Spatial Heterodyne Spectroscopy: Modeling and Interferogram Processing

by

Cara P. Perkins

B.S. Merrimack College, 2011

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Chester F. Carlson Center for Imaging Science
of the College of Science
Rochester Institute of Technology

July 29, 2013

Signature of the Author ________________________________

Accepted by ________________________________
Coordinator, M.S. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Cara P. Perkins
has been examined and approved by the
thesis committee as satisfactory for the
thesis required for the
M.S. degree in Imaging Science

____
Dr. John Kerekes, Thesis Advisor

____
Dr. Michael Gartley

____
Dr. Carl Salvaggio

____
Date

Spatial Heterodyne Spectroscopy: Modeling and Interferogram Processing

by
Cara P. Perkins

Submitted to the
Chester F. Carlson Center for Imaging Science
College of Science
in partial fulfillment of the requirements
for the Master of Science Degree
at the Rochester Institute of Technology

Abstract

This work presents a radiometric model of a spatial heterodyne spectrometer (SHS) and a corresponding interferogram-processing algorithm for the calculation of calibrated spectral radiance measurements. The SHS relies on Fourier Transform Spectroscopy (FTS) principles, and shares design similarities with the Michelson Interferometer. The advantages of the SHS design, including the lack of moving parts, high throughput, and instantaneous spectral measurements, make it suitable as a field-deployable instrument. Operating in the long-wave infrared (LWIR), the imaging SHS design example included provides the capability of performing chemical detection based on reflectance and emissivity properties of surfaces of organic compounds.

This LWIR SHS model takes into account the instruments entrance optics, interferometer, exit optics, and detection scheme to output realistic, interferometric data. The model serves as a tool to find the optimal SHS design parameters for the desired performance requirements and system application. It also assists in the data analysis and system characterization.

The interferogram-processing algorithm performs flat-fielding and phase corrections as well as apodization before recovering the measured spectral radiance from the recorded interferogram via the Inverse Fourier Transform (IFT). The model and processing algorithm are tested with a variety of spectra and the results are comparable to those in the literature with a noise-equivalent change in temperature of 0.2K. These results demonstrate the model's validity and the algorithm's performance. Additional experiments show the algorithm's real-time processing capability, indicating the LWIR SHS system presented is feasible.

Acknowledgements

Firstly, I would like to thank my thesis advisor, Dr. John Kerekes, for all his assistance and support and, of course, research funding. My career at the Center for Imaging Science (CIS) followed an unusual path and if it had not been for Dr. Kerekes, I would probably still be trying to figure out what my thesis topic is. I would also like to thank my thesis committee members, Dr. Michael Gartley for his advice on qualitative measurement comparisons and Dr. Carl Salvaggio for the use of his course lectures and his SHS model suggestions.

Thank you to Dr. María Helguera, who persuaded me to pursue my graduate career in imaging science and welcomed me as a part of her laboratory. Though our circumstances did not afford for us to work together, I am very appreciative for all her advice and encouragement along the way.

Finally, I would like to say thank you to all my friends and family, who have never once doubted me, and especially to Kyle for always listening to me babble.

Portions of the project or effort depicted herein were/are sponsored by the U.S. Army under Subcontract No. SC60721-1811-001 and W911SR-13-C-0005. The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement is or should be inferred.

I dedicate this work to my family -
for all the years of putting up with me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Fourier Transform Spectrometers (FTS) have long been used to measure a sample's spectrum [3, 19, 20, 21, 22]. The main advantage of the FTS over typical dispersive spectrometers, which use either diffraction gratings or prisms to separate incoming wavenumbers, is that it does not require the use of an entrance slit. Rather, it uses a wide, circular aperture. The term, optical throughput, is used to describe the amount of space through which light can enter an optical system. The large aperture of the FTS, as compared to the thin entrance slit of the dispersive spectrometer, provides a much higher optical throughput, allowing much more radiant energy to enter and, thus, increasing the signal-to-noise ratio.

A specific FTS, the Spatial Heterodyne Spectrometer (SHS) is known for having a very high throughput and is, therefore, often used to study emission lines [23, 24]. An SHS is very similar to a Michelson Interferometer (MI), a very well-known FTS [3]. A MI uses a beam-splitter to divide the incoming light into two interferometer arms, each containing a mirror to reflect the light to be recombined. After the separated wavefronts recombine at the detector, we have interference as a function of the difference in the optical path length of the two, separated beams. One mirrored-arm moves in increments to measure the interference at each optical path-length difference (OPD) within the system range. An SHS also uses a beam-splitter, but both its interferometer arms are fixed and contain diffraction gratings in place of mirrors, causing wavenumber-dependent tilts. The SHS design collects interference measurements at each OPD simultaneously and without any moving parts, making it more compact and robust [25].

Most current SHS designs operate in the visible and ultraviolet regions of the electromagnetic spectrum and have astronomical and atmospheric applications [18, 26, 27]. Of particular interest, however, is a long-wave infrared (LWIR) SHS. Because organic compounds have distinct spectral signatures in the thermal infrared (IR) region of the electromagnetic spectrum, the spectra collected by a LWIR SHS can be used to perform material detection processes. Efforts have been made to develop a long-wave infrared (LWIR) SHS [4, 14], but the particular area of research is, generally, understudied. The

Naval Research Laboratory has designed the first and, currently, the only, successful LWIR SHS [14].

In support of LWIR SHS research and general SHS instrument design processes, this thesis will develop an SHS model. The model generates realistic SHS interferograms. The interferogram, the Fourier transform of the input spectrum, is then converted back to calibrated spectral radiance by an interferogram processing algorithm, also developed in this thesis. The main processing tasks include apodization of the interferograms, phase-correction, and flat-fielding correction in addition to the inverse Fourier transform. The model can be used to choose design parameters for an instrument, given performance criteria, as well as assess the SHS design results and overall system performance. The algorithm should also be capable of real-time processing. The model and processing algorithm may be used as generic tools for a variety of SHS designs and applications, but the results in this thesis assume a LWIR SHS design, chosen due to the lack of spatial heterodyne spectroscopy research performed in the LWIR region.

Chapter 2

Project Overview

2.1 Research Goals

Because LWIR SHS theory is rather understudied, this thesis will provide two tools to assist with design optimization and characterization and data analysis. Possible instrument design parameters and assumptions will be used to model the LWIR SHS and generate raw interferograms. An interferogram processing algorithm will also be developed to convert the raw interferograms to calibrated radiance spectra. The model serves as a decision tool for SHS design parameters and system and performance analysis tool for any SHS design. The processing algorithm may also be applied to a wide variety of SHS systems.

2.2 Objectives

The objectives of this thesis are:

1. to create a model of a realistic SHS for generating raw interferograms,
2. to design an interferogram processing algorithm for converting SHS interferograms into calibrated radiance spectra with real-time capability,
3. to demonstrate design optimization and system characterization using the results of (1) and (2), and
4. to demonstrate the feasibility of measuring spectra with an SHS operating in the LWIR with results from (1) and (2).

2.3 Contributions to Knowledge

This research provides a general model of an SHS, which can assist with the optimization of design parameters based on system performance requirements and applications and the characterization of a designed SHS system. The data generated by the model is also used to test the interferogram processing algorithm, also contributed by this work. These tools can be used recursively to improve system design and data analysis results. Though the model and processing algorithm are intended to aid in any SHS development effort, this thesis emphasizes the use of the tools in LWIR SHS studies.

Chapter 3

Background

3.1 Spectroscopy

Spectral information has been used to determine material composition for many years. Some of the simplest tools for studying objects' spectra are dispersive spectrometers. A dispersive spectrometer measures the spectrum of a source by separating the incoming radiation into its different spectral components, using a prism (see Figure 3.1) or diffraction grating (see Figure 3.2) to bend the light as a function of wavelength. The sensor is made of one detector element that is moved to view the spectral components individually or an array of detector elements to view all the components at once [20].

A prism relies on the wavelength-dependent refractive index of a material and Snell's law:

$$n_1(\lambda)\sin\theta_1 = n_2(\lambda)\sin\theta_2, \tag{3.1}$$

where $n_1(\lambda)$ and $n_2(\lambda)$ are the refractive indices of the first and second material the light passes through, respectively, λ is the wavelength of light, and θ_1 and θ_2 are the incident and refracted angles of the light, respectively, with respect to the normal of the surface of the second material.

A diffraction grating derives the spectrum by:

$$\frac{1}{\lambda}[\sin(\theta_1) + \sin(\theta_2)] = m/d, m = 0, \pm 1, \pm 2, ... \tag{3.2}$$

where θ_1 and θ_2 are the incident and refracted angles of the light, respectively, m is the order of diffraction, and d is the distance between the grating grooves. Constructive interference for a given wavelength occurs at integer values of m, while destructive interference occurs at half-integer values of m.

The value of θ_2 depends on wavelength, such that the diffraction grating separates the input beam into its intensity contributions as a function of wavelength, as shown in

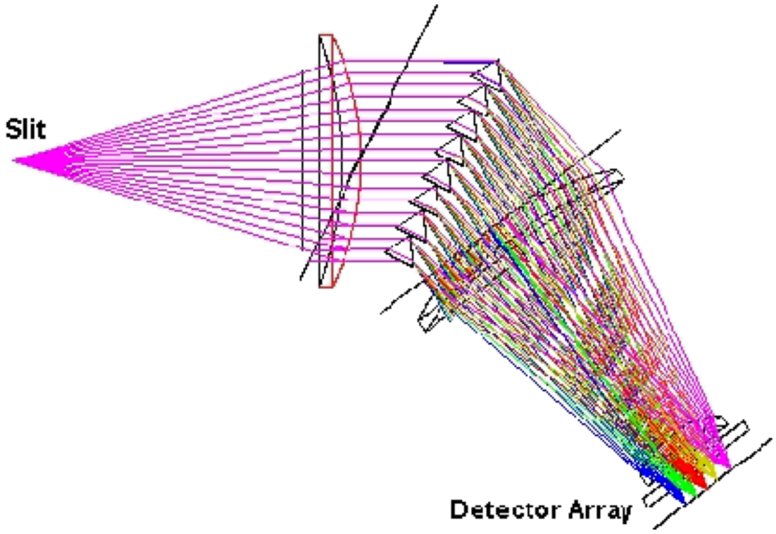


Figure 3.1: A dispersive spectrometer that uses prisms relies on Snell's law to disperse incoming light into its separate wavelength components [1].

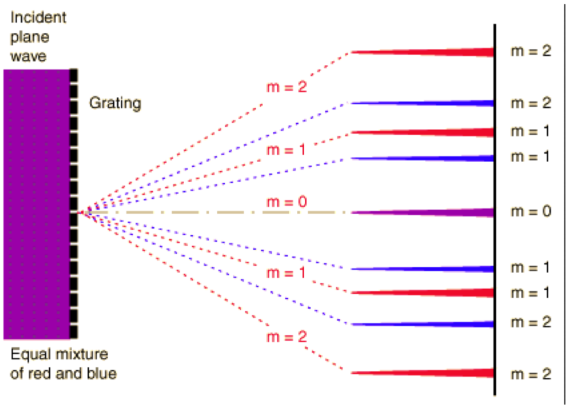


Figure 3.2: A diffraction grating disperses the incoming light beam into its separated wavelength constituents. Here, $\lambda_1 \neq \lambda_2 \neq \lambda_3$ [2].

Figure 3.2. Both the prism and diffraction grating-based dispersive spectrometers require thin entrance slits that are wide enough to let the observed wavelength pass through but narrow enough to maintain a reasonable spectral resolution.

3.2 Optical Throughput

Optical throughput, sometimes called etendue, describes the space through which light may enter an optical system. Throughput is defined by:

$$G_E = A_a \Omega_a, \tag{3.3}$$

where A_a is the area of the aperture and Ω_a is the field-of-view (FOV) subtended by the system aperture determined by the spectral resolving power, R:

$$\Omega_a = \frac{2\pi}{R}, \tag{3.4}$$

A larger system aperture or FOV will provide a greater optical throughput, allowing more light energy to enter the system. A high optical throughput is desirable because it increases the system sensitivity, therefore increasing the signal-to-noise ratio (SNR). In many optical systems, there is a trade-off between the optical throughput and the system resolution. For example, the dispersive spectrometer requires a rather narrow entrance slit to achieve a usable spectral resolution, but this creates a low optical throughput, decreasing system sensitivity and overall SNR.

3.3 Interferometry

Interferometry is a broad term used to describe any technique in which electromagnetic waves undergo interference in order to extract information about a radiant energy source. The tool used to cause the light interference is the interferometer.

3.3.1 Fourier Transform Spectroscopy

A very large family of interferometers exist. Of concern in this research is the subset of interferometers called Fourier Transform Spectrometers (FTS). As is explained in the following sections, a FTS measures a horizontal row of the interference pattern it creates from the input light beam. This measurement is the real part of the Fourier Transform (FT) of the source spectrum.

Both the dispersive spectrometer and the FTS can be used to measure a source spectrum. The FTS has the following advantages over the dispersive spectrometer:

- Fellgett Advantage: The FTS obtains a multiplex measurement rather than a direct measurement of spectra, resulting in an increased signal-to-noise ratio (SNR) [21, 3]

- Jacquinot Advantage: The FTS has notably high optical throughput due to its circular apertures (as opposed to the linear slits used in normal diffraction grating systems) [28], which also increases the SNR

- Connes Advantage: The FTS has a built-in HeNe laser for self-calibration [21]

3.3.2 Michelson Interferometer

The Michelson interferometer (MI) belongs to the FTS family. It consists of two interferometer arms and a beam-splitter to create two, separate wavefronts to travel down each of its arms (see Figure 3.3). Each beam travels along its respective path, reflects off the mirror at the end of the interferometer arm, and recombines with the other reflected beam at the sensor, creating part of the interference pattern. One mirror is moved to change the path length of one of the beams with respect to the other. The difference between the total lengths traveled by each beam is known as the optical path difference (OPD):

$$OPD = l_2 - l_1, \tag{3.5}$$

where l_1 and l_2 are the lengths of the travel paths in the first and second MI arms, respectively. The OPD between the recombining beams determines the phase difference between the two. The phase difference then determines the coherence of the interfering beams. For a monochromatic source, the phase difference between the two reflected beams is given by:

$$\Delta\phi = \frac{2\pi OPD}{\lambda}. \tag{3.6}$$

If

$$\Delta\phi = \pm 2(m+1)\pi,\ where\, m = 0, \pm 1, \pm 2, etc., \tag{3.7}$$

then the recombining beams are completely coherent and there is constructive interference. If

$$\Delta\phi = 2m\pi, m = 0, 1, 2, etc., \tag{3.8}$$

then the recombining beams are completely decoherent and there is destructive interference [3].

3.3.3 Interferogram Basics

The output signal of an interferometer, called an interferogram, is a single row of the interference pattern produced by the recombining wave-fronts at the detector. The interferogram is a measurement of intensity as a function of OPD, consisting of information from all the contributing frequencies in the input signal [21].

If we consider the interference of two wave-fronts with wave-vectors, $\vec{k_1}$ and $\vec{k_2}$, we can calculate their combined intensity by:

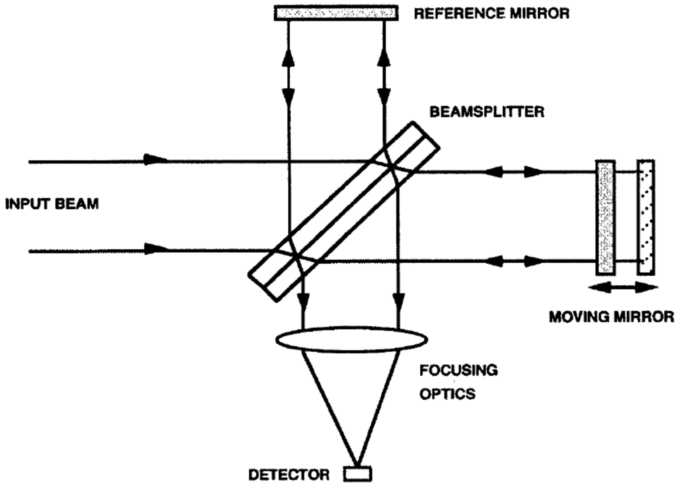


Figure 3.3: The MI separates incoming light into two beams using a beam-splitter. Each beam travels the length of one of the interferometer arms, reflects off the mirror, and recombines with the other at the detector [3].

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2}\cos\left[\vec{r}\cdot(\vec{k_1} - \vec{k_2}) + (\phi_1 - \phi_2)\right], \tag{3.9}$$

where I_1 and I_2 are the intensity magnitudes and ϕ_1 and ϕ_2 are the phases of the wavefronts corresponding to $\vec{k_1}$ and $\vec{k_2}$, respectively. If $I_1 = I_2$ (as is true in the interferometer since the two interfering beams are from the same source) and $\phi_1 = \phi_2 = 0$, then

$$I = 2I_1(1 + \cos\left[\vec{r}\cdot(\vec{k_1} - \vec{k_2})\right][8]. \tag{3.10}$$

Specifically, the cosine argument is the phase difference between the two waves recombining at the detector. Using Equation 3.6:

$$I(OPD) = \frac{1}{2}L(k)[1 + \cos(2\pi k \cdot OPD)], \tag{3.11}$$

where k is the wavenumber:

$$k = \frac{1}{\lambda} \tag{3.12}$$

and $L(k)$ is the spectral radiance of the source at a particular wavenumber. The 1/2 term is due to the fact that only half the energy is measured at the detector since the beams from each arm must travel through the beam-splitter a second time - half the light travels back through the input, while the other half travels toward the detector. Equation 3.11 indicates that part of the output signal is modulated by a function of the OPD, while the

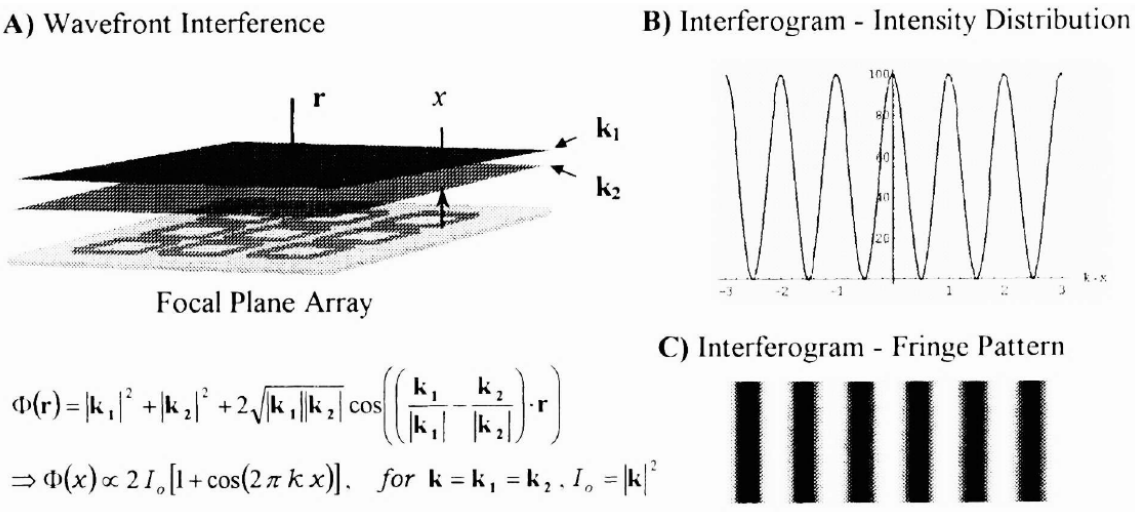

A) Wavefront Interference
r
x
$\mathbf{k}_1$
$\mathbf{k}_2$
Focal Plane Array
$\Phi(\mathbf{r}) = |\mathbf{k}_1|^2 + |\mathbf{k}_2|^2 + 2\sqrt{|\mathbf{k}_1||\mathbf{k}_2|}\cos\left(\left(\frac{\mathbf{k}_1}{|\mathbf{k}_1|} - \frac{\mathbf{k}_2}{|\mathbf{k}_2|}\right)\cdot\mathbf{r}\right)$
$\Rightarrow \Phi(x) \propto 2I_o[1+\cos(2\pi k x)]$, for $\mathbf{k} = \mathbf{k}_1 = \mathbf{k}_2$, $I_o = |\mathbf{k}|^2$
B) Interferogram - Intensity Distribution
100
80
60
40
20
-3
-2
-1
1
2
3
k·x
C) Interferogram - Fringe Pattern

Figure 3.4: The two monochromatic wave-fronts from the two interferometer arms recombine at the focal plane array (A.) and create the interference fringe pattern (B.). One row of the fringe pattern gives us the interferogram (C.) [4].

other part is not. The modulated portion of the signal is what provides the interferogram, by definition:

$$I(OPD) = 1/2 \cdot L(k) \cos(2\pi k \cdot OPD) \tag{3.13}$$

The above equation is for the ideal interferogram of a monochromatic source. A more realistic equation takes into account the reflection and transmission coefficients and self-emission of the internal optics (i.e. the beam-splitter and mirrors). We will describe this in more detail in Section 6.1. For a polychromatic source, the interferogram equation becomes:

$$I(OPD) = \int_{-\infty}^{\infty} 1/2 \cdot L(k) \cos(2\pi k \cdot OPD) dk. \tag{3.14}$$

The cosine term in the interferogram equation indicates that the interferogram is, essentially, a cosine transform (or the real part of the Fourier transform) of $L(k)$. To obtain the spectral radiance of the source, $L(k)$, we take the inverse Fourier Transform (IFT) of $I(OPD)$:

$$L(k) = \int_{-\infty}^{\infty} 1/2 \cdot I(OPD) e^{2\pi i k \cdot OPD} dOPD. \tag{3.15}$$

3.3.4 Sagnac Interferometer

Similar to the Michelson interferometer, the Sagnac interferometer divides the incoming light into two beams with a beam-splitter. Consider one common Sagnac interferometer geometry in Figure 3.5. One beam travels to the first, second, and third mirrors before the detector. The other beam travels to the third, second, and first mirrors before the detector. As implied in Figure 3.5, introducing a beam-splitter tilt angle, $\Delta\theta$, will create a phase difference between the two beams at the detector because each beam travels a slightly different path. The OPD in this Sagnac interferometer then becomes:

$$OPD = 4x\Delta\theta, \tag{3.16}$$

where x is the position along the detector. Substituting Equation 3.16 into Equation 3.11 gives the interferogram equation for the Sagnac interferometer for a monochromatic source:

$$I(x) = L(k)[1 + \cos(2\pi k \cdot 4x\Delta\theta)]. \tag{3.17}$$

From Equation 3.16, we see that the OPD is a function of x, meaning that the Sagnac interferometer collects the interference pattern spatially rather than temporally as in the MI. As we will soon see, the SHS also collects an interferogram spatially, but with a higher spectral resolution.

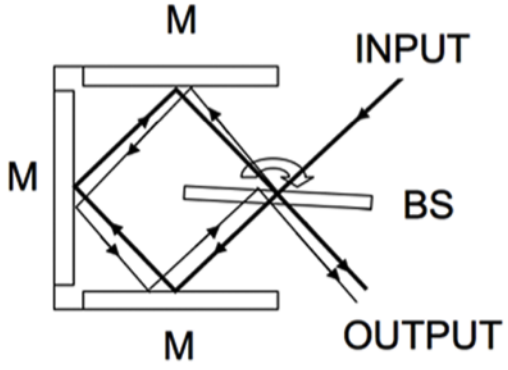


Figure 3.5: The Sagnac Interferometer usually consists of a beam splitter and three mirrors. The two, separated light beams travel to each mirror before reaching the detector, but in opposite directions. A beam-splitter tilt angle creates a phase difference between the two, recombining beams, creating the interference pattern [5].

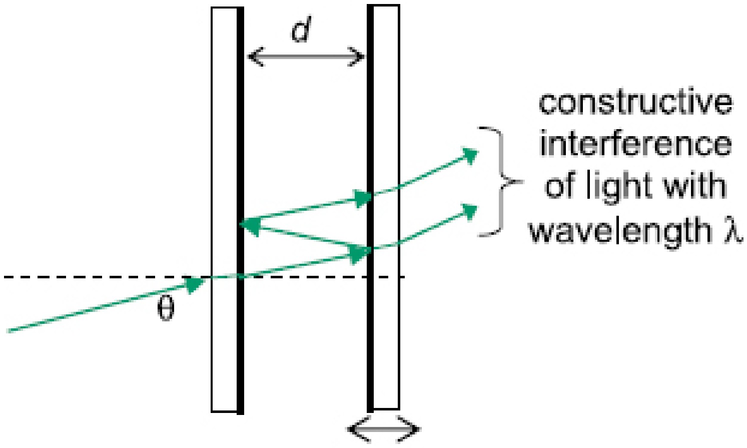


Figure 3.6: The Fabry-Perot Interferometer usually consists of two highly reflective plates that face each other. Incoming light is reflected and transmitted each time it is incident on a plate [6].

