  %   [ x1^2  x1  1 ] [ a ]   [ y1 ]
  %   [ x2^2  x2  1 ] [ b ] = [ y2 ]
  %   [ x3^2  x3  1 ] [ c ]   [ y3 ]
  %

  A = [ x(i)*x(i)      x(i)    1
        x(i+1)*x(i+1)  x(i+1)  1
        x(i+2)*x(i+2)  x(i+2)  1 ];

  Y = [ y(i)
        y(i+1)
        y(i+2) ];

  abc = A \ Y;

  a = abc(1);
  b = abc(2);
  c = abc(3);

  integral(i+1) = integral(i) + ...
      (1/3) * a * (x(i+1)^3-x(i)^3) + ...
      (1/2) * b * (x(i+1)^2-x(i)^2) + ...
      (1/1) * c * (x(i+1)-x(i));

  integral(i+2) = integral(i) + ...
      (1/3) * a * (x(i+2)^3-x(i)^3) + ....
      (1/2) * b * (x(i+2)^2-x(i)^2) + ...
      (1/1) * c * (x(i+2)-x(i));
end
```

E.4 Computation of Open Natural Cubic Splines: spline.m

```
function [xout,yout] = spline(xin,yin,pts)
%SPLINE
%
%       Open Cubic Spline Interpolation
%
%       SPLINE computes the coordinates of open natural parametric
```

```
%       cubic splines, for the control vertices specified by the xin
%       and yin vectors.
%
%       [xout,yout] = spline(xin,yin,pts)
%
%       xout:  output x coordinates
%       yout:  output y coordinates
%       xin:   input x coordinates
%       yin:   input y coordinates
%       pts:   number of points to be used for each curve segment
%              (optional: 10 points are assumed)
%

%-------------------------------------------------------------------------
%
%       Author: Mark S. Andersson.
%       Date:   September 30, 1991.
%       Copyright (c) 1991
%-------------------------------------------------------------------------

% Handle cases where optional arguments are not given:

if nargin == 2,
  pts   = 10;
elseif nargin < 2,
  error('Too few arguments given.');
elseif nargin > 3,
  error('Too many arguments given.');
end

%-------------------------------------------------------------------------
%
% 1. First create the basis curves that will form the spline.  These
%    are the four Hermite polynomials given in Appendix C.
%-------------------------------------------------------------------------

% Create the parametric variable t:

t = linspace(0,1,pts);

h = 0*ones(4,pts);

h(1,:) = (2*t + 1) .* (t - 1).^2;
h(2,:) = t.^2 .* (3 - 2*t);
h(3,:) = t .* (t - 1).^2;
h(4,:) = t.^2 .* (t - 1);

%-------------------------------------------------------------------------
%
% 2. Construct the A matrix to solve for the derivatives of the
%    x-coordinate and y-coordinate parametric functions of t.  These
%    are Dx = dX/dt and Dy = dY/dt.  (Di in Appendix C).
%
```

```
%    The matrix equation (C.104) will be in the form:
%
%    [ A ] { Dx } = { Bx }    (for the x-coordinates)
%    [ A ] { Dy } = { By }    (for the y-coordinates)
%-------------------------------------------------------------------------

% Make sure input vectors are in row vector form:

xin = xin(:)';
yin = yin(:)';

% Check to see whether they have the same number of elements:

if length(xin) ~= length(yin),
  error('Input vectors must contain the same number of elements.');
end

% Find length of input vector:

n = length(xin);

% Create the A matrix:

A = zeros(n,n);

% Specify the first row:

A(1,1) = 2;
A(1,2) = 1;

% Specify rows 2 up to and including row n-1.

for i = 2:n-1,
  A(i,i-1) = 1;
  A(i,i)   = 4;
  A(i,i+1) = 1;
end

% Specify the last row:

A(n,n-1) = 1;
A(n,n)   = 2;

% Create the Bx and By matrices:

Bx = zeros(n,1);
By = zeros(n,1);

% Specify the first row:

Bx(1,1) = 3 * (xin(2) - xin(1));
By(1,1) = 3 * (yin(2) - yin(1));
```

```
% Specify rows 2 up to and including row n-1.

for i = 2:n-1,
  Bx(i,1) = 3 * (xin(i+1) - xin(i-1));
  By(i,1) = 3 * (yin(i+1) - yin(i-1));
end

% Specify the last row:

Bx(n,1) = 3 * (xin(n) - xin(n-1));
By(n,1) = 3 * (yin(n) - yin(n-1));

%-------------------------------------------------------------------------
%
% 3. Solve for the derivatives of the x- and y-coordinates with respect
%    to the parametric variable t and the control points, Dx and Dy.
%-------------------------------------------------------------------------

Dx = A \ Bx;
Dy = A \ By;

%-------------------------------------------------------------------------
%
% 4. Compute the number of curve segments in the entire spline:
%-------------------------------------------------------------------------

n = length(xin) - 1;

% Allocate space for the output vectors:

xout = 0*ones(1,n*pts);
yout = 0*ones(1,n*pts);

%-------------------------------------------------------------------------
%
% 5. Sum the first Hermite basis vector multiplied by the beginning
%    coordinate, the second Hermite basis vector multiplied by the
%    ending coordinate, the third Hermite basis vector multiplied by
%    the beginning slope, and the fourth Hermite basis vector multiplied
%    by the ending slope, over the domain of the spline.  This is
%    equation (C.105) in Appendix C.
%-------------------------------------------------------------------------

for i=1:n,
  xout(pts*(i-1)+1:pts*i) = xout(pts*(i-1)+1:pts*i) + xin(i)*h(1,:);
  xout(pts*(i-1)+1:pts*i) = xout(pts*(i-1)+1:pts*i) + xin(i+1)*h(2,:);
  xout(pts*(i-1)+1:pts*i) = xout(pts*(i-1)+1:pts*i) + Dx(i)*h(3,:);
  xout(pts*(i-1)+1:pts*i) = xout(pts*(i-1)+1:pts*i) + Dx(i+1)*h(4,:);

  yout(pts*(i-1)+1:pts*i) = yout(pts*(i-1)+1:pts*i) + yin(i)*h(1,:);
  yout(pts*(i-1)+1:pts*i) = yout(pts*(i-1)+1:pts*i) + yin(i+1)*h(2,:);
  yout(pts*(i-1)+1:pts*i) = yout(pts*(i-1)+1:pts*i) + Dy(i)*h(3,:);
  yout(pts*(i-1)+1:pts*i) = yout(pts*(i-1)+1:pts*i) + Dy(i+1)*h(4,:);
```

```
end

%---------------------------------------------------------------------
```

E.5 Computation of Open B-Splines: bspline.m

```
function [xout,yout] = bspline(xin,yin,order,pts)
%BSPLINE
%
%       Open B-Spline Approximation
%
%       BSPLINE computes the coordinates of open B-splines of
%       order 2, 3, 4 and 5, for the control vertices specified
%       by the xin and yin vectors.
%
%       [xout,yout] = bspline(xin,yin,order,pts)
%
%       xout:  output x coordinates
%       yout:  output y coordinates
%       xin:   input x coordinates
%       yin:   input y coordinates
%       order: B-spline order (optional: order 2 is assumed)
%       pts:   number of points to be used for each curve segment
%              (optional: 10 points are assumed)
%

%---------------------------------------------------------------------
%
%       Author: Mark S. Andersson.
%       Date:   September 30, 1991.
%       Copyright (c) 1991
%---------------------------------------------------------------------

% Handle cases where optional arguments are not given:

if nargin == 2,
  order = 2;
  pts   = 10;
elseif nargin == 3,
  pts   = 10;
elseif nargin < 2,
  error('Too few arguments given.');
elseif nargin > 4,
  error('Too many arguments given.');
end

%---------------------------------------------------------------------
%
% 1. First create the basis curves that will form the B-spline.
%---------------------------------------------------------------------