3.3.5 Fabry-Perot Interferometer

Unlike the Michelson and Sagnac, the Fabry-Perot interferometer (FPI) is not, ordinarily, a FTS. The Fabry-Perot interferometer, as shown in Figure 3.6, consists of two parallel plates with highly reflecting surfaces that face each other. When light enters the area between the two plates, some of it is transmitted and some of it is reflected at each air-plate interface. Each reflection exhibited by a beam experiences a phase shift of:

$$\Delta\phi = 4\pi k \cdot nd\cos\theta, \tag{3.18}$$

where n is the index of refraction of the medium between the two reflective plates, d is the distance between the plates, and $OPD = 2d\cos\theta$. The wavenumber-dependent phase difference between the transmitted beams causes an interference pattern, consisting of rings, known as Newton's rings [29]. Sometimes, one thick plate with two reflecting surfaces is used in place of a pair of parallel reflecting plates. This device is more often called a Fabry-Perot etalon.

The total transmittance function of a Fabry-Perot instrument is:

$$\begin{aligned}\tau_{FP} &= \frac{(1-r)^2}{1+r^2-2r\cos\Delta\phi} \\ &= \frac{1}{(1+f\sin^2(\Delta\phi/2))},\end{aligned} \tag{3.19}$$

where r is the reflectance of the parallel plates or surfaces of the etalon plate, and f is the Fabry-Perot coefficient of finesse:

$$f = \frac{4r}{(1-r)^2}. \tag{3.20}$$

The quantity of finesse describes the sharpness of the etalons transmission peaks (i.e. the resolution) and is defined by:

$$F = \frac{\pi}{2\sin^{-1}(1/\sqrt{f})} \approx \pi/2\sqrt{f}[30]. \tag{3.21}$$

The output of the FPI is then a series of narrow transmission bands:

$$I_t = I_0 \frac{1}{1 + \frac{4r}{(1-r)^2}\sin^2(2\pi k \cdot \frac{OPD}{2})}[29], \tag{3.22}$$

where I_0 is the input beam intensity. Varying the plate separation distance, d, allows for the collection of intensity as a function of OPD, as in the MI.

3.3.6 Fabry-Perot Interferometer as a Fourier Transform Spectrometer

As mentioned in the previous section, the output of an FPI with highly reflecting plates is a pattern of Newton's rings (i.e. an Airy function). Meanwhile, the output of an MI is a cosine function. Performing an IFT on the output of the MI gives a single peak at a particular wavenumber for a monochromatic source. The IFT for the output of the FPI for that same source is a series of peaks. In other words, because the output of the FPI is not purely sinusoidal, the IFT of its output gives us a contaminated spectrum - the spectrum of the source and its harmonics. This concept is demonstrated in Figure 3.7.

Previous studies [29, 31], however, have demonstrated that the FPI can operate as a FTS. If the reflectance of the plates in the FPI is low, Equation 3.22 becomes:

$$\begin{aligned} I_t = I_0 \frac{1}{1 + \frac{4r}{(1-r)^2}\sin^2(2\pi k \cdot OPD)} &\approx I_0(1 - \frac{4r}{(1-r)^2}\sin^2(2\pi k \cdot OPD)) \\ &= I_0(\frac{4r}{2(1-r)^2} - \frac{4r}{2(1-r)^2}\cos(4\pi k \cdot OPD)). \end{aligned} \tag{3.23}$$

Therefore, the modulated part of the output of a low reflectance FPI consists of a cosine term and we can perform an IFT to recover the sample spectrum [31] as in the MI and other FTS instruments. Note, this type of FPI still requires moving one of the reflective plates to collect the entire interferogram.

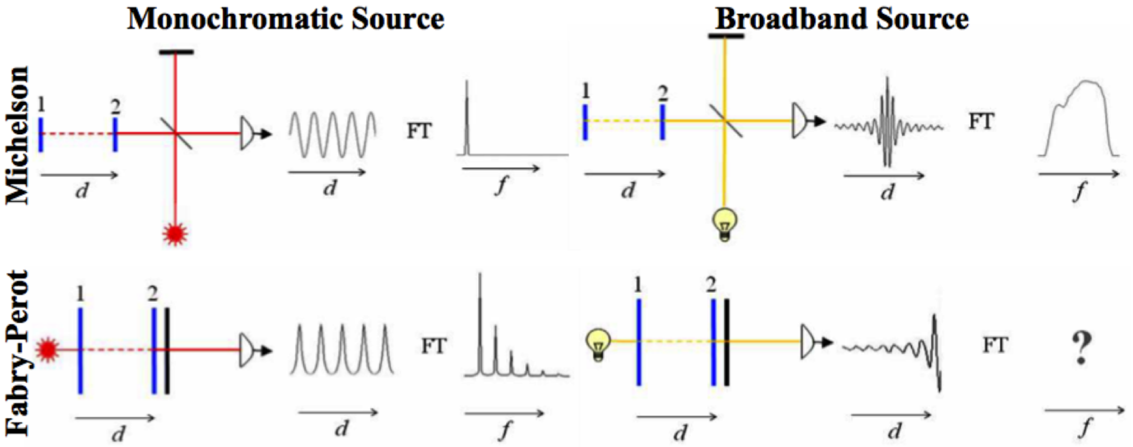


Figure 3.7: The Fabry-Perot interferometer produces a non-sinusoidal signal. Unlike that of the MI, the IFT of this signal gives the source spectrum contaminated with harmonics [7].

3.4 Spatial Heterodyne Spectroscopy

The spatial heterodyne spectrometer (SHS) is another type of interferometer that falls into the FTS category. The SHS is similar in design to the MI, but with diffraction gratings in place of mirrors in the arms (see Figure 3.8). Each diffraction grating creates a wave-front at a wavenumber-dependent spatial frequency, separated by a wavenumber-dependent angle for each wavenumber in the input beam. When the two wave-fronts recombine at the detector, they create interference fringes, known as Fizeau fringes, and represent the FT of the source's spectrum about the heterodyne wavenumber. The heterodyne wavenumber is that which produces parallel wave-fronts (i.e. the separation angle is zero) with zero spatial frequency, and is known as the Littrow wavenumber. The diffraction gratings allow us to create the entire interference pattern and record the interferogram without moving any interferometer parts [25].

Like dispersive spectrometers, the SHS uses diffraction gratings to separate the wave-fronts by their spatial frequencies. However, as noted above, FTS instruments have a much higher throughput - about 200 times larger than that of a conventional dispersive spectrometer [21] due to their circular apertures. In addition to the advantages of FTS listed in Section 3.3.1, the SHS has the following beneficial characteristics:

- Compact size
- Mechanical simplicity (with no moving parts)

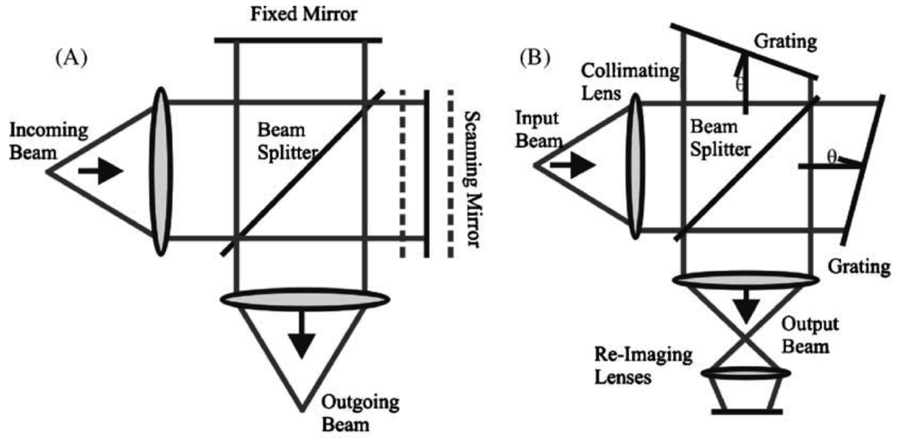


Figure 3.8: The SHS (B.) design is almost identical to that of the MI (A.). The main difference is that the mirrors in the MI are replaced by diffraction gratings tilted by the Littrow angle in the SHS [4].

- High resolution [32]

3.4.1 Heterodyne Mixing

As mentioned above, light energy with a wavenumber equal to that of the Littrow wavenumber associated with the diffraction gratings of the SHS is reflected in the direction normal to the detector. All other wavenumbers are dispersed at a wavenumber-dependent angle about the detector's normal vector. The Littrow wavenumber corresponds to zero spatial frequency such that the spectral frequency is in terms of the difference wavenumber, $k - k_L$, where k_L is the Littrow wavenumber.

A heterodyne frequency mixer with a characteristic local oscillator (LO) frequency accepts an input radio frequency (RF) and outputs a signal consisting of two frequencies. One of the frequencies is the sum of the RF and LO and the other is the difference of the RF and LO. In most applications, only one of these frequencies is needed and the other is filtered from the output.

Relating the concept of the heterodyne frequency mixer to the SHS, the Littrow wavenumber, k_L behaves like the LO and the wavenumber variable, k, behaves like the RF. The output signal has a frequency equal to their difference. Therefore, we say that the output of the SHS is heterodyned about the Littrow wavenumber and we call the instrument a spatial heterodyne spectrometer.

3.4.2 SHS Theory

The SHS separates the different wavenumber components of the input via the diffraction grating equation that we saw in Section 3.1:

$$k[\sin(\theta_L) + \sin(\theta_L - \gamma)] = m/d, \tag{3.24}$$

where k is the wavenumber of the incident light, θ_L is the Littrow angle (a property of the diffraction gratings), γ is the angle that the output wave-front makes with the normal to the detector, m is the order of diffraction, and $1/d$ is the density of the grating grooves [32]. If γ equals zero, the observed wavenumber is known as the Littrow wavenumber. Therefore, we can solve for the Littrow wavenumber by:

$$k_L = \frac{m/d}{2\sin(\theta_L)}[8]. \tag{3.25}$$

Alternatively, we can solve for the Littrow angle of the gratings if we know the Littrow wavenumber:

$$\theta_L = \sin^{-1}\frac{m}{2dk_L}. \tag{3.26}$$

We can also determine the necessary diffraction grating width for a desired number of interferogram samples, N:

$$W = \frac{N}{2(4(k_{max} - k_{min})\sin\theta_L}, \tag{3.27}$$

where $k_{max} - k_{min}$ is the spectral range of the SHS. Using this result, we determine the maximum position x on the detector:

$$x_{max} = \frac{1}{2}W\cos\theta_L. \tag{3.28}$$

Using the result of Equation 3.28, we can calculate the spectral resolution:

$$dk = \frac{1}{2(4(\tan\theta_L)x_{max})}. \tag{3.29}$$

Finally, using the result of Equation 3.29, we can calculate the interferogram sample spacing:

$$dx = \frac{1}{4(\tan\theta_L)Ndk}. \tag{3.30}$$

From Section 3.1, we know that a diffraction grating disperses the incident light into its separate wavenumber components. There will be overlap of the different wavenumber-components, such that the final output contains the interference of all the separated wavenumber constituents of the input, as in Figure 3.2. After the light leaves the gratings and is recombined at the exit optics, we have two wave-fronts for each wavenumber constituent, which interfere with one another to produce the Fizeau fringe patterns onto the detector. The output is then the interferogram of the input heterodyned about the Littrow wavenumber of the diffraction gratings [32]. The wavenumber-dependent Fizeau fringes have spatial frequencies:

$$\nu_F = 4(k - k_L)\tan(\theta_L) \tag{3.31}$$

with a resolving power of:

$$R_{SHS} = 4Wk\sin(\theta_L)[32]. \tag{3.32}$$

Equation 3.31 gives a number of fringes per centimeter on the detector. The Nyquist theorem states that in order to recover a non-aliased signal, we must sample at at least two times the maximum spatial frequency [33]. For example, assuming a detector width of one centimeter, we would need at least twice as many pixels as the number of fringes determined by Equation 3.31.

To determine the interference pattern created by the two recombining wavefronts in the SHS and better understand the Fizeau fringe frequency, we consider the two wave-fronts of wave-vectors, $\vec{k_1}$ and $\vec{k_2}$, from Section 3.3.3. A general wave-vector, $\vec{k_i}$ can be described in terms of its x, y, and z components:

$$\begin{aligned} k_{xi} &= 2\pi k\cos\phi_i\sin\beta_i \\ k_{yi} &= 2\pi k\sin\phi_i \\ k_{xi} &= 2\pi k\cos\phi_i\cos\beta_i, \end{aligned} \tag{3.33}$$

where ϕ_i and β_i are the angles between the outgoing wave-front and dispersion plane (the y-axis) and the outgoing wave-front and z-axis, respectively, as in Figure 3.9. We will call the incoming wave-front angles ϕ and β. Using these variables, the diffraction grating equation (Equation 3.2) becomes:

$$\sin(\theta_L \pm \beta_i) = \frac{2k_L\sin\theta_L}{k\cos\phi} - \sin(\theta_L \pm \beta). \tag{3.34}$$

Assuming the incoming wave-front is on axis (i.e. parallel to the interferometer arm), $\beta = \phi = 0$, which gives:

$$\sin(\theta_L \pm \beta_i) = \frac{2(k_L - k)\sin\theta_L}{k}. \tag{3.35}$$

Expanding the sine term, we calculate:

$$\beta_i = \pm 2\tan\theta_L\frac{k - k_L}{k}. \tag{3.36}$$

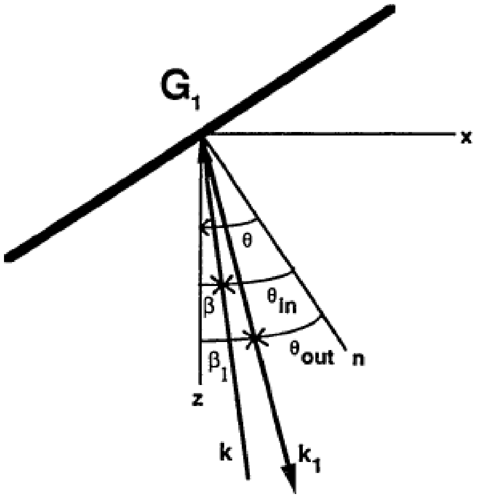


Figure 3.9: One diffraction grating, G1, in the SHS. The z-axis runs parallel to the interferometer arm containing G1. An incoming wave-front that is not traveling parallel to the interferometer arm is an angle β from the z-axis. The y-axis is not shown in the figure as it is coming out of the page. The x-axis runs perpendicular to the z and y-axes. Here, θ is the Littrow angle [8].

Because we want the interference pattern as a function of x, we compute the x-components of the two wave-vectors from their respective gratings:

$$\begin{aligned} k_{x1} &= 2\pi k \sin(\pm 2\tan\theta_L \frac{k-k_L}{k}) \\ k_{x2} &= 2\pi k \sin(\pm 2\tan\theta_L \frac{k-k_L}{k}), \end{aligned} \tag{3.37}$$

where we have assumed that $\phi_i = \phi$. These components add together when they recombine at the detector:

$$k_{x1} + k_{x2} = 2\pi \cdot 4(k-k_L)\tan\theta_L, \tag{3.38}$$

where we see the Fizeau spatial frequency term.

In typical Fourier transform IR (FTIR) spectroscopy, the interferogram is a function of OPD and is collected over time by moving a mirror, as in the MI. In the SHS, the interferogram is also a function of OPD, but it is collected over space in the x-dimension along the detector. The SHS interferogram may be recorded simultaneously because the dispersion caused by the diffracting gratings simulates the effect of varying the OPD in an interferometer. We can relate the value of x with the value of the OPD by:

$$OPD = 4x\tan(\theta_L). \tag{3.39}$$

The spectral variable in the SHS is also different from typical FTIR instruments. Instead of our interferogram being a function of wavenumber, k, it is a function of the difference wavenumber, $k - k_L$. The phase difference between the recombining beams in the SHS is then:

$$\Delta\phi = 2\pi \cdot 4(k - k_L)\tan\theta_L x \tag{3.40}$$

and the interferogram as a function of position, x, along the detector for a monochromatic source is given by:

$$I(x) = \frac{1}{2}L(k)[1 + \cos(2\pi \cdot 4(k - k_L)\tan\theta_L x). \tag{3.41}$$

For a poly-chromatic source, the interferogram becomes:

$$I(x) = \int_{-\infty}^{\infty} \frac{1}{2}L(k)[1 + \cos(2\pi \cdot 4(k - k_L)\tan\theta_L x)]dk, \tag{3.42}$$

where $L(k)$ is the spectral radiance measured at the input and $L(-k)$ is the same function as $L(k)$ but flipped [8]. As in the other FTS instruments, the IFT of the interferogram gives the spectrum of the input.

3.4.3 SHS Limitations

The high resolution and throughput, lack of moving parts, and compact size of the SHS make it a very robust and versatile instrument. Yet, the SHS does have its limitations.

The resolving power of the SHS depends on the width of the diffraction grating. This, in turn, limits the observable bandpass. A larger grating is needed to view a wider bandpass region. Also, at low resolving powers, the accepted angle of light hitting the detector may be greater than the angle between the orders of diffraction, allowing undesired light, or noise, to reach the detector [25].

Because each pixel in the SHS detector sees a different interferometer path, the arms and detector must be aligned very carefully. The system must also be flat-fielded so that the gain and dark current of each pixel is taken into account during the measurement process. These restrictions also increase the computation time for the SHS system. The necessary flat-fielding and phase corrections must be done thoroughly. Signal apodization is also needed to avoid truncation errors in the spectra (see Section 5.2.4), which also reduces the resolving power [25].

The main detector figure of merit used to describe an SHS system is the noise-equivalent spectral radiance (NESR). The NESR is the average noise in a single measurement expressed as a radiance value. It defines the just-noticeable difference (JND) in radiance. The ideal

NESR is achieved when the only noise source in the system is shot noise, i.e. the noise associated with the arrival of photons at a detector. This ideal NESR is known as the shot-noise-limited NESR [34].

3.4.4 Imaging SHS

Thus far, our discussion of the SHS has only been concerned with the interferometric applications of the SHS. However, the SHS can be used to collect hyperspectral imagery using additional optics [9]. An anamorphic telescope at the entrance aperture of the SHS images one dimension of the spatial scene onto the diffraction gratings while defocusing the other dimension. Depending on how your system is oriented, the anamorphic telescope looks at either a row or column of pixels at a time as in Figure 3.10. Each pixel in the row or column can be considered a point source in the scene. Each point source creates a line image across the diffraction gratings, perpendicular to the grating grooves. This results in an individual interferogram corresponding to each imaged pixel across the detector. Thus, the spatial information is recorded along one dimension of the detector array, while the spectral information is recorded in the other as in Figure 3.10. By scanning the scene by either columns or rows or pixels, we can collect an entire hyperspectral data cube.

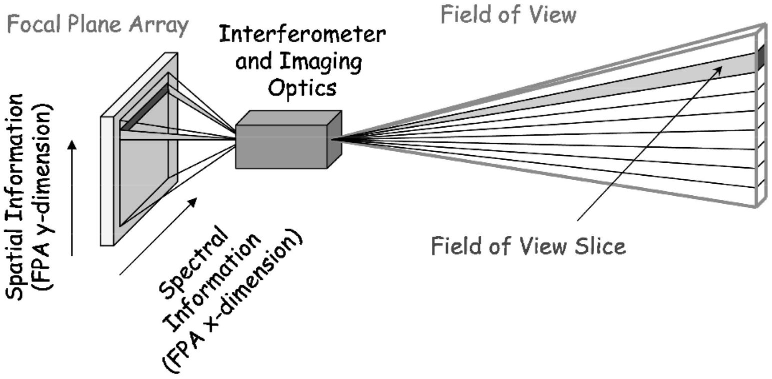

Focal Plane Array
Interferometer and Imaging Optics
Field of View
Spatial Information (FPA y-dimension)
Spectral Information (FPA x-dimension)
Field of View Slice

Figure 3.10: By means of an anamorphic telescope, the SHS can collect rows or columns of pixels at a time, recording the spatial information in one dimension of the detector and spectral information in the other dimension. After scanning for a desired number of rows or columns, the result is a complete hyperspectral data cube [9].

Chapter 4

Previous LWIR Interferometry Technologies

Spatial heterodyne spectroscopy is still a relatively new field and, therefore, other interferometers have been used to measure and exploit spectra in the past. As mentioned earlier, several SHS instruments have been created for the UV-VIS regions for astronomical purposes. However, only one SHS instrument has been successfully designed and built to operate in the LWIR [14]. This is mainly due to the high complexity and expense of LWIR-transmitting optical materials. Below, we describe some specific instruments that are similar in design, theory, or application to the instrument design used in the examples in this thesis. All the instruments discussed in this section operate in the LWIR because the results in this thesis are for a LWIR SHS design example. For a summary of the different technologies, see Appendix A.

4.1 FIRST (Field-portable Imaging Radiometric Spectrometer Technology)

The Field-portable Imaging Radiometric Spectrometer (FIRST) is developed for standoff chemical detection. Developed by Telops, FIRST operates over 8 to 11.5 microns. The spectral resolution, determined by user-specified parameters, ranges from 0.25 to 150 cm^{-1}. FIRST consists of a Michelson interferometer (Figure 4.1) and is capable of displaying either a sample interferogram or the calibrated spectral signature. Recall that the Michelson interferometer must collect the interferogram temporally while changing the OPD. FIRST collects an 128x128 image of interferograms or spectra in approximately 2 seconds at a spectral resolution of 4 cm^{-1}.

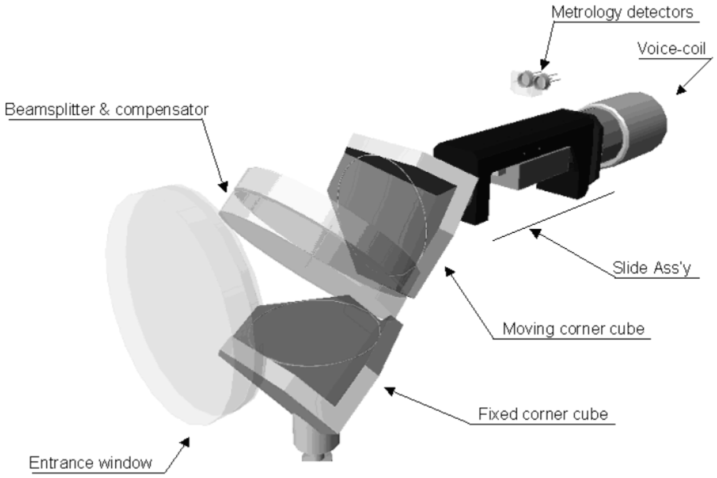


Figure 4.1: FIRST by Telops is an imaging Michelson interferometer operating in the LWIR. Depicted here is the interferometer subsystem [10].

4.2 LIFTIRS (Livermore Imaging Fourier Transform Infrared Spectrometer)

The Livermore Imaging Fourier Transform Infrared Spectrometer (LIFTIRS) is an imaging Michelson interferometer that can be used in the 3 to 5 micron or the 8 to 12 micron region. It has a variety of applications, including the "identification and mapping of gaseous effluents, the remote measurement of surface temperatures and the remote classification of surface material types based on their emissivity in the infrared" [11]. Like FIRST, the spectral resolution is defined by the user and can be as high as 0.25 cm^{-1}. An 128x128 hyperspectral image (HSI) with a spectral resolution of 4 cm^{-1} takes about 5 seconds to collect (as in Figure 4.2), more than twice the collection time of FIRST [11].

4.3 THI (Thermal Hyperspectral Imager)

The Thermal Hyperspectral Imager (THI) developed by Wright et al. [12] uses a three-mirror-geometry Sagnac interferometer to collect hyperspectral imagery over 7.5 to 13.5 microns. It is contained in a small satellite and meant to collect information about the Earth and its atmosphere. The THI acquires an entire 128x128 hyperspectral image in about 2 seconds. The THI, however, only has a spectral resolution of 16 cm^{-1}. The THI is shown in Figure 4.3.

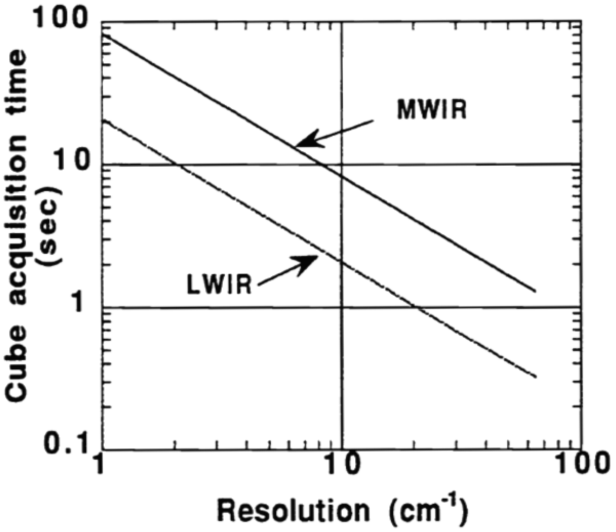


Figure 4.2: The LIFTIRS collection time is approximately twice that of the FIRST collection time while using the same image size and spectral resolution [11].

4.4 AIRIS (Adaptive Infrared Imaging Spectroradiometer)

The Adaptive Infrared Imaging Spectroradiometer (AIRIS) built by Physical Sciences, Inc. (PSI) is a LWIR, imaging, low-order, tunable Fabry-Perot etalon spectrometer (Figure 4.4). AIRIS provides enough resolution to distinguish spectral emissivities of chemical vapors in the air with a spectral coverage of 8 to 11 microns. The tuning system allows for the collection of an interference pattern at each observed wavelength with an NESR of about two times the shot-noise-limited NESR. Gittins et al. [13] [35] have shown that AIRIS can detect multi-species chemical vapor plumes as well as clouds of biological stimulants, using target detection algorithms. The AIRIS system is capable of detecting toxic industrial chemicals (TICs) and chemical warfare agents (CWAs) based on known spectral features with a detection accuracy as high as 85% [36]. AIRIS captures a 128x128 image at a particular wavenumber in about 1 minute. Again, the FPI requires moving one parallel plate to collect the transmitted intensity at a variety of wavenumbers [37].

4.5 IRISHS (Infrared Imaging Spatial Heterodyne Spectrometer)

The Infrared Imaging Spatial Heterodyne Spectrometer (IRISHS) [4] was meant for the identification of gases in the atmosphere when viewed from above the Earth. The IRISHS

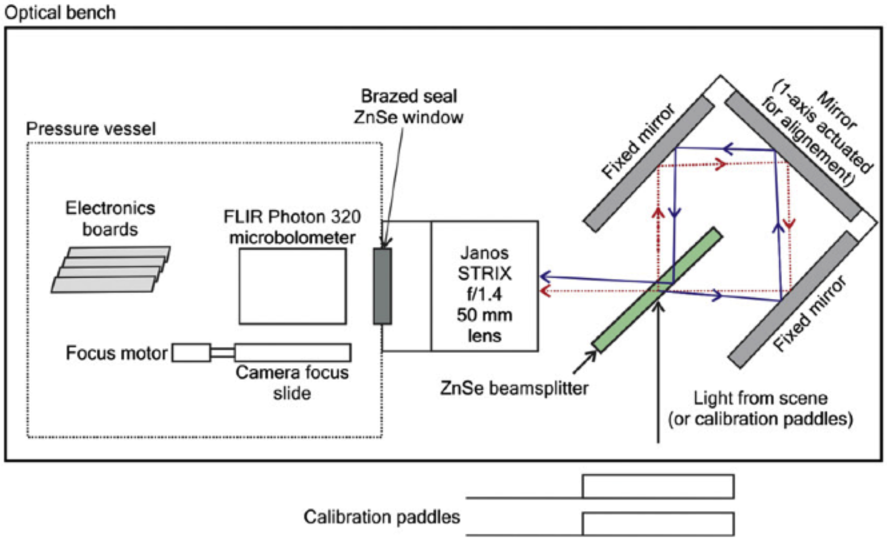


Figure 4.3: The THI uses a Sagnac interferometer to collect HSI at a spectral resolution of about 16 cm^{-1} [12].

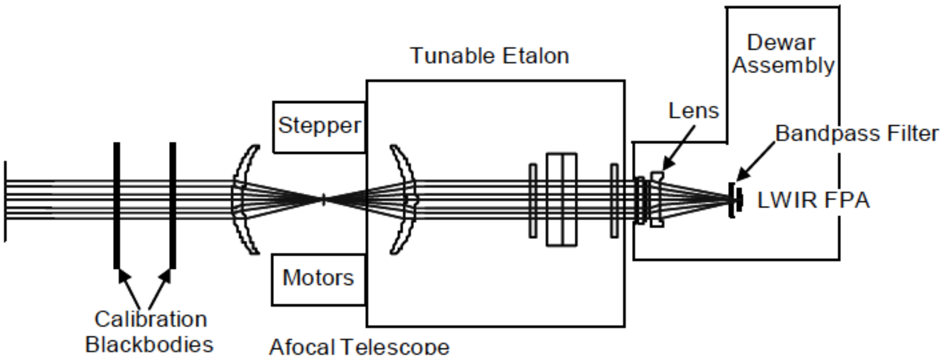


Figure 4.4: The AIRIS is a tunable Fabry-Perot interferometer with built-in hot and cold blackbodies for internal calibration [13].

combines an SHS and a Cassegrain telescope to image the scene with a Mercury Cadmium Telluride (HgCdTe) focal plane array (FPA) and an f/2 camera system (Figure 4.5). The IRISHS optical system is designed for the thermal LWIR - 8 to 12 microns, while the camera only operates from 8 to 9.5 microns. The IRISHS research and development effort ended before a use-able instrument was produced [4].

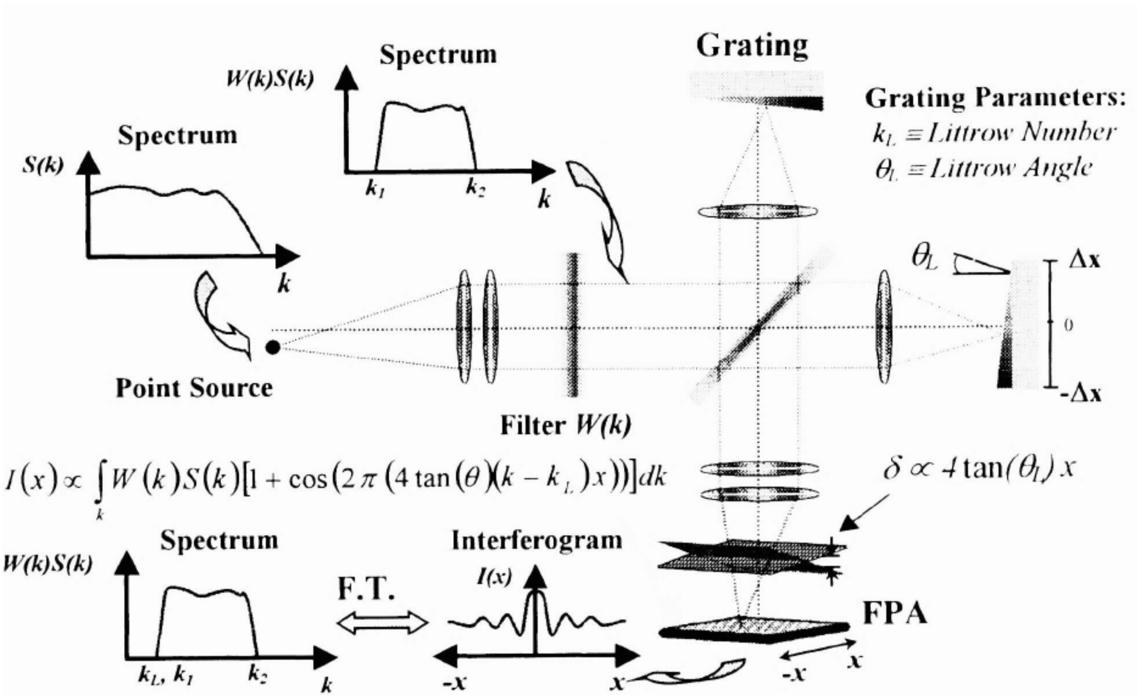


Figure 4.5: The IRISHS uses an IR SHS with a FPA to record an image of the sampled scene and an interferogram for each pixel in the image [4].

4.6 SHIMCAD (Spatial Heterodyne Imager for Chemical and Atmospheric Detection)

The US Naval Research Laboratory's Englert et al. [14] recorded the first broadband LWIR measurements from an SHS. Their instrument, the Spatial Heterodyne Imager for Chemical and Atmospheric Detection (SHIMCAD), uses an anamorphic telescope to image one dimension of the scene onto the interferometer aperture, an SHS to create the Fizeau fringes, and a FPA to collect the image and pixel interferograms. The telescope has a field of view (FOV) of 4 by 4.5 degrees, a 61 by 20 mm aperture size, and a wavenumber range of 1196 to 862 cm^{-1}, corresponding to a wavelength range of about 8.4 to 11.6 microns. The interferometer has a Littrow wavelength of 8.4 microns (or wavenumber of 1190 cm^{-1}), a passband range of 1190 to 870 cm^{-1}, a grating groove density of 30 mm^{-1}, a grating area of 9.6 by 7.2 mm, and a grating FOV of 28.07 by 9.08 degrees. The interferometer processing system uses a Hamming apodization function, resulting in a resolution of about 4 cm^{-1}. The SHIMCAD imaging system uses an f/2 lens and a 320 by 240 pixel HgCdTe focal plane array. The first laboratory tests used methanol and Kapton®foil samples. The recorded SHIMCAD spectra show similarities to that recorded with a traditional FTIR spectrometer (with a resolution of 1 cm^{-1}). The measurements provide evidence of the feasibility of a LWIR SHS for field measurements [14].

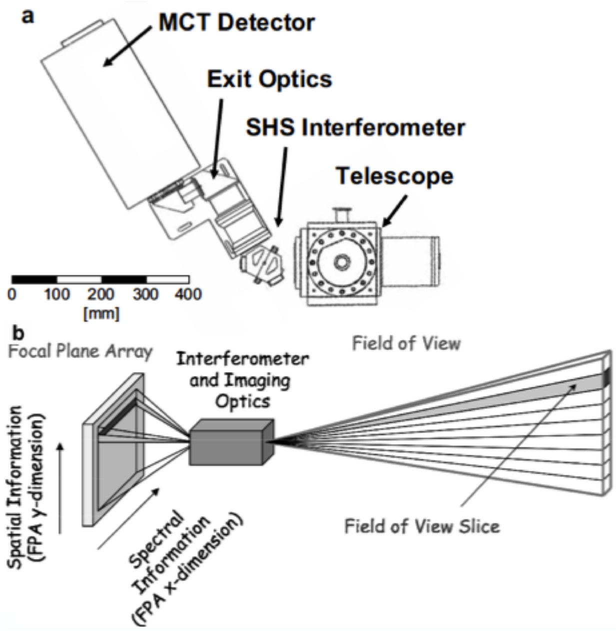


Figure 4.6: SHIMCAD is an imaging IR SHS like IRISHS, using an anamorphic telescope to image the scene onto the SHS aperture and a FPA to collect the scene's spatial and spectral information [14].

Chapter 5

Design Objectives

As stated above, this thesis aims to promote the study and advancement of the SHS technology. In support of this effort, an SHS model is developed. Although the results in this thesis are for a particular design example in the LWIR, the model employs general SHS concepts and can be used for a variety of applications and design requirements. Similarly, the interferogram processing algorithm may be applied to other SHS instruments. The model and algorithm are developed focusing purely on the interferometry aspects of the SHS, though they can be adapted to incorporate imaging aspects as well.

5.1 SHS Modeling: Errors to Consider

The SHS model developed in this thesis will rely on the basic theory in Section 3.4.2. The general model concepts used are outlined by Cooke et. al [4]. Other factors to consider in generating realistic interferometric data are:

- A wavelength-dependent optical transmission function.
- Phase errors.
- System vibrations.
- Noise.

5.1.1 Optical Transmission Function

The optical transmission function depends on the materials used in the instrument. The interferometer walls, beam-splitter, lenses, diffraction gratings, etc. respond to each wavelength differently. I.e. the beam-splitter might transmit one wavelength within the system bandpass more efficiently than another.

5.1.2 Phase Errors

There are two main components for the SHS system phase error. One part is wavenumber-independent while the other is wavenumber-dependent. The wavenumber-dependent phase error is usually rather complicated. Typically, this phase error is characterized after the instrument is built (or after the final design is known, using ray-tracing software) and accounted for in the interferogram processing algorithm or by methods outlined in [38] and [39].

The wavenumber-independent phase error is easy to determine from the data and, therefore, simple to correct. It refers to the interferogram's center-burst shift from the zero-location on the detector. The position, x, along the detector, ranges from $-x_{max}[cm]$ to $+x_{max}[cm]$. Because the SHS interferogram is heterodyned about the Littrow wavenumber, it should be symmetric and the location of its highest magnitude, i.e. the "center-burst", should occur at $x = 0cm$. Due to alignment errors in the instrument, this usually does not happen. A simple use of the FT shift theorem, however, can correct for this and place the center-burst at its correct location [4] as we will see in Section 5.2.2.

5.1.3 System Vibrations

System vibrations can be caused by thermal energy or movement of the instrument platform. These can be modeled as random events. Thermal and motion control capabilities can be built into the system to eliminate as much of these effects as possible. Thermal fluctuations mostly affect the spectral radiance calibration process. The SHS model has built-in hot and cold blackbodies for internal calibration, but if the blackbodies have fluctuating temperatures, this introduces an error in the calibration process. Motion within the instrument affects the interferogram sample spacing precision and the Littrow angle precision.

5.1.4 Noise

There are many different types of noise that can affect a system's output. We can calculate a standard deviation for each noise source and add it to our measured signal in a Gaussian fashion. The most significant noise source, in many cases, is shot noise, which refers to the statistical variations in the number of photons converted to photo-electrons at the detector. We assume that the number of electrons we measure is large enough such that the shot noise has a Gaussian distribution rather than a Poisson distribution. If our measured interferogram, which contains signal contributions from the system self-emission, is $I(x)$ with irradiance units of W/m^2, we can convert this to a number of electrons incident on the detector by:

$$S_{meas}(x) = \frac{I(x) A_d t_{int} \cdot QE}{(hck)}[40], \tag{5.1}$$

where t_{int} is the integration time, A_d is the unit detector area, QE is the detector quantum efficiency, h is Planck's constant, and c is the speed of light. Using our signal in electrons, we can calculate the Gaussian standard deviations associated with our noise sources in terms of electrons, starting with the shot noise:

$$\sigma_{shot} = \left[\sqrt{S_{meas}(x)}\right]_{avg}. \tag{5.2}$$

The next noise source we will consider is calibration noise. In Section 5.1.3 we said that temperature fluctuations in the internal blackbodies can cause calibration errors. We can capture these errors with an approximate calibration error rate, ϵ_c, giving a calibration noise standard deviation of:

$$\sigma_{calib} = \epsilon_{calib}\sigma_{shot}^2. \tag{5.3}$$

There is also a noise source associated with the detector and readout noise:

$$\sigma_{det} = \frac{2}{A_a \Omega dk D^* \tau_{tot}\sqrt{A_d T}} \cdot C_{avg}, \tag{5.4}$$

where A_a is the area of the system aperture, D^* is the specific detectivity, τ_{tot} is the total optical transmission function, which accounts for transmission functions, beam-splitter and grating efficiencies, etc., T is the frame rate, and C_{avg} is the constant used to convert from spectral radiance to number of electrons in the interferogram domain:

$$C(k) = \frac{A_d \Omega t_{int} \cdot QE}{(hc/k)\delta k} \cdot \sqrt{\frac{N}{2}}, \tag{5.5}$$

where N is the number of interferogram samples and the factor of $\sqrt{\frac{N}{2}}$ gives us the noise standard deviation in the interferogram domain rather than the spectral domain [41]. C_{avg} is the mean value of C. The electronics noise is caused by components following the detector, mainly the pre-amplifier, and has a standard deviation of:

$$\sigma_{electronics} = \frac{4x_{max} NPSD}{A_a \Omega R_D \tau_{tot}\sqrt{T}} \cdot C_{avg}, \tag{5.6}$$

where $NPSD$ is the noise-power spectral density, sometimes called the input to the pre-amplifier, and R_D is the detector responsivity. The next noise source to consider is that associated with the bit error rate, ϵ_B:

$$\sigma_{bit} = \sqrt{\frac{\epsilon_{bit}}{M} \sum_{q=0}^{q=M-1} \left(\frac{2^q L_{max}}{2^M - 1}\right)^2} \cdot C_{avg}, \tag{5.7}$$

where M is the number of bits used in the system and L_{max} is the maximum accepted spectral radiance (found by converting the detector electron saturation level to spectral radiance units). Finally, we consider the quantization noise. Quantization is the process of rounding measurements values to a certain precision. The level of precision depends on the number of bits used in the system. The quantization noise has a standard deviation of:

$$\sigma_{quant} = \frac{L_{max}}{\sqrt{12}(2^M - 1)} \cdot C_{avg}. \tag{5.8}$$

The total noise standard deviation is:

$$\sigma_{tot} = \sqrt{\sum_i \sigma_i^2}. \tag{5.9}$$

As mentioned in Section 3.4.3, an important figure of merit of our system is the overall noise-equivalent spectral radiance (NESR). We define the NESR as:

$$NESR(k) = \frac{\sigma_{tot}}{C(k)}. \tag{5.10}$$

We can then use the NESR to calculate our SNR and a noise-equivalent change in temperature (NEdT). Our SNR is:

$$SNR(k) = \frac{L_{recovered}(k)}{NESR(k)}, \tag{5.11}$$

where $L_{recovered}(k)$ is the recovered spectral radiance signal. Our NEdT is:

$$NEdT(k) = \frac{NESR(k)}{\frac{dL_{BB}}{dT}}, \tag{5.12}$$

where $\frac{dL_{BB}}{dT}$ is the derivative of the Planck equation for the spectral radiance of a blackbody with respect to temperature, T. In this work, we assume a temperature of 300K.

Another form of noise we must consider in the SHS model is the pixel-to-pixel sensitivity variation in our detector. Each detector element has an individual dark current and overall sensitivity to incoming photons. We can account for the differences in sensitivity across our detector once it is built [42].

5.2 Interferogram Processing Concepts

The errors listed above, among others, must be accounted for in the interferogram processing algorithm. Specifically, the algorithm must perform the following tasks:

- Flat-fielding correction
- Phase correction

- Bias removal
- Apodization
- Inverse Fourier Transform
- Noise reduction
- Spectral Radiance Calibration

5.2.1 Flat-fielding

Flat-fielding describes the process of correcting sensor data for variations in the pixel-to-pixel sensitivity. These variations are constant and need only to be characterized after the instrument is built. Generally, the flat-fielding correction process corrects for sensitivity measurements in the detector as well as alignment errors in the interferometer arms. There are a number of ways to perform flat-fielding correction. Often, the flat-fielding correction is done in conjunction with the calibration process. However, the sensitivity variation in the detector can significantly affect the SNR, decreasing the overall quality of the retrieved spectra. Therefore, it is often desirable to characterize these errors in the laboratory for a built-in flat-fielding correction process, separate of the system calibration [42].

Englert and Harlander [42] outline three flat-fielding processes for SHS. The first assumes that the interferometer arms are exactly aligned, the second assumes misaligned arms, and the third assumes misaligned arms and corrects for phase shifts. We will explain the latter two, since there is always some error in the arm-to-arm alignment.

For the misaligned-arm correction, the flat-fielding process requires two measurements of the same source. One measurement is made with one interferometer arm blocked and the second measurement is made with the opposite arm blocked. Using these two measurements, we can correct future measured interferograms. First, we calculate the biased, partially corrected interferogram:

$$I_{C'}(x) = \frac{I(x)}{I_a(x) + I_b(x)}, \tag{5.13}$$

where $I(x)$ is any measured interferogram, $I_a(x)$ is the interferogram of the known source with one interferometer arm blocked, $I_b(x)$ is the interferogram of the same source with the opposite arm blocked, and $I_{C'}(x)$ has a modulated and unmodulated term as in Equation 3.11. Next, we remove the unmodulated term. The flat-field corrected interferogram is then:

$$I_{ff}(x) = I_C(x) \cdot \frac{I_A(x) + I_B(x)}{2\sqrt{I_A(x) I_B(x)}}, \tag{5.14}$$

where $I_C(x)$ is the modulated term of $I_{C'}(x)$.

The misaligned-arm and phase shift correction technique requires a monochromatic source for characterization and the ability to change the length of one of the arms. Three interferograms are recorded using the same monochromatic source, but three different interferometer arm lengths. Each interferogram has a different phase. We can determine the three phases by:

$$\phi_i(k_0, x) = \arctan \frac{Im[I_{i,k_0}(x)]}{Re[I_{i,k_0}(x)]} - k_0 x, \tag{5.15}$$

where $i = 1, 2, 3$, corresponding to each of the three measurements, and k_0 is the wavenumber of the monochromatic light source. As expected, the interferograms have a modulated and unmodulated term. The modulated term is calculated by:

$$M(k_0, x) = \frac{I_{i,k_0}(x) - I_{j,k_0}(x)}{\cos \phi_i(k_0, x) - \cos \phi_j(k_0, x)}, \; i \neq j, \tag{5.16}$$

where i and j are chosen on a pixel-by-pixel basis, such that the denominator is never zero. The unmodulated term is:

$$N(k_0, x) = I_{1,k_0}(x) - M(k_0, x) \cos \phi_1(x). \tag{5.17}$$

To remove the unmodulated term, we divide our measured interferogram by $N(k_0, x)$ and then subtract the mean. To remove the phase errors, we then divide by $M(k_0, x)$. Now, we have a flat-fielded and phase-corrected interferogram. This phase-correction should correct both phase errors described in Section 5.1.2.

5.2.2 Phase Correction

From Section 5.1.2, we know that there are two different phase errors to be corrected in the processing algorithm. The wavenumber-dependent phase error can be characterized once the instrument is built [42, 39]. The wavenumber-independent phase error causes the shift of the interferogram center-burst from the zero path difference (ZPD) location. We can correct this by implementing the FT shift theorem.

If the center-burst is located a distance, $+x_c$ from the ZPD, then we can shift the interferogram to be centered at ZPD by:

$$I(x)_p = I(x) * \delta(x + x_c), \tag{5.18}$$

where $\delta(x + x_c)$ is the dirac delta function centered at x_c and $*$ denotes the convolution. By means of the Fourier transform modulation theorem, we can re-write this in terms of the FT pairs:

$$I_p(x) = IFT\{FT\{I(x)\} \cdot FT\{\delta(x + x_c)\}\}[33], \tag{5.19}$$

where the FT of $\delta(x + x_c)$ is $\exp(2\pi \cdot x_c \cdot x)$.

Because the recorded interferogram is discrete, we can only shift the function by an integer number of sample spaces using the convolution method. In the more common case, when our interferogram is shifted by a non-integer number of sample spaces, Ben-David et al. [43] derive an approximation of the location of x_c by:

$$x_c = dx \left[\frac{\sum_i i|I(x_i)|}{\sum_i |I(x_i)|} - i_0 \right], \tag{5.20}$$

where i_0 is the current sample index of the apparent ZPD in the recorded interferogram. An estimate of i_0 can be found by locating the sample of the maximum of the recorded interferogram. Using this value for x_c, we can shift the interferogram by the FT method.

Even after the interferogram is shifted such that the ZPD is located at the center, the interferogram is still usually not perfectly symmetric. As stated earlier, to retrieve the sample spectrum, we must take the IFT of the interferogram. Our recorded interferogram will never have imaginary parts because we cannot make imaginary measurements. If the recorded interferogram is also symmetric, the IFT will also be real and symmetric. If the interferogram, however, is anti-symmetric, the IFT will have both real and imaginary parts and be odd. A signal with real and imaginary parts will have an additional phase, which we must also correct. We will explain this phase correction in Section 5.2.5.

5.2.3 Bias Removal

From Equation 3.11, we know the measured interferogram is a sum of a modulated and unmodulated term. The modulated term is the interferogram, by definition, whereas the unmodulated term may be neglected. We determine the interferogram bias, the unmodulated term, by averaging the samples on either side of the interferogram centerburst. Once the bias is determined, we subtract it from the measured interferogram so that we are left with only the modulated term.

5.2.4 Apodization

At this point, we have been describing the interferogram equation for a polychromatic source as an infinite integral. In reality, we cannot achieve infinite values for the x_{max}. The x_{max} is actually limited by the detector width and can be described by a rectangular function:

$$RECT(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq x_{max} \\ 0 & \text{otherwise} \end{cases} \tag{5.21}$$

where x_{max} is the maximum achievable position, x, along the detector within a given system. Figure 5.1 shows a plot of the one-dimensional RECT function that describes the detector size limitation. Here, the function is centered about the origin and the horizontal axis is the scale of the x_{max} term. The SHS output is the infinite interferogram multiplied by the detector widths RECT function. The interferogram data is cut off abruptly at the $-x_{max}$ and $+x_{max}$ values. In signal processing, this is known as truncation [33].