% Create the parametric variable t:
```

```
t = linspace(0,1,pts);

if order == 2,
  % Create basis curves for B-splines of order 2:

  b2 = 0*ones(3,pts);

  b2(1,:) = (1/2) * (t.^2);
  b2(2,:) = (1/2) * (1 + 2*t - 2*t.^2);
  b2(3,:) = (1/2) * (1 - 2*t + t.^2);

elseif order == 3,
  % Create basis curves for B-splines of order 3:

  b3 = 0*ones(4,pts);

  b3(1,:) = (1/6) * (t.^3);
  b3(2,:) = (1/6) * (1 + 3*t + 3*t.^2 - 3*t.^3);
  b3(3,:) = (1/6) * (4 - 6*t.^2 + 3*t.^3);
  b3(4,:) = (1/6) * (1 - 3*t + 3*t.^2 - t.^3);

elseif order == 4,
  % Create basis curves for B-splines of order 4:

  b4 = 0*ones(5,pts);

  b4(1,:) = (1/24) * (t.^4);
  b4(2,:) = (1/24) * (1 + 4*t + 6*t.^2 + 4*t.^3 - 4*t.^4);
  b4(3,:) = (1/24) * (11 + 12*t - 6*t.^2 - 12*t.^3 + 6*t.^4);
  b4(4,:) = (1/24) * (11 - 12*t - 6*t.^2 + 12*t.^3 - 4*t.^4);
  b4(5,:) = (1/24) * (1 - 4*t + 6*t.^2 - 4*t.^3 + t.^4);

elseif order == 5,
  % Create basis curves for B-splines of order 5:

  b5 = 0*ones(6,pts);

  b5(1,:) = (1/120) * (t.^5);
  b5(2,:) = (1/120) * (1 + 5*t + 10*t.^2 + 10*t.^3 + 5*t.^4 - 5*t.^5);
  b5(3,:) = (1/120) * (26 + 50*t + 20*t.^2 - 20*t.^3 -20*t.^4 + 10*t.^5);
  b5(4,:) = (1/120) * (66 - 60*t.^2 + 30*t.^4 - 10*t.^5);
  b5(5,:) = (1/120) * (26 - 50*t + 20*t.^2 + 20*t.^3 - 20*t.^4 + 5*t.^5);
  b5(6,:) = (1/120) * (1 - 5*t + 10*t.^2 - 10*t.^3 + 5*t.^4 - t.^5);
end

%-------------------------------------------------------------------------
%
% 2. Extend the xin and yin vectors to include enough copies of
%    the starting and ending coordinates so that the B-spline curve
%    will explicitly connect to the starting and ending coordinates
%-------------------------------------------------------------------------

% Make sure input vectors are in row vector form:
```

```
xin = xin(:)';
yin = yin(:)';

% Check to see whether they have the same number of elements:

if length(xin) ~= length(yin),
  error('Input vectors must contain the same number of elements.');
end

% Find length of input vector:

n = length(xin);

% Extend the input vectors:

if (order == 2) & (n > 2),
  xx = [ xin(1) xin xin(n) ];
  yy = [ yin(1) yin yin(n) ];
elseif (order == 3) & (n > 3),
  xx = [ xin(1) xin(1) xin xin(n) xin(n) ];
  yy = [ yin(1) yin(1) yin yin(n) yin(n) ];
elseif (order == 4) & (n > 4),
  xx = [ xin(1) xin(1) xin(1) xin xin(n) xin(n) xin(n) ];
  yy = [ yin(1) yin(1) yin(1) yin yin(n) yin(n) yin(n) ];
elseif (order == 5) & (n > 5),
  xx = [ xin(1) xin(1) xin(1) xin(1) xin xin(n) xin(n) xin(n) xin(n) ];
  yy = [ yin(1) yin(1) yin(1) yin(1) yin yin(n) yin(n) yin(n) yin(n) ];
else
  error('Order too high for number of points given.');
end

%----------------------------------------------------------------------
%
% 3. Compute the number of curve segments in the entire B-spline:
%----------------------------------------------------------------------

n = length(xx) - order;

% Allocate space for the output vectors:

xout = 0*ones(1,n*pts);
yout = 0*ones(1,n*pts);

%----------------------------------------------------------------------
%
% 4. Sum the basis vectors over the domain of the B-spline:
%----------------------------------------------------------------------

for i=1:n,
  for j=1:order+1,
    if order == 2,
      xout(pts*(i-1)+1:pts*i) = xout(pts*(i-1)+1:pts*i) + xx(i+j-1)*b2(4-j,:);
```

```
      yout(pts*(i-1)+1:pts*i) = yout(pts*(i-1)+1:pts*i) + yy(i+j-1)*b2(4-j,:);
    elseif order == 3,
      xout(pts*(i-1)+1:pts*i) = xout(pts*(i-1)+1:pts*i) + xx(i+j-1)*b3(5-j,:);
      yout(pts*(i-1)+1:pts*i) = yout(pts*(i-1)+1:pts*i) + yy(i+j-1)*b3(5-j,:);
    elseif order == 4,
      xout(pts*(i-1)+1:pts*i) = xout(pts*(i-1)+1:pts*i) + xx(i+j-1)*b4(6-j,:);
      yout(pts*(i-1)+1:pts*i) = yout(pts*(i-1)+1:pts*i) + yy(i+j-1)*b4(6-j,:);
    elseif order == 5,
      xout(pts*(i-1)+1:pts*i) = xout(pts*(i-1)+1:pts*i) + xx(i+j-1)*b5(7-j,:);
      yout(pts*(i-1)+1:pts*i) = yout(pts*(i-1)+1:pts*i) + yy(i+j-1)*b5(7-j,:);
    end
  end
end

%-------------------------------------------------------------------------
```

E.6 Chebyshev Integration: chebyshev.m

```
function [integral] = chebyshev(x,y,n,initial)
%CHEBYSHEV
%
%       Integration by interpolation by Chebyshev polynomials.
%
%       CHEBYSHEV computes the running integral of the curve (x,y)
%       using the Chebyshev integration rule:
%
%       integral = chebyshev(x,y,n,initial)
%
%       integral: Running integral of the curve (x,y).
%       n:        Number of polynomials to use.
%       x:        Horizontal abscissae of curve.
%       y:        Vertical values of curve.
%       initial:  Initial value of integral at the start of the interval.

%-------------------------------------------------------------------------
%
%       Author: Mark S. Andersson.
%       Date:   January 10, 1992.
%       Copyright (c) 1992
%-------------------------------------------------------------------------

[mx,nx] = size(x);
[my,ny] = size(y);

if((mx ~= 1) & (nx ~= 1))
  error('The x vector must be a column or row vector.')
end

if((my ~= 1) & (ny ~= 1))
  error('The y vector must be a column or row vector.')
end

% Check that x and y are the same size:
```

```
x = x(:)';
y = y(:)';

if (length(x) ~= length(y))
        error('x and y must be the same size.');
end

m = length(x);

if (n > m)
  error('Number of polynomials must be less than or equal to the number of abscissae.');
end

% First must fit n polynomials to the m values.
%
%   a T1(x1) + b T2(x1) + c T3(x1) + ... y Tn-1(x1) + z Tn(x1) = y1
%   a T1(x2) + b T2(x2) + c T3(x2) + ... y Tn-1(x2) + z Tn(x2) = y2
%   a T1(x3) + b T2(x3) + c T3(x3) + ... y Tn-1(x3) + z Tn(x3) = y3
%      :          :          :              :            :       :
%   a T1(xn) + b T2(xn) + c T3(xn) + ... y Tn-1(xn) + z Tn(xn) = yn
%
% etc.