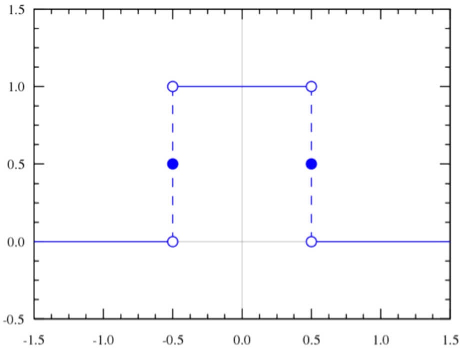


Figure 5.1: The RECT function has unit amplitude between $-x_{max}$ and x_{max} [15].

In the finite (realistic) case, the spectral radiance is:

$$L_{recovered}(k) = \int_{-\infty}^{\infty} \frac{1}{2} \cdot I(x) \cdot RECT(x) \cos\left(2\pi\nu_F x\right) dk[4]. \tag{5.22}$$

By the modulation theorem, the Fourier transform (FT) of the product of two functions is equal to the convolution of the FT'ed factors:

$$FT\left\{a \cdot b\right\} = FT\left\{a\right\} * FT\left\{b\right\} = A * B[33]. \tag{5.23}$$

The finite-x spectral radiance is:

$$L_{recovered}(k) = L(k) * SINC(k), \tag{5.24}$$

where the SINC function is the Fourier Transform of the RECT function in Equation 5.21. The theoretical maximum resolution of an interferometer is given by the full-width half-maximum (FWHM) of the characteristic SINC function, which can be calculated by:

$$R = \frac{1.207}{2x_{max}} \tag{5.25}$$

and can be improved by making a larger interferometer with a greater maximum achievable x_{max} [21].

The convolution of our spectral radiance signature with the detector SINC function causes non-zero amplitudes belonging to the lower frequencies located at the higher frequencies in our signal. This phenomenon is called leakage. To better describe leakage, let us look at the FT of the RECT function, the SINC function:

$$SINC(k) = \frac{\sin \pi k}{\pi k}[33]. \tag{5.26}$$

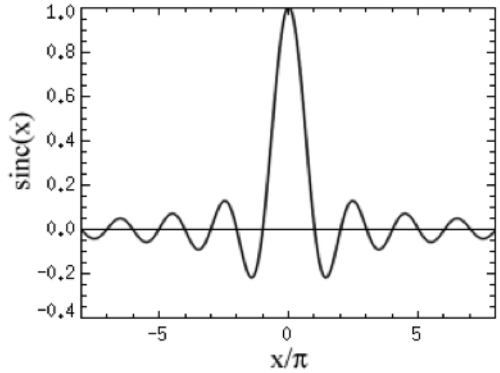


Figure 5.2: The SINC function has a center peak with oscillations on either side that eventually go to zero amplitude [16].

A plot of the SINC function is shown in Figure 5.2. As is shown in the plot, the SINC function consists of one large central and many side lobes, some with amplitudes greater than 0.2. If the interferogram is truncated with the RECT function, the data points at either end of the truncated plot will not have an amplitude of zero. The IFT of the truncated interferogram will cause the SINCs many side lobes to spread the spectral energy from the lower frequencies to the higher frequencies [28].

Because leakage is due to the signal truncation, or the abrupt detector cutoff, we can reduce its effects by smoothing out the ends of the RECT function. This, in turn will decrease the number of and amplitude of the SINCs many side lobes. This is done by multiplying the discrete, finite interferogram by an apodization function that slowly decays to zero at the endpoints. There are a number of different apodization functions that are often used, depending on the application. The first column in Figure 5.3 shows some of the more well-known apodization functions. The second column shows the FTs of the apodization functions and the third column shows a magnified view of the FTs side lobes. Here, the unity function is what we have referred to as the RECT function. One can see that the amplitudes of its FTs side lobes are much greater than the other functions. The smaller-amplitude side lobes are less effective in spreading information into the wrong frequencies [33].

Earlier, we said that the resolution of the interferometer is inversely proportional to the FWHM of the SINC function. As we see in Figure 5.3, the FWHM of the central lobes of the apodization functions are larger than for the SINC function. Therefore, the increase in central lobe width causes a decrease in the system spectral resolution [28].

5.2.5 Inverse Fourier Transform

As previously mentioned, taking the IFT of the measured interferogram gives the measured spectrum. We know that the measured interferogram is real and should be symmetric. The FT of a real and symmetric function is also real and symmetric. The FT of a real and anti-symmetric function, however, has both real and imaginary parts and is odd. We know that measured spectral radiance should be a real function. Also, we said earlier that the SHS interferogram is a function of $L(k)$ where k can be negative or positive and $L(-k)$ is equal in magnitude to $L(k)$. This means, our interferogram is the FT of the mirrored radiance spectrum. We want the IFT of the interferogram to be real and symmetric. If it is not, it will have an induced phase error we need to correct. We do this by taking the magnitude of the spectrum after the IFT:

$$L_{recovered}(k) = \sqrt{L_{recovered,R}(k)^2 + L_{recovered,I}(k)^2}, \tag{5.27}$$

where $L_{recovered,R}(k)$ and $L_{recovered,I}(k)$ are the real and imaginary parts of the IFT of the interferogram, respectively [44].

It is necessary to take the magnitude of the IFT, rather than just throw the imaginary part away because we want to maintain the spectrum power. From Parseval's theorem [33], we know that:

$$\int_{-\infty}^{\infty} |f(x)|^2 dx = \int_{-\infty}^{\infty} |F(k)|^2 dk, \tag{5.28}$$

where $|f(x)|^2$ is the power of the sample spectrum and $|F(k)|^2$ is the power spectrum of the sample spectrum. Simply removing the imaginary part of the IFT of our interferogram would reduce the power spectrum, reducing the power of the recovered spectrum as well. Therefore, taking the magnitude of the IFT ensures we do not lose any necessary information about our spectral signal.

Finally, we just said that the measured interferogram is actually the FT of the mirrored spectrum. We only need half the recovered information, mainly the half that is a function of $+k$. Therefore, we can discard the first $N/2$ samples.

5.2.6 Noise Reduction

After taking the IFT of the interferogram, we have the noisy spectrum. To reduce the noise in the signal, we can apply a moving filter or simply average multiple sample measurements together. Two filters often used for noise reduction are the mean and median filters. The mean filter replaces the sample at its current position with the average of the nearby samples. The median filter replaces its current sample with the median of the nearby samples. The mean filter is sometimes called a smoothing filter because it smooths the signal, whereas the median filter is better at preserving edges. This characteristic of the median filter makes it slightly more appropriate for signal processing in the spectral domain because it preserves the emission and absorption lines in the spectra [28].

5.2.7 Spectral Radiance Calibration

As we have mentioned, the spectral radiance measured by the SHS is the sum of the scene spectral radiance and instrument self-emission terms. The spectrum recovered by the IFT is the measured spectral radiance, so we must remove the instrument self-emission to retrieve the true spectral radiance of the imaged scene. The measured spectral radiance includes one more term, the instrument responsivity. The responsivity is a function of wavenumber and describes how sensitive the system is to each wavenumber. Therefore, the actual measured spectral radiance is:

$$L(k) = r(k)[L_{scene}(k) + g(k)] \tag{5.29}$$

and if we know the responsivity, $r(k)$, and the system offset, $g(k)$, we can calibrate the measured spectral radiance to the scene spectral radiance by:

$$L_{scene}(k) = \frac{L(k)}{r(k)} - g(k). \tag{5.30}$$

The instrument offset varies with instrument temperature and must be calculated often (usually between each measurement). We use a hot and a cold blackbody (BB) source for spectral calibration calculations. These calculations are usually performed within the instrument on a measurement-by-measurement basis using built-in hot and cold BB sources, as in AIRIS [36]. The hot BB has a temperature just above the ambient temperature, while the cold BB has a temperature of approximately ambient temperature, or just below ambient temperature. If the respective radiance measurements of the hot and cold BB's are $L_H(k)$ and $L_C(k)$ at temperatures, T_H and T_C, the responsivity is:

$$r(k) = \frac{L_H(k) - L_C(k)}{L_{BB}(T_H, k) - L_{BB}(T_C, k)}, \tag{5.31}$$

where $L_{BB}(T_H, k)$ and $L_{BB}(T_C, k)$ are the expected BB radiation measurements of a hot and cold BB at temperatures T_H and T_C, respectively. The instrument offset is then:

$$g(k) = \frac{L_C(k)}{r(k)} - L_{BB}(T_C, k). \tag{5.32}$$

The calibrated spectral radiance can then be calculated by Equation 5.30 [19].

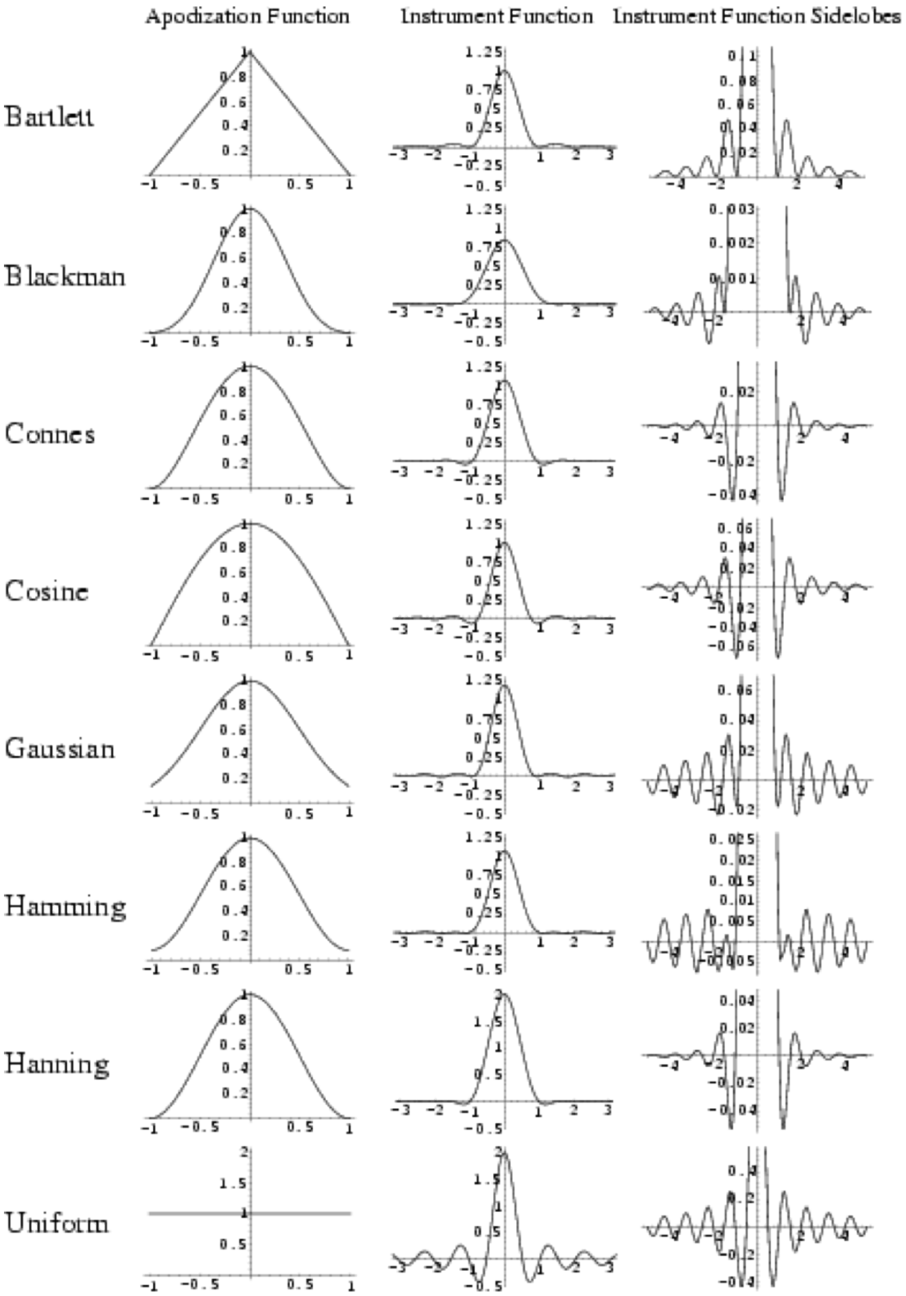


Figure 5.3: The first column shows common apodization functions, the second column shows their FT's, and the last column shows a magnified view of their side lobes [17].

Chapter 6

Experimental Approach

The first objective of this research is to create a radiometric model of the SHS. The second objective is to develop a processing algorithm, with real-time capability, to convert N-point SHS interferograms into calibrated spectral radiance. The experimental approach consists of simulating realistic interferometric data with the model and then using the simulated interferograms to test the algorithm. This process is repeated until desired results are achieved. Both the model and algorithm are implemented in MATLAB.

6.1 SHS Model and Interferogram Simulation

The SHS system developed in this research is shown in Figure 6.1. The sample spectrum provides the system input. The SHS measures its interferogram which is then processed to best-recover the input spectral radiance. Post-processing on the retrieved spectral data is possible for future work, depending on the SHS application.
The general SHS Model developed in this thesis has the following blocks:

1. System Input
2. Entrance Optics
3. Interferometer
4. Exit Optics
5. Detection
6. System Output

This organization is based on the model by Cooke et. al [4], shown in Figure 6.2.

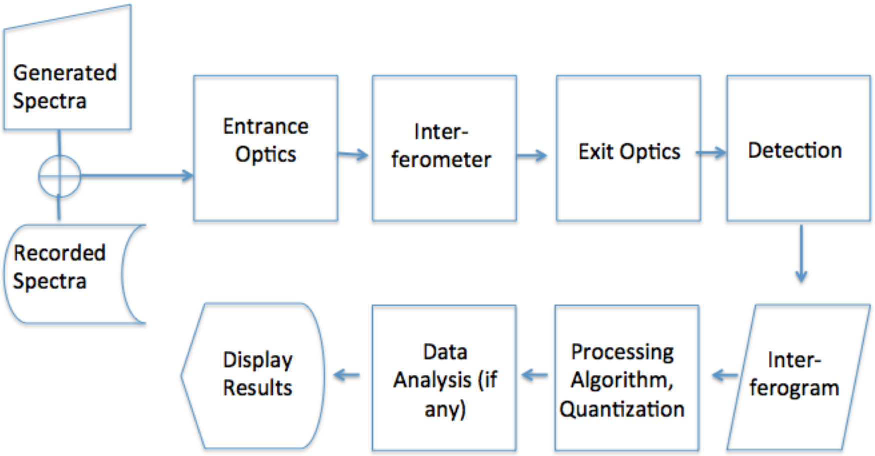


Figure 6.1: The entire SHS system consists of the input, or the scene spectral radiance, the SHS itself, and processing software to determine the measured spectral radiance.

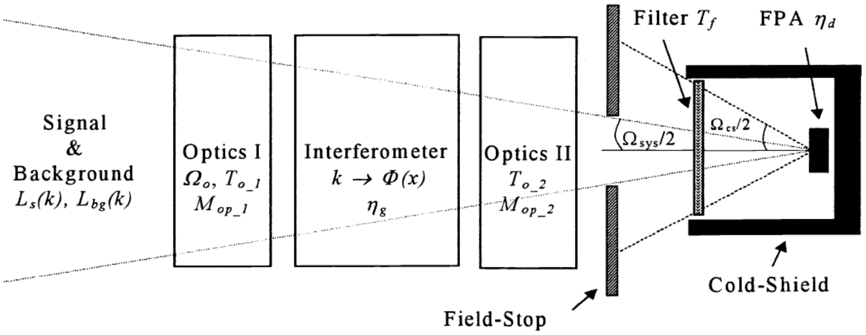

Signal & Background $L_s(k)$, $L_{bg}(k)$
Optics I Ω_o, T_{o_1} M_{op_1}
Interferometer $k \rightarrow \Phi(x)$ η_g
Optics II T_{o_2} M_{op_2}
Filter T_f
FPA η_d
$\Omega_{sys}/2$
$\Omega_{cs}/2$
Cold-Shield
Field-Stop

Figure 6.2: Cooke et. al's model accepts the sample radiance with added background radiance. The input undergoes the entrance optics, Optics I, the SHS configuration, Interferometer, the exit optics, Optics II, and the detection scheme (Field-Stop, Filter, FPA, and Cold Shield) [4].

6.1.1 SHS Parameters

We begin with the user-defined SHS operational parameters:

- minimum accepted wavenumber, k_{min}
- maximum accepted (Littrow) wavenumber, k_L
- high resolution number of interferogram samples, N_H

- number of interferogram samples, N
- diffraction grating order, m
- diffraction grating groove density, d

As we will soon see, the model generates a high resolution interferogram with N_H number of samples before the detection scheme. The detection scheme then resamples the signal at N samples. Using the above parameters, we calculate additional SHS parameters:

- Littrow angle of the diffraction gratings, θ_L, from Equation 3.26
- diffraction grating width, W, from Equation 3.27
- maximum value of position x along the detector, x_{max}, from Equation 3.28
- high resolution number of samples, N_H
- high spectral resolution, dk_H
- high resolution sample spacing, dx_H
- spectral resolution, dk, from Equation 3.29
- sample spacing, dx, from Equation 3.30

Again, we have defined spectral resolution and sample spacing parameters for both the high resolution interferogram generated by the model and the lower resolution interferogram sampled by the model's detection scheme. We calculate dk_H and dx_H from:

$$dk_H = \frac{2 \cdot k_L}{N_H} \tag{6.1}$$

and

$$dx_H = \frac{2 \cdot x_{max}}{N_H}, \tag{6.2}$$

respectively, where N_H is:

$$N_H = 2(k_{max} - k_{min}). \tag{6.3}$$

6.1.2 Input

Now, we are ready to run our spectral data through the model. The model accepts either generated spectral radiance or real, recorded spectral radiance measurements (units of $\frac{W}{m^2 sr cm^{-1}}$). The user places spectral radiance files into the data folder to be processed and the model runs the data through its various block processes.

6.1.3 Entrance Optics

The first block is the Entrance Optics block. The Entrance Optics incorporate the instrument self-emission, FOV, and optical transmission. The FOV is a constant and we can factor it out later. As mentioned earlier, the instrument self-emission is added to the scene radiance. The self-emission term is modeled as a BB radiator with unit emissivity and temperature of 290K, calculated by the Planck equation:

$$L_{BB}(k) = 2 \cdot 10^8 hc^2 k^3 \frac{1}{\exp\left[\frac{hc(100k)}{k_B T} - 1\right]}, \tag{6.4}$$

where wavenumber, k, has units of cm^{-1}, k_B is Boltzmann's constant, and T is the temperature of the blackbody. Finally, the transmission of the Entrance Optics is assumed to be wavenumber-dependent. The output of the Entrance Optics is, therefore:

$$L'(k_n) = \Omega \cdot \tau_1(k) L_{scene}(k_n) + L_{self,1}(k_n, 290), \tag{6.5}$$

where Ω is the system FOV (not the aperture FOV described earlier), $\tau_1(k)$ is the transmission function of the Entrance Optics, k_n is the discrete variable for wavenumber, $L_{scene}(k_n)$ is the input sample spectrum, and $L_{self,1}(k_n, 290)$ is the self-emission term of the entrance optics. The system FOV is found by:

$$\Omega = \frac{\pi}{4(f/\#)^2 + 1}, \tag{6.6}$$

where $f/\#$ is the system F-number.

6.1.4 SHS Interferometer

The next block is the SHS Interferometer block. Here, we employ the interferogram equation, Equation 3.42. Now, however, we must account for the diffraction grating efficiencies and vibrational and phase errors. The measured radiance at the detector is now:

$$L''(k) = (\frac{1}{2}\eta_A + \frac{1}{2}\eta_B) L'(k), \tag{6.7}$$

where η_A and η_B are the efficiencies of diffraction gratings A and B, respectively. After accounting for the vibrational and phase errors, the Fizeau frequency becomes:

$$\nu_F = 2(k - k_L)\tan\theta_L + 2(k - (k_L + k_e))\tan(\theta_L + \theta_e), \tag{6.8}$$

where θ_e is the angle of error between the Littrow angles of the two gratings and k_e is the Littrow wavenumber associated with the diffraction grating Littrow angle, $\theta_L + \theta_e$. Therefore, the interferogram calculated in the Interferometer block is:

$$I(x_m) = \sum_{n=-N_H/2}^{N_H/2-1} \frac{1}{2} L''(k_n)[1 + \cos(2\pi\nu_F(x_m + x_e) + \phi(k,x))]dk_H, \tag{6.9}$$

where x_m is the discrete variable for position on the detector, ranging from $-x_{max}$ to $+x_{max}$, and x_e and $\phi(k,x)$ are the errors associated with vibrations along the detector and phase shifts, respectively.

6.1.5 Exit Optics

The result of Equation 6.9, the high resolution interferogram, is then passed through the Exit Optics, which includes a second transmission function and another self-emission term, also modeled as a blackbody but with a temperature of 280K. The output of the Exit Optics is then:

$$I'(x_m) = \tau_2(k)_{avg} I(x_m) + \Omega\tau_2(k)_{avg} \sum_n L_{self,2}(k_n, 280)dk, \tag{6.10}$$

where $\tau_2(k)_{avg}$ is the mean of the Exit Optics transmission.

6.1.6 Detection

The last block in the SHS model is the Detection block. Here, we re-sample the high resolution interferogram. The re-sampling occurs at the user-specified number of samples, N, spectral resolution, dk, and sample spacing, dx, where dk and dx are calculated by Equations 3.29 and 3.30, respectively. We also account for the sensitivity variation across the detector and add the system noise, such that the output becomes:

$$I''(x_m) = S_{FPA}(x_m) I'_{resampled}(x_m) + I_{noise}(x_m), \tag{6.11}$$

where $S_{FPA}(x_m)$ describes the pixel-to-pixel sensitivity variation and $I_{noise}(x)$ is the system noise in the interferogram domain. We calculate $I_{noise}(x)$ using the the standard deviations defined in Section 5.1.4 and MATLAB's random number generator, randn(x), to calculate the noise electrons in a Gaussian fashion. The noise signal in electrons is:

$$Noise(x_m) = \sum_i \pm n(x_m) \cdot \sigma_i. \tag{6.12}$$

The noise in irradiance (i.e. interferogram) units (W/m^2) is:

$$I_{noise}(x_m) = \frac{hck}{A_d t_{int} \cdot QE} Noise(x_m). \tag{6.13}$$

The output of the SHS model is realistic, interferometric data.

6.2 Interferogram Processing Algorithm

As we stated in Section 5.2, the interferogram processing algorithm is as follows:

1. Flat-fielding correction
2. Phase correction
3. Bias removal
4. Apodization
5. Inverse Fourier Transform
6. Noise reduction
7. Spectral Radiance Calibration

Our flat-fielding technique is similar to the first technique described in Section 5.2.1. We also take two measurements of the same known source - one measurement with one interferometer arm blocked and the second with the other arm blocked. We then average the two measurements together and normalize the result to unity. The flat-field corrected interferogram is then:

$$I_{ff}(x) = \frac{I(x)}{\left(\frac{I_A(x)+I_B(x)}{2}\right)_{normalized}}. \tag{6.14}$$

The phase correction performed by the algorithm uses the methods outlined in Section 5.2.2. Next, we perform apodization with the Hamming window, as in [14]. The Hamming window reduces our spectral resolution by a factor of about 1.82 [45]. As described in Section 5.2.5, after apodization we then take the magnitude of the IFT of the interferogram to recover the spectral radiance. We also need to scale for the instrument optics and resolution such that the raw recovered spectral radiance is:

$$L_{recovered}(k_n) = \frac{2}{\Omega dk_n}|IFT(I(x_m))|, \tag{6.15}$$

where the factor of 2 accounts for the fact that as the diffracted beams travel back through the beam-splitter, only half the total radiance reaches the detector, while the other half travels back toward the system aperture. At this time, we can use noise reduction filters to eliminate the system noise. The noise reduction method used in this thesis, however, is sample averaging. We perform averaging as we record the interferogram samples, so no filtering is necessary. Finally, the last step of the algorithm is the spectral radiance calibration, which is performed using the methods described in Section 5.2.7.

Chapter 7

Results and Discussion

As mentioned, our SHS model and interferogram processing algorithm are used recursively to optimize the other. The example SHS design in this thesis is a LWIR SHS with a spectral range of 950 to 1250 cm^{-1}. We incorporate the example design plans in our SHS model and simulate realistic interferograms.

The optimal SHS model is one that incorporates as many errors as we would normally find in a real SHS instrument. The errors we are considering are described in Section 5.1. As mentioned earlier, our SHS model is formulated using the basic framework of the IRISHS model. For a summary of comparison between our model and IRISHS, see Appendix C. The model is optimized by beginning with a simplified version of our design and slowly building up the model. The optimal interferogram processing algorithm, on the other hand, is one that reduces as many errors as possible in our recovered spectra.

We began with the interferogram equation as our SHS model. We generated interferograms with this model and then ran them through our interferogram processing algorithm to check that we could recover the input spectra. We modified the algorithm until we achieved the best possible results (i.e. those with the fewest errors in the recovered spectra). Once we were able to recover the input spectra with our simplified model, we began adding errors, one at a time, repeating the same process of generating data and modifying the processing algorithm for the best spectra recovery.

Our current ”optimized” model and processing algorithm are described in Chapter 6. We have chosen an SHS design example with the SHS operational parameters listed in Table 7.1. The model then calculates the other SHS parameters necessary to run the program in MATLAB (Table 7.2). (For a complete list of SHS model parameters, see Appendix D.) The next few sections show the results of the model and interferogram processing algorithm using this SHS design.

Table 7.1: The user-specified operational parameters for the LWIR SHS design example.

k_{min} $[cm^{-1}]$	950
k_L $[cm^{-1}]$	1250
N	128
m	1
d $[cm^{-1}]$	143

Table 7.2: The additional parameters calculated by the SHS model using the user-specified operational parameters.

N_H	600
θ_L $[°]$	3.2791
W $[cm]$	0.9324
x_{max} $[cm]$	0.4654
dk_H $[cm^{-1}]$	1.00
dx_H $[cm]$	0.0016
dk $[cm^{-1}]$	4.6875
dx $[cm^{-1}]$	0.0073

As we stated in Section 2.2, this research provides two tools to better understand the SHS: the model and interferogram processing algorithm. The model generates realistic data for testing the processing algorithm. The model and processing algorithm may also be used together in design optimization or characterization efforts. We will show results for all these uses.

The SHS model and interferogram processing algorithm are tested using a variety of input spectra. Specifically, we input:

- MODTRAN-generated atmospheric spectral radiance spectra,
- Recorded spectral radiance spectra of sand of sizes 425 to 1000 and 1000 to 1400 microns with and without SF96 (oil) measured at nadir and 45° from nadir,
- microDIRSIG-generated spectral radiance spectra of sand of sizes 425 to 1000 and 1000 to 1400 microns with and without SF96 (oil) at nadir and 45° from nadir.

The atmospheric spectral radiance spectrum is generated using MODTRAN software. The simulated data is for Rochester, NY (Latitude = 43.26, Longitude = 77.61), where the altitude, relative to sea level is 260 feet. The sensor is imaging at an altitude of 67,365 feet, pointing down (i.e., initial zenith angle is 180°. The assumed collection occurred on May 5, 1999 at 12:40 EDT. The spectrum is generated from 600 cm^{-1} to 1600 cm^{-1} in $1cm^{-1}$ increments with a slit function with FWHM of $1cm^{-1}$.

The recorded sand spectral radiance data were measured on the roof of the Chester F. Carlson Center for Imaging Science at Rochester Institute of Technology (RIT) in Rochester, NY in June of 2012. The measurements were collected with a Designs & Prototypes (D&P) Model 102 FTIR non-imaging spectrometer. The sample contents were of pristine sand with particle diameters between 425 to 1000 microns, pristine sand of 1000 to 1400 microns, sand of 425 to 1000 microns mixed with oil, and sand of 1000 to 1400 microns mixed with oil. The four samples were recorded at two different viewing angles: nadir and 45° from nadir.

The simulated sand spectral radiance data were generated by RIT's microDIRSIG software. As an initial estimate, the sand particle shape was assumed to be spherical. Spectra were generated for the same four sand samples described above at nadir and 45° from nadir.

7.1 Spectra Recovery Results

First, we will provide examples of the interferograms generated by the model and the corresponding spectra recovered by the processing algorithm. We include one MODTRAN result, one recorded sand result (pristine, 425 to 1000 microns at nadir), and one simulated sand result (pristine, 425 to 1000 microns at nadir). In all results, we recover the same approximate shape and magnitude of the input spectra, but at a slightly lower spectral resolution. The SNR is about 300 and the NEdT is about 0.2K.

7.2 Design Optimization Results

Design optimization refers to the selection of design parameters that best achieve desired results based on the application. Typically, during the design optimization process, we are interested in system trade-offs. We demonstrate a handful of these system trade-offs with our SHS model and the specific design described above. The trade-offs we investigate are:

- Spectral range versus spectral resolution,
- Diffraction grating width versus spectral resolution,
- Throughput versus SNR.

From Section 3.4.2, we know that the spectral resolution, dk, is inversely proportional to the spectral range, Δk. This relationship is also portrayed in the plot in Figure 7.4. In certain applications, the system spectral resolution may be more important than the range. For example, when looking for emission and absorption lines, a higher spectral resolution will provide a more distinct peak. In cases such as this, a wide spectral range is unnecessary

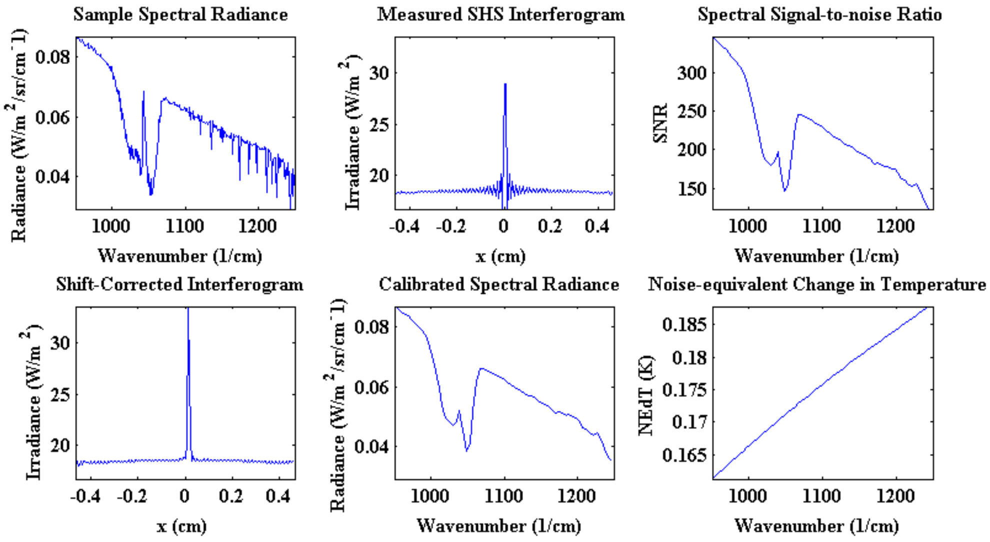


Figure 7.1: Results of the MODTRAN test: in clockwise rotation from top left, the input atmospheric spectral radiance spectrum, measured interferogram, wavenumber-dependent SNR, shift-corrected interferogram, recovered spectrum, and wavenumber-dependent NEdT.

because we only need information from a narrow spectral region. In other cases, however, we need more spectral coverage to make sense of the data. For example, if we are collecting an HSI data cube in order to perform image classification, a larger number of observed wavenumbers provides more statistical information for our classification algorithm.

Section 3.4.2 also tells us that as the number of samples increases, the diffraction grating width increases, improving the spectral resolution. This is demonstrated in the plot in Figure 7.5. Therefore, if you need a higher spectral resolution, use a wider diffraction grating. It sounds simple, but recall that one of the advantages of the SHS was its compact size due to its mechanical simplicity. If the instrument size requirement is not restricted, trading a larger size for a higher spectral resolution is very beneficial. If, however, the instrument is meant to be field-deployable, a smaller size may be necessary. We also need to consider the fact that increasing the diffraction grating width also increases the detector size and the number of pixels. Again, the increase in detector size will make the instrument less portable. The larger number of pixels will also increase the interferogram computation time. In the case of a portable instrument with real-time processing, a high spectral resolution may not be as important.

In Section 3.4, we said that one advantage of the SHS was its high optical throughput. Throughput describes the amount of space through which an instrument can measure

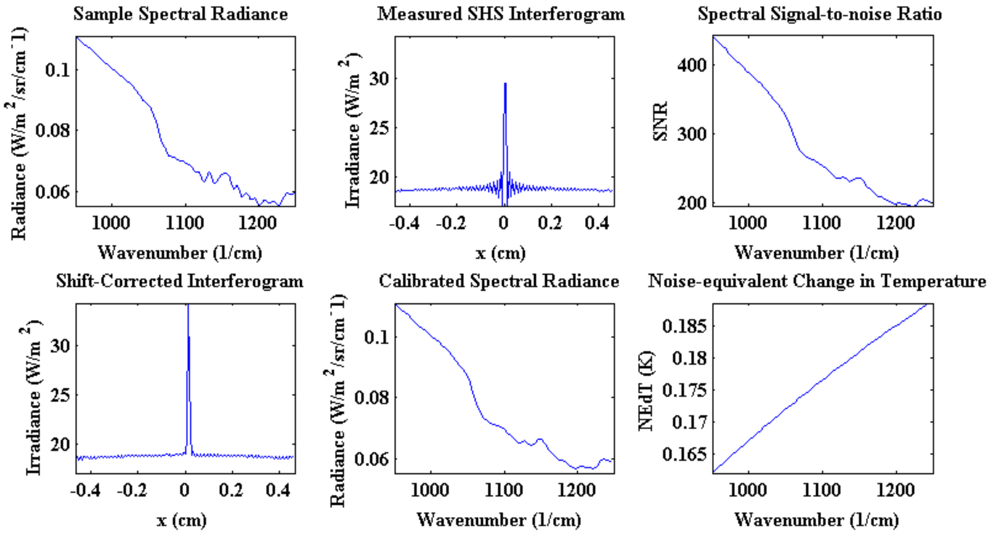


Figure 7.2: Results of the recorded sand test: in clockwise rotation from top left, the input atmospheric spectral radiance spectrum, measured interferogram, wavenumber-dependent SNR, shift-corrected interferogram, recovered spectrum, and wavenumber-dependent NEdT.

radiance propagation. A larger throughput means more radiant energy is passing through our system, increasing our sensitivity. As with most optical systems, an increase in sensitivity will cause an increase in SNR. This relationship is depicted for the SHS in Figure 7.6. Again, we have a tradeoff between an improved signal and instrument size. From Equation 3.3, we know that in order to increase the throughput we must increase the instrument aperture and/or the FOV, increasing the overall size of our instrument.

7.3 System Characterization Results

In using the model and processing algorithm for system characterization, we are interested in observing the overall system performance as a function of the sources of error. For this demonstration, we consider the following sources of system error:

- Littrow angle error, θ_e,
- Phase error, $\phi(k, x)$,
- Calibration error rate,
- Detector position error, x_e.

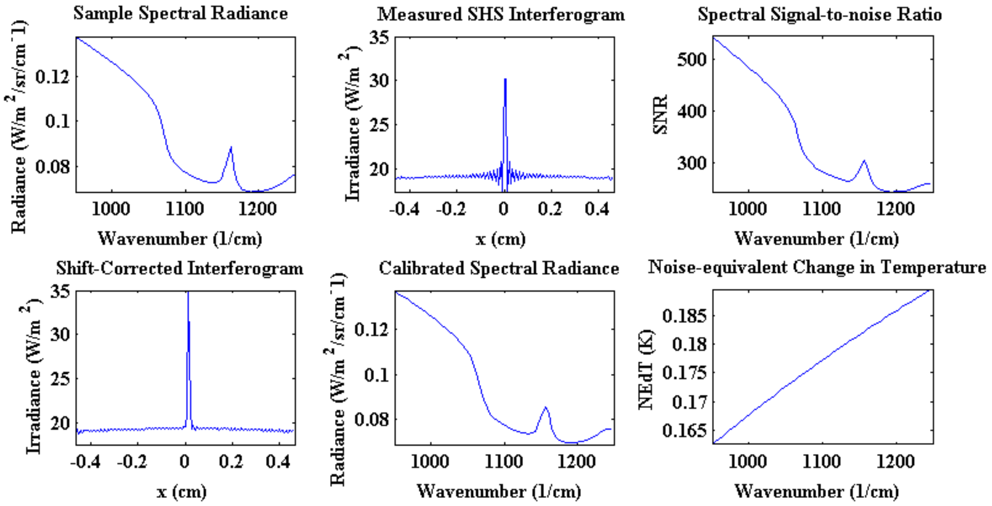


Figure 7.3: Results of the simulated sand test: in clockwise rotation from top left, the input atmospheric spectral radiance spectrum, measured interferogram, wavenumber-dependent SNR, shift-corrected interferogram, recovered spectrum, and wavenumber-dependent NEdT.

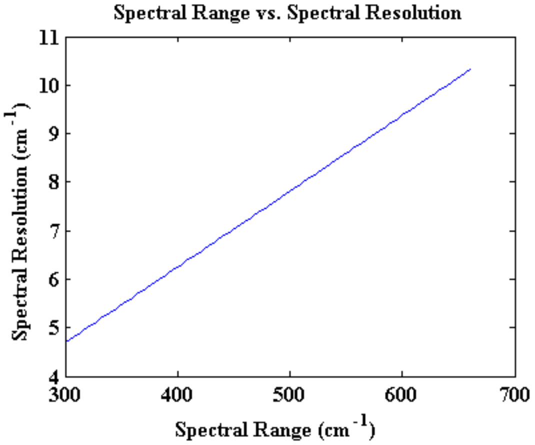


Figure 7.4: Results of the system trade-off between spectral range and spectral resolution.

We use system performance metrics such as the root-mean-square error (RMSE) between the input spectra and the recovered spectra, the percent accuracy between the input and recovered spectra, the signal-to-distortion ratio (SDR), and the separation angle determined by the spectral angle mapper (SAM) algorithm. All the results in this Section are averaged

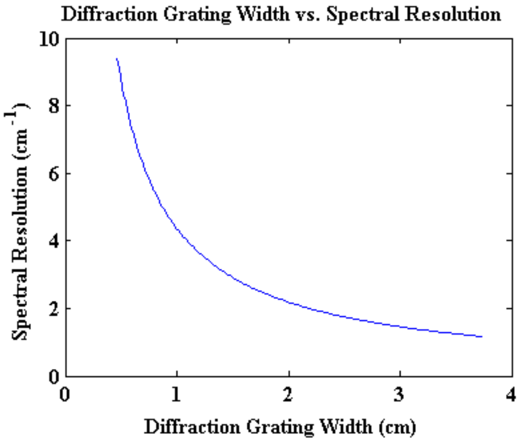


Figure 7.5: Results of the system trade-off between diffraction grating width and spectral resolution.

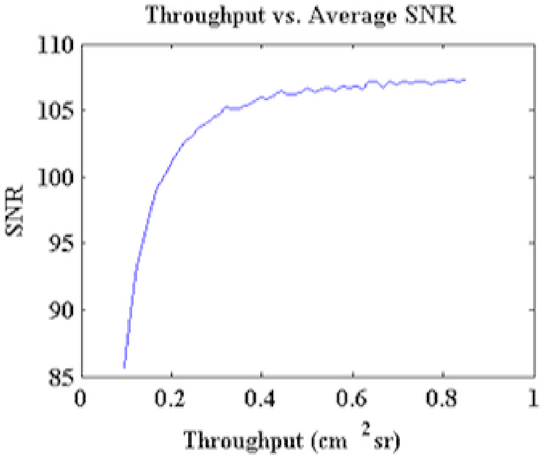


Figure 7.6: Results of the system trade-off between throughput and SNR.

over all the sample spectra described earlier. The RMSE describes the difference in the two spectra averaged over all wavenumbers and is defined as:

$$RMSE = \sqrt{\frac{\sum_k (L_{input}(k) - L_{recovered}(k))^2}{N_L}}, \tag{7.1}$$

where $L_{input}(k)$ is the input spectral radiance and N_L is the number of samples of the recovered spectral radiance function. Note that we must interpolate our input spectrum such that it also has N_L samples. The percent accuracy is calculated by:

$$\%Accuracy = 100\% - \frac{100\%}{N_L}\sum_k \frac{|L_{recovered}(k) - L_{input}(k)|}{L_{input}(k)}, \tag{7.2}$$

The SDR is calculated by:

$$SDR = \frac{1}{N_L}\sum_k \frac{L_{input}(k)}{RMSE}. \tag{7.3}$$

Finally, the separation angle is defined:

$$\alpha = \cos^{-1}\left[\frac{\vec{L_{recovered}}(k) \cdot \vec{L_{input}}(k)}{||\vec{L_{recovered}}(k)||||\vec{L_{input}}(k)||}\right], \tag{7.4}$$

where $L_{input}(k)$ is the input spectral radiance corresponding to a particular recovered radiance, $L_{recovered}(k)$, $L_{recovered}(k) \cdot L_{input}(k)$ is their dot product, and $||L_{recovered}(k)||$ and $||L_{input}(k)||$ are their magnitudes.
Recall that the Littrow angle in our example design is 3.2791°. We incorporate a Littrow angle of up to 0.25° and observe the system performance. The RMSE and separation angle steadily increase while the percent accuracy steadily decreases and the SDR decreases quickly but does not decrease very drastically over the Littrow angle error range of 0.1 to 0.25° (see Figure 7.7. We see similar results with the phase error (Figure 7.8), where the SDR decreases dramatically except for over the range of 0.1 to 0.3 radians. Next, we consider a calibration error rate of up to 0.3. Recall from Section 5.1.4 that the calibration noise in our measured signal will be proportional to the calibration error rate. Again the calibration error rate does not affect the percent accuracy as much as the Littrow error. In Figure 7.9, we see that the SDR does not vary significantly over the calibration error rate range of 0.015 to 0.03. In considering the detector position error, we should keep in mind that the spatial sample spacing is 0.0073 *cm*. The RMSE, percent accuracy, and SDR remain approximately constant up to a position error of about 0.00035 *cm*, as in Figure 7.10. If we can confine the vibrational detector position error to about one twentieth the sample spacing, we could possibly eliminate any such errors in our recovered spectrum.

In addition to observing system performance as a function of the system errors, we can also make an effort to characterize our system by observing the average spectrum recovery percent accuracy as a function of SNR (Figure 7.11. As stated above, the noise is proportional to the calibration error rate. We can easily change our SNR by changing the calibration error rate. Figure 7.11 is a very useful plot in characterizing a system. If we are required to have a specific overall system accuracy, for example, we can determine the minimum SNR needed to achieve that accuracy level and us that information to determine the necessary integration time or number of sample averages, etc.

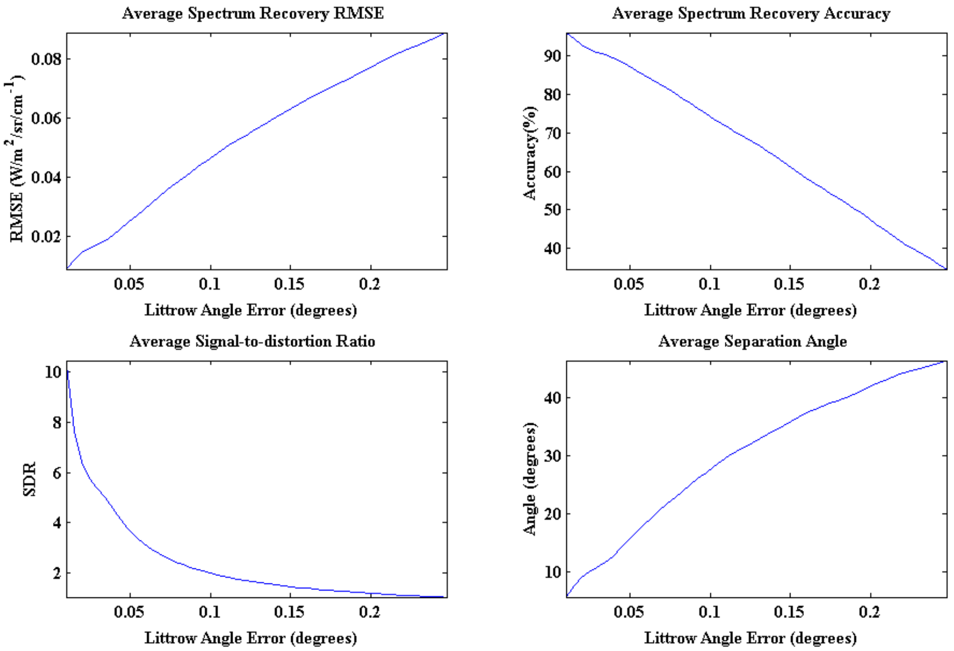


Figure 7.7: Results of the Littrow angle error, clockwise from top left: RMSE, percent accuracy, SDR, and separation angle.

7.4 Model Validation

The purpose of this research is to provide two tools for SHS studies: the SHS model and the interferogram processing algorithm. In order for these to be useful, we need to ensure that the model is outputting realistic data and that the interferogram processing algorithm has real-time capability.

7.4.1 Performance Metrics

As we have said, there are a number of performance metrics we can use to describe our system: NESR, NEdT, and SNR. We usually expect an SNR of a few to several hundreds and an NEdT within 0.1 to 0.5K for LWIR systems [36]. From the plots in Section 7.1, we know that we are achieving an average SNR of about 300 and an NEdT of about 0.2K. As initial tests, these results demonstrate the validity of our realistic SHS model.

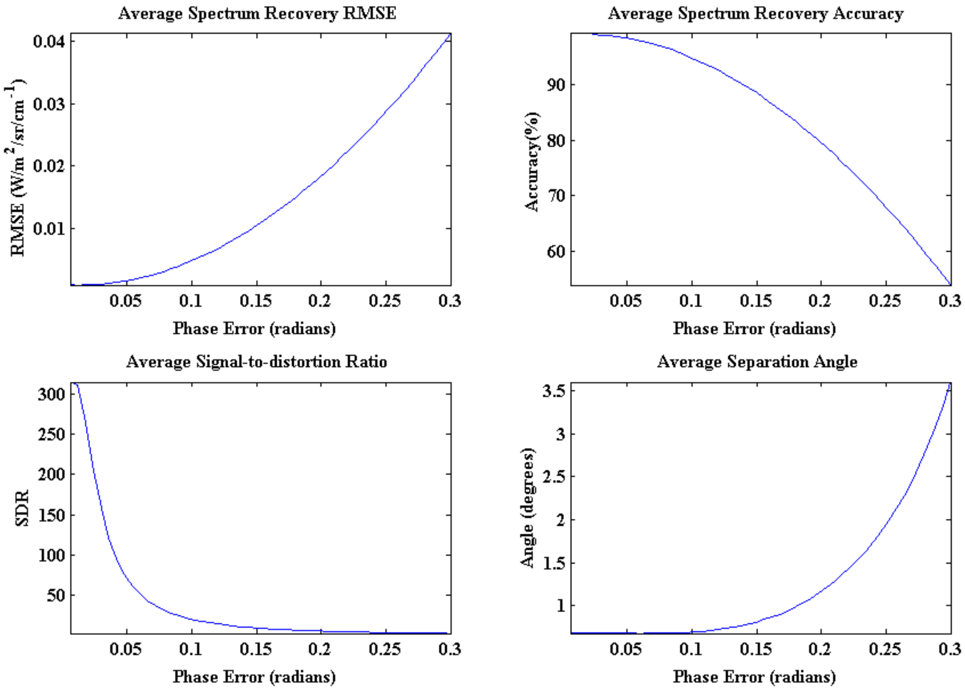


Figure 7.8: Results of the phase error, clockwise from top left: RMSE, percent accuracy, SDR, and separation angle.

7.4.2 Comparison of Results

A more in depth analysis of our model's validity would be to compare our results with results of a real LWIR SHS. As we have already noted, only one LWIR SHS exists today: SHIMCAD. Enlgert et al. [14] measured the transmittance of methanol gas samples placed in front of a 100°C blackbody with the SHIMCAD model. Sample pressures ranged from 6.9 to 59.9 hPa. We use methanol transmittance spectra at 93.3 hPa provided by the Coblentz Society's IR spectra collection [46] to simulate similar measurements with our system. Multiplying the downloaded transmittance spectra for our SHS accepted spectral range by the blackbody radiance at $373.15K$ gives us a measurement of spectral radiance to input to the model. After the conversion back to calibrated spectral radiance, we can divide by the same blackbody function to recover the transmittance.