% Form the T matrix:

T = zeros(m,n);

for i = 1:m,
  for j = 1:n,
    T(i,j) = cheby(x(i),j-1);
  end
end

% Solve for the coefficients (a, b, c, etc.):

c = T \ y;

% Compute the length of the interpolated curve:

len = 10 * (m - 1) + 1;

% Create the interpolated curve:

xx = linspace(x(1),x(m),len);
yy = 0*xx;

% Sum up the Chebyshev polynomials used to create the curve:

for i = 1:len,
  for j = 1:n,
    yy(i) = yy(i) + c(j) * cheby(xx(i),j-1);
  end
```

```
end

% Integrate the interpolated curve:

i = trapezoidalr(xx,yy,0);

% Sample the running integral where we need values:

integral = samplevec(x,xx,i);
```

E.7 Computation of Chebyshev Polynomials: cheby.m

```
function [t] = cheby(x,n)
%CHEBY
%
% Evaluate Chebyshev polynomials using recurrence relation.
%
%       t = cheby(x,n)
%
%       x: points at which to evaluate the nth Chebyshev polynomial, Tn(x)
%       n: order of Chebyshev polynomial

%-------------------------------------------------------------------------
%
%       Author: Mark S. Andersson.
%       Date:   May 28, 1992.
%       Copyright (c) 1992
%-------------------------------------------------------------------------

% Ensure that x is a row vector:

x = x(:).';

% Find number of points at which to evaluate Tn(x)

m = length(x);

% Allocate space for the output vector of Chebyshev polynomials:

t = ones(1,m);

% Handle trivial cases:

if n == 0,
  return;
end

if n == 1,
  t = x;
end
```

```
% Use Chebyshev polynomial recurrence formula to compute values of Tn(x)
% for n > 1:

t0 = ones(1,m);
t1 = x;

for i=2:n,
  t = 2.*x.*t1 - t0;
  t0 = t1;
  t1 = t;
end
```

E.8 Global Dynamic Shape Function Fitting: gdsf.m

```
function wtip = gdsf(x,weightings,curvature,shapes);
%GDSF
%
%       Fits a number of global curvature shape functions to the outputs
%       of an array of sensors, integrates them, and returns an
%       estimate of the deflection.
%
%       wtip = gdsf(gagex,gages,k,wt)
%
%       wtip:       Tip deflection
%       x:          Horizontal abscissae for sensor weightings
%       weightings: Sensor weightings as a function of x (there is one
%                   row for each sensor)
%       curvature:  Curvature of the beam
%       shapes:     Shape functions to fit sensor outputs to (there is
%                   one row for each shape functions)
%

%-------------------------------------------------------------------------
%
%       Author: Mark S. Andersson.
%       Date:   November 11, 1992.
%       Copyright (c) 1992
%-------------------------------------------------------------------------


% Find n, the number of sensors:

[n,p] = size(weightings);

% Find the outputs of the sensors by performing the n integrals:
%
%            / L
%           |
% outputs = | weighting(x) curvature(x) dx
%           |
%          / 0
```

```
%
% for each sensor in the array.

for i = 1:n,
  outputs(i) = trapezoidal(linspace(0,1,p),weightings(i,:) .* curvature);
end

% Find the number of shape functions to fit to the sensor output data:

[m,p] = size(shapes);

% Perform shape function fitting by constructing the phi matrix for the
% matrix equation:
%
% [ phi ] [ coefficients ] = [ outputs ]

phi = zeros(n,m);

for i = 1:n,
  for j = 1:m,
    phi(i,j) = trapezoidal(x,weightings(i,:) .* shapes(j,:));
  end
end

% Solve for the coefficients in the matrix equation.  If the number of
% sensors n is the same as the number of shape functions m, the column
% vector [coefficients] is determined by inversion of [phi].   If the
% number of sensors n is less than the number of shape functions m,
% the vector [coeffivients] is computed in a least squares sense.

coefficients = phi \ output;

% Now add up the tip deflection of each of the shape functions given,
% multiplied by the coefficient:

wtip = 0;

for i = 1:n,
  wtip = wtip + coefficients(i) * trapezoidal(x,shapes(i,:));
end
```