Figure 7.12 shows the results of this process at our specified SHS design. Recall that our design's spectral resolution is approximately half that of the SHIMCAD design. For comparison, Figure 7.13 shows the result of the 59.9 hPa methanol transmittance sample

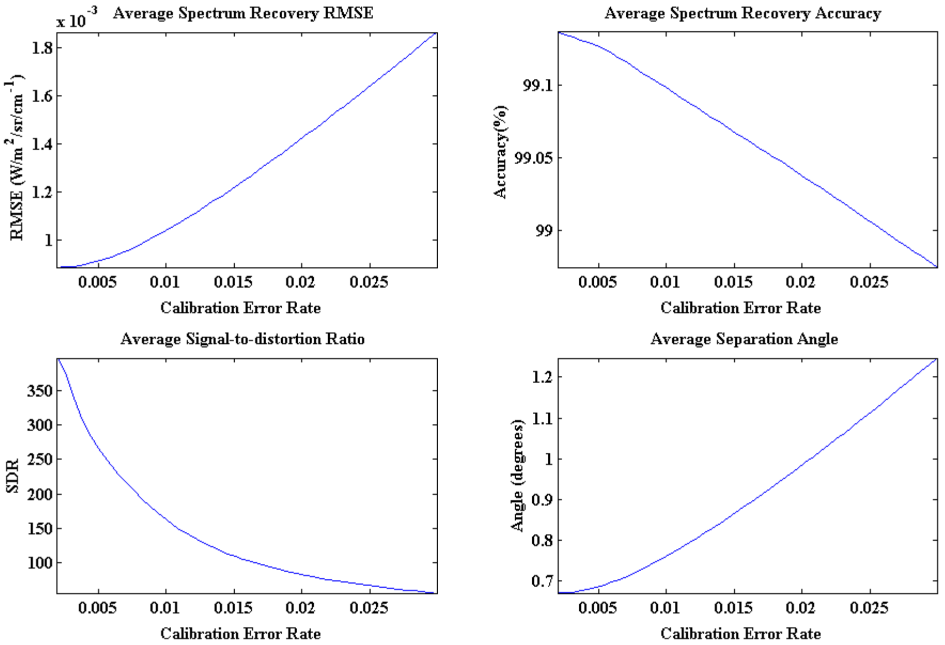


Figure 7.9: Results of the calibration error rate, clockwise from top left: RMSE, percent accuracy, SDR, and separation angle.

measured by SHIMCAD. As you can see, we barely recover the absorption peak at 1040 cm^{-1}, whereas the peak is very well-defined in the SHIMCAD measurement. By doubling the number of pixels (i.e. doubling the diffraction grating width), we can simulate a measurement more comparable to that achieved by SHIMCAD. Figure 7.14 shows this result. Now, we recover much more of the absorption peak, but not as well as SHIMCAD.

Although our results still do not match the SHIMCAD results, we must pay attention to the input transmittance function. As shown in Figures 7.12 and 7.14, the input transmittance data we downloaded from the Coblentz Society's IR spectra collection is a lower resolution than the sample measured by SHIMCAD. Increasing our spectral resolution still would not give us a result identical to that of the SHIMCAD model because our input spectrum is at a reduced resolution. Also, recall that the data provided by the Coblentz Society is for a methanol gas sample at 93.3 hPa, while the largest observed sample pressure in the SHIMCAD results is only 59.9 hPa. This pressure difference will also lead to a difference in measurements.

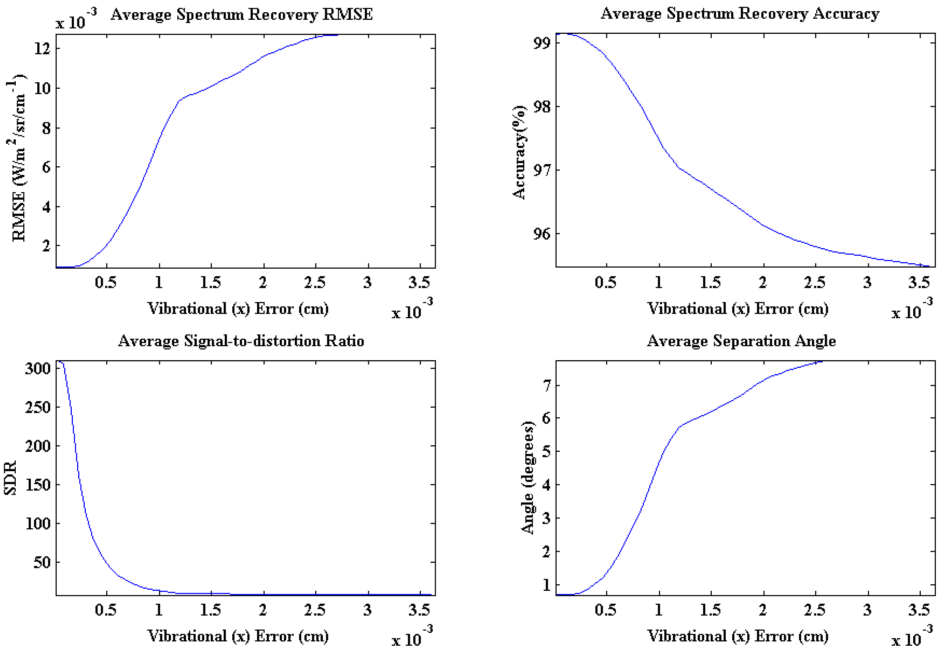


Figure 7.10: Results of the detector position error, clockwise from top left: RMSE, percent accuracy, SDR, and separation angle.

7.4.3 Real-time Processing Capability

Because advantages of the SHS technology include its mechanical simplicity, compact size, and low collection time, the SHS is suitable as a portable, field-deployable instrument. For on-the-move measurements, specifically, it is helpful (and sometimes required, depending on the application) to perform real-time interferogram processing.

The work presented in this thesis is implemented using MATLAB Release 2011a on a MacBook Pro with an Intel Core i5 processor at 2.3 GHz. The NovaBench software is used to determine that the MacBook Pro processor operates at 117 mega floating-point operations per second (MFLOPS). The Tesla C2705 GPU by Nvidia is advertised as operating at 515 GFLOPS [47]. If we implement the interferogram processing algorithm on the Tesla GPU, we increase our processing speed by a factor of 4,444. Currently, the algorithm takes about 0.2 seconds per interferogram on the MacBook Pro. Theoretically, the algorithm should only take about 45 microseconds per interferogram on the Tesla GPU. Assuming that our SHS platform is moving at 30mph with a pixel ground sample distance of 7.5cm, an integration time of 500 microseconds, and an image size of 128 by

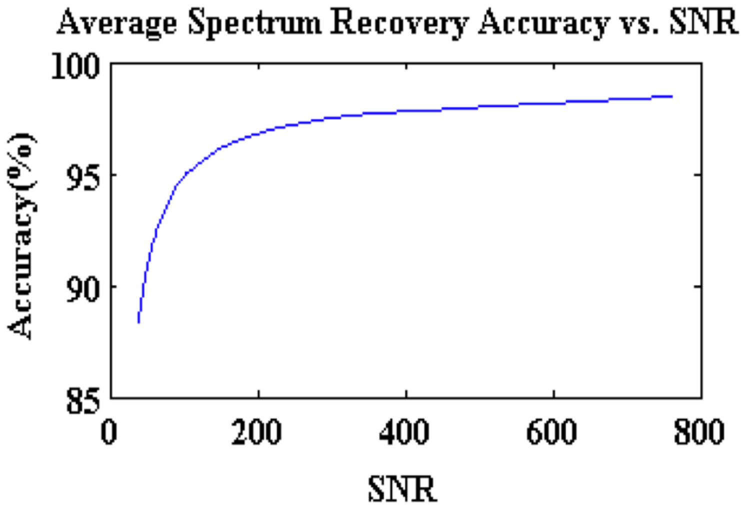


Figure 7.11: Results of the average spectrum recovery percent accuracy as a function of SNR.

128 pixels, we would need the algorithm to convert one row of pixel interferograms in less than one second. Using the Tesla GPU, we should be able to convert an entire image of interferograms into calibrated spectral data in about 0.74 seconds, achieving our desired processing speed. This particular GPU is certainly not the fastest on the market and it is also possible to use multiple GPUs at one time. Therefore, it is reasonable to say that the interferogram processing algorithm has real-time capability.

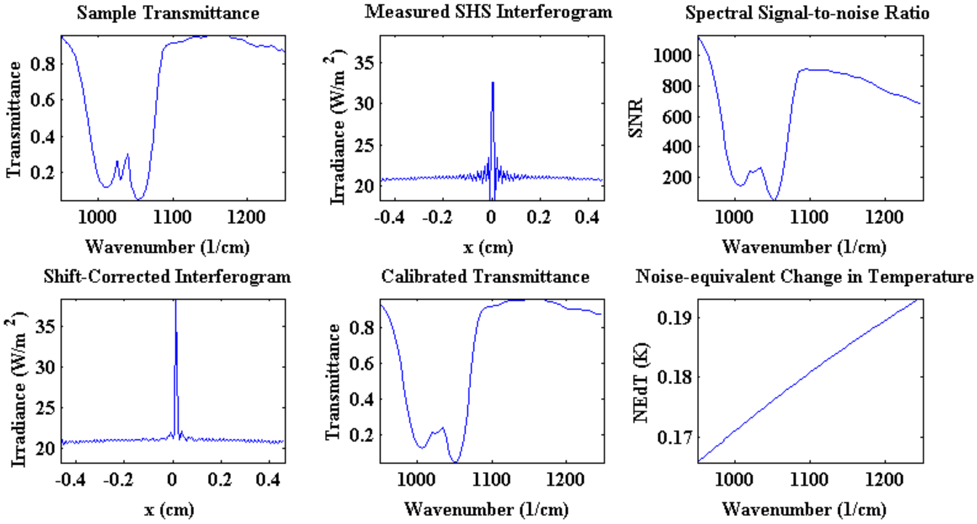


Figure 7.12: Results of the methanol (93.3 hPa) test at 128 samples: in clockwise rotation from top left, the input methanol gas transmittance spectrum, measured interferogram, wavenumber-dependent SNR, shift-corrected interferogram, recovered spectrum, and wavenumber-dependent NEdT.

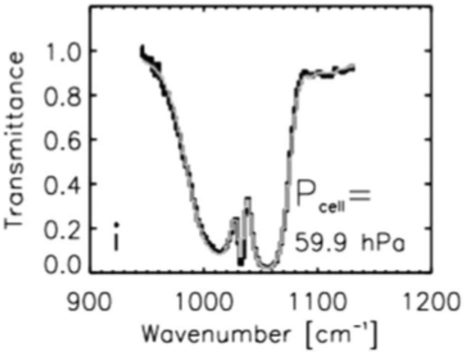


Figure 7.13: Results of the SHIMCAD measurement of methanol transmittance (59.9 hPa).

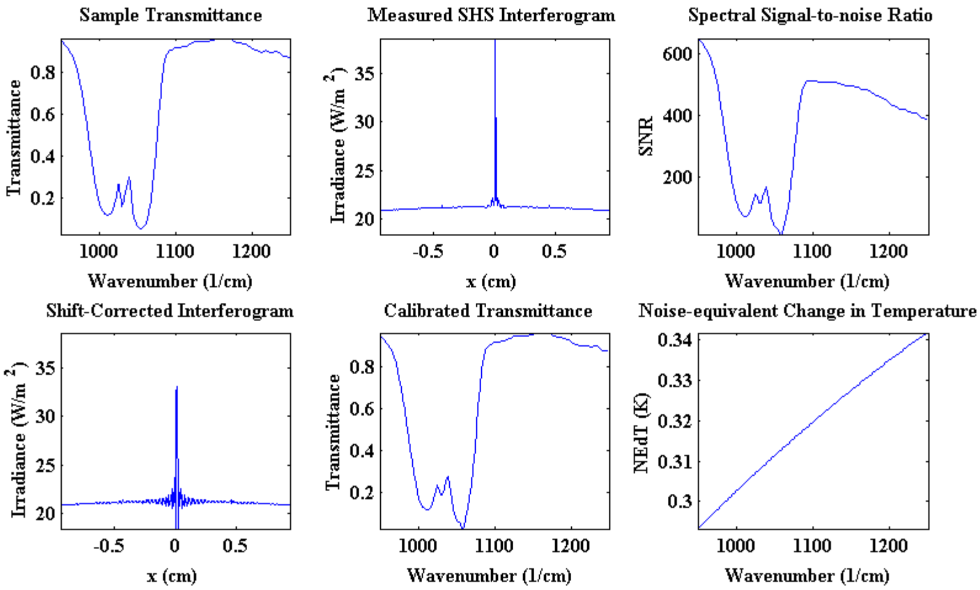


Figure 7.14: Results of the methanol (93.3 hPa) test at 256 samples: in clockwise rotation from top left, the input methanol gas transmittance spectrum, measured interferogram, wavenumber-dependent SNR, shift-corrected interferogram, recovered spectrum, and wavenumber-dependent NEdT.

Chapter 8

Conclusions

Though the class of FTS has been a major area of research and development for years, the field of spatial heterodyne spectroscopy is still fairly new. Specifically, the area of LWIR spatial heterodyne spectroscopy is mainly understudied. In fact, only one, working LWIR SHS exists today. In an effort to encourage and assist LWIR SHS studies, this research provides a general SHS model and interferogram processing algorithm that can be used to analyze LWIR designs as well as any other SHS designs. The examples in this thesis are specific to a LWIR design.

The results presented indicate mostly accurate spectral recovery with realistic quantities for SNR and NEdT. We demonstrate the use of the model and algorithm to perform design optimization and system characterization. Our model is validated with methanol transmittance spectra, achieving results comparable to SHIMCAD, the current LWIR SHS. Our interferogram processing algorithm demonstrates real-time capability. Overall, we determine our model and algorithm simulate realistic, useful results. We also indicate the feasibility of a LWIR SHS design with accurate spectral recovery results.

Chapter 9

Suggested Further Work

In Section 2.2, we said we wish to provide two tools for SHS studies: a general SHS model and an interferogram processing algorithm. The model is made as general as possible, but results in this thesis are demonstrated for a specific design in the LWIR. Though the model does account for wavenumber-dependencies and some system errors, there is always room for improvement, no matter the SHS design or application.

Some suggestions for further work are:

- Including actual optical responsivity functions
- Incorporating field-widening prisms
- Expanding the model to perform multiple-order analyses
- Including imaging SHS aspects
- Creating a GUI for easy-use

In depth analyses of the optics used in an SHS design (beam-splitter, lenses, diffraction gratings, etc.) would give actual wavenumber-dependencies to be used in the model to produce data specific to an SHS design. Wavenumber-dependencies include optical transmissions, efficiencies, absortances, and reflectivities. Ray-tracing software could also prove helpful in following the radiant photons from the source to the detector.

Previous studies [4, 8, 14, 18, 26, 23] have shown that including prisms between the SHS diffraction gratings and the beam-splitter (see Figure9.1) cause field-widening without any additional size or moving parts. This increases the system throughput without affecting the other design parameters.

As previously stated in Section 3.4.4, using an anamorphic telescope at the SHS entrance aperture allows for the collection of multiple interferograms. Recording the spatial information of a row or column of pixels in one dimension and the spectral information in

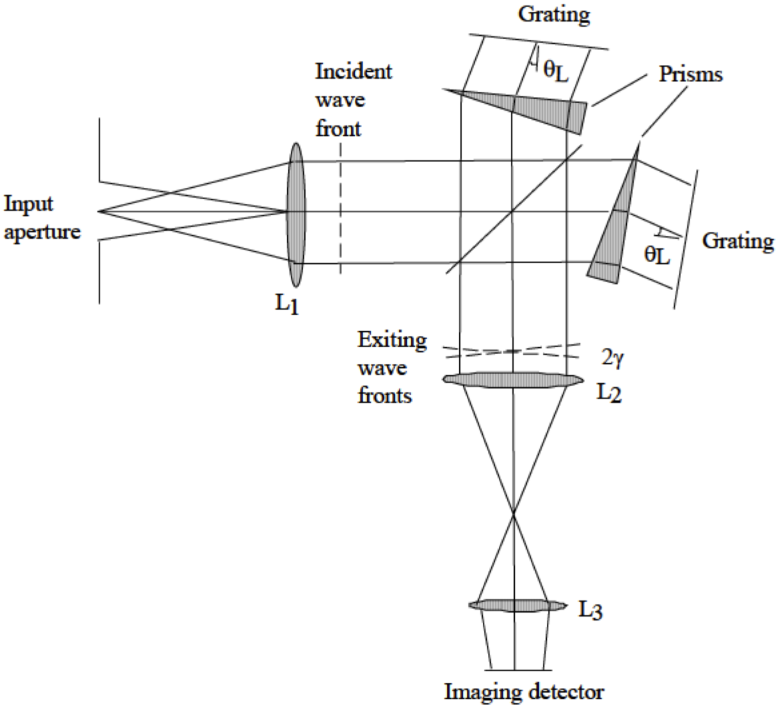


Figure 9.1: A depiction of the field-widened SHS with prisms placed between the diffraction gratings and beam-splitter [18].

the other dimension of our detector allows us to collect an entire hyperspectral data cube by scanning the imaged scene. The model presented focuses only on the interferometry aspects of the SHS, but it may be useful to expand the model to include these imaging aspects as well.

The SHS concepts explained in Section 3.4.2 and implemented in the model in Section 6.1 assume a Fizeau frequency pattern resulting from the one specified diffraction grating order. In Section 3.1, however, we said that the diffraction grating outputs light of multiple orders, all interfering with one another. The interferogram in a real SHS system, therefore, will be a result of all diffraction grating orders [8]. Though this multiple-order analysis is outside the scope of this research, including such results in the model would make it more realistic and better predict system performance metrics for design optimization and characterization.

Finally, we need to consider the user of our model and processing algorithm. Making the tools more user-friendly and easy to employ for a range of SHS applications would make them more helpful. Perhaps, a graphical user-interface (GUI) would be useful. A GUI would allow a user to select design parameters and error rates and instantaneously see the resulting interferogram and recovered spectrum.

Appendix A

Summary of LWIR Interferometry Technologies

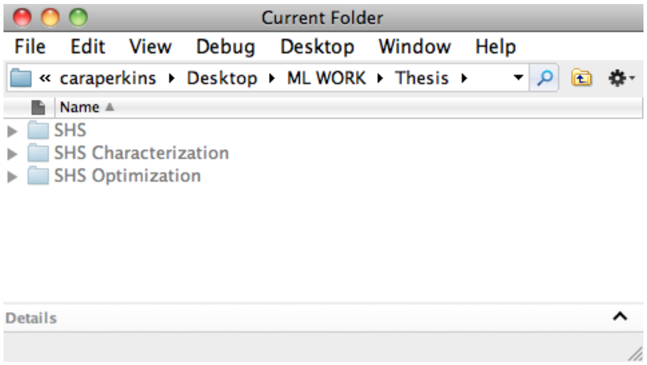
Table A.1: The technology summary and comparison of the instruments discussed in Chapter 4.

Name	**Type**	**FTS**	**Requires Moving Parts**	**Data Collection Method**	**Spectral Range (microns)**	**Spectral Resolution (cm^{-1})**	**Collection Time**
FIRST	Michelson	Yes	Yes	Temporal	7.7 - 12.0	0.125 - 150	2s at $4cm^{-1}$
LIFTIRS	Michelson	Yes	Yes	Temporal	8.0 - 12.0	0.125 - 150	5s at $4cm^{-1}$
THI	Sagnac	Yes	No	Spatial	7.5 - 12.5	16	2s at $16cm^{-1}$
AIRIS	Fabry-Perot	No	Yes	Temporal	8.0 - 11.0	10	9s at $10cm^{-1}$
SHIMCAD	SHS	Yes	No	Spatial	8.4 - 11.5	2	0.16s at $2cm^{-1}$

Appendix B

MATLAB Code

There are three SHS system folders: one for generating SHS results, one for generating system characterization results, and one for generating design optimization results (see Figure B.1). Each folder contains the necessary programs and functions and a SPECTRAL DATA folder containing the spectral radiance and transmittance data presented in Section 7 (see Figure B.2). The user puts the spectral data into this folder and runs the desired program.



« caraperkins ▸ Desktop ▸ ML WORK ▸ Thesis ▸

SHS
SHS Characterization
SHS Optimization

Figure B.1: There are three system folders. SHS contains code to generate interferometric results. SHS Characterization contains code to generate system characterization results. SHS Optimization contains code to generate design optimization results.

B.1 All Folders

All three SHS system folders contain several identical functions.

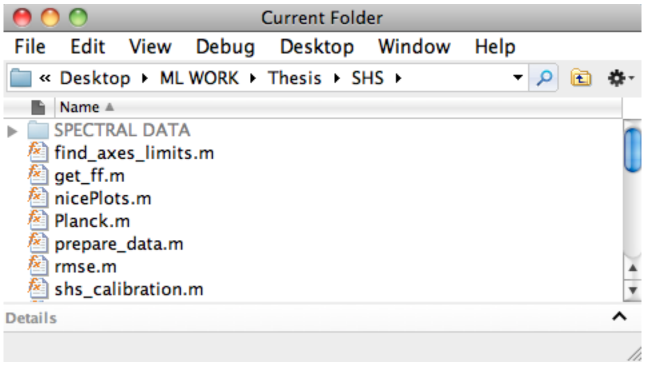


Figure B.2: Each of the three system folders contains the necessary functions and programs needed to generate the results and a SPECTRAL DATA folder containing the spectral radiance and transmittance data presented in Section 7.

B.1.1 find_axes_limits.m

```
function [ output ] = find_axes_limits( input1, input2 )

% This function finds the minima and maxima of two inputs so that we can
% plot two similar functions on the same-sized axes for comparison.

output.min = min ( min(input1), min(input2) );
output.max = max ( max(input1), max(input2) );

end
```

B.1.2 get_ff.m

```
function [ output ] = get_ff( shs )

% This function calculates the flat-field correction function. The known
% source is the built-in hot blackbody so that this correction may be
% performed on a measurement-by-measurement basis.
%
% The function input is the "shs" struct. The ouput is the x-dependent
% correction factor.

bb_h = Planck(shs.wavenumber, shs.hot_bb);
bb_h = 1/2*[fliplr(bb_h(1,2:end-1)) bb_h];
bb_h = shs_entrance_optics(bb_h, shs);
```

```
[ia ib] = shs_ff_interferogram(bb_h, shs);
ia = shs_exit_optics(ia, shs);
ib = shs_exit_optics(ib, shs);
% Interpolate the high-resolution interferogram at the sampling frequency:
ia = interp1(shs.x, ia, shs.meas_x);
ib = interp1(shs.x, ib, shs.meas_x);
% Multiply the interferogram by the detector sensitivity function:
ia = ia.*shs.cp;
ib = ib.*shs.cp;
ia = shs_shift(ia, shs);
ib = shs_shift(ib, shs);
ia = 1/2*(ia+ib);
output = ia / max(max(ia));

end
```

B.1.3 nicePlots.m

```
function nicePlots( )

% This function changes some of the default settings for MATLAB plots to
% automatically make nice-looking plots.

set(0, 'DefaultAxesFontSize', 16,...
'DefaultLineLineWidth', 1,...
'DefaultTextFontSize', 14,...
'DefaultAxesFontWeight', 'bold',...
'DefaultAxesColor',[1 1 1],...
'DefaultFigureColor', 'w',...
'DefaultFigurePaperType', 'a4letter',...
'DefaultAxesDrawmode', 'fast',...
'DefaultAxesFontUnits', 'points',...
'DefaultAxesFontName', 'Times',...
'DefaultAxesGridLineStyle', ':',...
'DefaultAxesUnits', 'normalized',...
'DefaultLineColor', 'Red',...
'DefaultTextFontUnits', 'Points',...
'DefaultTextFontName', 'Times');

end
```

B.1.4 Planck.m

```
function [ L ] = Planck( nu, T, units )
```

```
% This function calculates the blackbody spectrum for a given temperature
% and range of wavenumber values using the Planck equation.

% nu - (1/cm)
% output - (W/m^2/sr/cm)

h = 6.26*10^(-34); % Planck's constant (Js)
c = 3*10^8; % speed of light (m/s)
k = 1.38*10^(-23); % Boltzman constant (J/K)

% blackbody radiance:
L = 2*10^8*h*c^2 * ((nu).^3) .* (exp(h*c*(nu*100).*(k*T).^(-1)) - 1).^(-1);

if nargin == 3
    if strcmp(units, 'cm2') == 1
        L = L * 1/10000; % W/cm^2/sr/cm
    elseif strcmp(units, 'um') == 1
        L = L * 1/10000; % W/m^2/sr/um
    end
end

end
```

B.1.5 prepare_data.m

```
function [spectrum] = prepare_data(file, shs)

% This function prepares the spectral data in "file" to be run through the
% model. It outputs the prepared sample spectrum.

% Load the input file and check the orientation of the data arrays:
load(file, 'wavenumber', 'spectrum')
[M N] = size(wavenumber);
if M > 1
    wavenumber = wavenumber';
    spectrum = spectrum';
end

% Interpolate the spectral information for the system's wavenumber array:
wave_new = shs.kmin:(shs.kmax-shs.kmin)*2/(shs.N):shs.kmax;
spectrum = interp1(wavenumber, spectrum, wave_new);
wavenumber = wave_new;

% Eliminate the data outside of our system spectral range:
ind = find(wavenumber >= 950 & wavenumber <= 1260);
wavenumber = wavenumber(ind);
spectrum = spectrum(ind);
```

```
end
```

B.1.6 rmse.m

```
function [ output ] = rmse( A, B )

% This function calculates the RMSE of two inputs and prints it.

% Calculate their RMSE:
N = numel(A);
output = sqrt( sum(sum((A-B).^2)) / N );

% Print the answer:
disp('RMSE = ');
disp(output);

end
```

B.1.7 shs_detector.m

```
function [ output shs ] = shs_detector( input, shs)

% This function calculates te output of the Detector block. The input is
% the high-resolution interferogram and the output is the interferogram
% sampled by the MCT detector.

% Interpolate the high-resolution interferogram at the sampling frequency:
output = interp1(shs.x, input, shs.meas_x);

% Multiply the interferogram by the detector sensitivity function:
output = output.*shs.cp;

[output shs] = shs_noise_model(output, shs);

end
```

B.1.8 shs_entrance_optics.m

```
function [ output ] = shs_entrance_optics( input, shs )

% This function gives the output of the SHS entrance optics. The inputs are
% the input sample radiance and "shs".
% Get total optical effects:
shs.optics1 = shs.transmission1*shs.FOV;

% Calculate the spectrum after the Entrance Optics:
output = input.*shs.optics1 + 1/2*[fliplr(shs.self_emission1(1,2:end-1)) ...
    shs.self_emission1];

end
```

B.1.9 shs_exit_optics.m

```
function [ output ] = shs_exit_optics( input, shs )

% This function calculates the output of the Exit Optics, where the input
% is the high-resolution interferogram.

output = (input + sum(sum(1/2*shs.dk*shs.FOV*shs.self_emission2)))...
    .*mean(shs.transmission2);

end
```

B.1.10 shs_ff_interferogram.m

```
function [ ia ib ] = shs_ff_interferogram( bb_h, shs )

% This function calculates the two measurements used to calculate the
% flat-field correction function.
%
% The input is the known source and the "shs" struct. The outputs are the
% measurments with arm a or b blocked.

% Create a double-sided differential wavenumber array:
shs.kk = [-fliplr(shs.k(1,2:end-1)) shs.k];
%
% Calculate the "interferogram". Really, this is a measurement with one arm
% blocked, so there is no modulation (i.e. it is not an interferogram).:
ia = sum(sum(bb_h.*shs.Nga*shs.dk*(2*tand(shs.ltheta))...
    .*shs.kk))*ones(size(shs.kk));
ib = sum(sum(bb_h.*shs.Ngb*shs.dk*(2*tand(shs.ltheta))...
    .*shs.kk))*ones(size(shs.kk));

end
```

B.1.11 shs_model.m

```
function [ interferogram, shs ] = shs_model( input, shs )

% This function runs the input radiance through the SHS model. The model
% has blocks: Entrance Optics, Interferometer, Exit Optics, and Detector.
%
% The input is the sample spectral radiance and "shs" struct. The outputs
% are an updated version of "shs" and the interferogram measured by the SHS.


% Generate double-sided spectral data (1/2 term is due to the fact that
% half the radiance exits the interferometer through the entrance aperature
% after recombining at the beam-splitter).

input = 1/2*[fliplr(input(1,2:end-1)) input];



% 1. ENTRANCE OPTICS

spectrum = shs_entrance_optics(input, shs);
% ********************************************



% 2. INTERFEROMETER:

interferogram = shs_interferometer(spectrum, shs);
% ********************************************



% 3. EXIT OPTICS:

% generate a high-resolution interferogram:
interferogram = shs_exit_optics(interferogram, shs);
% ********************************************



% 4. DETECTOR:

% sampled interferogram:
[interferogram shs] = shs_detector(interferogram, shs);
% ********************************************



end
```

B.1.12 shs_quantization.m

```
function [ output ] = shs_quantization( input, shs )

% This function quantizes the measured uncalibrated radiance for a number
% of bits and saturation level in spectral radiance.

% Determine the quantization levels:
dL = shs.Lmax/(2^shs.bits - 1);
levels = dL:dL:shs.Lmax;
N = numel(levels);
M = numel(input);
k=0;

% Quantize:
for i = 1:N
    if i == 1
        ind = find(input <= levels(i));
        input(ind) = levels(i);
    elseif i == N
        ind = find(input > levels(i-1));
        input(ind) = levels(i);
    else
        ind = find(input > levels(i-1) & input <= levels(i));
        input(ind) = levels(i);
    end
    if isempty(ind) == 0
        k = k + numel(ind);
        if k == M
            break
        end
    end
end
   output = input;
end
```

B.2 SHS

The SHS folder contains two programs, shs_system.m for generating spectral radiance results and shs_methanol.m for generating methanol transmittance results to compare with those from SHIMCAD, and the necessary functions.

B.2.1 shs_system.m

```
% This is a program to run spectral data through the SHS model and
% interferogram processing algorithm.
%
% The user must define the SHS operational parameters in Step 1.
%
% The spectral data must be specified in Step 5. of the program. The
% data should be saved in a ".mat" file. The ".mat" file should contain one
% variable called "spectrum" with the spectral data and another called
% "wavenumber" with the corresponding wavenumber values. Currently, the
% units are set to spectral radiance values of [W/m^2/sr/cm^-1] and
% wavenumber values of [cm^-1].


% THIS MODEL INCLUDES THE GRATING ANGLE ERROR.



clear all
close all
clc
nicePlots;



% 1. **USER-DEFINED** SHS OPERATIONAL PARAMETERS:

% wavenumber range:
shs.kmin = 950; % (1/cm)
shs.kmax = 1250; % (1/cm) --> Littrow wavenumber
%
% diffraction grating characteristics:
shs.m = 1; % order number
shs.d = 1/143; % grating groove separation (cm)

% number of interferogram samples:
shs.meas_N = 128;
% *******************************************



% 2. PARAMETERS CALCULATED FROM OPERATIONAL PARAMETERS:

% Littrow settings of the gratings:
shs.lk = shs.kmax; % Littrow wavenumber (1/cm)
shs.ltheta = asind( shs.m / (2*shs.d*shs.lk)); % Littrow angle (degrees)

% number of high-resolution samples:
shs.N = 2^(nextpow2(2*(shs.kmax-shs.kmin)));
```

```
% grating width:
shs.w = shs.meas_N / (2*4*(shs.kmax - shs.kmin)*sind(shs.ltheta)); % (cm)

% maximum position, x, along the detector:
shs.xmax = 1/2*shs.w*cosd(shs.ltheta); % (cm)

% high-res interferogram sample spacing:
shs.dx = 2*shs.xmax / shs.N; % (cm)

% high-res spectral resolution:
shs.dk = 2*(shs.kmax-shs.kmin) / shs.N; % (1/cm)

% high-res position x and difference wavenumber (k - l.k) arrays:
shs.x = -shs.xmax:shs.dx:shs.xmax-shs.dx; % (cm)
shs.k = 0:shs.dk:shs.kmax-shs.kmin; % (1/cm)

% high-res spectral range array:
shs.wavenumber = shs.kmin:(shs.kmax-shs.kmin)*2/(shs.N):shs.kmax;
% ********************************************



% 3. DETECTION PARAMETERS:

% spectral resolution:
shs.meas_dk = 1 / (2*4*tand(shs.ltheta)*shs.xmax); % (1/cm)

% interferogram sample spacing:
shs.meas_dx = 1 / (4*tand(shs.ltheta)*shs.meas_dk*shs.meas_N); % (cm)

% position x and difference wavenumber (k - l.k) arrays
shs.meas_x = -shs.xmax:shs.meas_dx:shs.xmax-shs.meas_dx; % (cm)
shs.meas_k = 0:shs.meas_dk:(shs.kmax-shs.kmin-shs.meas_dk); % (1/cm)

% measured spectral range array:
shs.meas_wavenumber = shs.kmin:(shs.kmax-shs.kmin)*2/(shs.meas_N):(shs.kmax-shs.meas_dk);
% ********************************************



% 4. CALIBRATION MEASUREMENTS:

% First, get the full SHS model description:
shs = shs_model_description(shs);
% Perform flat-fielding measurements:
shs.ff = get_ff(shs);
% Calculate system gain and responsivity:
[shs.r shs.g] = shs_calibration(shs);
% ********************************************
% 5. RUN DATA THROUGH SYSTEM:

% Get the SHS Spectral Data folder directory.
toolName = 'SHS';
toolDirectory = mfilename('fullpath');
toolDirectory(strfind(toolDirectory, toolName) + length(toolName) :  end) = [];
dataDirectory = fullfile(toolDirectory, 'SPECTRAL DATA');
addpath(genpath(dataDirectory));
data = what(dataDirectory);
num_files = numel(data.mat);
% *******************************************


for i = 1:num_files



    % a. PREPARE DATA FOR SHS MODEL:
    char(data.mat(i)) % prints the file name
    spectrum = prepare_data(char(data.mat(i)), shs);
    % *******************************************



    % b. GENERATE THE INTERFEROGRAM:

    [interferogram shs] = shs_model(spectrum, shs);
    % *******************************************



    % c. CONVERT THE RAW INTERFEROGRAM TO CALIBRATED SPECTRAL RADIANCE:

    % Convert back to the radiance domain:
    [converted shs] = shs_conversion(interferogram, shs);
    % Quantize:
    recovered = shs_quantization(converted, shs);
    % *******************************************



    % d. CALCULATE NOISE FIGURES OF MERIT:

    SNR = recovered ./ shs.NESR;
    NEDT = shs.NEDT;
    % *******************************************
    % e. PLOT RESULTS:

    % Determine axes limits:
    spectral_limits = find_axes_limits(spectrum, recovered);
    spatial_limits = find_axes_limits(interferogram, shs.shifted);

    % i. Input radiace
    figure; subplot(2,3,1), plot(shs.wavenumber, spectrum);
    xlabel('Wavenumber (1/cm)');
    ylabel('Radiance (W/m^2/sr/cm^-1)');
    title('Sample Spectral Radiance');
    axis([shs.kmin shs.kmax spectral_limits.min spectral_limits.max]);

    % ii. Measured interferogram:
    subplot(2,3,2), plot(shs.meas_x, interferogram);
    xlabel('x (cm)');
    ylabel('Irradiance (W/m^2)');
    title('Measured SHS Interferogram');
    axis([-shs.xmax shs.xmax spatial_limits.min spatial_limits.max]);

    % iii. Centered interferogram:
    subplot(2,3,4), plot(shs.meas_x, shs.shifted);
    xlabel('x (cm)');
    ylabel('Irradiance (W/m^2)');
    title('Shift-Corrected Interferogram');
    axis([-shs.xmax shs.xmax spatial_limits.min spatial_limits.max]);

    % iv. Calibrated recovered radiance:
    subplot(2,3,5), plot(shs.meas_wavenumber, recovered);
    xlabel('Wavenumber (1/cm)');
    ylabel('Radiance (W/m^2/sr/cm^-1)');
    title('Calibrated Spectral Radiance');
    axis([shs.kmin shs.kmax spectral_limits.min spectral_limits.max]);

    % v. SNR
    subplot(2,3,3), plot(shs.meas_wavenumber, SNR);
    xlabel('Wavenumber (1/cm)');
    ylabel('SNR');
    title('Spectral Signal-to-noise Ratio');
    axis([shs.kmin shs.kmax min(SNR) max(SNR)]);

    % vi. NEdT
    subplot(2,3,6), plot(shs.meas_wavenumber, shs.NEDT);
    xlabel('Wavenumber (1/cm)');
    ylabel('NEdT (K)');
    title('Noise-equivalent Change in Temperature');
    axis([shs.kmin shs.kmax min(shs.NEDT) max(shs.NEDT)]);
    % *******************************************
```

```
end
```

B.2.2 shs_methanol.m

```
% This is a program to run methanol transmittance data through the
% SHS model and interferogram processing algorithm so that we can compare
% our results with the SHIMCAD results [1].
%
% The user must define the SHS operational parameters in Step 1.
%
% The spectral data must be specified in Step 5. of the program. The
% data should be saved in a ".mat" file. The ".mat" file should contain one
% variable called "transmission" with the spectral data and another called
% "wavenumber" with the corresponding wavenumber values. Currently, the
% units are set to wavenumber values of [cm^-1].


% THIS MODEL INCLUDES THE GRATING ANGLE ERROR.



clear all
close all
clc
nicePlots;



% 1. **USER-DEFINED** SHS OPERATIONAL PARAMETERS:

% wavenumber range:
shs.kmin = 950; % (1/cm)
shs.kmax = 1250; % (1/cm) --> Littrow wavenumber
%
% diffraction grating characterisitics:
shs.m = 1; % order number
shs.d = 1/143; % grating groove separation (cm)

% number of interferogram samples:
shs.meas_N = 256;
% ******************************************



% 2. PARAMETERS CALCULATED FROM OPERATIONAL PARAMETERS:
```

```
% Littrow settings of the gratings:
shs.lk = shs.kmax; % Littrow wavenumber (1/cm)
shs.ltheta = asind( shs.m / (2*shs.d*shs.lk)); % Littrow angle (degrees)

% number of high-resolution samples:
shs.N = 2^(nextpow2(2*(shs.kmax-shs.kmin)));

% grating width:
shs.w = shs.meas_N / (2*4*(shs.kmax - shs.kmin)*sind(shs.ltheta)); % (cm)

% maximum position, x, along the detector:
shs.xmax = 1/2*shs.w*cosd(shs.ltheta); % (cm)

% high-res interferogram sample spacing:
shs.dx = 2*shs.xmax / shs.N; % (cm)

% high-res spectral resolution:
shs.dk = 2*(shs.kmax-shs.kmin) / shs.N; % (1/cm)

% high-res position x and difference wavenumber (k - l.k) arrays:
shs.x = -shs.xmax:shs.dx:shs.xmax-shs.dx; % (cm)
shs.k = 0:shs.dk:shs.kmax-shs.kmin; % (1/cm)

% high-res spectral range array:
shs.wavenumber = shs.kmin:(shs.kmax-shs.kmin)*2/(shs.N):shs.kmax;
% *******************************************



% 3. DETECTION PARAMETERS:

% spectral resolution:
shs.meas_dk = 1 / (2*4*tand(shs.ltheta)*shs.xmax); % (1/cm)

% interferogram sample spacing:
shs.meas_dx = 1 / (4*tand(shs.ltheta)*shs.meas_dk*shs.meas_N); % (cm)

% position x and difference wavenumber (k - l.k) arrays
shs.meas_x = -shs.xmax:shs.meas_dx:shs.xmax-shs.meas_dx; % (cm)
shs.meas_k = 0:shs.meas_dk:(shs.kmax-shs.kmin-shs.meas_dk); % (1/cm)

% measured spectral range array:
shs.meas_wavenumber = shs.kmin:(shs.kmax-shs.kmin)*2/(shs.meas_N):(shs.kmax-shs.meas_dk);
% *******************************************


% 4. CALIBRATION MEASUREMENTS:

% First, get the full SHS model description:
```

```
shs = shs_model_description(shs);
% Perform flat−fielding measurements:
shs.ff = get_ff(shs);
% Calculate system gain and responsivity:
[shs.r shs.g] = shs_calibration(shs);
% ******************************************



% 5. RUN DATA THROUGH SYSTEM:

% Get the SHS Spectral Data folder directory.
toolName = 'SHS';
toolDirectory = mfilename('fullpath');
toolDirectory(strfind(toolDirectory, toolName) + length(toolName) :  end) = [];
dataDirectory = fullfile(toolDirectory, 'SPECTRAL DATA');
addpath(genpath(dataDirectory));
data = what(dataDirectory);
num_files = numel(data.mat);
% ******************************************





% a. PREPARE DATA FOR SHS MODEL:

char(data.mat(1)) % prints the file name
transmission = prepare_data(char(data.mat(1)), shs);
shs.blackbody1 = Planck(shs.wavenumber, 373.15);
shs.blackbody2 = Planck(shs.meas_wavenumber, 373.15);
spectrum = transmission.*shs.blackbody1; % places the gas sample in front
                                         % of a blackbody
% ******************************************



% b. GENERATE THE INTERFEROGRAM:

[interferogram shs] = shs_model(spectrum, shs);
% ******************************************



% c. CONVERT THE RAW INTERFEROGRAM TO CALIBRATED SPECTRAL RADIANCE:

% Convert back to the radiance domain:
[converted shs] = shs_conversion(interferogram, shs);
% Quantize:
recovered = shs_quantization(converted, shs);
```

```
recovered_t = recovered ./ shs.blackbody2;
% *******************************************



% d. CALCULATE NOISE FIGURES OF MERIT:

SNR = recovered ./ shs.NESR;
NEDT = shs.NEDT;
% *******************************************



% e. PLOT RESULTS:

% Determine axes limits:
spectral_limits = find_axes_limits(transmission, recovered_t);
spatial_limits = find_axes_limits(interferogram, shs.shifted);

% i. Input transmittance:
figure; subplot(2,3,1), plot(shs.wavenumber, transmission);
xlabel('Wavenumber (1/cm)');
ylabel('Transmittance');
title('Sample Transmittance');
axis([shs.kmin shs.kmax spectral_limits.min spectral_limits.max]);

% ii. Measured interferogram:
subplot(2,3,2), plot(shs.meas_x, interferogram);
xlabel('x (cm)');
ylabel('Irradiance (W/m^2)');
title('Measured SHS Interferogram');
axis([−shs.xmax shs.xmax spatial_limits.min spatial_limits.max]);

% iii. Shifted interferogram
subplot(2,3,4), plot(shs.meas_x, shs.shifted);
xlabel('x (cm)');
ylabel('Irradiance (W/m^2)');
title('Shift−Corrected Interferogram');
axis([−shs.xmax shs.xmax spatial_limits.min spatial_limits.max]);

% iv. Calibrated recovered transmittance
subplot(2,3,5), plot(shs.meas_wavenumber, recovered_t);
xlabel('Wavenumber (1/cm)');
ylabel('Transmittance');
title('Calibrated Transmittance');
axis([shs.kmin shs.kmax spectral_limits.min spectral_limits.max]);

% v. SNR
subplot(2,3,3), plot(shs.meas_wavenumber, SNR);
xlabel('Wavenumber (1/cm)');
```

```
ylabel('SNR');
title('Spectral Signal-to-noise Ratio');
axis([shs.kmin shs.kmax min(SNR) max(SNR)]);

% vi. NEdT
subplot(2,3,6), plot(shs.meas_wavenumber, shs.NEDT);
xlabel('Wavenumber (1/cm)');
ylabel('NEdT (K)');
title('Noise-equivalent Change in Temperature');
axis([shs.kmin shs.kmax min(shs.NEDT) max(shs.NEDT)]);
% ********************************************

% [1] C. R. Englert, D. D. Babcock, and J. M. Harlander, "Spatial heterodyne
% spectroscopy for long-wave infrared: first measurements of broadband
% spectra," SPIE, vol. 48(10), October 2009.
```

B.2.3 shs_calibration.m

```
function [ r g ] = shs_calibration( shs )

% This function calculates the system responsivity fucntion (not to be
% confused with the detector responsivity term) and gain function to use
% for spectral radiance calibration, later. The built-in calibrating
% blackbodies are used to perform the calculations.
%
% The input is the "shs" struct and the outputs are the system responsivity
% (r) and gain (g) functions.


bb_h = shs.emissivity.*Planck(shs.wavenumber, shs.hot_bb);
bb_c = shs.emissivity.*Planck(shs.wavenumber, shs.cold_bb);
bb_h_i = shs_model(bb_h, shs);
bb_c_i = shs_model(bb_c, shs);

shs.r = 1;
shs.g = 0;
bb_h_rec = shs_conversion(bb_h_i, shs);
bb_c_rec = shs_conversion(bb_c_i, shs);
bb_h_rec = shs_quantization(bb_h_rec, shs);
bb_c_rec = shs_quantization(bb_c_rec, shs);

bb_h = interp1(shs.wavenumber, bb_h, shs.meas_wavenumber);
bb_c = interp1(shs.wavenumber, bb_c, shs.meas_wavenumber);

clearvars shs.r shs.g
r = (bb_h_rec-bb_c_rec)./(bb_h-bb_c);
g = (bb_h_rec./r - bb_h);
```

```
end
```

B.2.4 shs_conversion.m

```
function [ output shs] = shs_conversion( input, shs )

% This function converts the measured interferogram values back into
% spectral radiance values. Half the recovered signal is thrown away
% because the recovered spectrum is actual a mirrored image of the input
% spectrum. The output is not yet calibrated.

N = numel(input);

% 1. FLAT-FIELDING:
input = input./shs.ff;
% ******************************************



% 2. SHIFT CORRECTION:

input = shs_shift(input, shs);
shs.shifted = input;
% ******************************************



% 3. BIAS REMOVAL:

DC = mean(input(1,1:round(N/4)) + input(1,end-round(N/4)+1:end))/2;
input = input - ones(size(input))*DC;
% ******************************************



% 4. APODIZATION:

m = N;                          % define the width of the apod. window
q = round(N/2) - round(m/2);    % determine the necessary window padding
input = (padarray(hamming(m),q)'.*input);
% ******************************************



% 5. IFT:

spectrum = fliplr(1/shs.meas_dk/shs.FOV*2*(abs(ifftshift(ifft(ifftshift...
    (input))))));
```

```
output = spectrum(1, round(N/2)+1:end);
% *******************************************



% 6. CALIBRATION:

output = (output./shs.r - shs.g);
% *******************************************



end
```

B.2.5 shs_interferometer.m

```
function [ output ] = shs_interferometer( input, shs )

% This function calculates the high-resolution interferogram from the
% output of the Entrance Optics.

% Count the number of elements
num_input = numel(input);

% Determine the radiance incident at the detector (after the split beams
% interact with their respective diffraction grating).
input = 1/2*input.*shs.Nga + 1/2*input.*shs.Ngb;

% Calculate any Littrow angle error and the corresponding difference
% wavenumber array:
angle_error = shs.littrow_error.*randn(1,shs.num_samples);
k_error = shs.m/shs.d ./ (2*sind(shs.ltheta+angle_error));

% Calculate the position, x, and phase error.
x_e = shs.vib_x*randn(1,shs.num_samples);
p_e = shs.phase_error*randn(1,shs.num_samples);

% THE LITTROW ANGLE, x, AND PHASE ERRORS ARE RANDOMIZED FOR EACH SAMPLE
% MEASUREMENT TO MODEL SYSTEM VIBRATIONS.

% Calculate the interferogram averaged over all samples.
for j = 1:shs.num_samples

    shs.kk_error = shs.kk + ones(size(shs.kk)).*(shs.lk-k_error(1,j));
    fizeau_freq = 2*shs.kk*tand(shs.ltheta)+2*shs.kk_error*...
    tand(shs.ltheta+angle_error(1,j));

    for i = 1:num_input
```

```
        I = shs.dk*input.*(1 + ...
            sqrt(shs.Nga.*shs.Ngb).*cos(2*pi*fizeau_freq.*(shs.x(1,i) - ...
            x_e(1,j) - shs.shift*shs.dx)...
            - p_e(1,j)));...
        output(j,i) = (sum(I));
    end

end

output = mean(output,1);

end
```

B.2.6 shs_model_description.m

```
function [ shs ] = shs_model_description( shs )

% This is a complete description of the SHS model. All necessary parameters
% are defined, here. This includes constants, system error rates,
% parameters describing the Entrance Optics, Interferometer, Exit Optics,
% and Detection blocks, and the built-in calibrating blackbodies'
% description.
%
% The function input is the struct called "shs". shs already has the field
% parameters defined by the user and calculated by the model to describe
% the SHS basics. The output is the same struct, "shs", but with additional
% parameters to create a complete description.

% Constants:
shs.q = 1.602*10^(-19); % Coulomb (C = A/s)
shs.h = 6.26*10^(-34);  % Planck's constant (Js)
shs.c = 3*10^(10);      % speed of light (cm/s)
shs.kb = 1.38*10^(-23); % Boltzman constant (J/K)

% System Error Rates:
shs.cal_e = 10^(-6);         % calibration error rate
shs.bit_e = 10^(-9);        % bit error rate
shs.vib_x = 10^(-6);        % position, x, error rate (cm)
shs.littrow_error = 0.001;  % Littrow angle error rate (degrees)
shs.shift = 4;              % center-burst phase shift (number of dx's)
shs.phase_error = 0.006;   % k-dependent phase error rate (radians)

% Create a double-sided differential wavenumber array:
shs.kk = [-fliplr(shs.k) shs.k(1,2:end-1)];

% Entrance Optics:
shs.transmission1 = 0.85*exp(-(... % entrance transmission function
    (shs.kk+shs.kmax-shs.kmin)...
    /(2*shs.kmax-shs.kmin)).^2);
shs.T1 = 290;                      % entrance optics temperature (K)
shs.self_emission1 = Planck(...    % entrance emission (W/m^2/sr/cm^-1)
    shs.wavenumber, shs.T1);
shs.f_num = 2.4;                   % system f/#
shs.FOV = pi/(4*shs.f_num^2+1);    % system FOV (radians)

% Interferometer:
shs.Nga = 0.8 + 0.01*cos(0.2*pi*shs.kk);  % grating efficiency in arm, a
shs.Ngb = 0.85 + 0.01*cos(0.2*pi*shs.kk); % grating efficiency in arm, b
shs.num_samples = 100;                    % number of sample averages

% Exit Optics:
shs.transmission2 = 0.8*exp(-(... % exit transmission function
    (shs.kk+shs.kmax-shs.kmin)...
    /(8*shs.kmax-shs.kmin)).^2);
shs.T2 = 280;                     % exit optics temperature (K)
shs.self_emission2 = Planck(...
    shs.meas_wavenumber, shs.T2); % exit emission (W/m^2/sr/cm^-1)

% Detector (assumed to be MCT):

shs.tot_eff = mean(...        % total system efficiency
    shs.transmission1.*shs.transmission2.*...
    (shs.Nga+shs.Ngb)/2);
shs.spectral_bandwidth = ...  % spectral bandwidth (cm^-1)
    shs.meas_dk;
shs.resolving_power = 4*...   % Resolving power derived by Harlander [1]
    shs.w*shs.lk*sind(shs.ltheta);
shs.FOVmax = 2*pi/(4*shs.w*...% Limiting FOV derived by Harlander [1]
    shs.lk*sind(shs.ltheta));

shs.pixel_pitch = 40*10^(-6); % pixel pitch (m)
shs.pixel_array = ...         % detector width (cm)
    shs.meas_N*shs.pixel_pitch*10^2;
shs.Adu = shs.pixel_pitch^2;  % detector element area (m^2)
shs.Aa = 60*21/10^6;          % aperature area (m^2)
shs.tint = 1/2000;            % integration time (s)
shs.frame_rate = 114.9;       % frame rate (frame / sec)
shs.QE = 0.70*exp(-(...       % quantum efficiency
    1-shs.meas_wavenumber/shs.lk)/2);
shs.Dstar = 2*10^10;          % D* (cm*sqrt(Hz)/W)
shs.R = 100;                  % Responsivity (V/W)
shs.NPSD = 91.6*10^(-9);      % Noise power spectral density (V/sqrt(Hz))
shs.bits = 17;                % number of bits

shs.p0 = 0.999;               % peak FPA sensitivity
shs.pc = 1/0.15*shs.pixel_array;% width of sensitivity Gaussian
shs.pe = 2;                   % exponential coefficient
shs.cp = shs.p0*exp(-(abs...  % sensitivity across FPA
    (shs.meas_x*1/shs.pc)...
    .^shs.pe)) + 0.001*cos(shs.meas_x*200*pi)...
    .*sin(shs.meas_x*300*pi);

shs.C1 = ...                  % conversion factor for spectral radiance to
    (shs.spectral_bandwidth...% number of electrons
        *shs.Adu*shs.FOV*shs.tint*shs.QE...
    ./(shs.h*shs.c*shs.meas_wavenumber));
shs.C2 = shs.Adu.*shs.tint... % conversion factor for irradiance to # e-
    ./(shs.h.*shs.c.*mean(shs.meas_wavenumber))*mean(shs.QE);
shs.C3 = sqrt(2/shs.meas_N);% conversion factor from k noise to x noise

shs.saturation = 32*10^6;     % saturation level (e-)
shs.Lmax = shs.saturation...  % saturation level (W/m^2/sr/cm^-1)
    ./shs.C1;
shs.throughput = shs.Aa*shs...% optical throughput (sr*cm^2)
    .FOVmax*10^4;
% ********************************************

% Calibrating Blackbodies:
shs.hot_bb = 380;       % hot blackbody temperature (K)
shs.cold_bb = 290;      % cold blackbody temperature (K)
shs.emissivity = 0.994;  % emissivity of blackbodies
% ********************************************

end

% [1] J. M. Harlander, "Spatial heterdyne spectroscopy: Interferometric
%      performance at any wavelength without scanning," Ph.D.
%      dissertation, The University of Wisconsin - Madison, Madison, WI,
%      1991.
```

B.2.7 shs_noise_model.m

```
function [ output shs ] = shs_noise_model( input, shs )

% This function calculates the various noise sources and adds them to the
% measured radiance spectrum and outputs the noisy signal and new "shs"
% terms that describe the noise figures of merit.

new_N = numel(input);

% Convert the measured sample radiance to #e-:
signal_e = input.*shs.C2;

% Noise sources in #e- (standard deviations):

% IN THE INTERFEROGRAM DOMAIN:
```

```
% Shot noise:
shot_noise = mean(sqrt(signal_e));

% Calibration noise:
calibration_noise = shs.cal_e*(shot_noise).^2;

% IN THE SPECTRAL DOMAIN:

% Detector noise:
det_noise = mean(2/(shs.Aa*shs.FOV*shs.meas_dk*shs.Dstar)*...
    sqrt(shs.Adu*10^4*shs.frame_rate)/shs.tot_eff.*shs.C1);

% Electronics noise:
electronics_noise = mean(4*shs.xmax*shs.NPSD*...
    sqrt(shs.frame_rate)/(shs.Aa*shs.FOV*shs.R)/shs.tot_eff.*shs.C1);

% Quantization noise:
quantization_noise = mean(shs.Lmax/(sqrt(12)*2^(shs.bits) - 1).*shs.C1);

% Bit error noise:
bit_noise = 0;
for q = 0:shs.bits-1;
    bit_noise = bit_noise + (2^q*(shs.Lmax)./(2^shs.bits - 1)).^2;
end
bit_noise = mean(sqrt(shs.bit_e/shs.bits*bit_noise).*shs.C1);


% Calculate total noise in #e- and average over all samples:
for i = 1:shs.num_samples
    noise1(i,:) = (randn(1,new_N).*shot_noise) +...
        (randn(1,new_N).*calibration_noise);
    noise2(i,:) = (randn(1,new_N).*det_noise +...
        randn(1,new_N).*electronics_noise +...
        randn(1,new_N).*quantization_noise +...
        randn(1,new_N).*bit_noise)./shs.C3;
end
% Convert spectral noise to interferogram domain and add:
noise = mean(noise1,1) + mean(noise2,1);

% Add noise and convert back to interferogram values:
output = (signal_e + noise)./shs.C2;


% Calculate noise-equivalent change in temperature:

% First, get the total noise standard deviation (#e-):
total_noise = sqrt((shot_noise*shs.C3).^2 + (det_noise).^2 +...
    (electronics_noise).^2 +...
    (quantization_noise).^2 + (calibration_noise*shs.C3).^2 ...
    + (bit_noise).^2);


% Then, get the noise-equivalent spectral radiance (W/cm&2/sr/cm^-1):
shs.NESR = (total_noise).*shs.C1.^(-1);
avg_k = shs.meas_wavenumber;
avg_T = 300;

% Get the derivative of the Planck equation for a given temperature:
dLBdT = 100^2*(2*shs.h^2*shs.c^3*avg_k.^4.*exp(shs.h*shs.c*avg_k./...
    (shs.kb*avg_T)))/(shs.kb*avg_T^2*(exp(shs.h*shs.c.*avg_k./...
    (shs.kb*avg_T)) - 1).^2); % derivative of Planck eq. w.r.t T

shs.NEDT = (shs.NESR./dLBdT); % noise-equivalent change in temperature (K)

end
```

B.2.8 shs_shift.m

```
function [ output ] = shs_shift( input, shs )

% This function performs shift-correction on the measured interferogram.
% The center-burst of the measured interferogram is usually not perfectly
% aligned with the x=0cm position on the detector. This function shifts it
% such that the center-burst IS at x=0cm.
%
% The input is the interferogram sampled by the detector and the output is
% the shift-corrected interferogram.

N = numel(input);

% Get a good first estimate of the peak location:
ind = find( abs(input) == max(max(abs(input))) );
shift = ind - round(N/2) - 2;

% Shift by the convolution method to get a good first estimate:

% dd = zeros(size(input));
% dd(1, round(N/2) - shift) = 1;
% I = ((fft((input))));
% DD = ((fft((dd))));
% input = (ifftshift(ifft((I.*DD))));


% Shift by the FT method derived by Ben-david et al. [1] for better
% precision:
% ind = find( abs(input) == max(max(abs(input))) );
% shift = ind - round(N/2) - 1;
```

```
i = 1:N;
delta = (sum(i.*abs(input))./sum(abs(input)) - shift - round(N/2));

I = fft(input);
W = exp(-1i*2*pi*delta*[0:floor(N/2)-1 floor(-N/2):-1]/N);
if mod(N, 2) == 0
    W(N/2+1) = real(W(N/2+1));
end
Y = I .* W;

% Invert the FFT.
input = ifft(Y);

% Take magnitude.
output = real(input);

end

% [1] A. Ben-David and A. Ifarraguerri, "Computation of a spectrum from a
% single-beam fourier-transform infared interferogram," Optical Society of
% America, vol. 41, no. 6, February 2002.
```

B.3 SHS Characterization

The SHS Characterization folder contains the program: shs_c_system.m and the necessary functions not already listed.

B.3.1 shs_c_system.m

```
% This is a program to run spectral data through the SHS model and
% interferogram processing algorithm to perform system characterization.
%
% We study the following system error sources:
% - Littrow angle error
% - phase error
% - position, x, error
% - calibration error rate
%
% The user must define the SHS operational parameters in Step 1.
%
% The spectral data must be specified in Step 5. of the program. The
% data should be saved in a ".mat" file. The ".mat" file should contain one
% variable called "spectrum" with the spectral data and another called
% "wavenumber" with the corresponding wavenumber values. Currently, the
% units are set to spectral radiance values of [W/m^2/sr/cm^-1] and
% wavenumber values of [cm^-1].
% THIS MODEL INCLUDES THE GRATING ANGLE ERROR.



clear all
close all
clc
nicePlots;



% 1. **USER-DEFINED** SHS OPERATIONAL PARAMETERS:

% wavenumber range:
shs.kmin = 950; % (1/cm)
shs.kmax = 1250; % (1/cm) --> Littrow wavenumber
%
% diffraction grating characteristics:
shs.m = 1; % order number
shs.d = 1/143; % grating groove separation (cm)

% number of interferogram samples:
shs.meas_N = 128;
% ******************************************



% 2. PARAMETERS CALCULATED FROM OPERATIONAL PARAMETERS:

% Littrow settings of the gratings:
shs.lk = shs.kmax; % Littrow wavenumber (1/cm)
shs.ltheta = asind( shs.m / (2*shs.d*shs.lk)); % Littrow angle (degrees)

% number of high-resolution samples:
shs.N = 2^(nextpow2(2*(shs.kmax-shs.kmin)));

% grating width:
shs.w = shs.meas_N / (2*4*(shs.kmax - shs.kmin)*sind(shs.ltheta)); % (cm)

% maximum position, x, along the detector:
shs.xmax = 1/2*shs.w*cosd(shs.ltheta); % (cm)

% high-res interferogram sample spacing:
shs.dx = 2*shs.xmax / shs.N; % (cm)

% high-res spectral resolution:
shs.dk = 2*(shs.kmax-shs.kmin) / shs.N; % (1/cm)
```

```
% high-res position x and difference wavenumber (k - l.k) arrays:
shs.x = -shs.xmax:shs.dx:shs.xmax-shs.dx; % (cm)
shs.k = 0:shs.dk:shs.kmax-shs.kmin; % (1/cm)

% high-res spectral range array:
shs.wavenumber = shs.kmin:(shs.kmax-shs.kmin)*2/(shs.N):shs.kmax;
% *******************************************



% 3. DETECTION PARAMETERS:

% spectral resolution:
shs.meas_dk = 1 / (2*4*tand(shs.ltheta)*shs.xmax); % (1/cm)

% interferogram sample spacing:
shs.meas_dx = 1 / (4*tand(shs.ltheta)*shs.meas_dk*shs.meas_N); % (cm)

% position x and difference wavenumber (k - l.k) arrays
shs.meas_x = -shs.xmax:shs.meas_dx:shs.xmax-shs.meas_dx; % (cm)
shs.meas_k = 0:shs.meas_dk:(shs.kmax-shs.kmin-shs.meas_dk); % (1/cm)

% measured spectral range array:
shs.meas_wavenumber = shs.kmin:(shs.kmax-shs.kmin)*2/(shs.meas_N):(shs.kmax-shs.meas_dk);
% *******************************************



% 4. CALIBRATION MEASUREMENTS:

% First, get the full SHS model description:
shs = shs_c_model_description(shs);
% Define System Error Rates:
shs.littrow_error = 0;
shs.phase_error = 0;
shs.cal_e = 0;
shs.vib_x = 0;
% Perform flat-fielding measurements:
shs.ff = get_ff(shs);
% Calculate system gain and responsivity:
[shs.r shs.g] = shs_c_calibration(shs);
% *******************************************



% 5. RUN DATA THROUGH SYSTEM:

% Get the SHS Spectral Data folder directory.
toolName = 'SHS Characterization';
toolDirectory = mfilename('fullpath');
```

```
toolDirectory(strfind(toolDirectory, toolName) + length(toolName) :  end) = [];
dataDirectory = fullfile(toolDirectory, 'SPECTRAL DATA');
addpath(genpath(dataDirectory));
data = what(dataDirectory);
num_files = numel(data.mat);
% *******************************************

% Define the parameter ranges we want to study:
study(1,:) = 0.01:0.00485:0.25;           % Littrow angle error (deg)
study(2,:) = 0.006:0.006:0.3;            % Phase error (rad)
study(3,:) = 2*10^(-4):6*10^(-3):0.3; % Calibration error rate
study(4,:) = 0.00001:0.000074:0.00365;    % Position, x, error (cm)

% Loop through the SHS system to study each parameter:
for m = 1:size(study,1)

    parameter = m;

    shs.littrow_error = 0;
    shs.phase_error = 0;
    shs.cal_e = 0;
    shs.vib_x = 0;

    % Loop through all the data files so we can average the error effects
    % over a variety of samples:
    start = 2;
    for i = start:num_files



        % a. PREPARE DATA FOR SHS MODEL:
        char(data.mat(i)) % prints file name
        spectrum = prepare_data(char(data.mat(i)), shs);
        % *******************************************


        % Loop through the SHS system to observe effect of every parameter
        % value within the study range:
        for j = 1:size(study,2)


            % Define the parameter we are currently observing:
            switch parameter
                case 1
                    shs.littrow_error = study(1,j);
                    plot_data = (study(1,:));
                    error_rate = 'Littrow Angle Error (degrees)';
                case 2
                    shs.phase_error = study(2,j);
                    plot_data = study(2,:);
                    error_rate = 'Phase Error (radians)';
                case 3
                    shs.cal_e = study(3,j);
                    plot_data = study(3,:);
                    error_rate = 'Calibration Error Rate';
                case 4
                    shs.vib_x = study(4,j);
                    plot_data = study(4,:);
                    error_rate = 'Vibrational (x) Error (cm)';
            end




        % b. GENERATE THE INTERFEROGRAM:

        [interferogram shs] = shs_c_model(spectrum, shs);
        % ******************************************



        % c. CONVERT THE RAW INTERFEROGRAM TO CALIBRATED SPECTRAL RADIANCE:

        % Convert back to the radiance domain:
        [converted shs] = shs_c_conversion(interferogram, shs);
        % Quantize:
        recovered = shs_quantization(converted, shs);
        % ******************************************



        % d. CALCULATE NOISE FIGURES OF MERIT:

        SNR = recovered ./ shs.NESR;
        NEDT = shs.NEDT;
        % ******************************************



        % e. CALCULATE PERFORMANCE METRICS:

        spectrum1 = interp1(shs.wavenumber, spectrum, shs.meas_wavenumber);
        % RMSE
        RMSE(i-start+1,j) = rmse(spectrum1, recovered);
        % SDR
        SDR(i-start+1,j) = mean(spectrum1./RMSE(i-start+1,j));
        % Recovery Percent Accuracy
        ACC(i-start+1,j) = accuracy(spectrum1, recovered);
        % SNR
```

```
        SNR1(i-start+1,j) = mean(SNR);
        % SAM
        sam(i-start+1,j) = SAM(spectrum1, recovered);
        % *******************************************


        end
    end


    % f. PLOT RESULTS:

    % i. Average RMSE
    figure, subplot(2,2,1), plot(plot_data, mean(RMSE,1));
    xlabel(error_rate);
    ylabel('RMSE (W/m^2/sr/cm^-^1)');
    title('Average Spectrum Recovery RMSE');

    % ii. Average Recovery Percent Accuracy
    subplot(2,2,2), plot(plot_data, mean(ACC,1));
    xlabel(error_rate);
    ylabel('Accuracy(%)');
    title('Average Spectrum Recovery Accuracy');

    % iii. Average SDR
    subplot(2,2,3), plot(plot_data, mean(SDR,1));
    xlabel(error_rate);
    ylabel('SDR');
    title('Average Signal-to-distortion Ratio');

    % iii. Average Separation Angle
    subplot(2,2,4), plot(plot_data, mean(sam,1));
    xlabel(error_rate);
    ylabel('Angle (degrees)');
    title('Average Separation Angle');

    if m ==3
        % iv. Average Accuracy vs SNR
        figure, plot(mean(SNR1,1), mean(ACC,1));
        xlabel('SNR');
        ylabel('Accuracy(%)');
        title('Average Spectrum Recovery Accuracy vs. SNR');
    end



end
```

B.3.2 accuracy.m

```
function [ output ] = accuracy( input1, input2 )

% This function calculates the percent accuracy of the measurement, input2,
% w.r.t. the theoretical result, input1.

output = mean(100 — 100*abs(input1 — input2)./input1);

end
```

B.3.3 SAM.m

```
function angle = SAM(a,b)
% This function calculates the separation angle between two vectors by the
% SAM algorithm.

a = a./norm(a);
b = b./norm(b);

angle = acosd(a*b');

end
```

B.3.4 shs_c_calibration.m

```
function [ r g ] = shs_c_calibration( shs )

% This function calculates the system responsivity function (not to be
% confused with the detector responsivity term) and gain function to use
% for spectral radiance calibration, later. The built—in calibrating
% blackbodies are used to perform the calculations.
%
% The input is the "shs" struct and the outputs are the system responsivity
% (r) and gain (g) functions.


bb_h = shs.emissivity.*Planck(shs.wavenumber, shs.hot_bb);
bb_c = shs.emissivity.*Planck(shs.wavenumber, shs.cold_bb);
bb_h_i = shs_c_model(bb_h, shs);
bb_c_i = shs_c_model(bb_c, shs);

shs.r = 1;
shs.g = 0;
bb_h_rec = shs_c_conversion(bb_h_i, shs);
bb_c_rec = shs_c_conversion(bb_c_i, shs);
```

```
bb_h = interp1(shs.wavenumber, bb_h, shs.meas_wavenumber);
bb_c = interp1(shs.wavenumber, bb_c, shs.meas_wavenumber);

clearvars shs.r shs.g
r = (bb_h_rec-bb_c_rec)./(bb_h-bb_c);
g = (bb_h_rec./r - bb_h);

end
```

B.3.5 shs_c_conversion.m

```
function [ output shs] = shs_c_conversion( input, shs )

% This function converts the measured interferogram values back into
% spectral radiance values. Half the recovered signal is thrown away
% because the recovered spectrum is actual a mirrored image of the input
% spectrum. The output is not yet calibrated.


N = numel(input);

% 1. FLAT-FIELDING:

input = input./shs.ff;
% *******************************************



% 2. SHIFT CORRECTION:

input = shs_c_shift(input, shs);
shs.shifted = input;
% *******************************************



% 3. BIAS REMOVAL:

DC = mean(input(1,1:round(N/4)) + input(1,end-round(N/4)+1:end))/2;
input = input - ones(size(input))*DC;
% *******************************************



% 4. APODIZATION:

m = N;                          % define the width of the apod. window
```

```
q = round(N/2) - round(m/2);    % determine the necessary window padding
input = (padarray(hamming(m),q)'.*input);
% ********************************************



% 5. IFT:

spectrum = fliplr(1/shs.meas_dk/shs.FOV*2*(abs(ifftshift(ifft(ifftshift...
    (input))))));
output = spectrum(1, round(N/2)+1:end);
% ********************************************



% 6. CALIBRATION:

output = (output./shs.r - shs.g);
% ********************************************



end
```

B.3.6 shs_c_interferometer.m

```
function [ output ] = shs_c_interferometer( input, shs )

% This function calculates the high-resolution interferogram from the
% output of the Entrance Optics.

% Count the number of elements
num_input = numel(input);

% Determine the radiance incident at the detector (after the split beams
% interact with their respective diffraction grating).
input = 1/2*input.*shs.Nga + 1/2*input.*shs.Ngb;

% Calculate any Littrow angle error and the corresponding difference
% wavenumber array:
angle_error = shs.littrow_error;%.*randn(1,shs.num_samples);
k_error = shs.m/shs.d ./ (2*sind(shs.ltheta+angle_error));
shs.kk_error = shs.kk + ones(size(shs.kk)).*(shs.lk-k_error);

% Calculate the position, x, and phase error.
x_e = shs.vib_x;%*randn(1,shs.num_samples);
p_e = shs.phase_error;%*randn(1,shs.num_samples);
```

```
% THE LITTROW ANGLE, x, AND PHASE ERRORS ARE RANDOMIZED FOR EACH SAMPLE
% MEASUREMENT TO MODEL SYSTEM VIBRATIONS.

% Calculate the Fizeau frequency:
fizeau_freq = 2*shs.kk*tand(shs.ltheta)+2*shs.kk_error*...
    tand(shs.ltheta+angle_error);
% Calculate the interferogram averaged over all samples.
for j = 1:1

    for i = 1:num_input
        I = shs.dk*input.*(1 + ...
            sqrt(shs.Nga.*shs.Ngb).*cos(2*pi*fizeau_freq.*(shs.x(1,i) - ...
            x_e(1,j) - shs.shift*shs.dx)...
           - p_e(1,j)));...
        output(j,i) = (sum(I));
    end

end

output = mean(output,1);

end
```

B.3.7 shs_c_model.m

```
function [ interferogram, shs ] = shs_c_model( input, shs )

% This function runs the input radiance through the SHS model. The model
% has blocks: Entrance Optics, Interferometer, Exit Optics, and Detector.
%
% The input is the sample spectral radiance and "shs" struct. The outputs
% are an updated version of "shs" and the interferogram measured by the SHS.


% Generate double-sided spectral data (1/2 term is due to the fact that
% half the radiance exits the interferometer through the entrance aperature
% after recombining at the beam-splitter).

input = 1/2*[fliplr(input(1,2:end-1)) input];



% 1. ENTRANCE OPTICS

spectrum = shs_entrance_optics(input, shs);
% *******************************************
% 2. INTERFEROMETER:

interferogram = shs_c_interferometer(spectrum, shs);
% ********************************************



% 3. EXIT OPTICS:

% generate a high-resolution interferogram:
interferogram = shs_exit_optics(interferogram, shs);
% ********************************************



% 4. DETECTOR:

% sampled interferogram:
[interferogram shs] = shs_detector(interferogram, shs);
% ********************************************



end
```

B.3.8 shs_c_model_description.m

```
function [ shs ] = shs_c_model_description( shs )

% This is a complete description of the SHS model. All necessary parameters
% are defined, here. This includes constants, system error rates,
% parameters describing the Entrance Optics, Interferometer, Exit Optics,
% and Detection blocks, and the built-in calibrating blackbodies'
% description.
%
% The function input is the struct called "shs". shs already has the field
% parameters defined by the user and calculated by the model to describe
% the SHS basics. The output is the same struct, "shs", but with additional
% parameters to create a complete description.

% Constants:
shs.q = 1.602*10^(-19); % Coulomb (C = A/s)
shs.h = 6.26*10^(-34);  % Planck's constant (Js)
shs.c = 3*10^(10);      % speed of light (cm/s)
shs.kb = 1.38*10^(-23); % Boltzman constant (J/K)

% System Error Rates:
```

```
shs.bit_e = 10^(-9);        % bit error rate
shs.shift = 0;              % center-burst phase shift (number of dx's)

% Create a double-sided differential wavenumber array:
shs.kk = [-fliplr(shs.k) shs.k(1,2:end-1)];

% Entrance Optics:
shs.transmission1 = 0.85*exp(-(... % entrance transmission function
    (shs.kk+shs.kmax-shs.kmin)...
    /(2*shs.kmax-shs.kmin)).^2);
shs.T1 = 290;                      % entrance optics temperature (K)
shs.self_emission1 = Planck(...    % entrance emission (W/m^2/sr/cm^-1)
    shs.wavenumber, shs.T1);
shs.f_num = 2.4;                   % system f/#
shs.FOV = pi/(4*shs.f_num^2+1);    % system FOV (radians)

% Interferometer:
shs.Nga = 0.8 + 0.01*cos(0.2*pi*shs.kk);  % grating efficiency in arm, a
shs.Ngb = 0.85 + 0.01*cos(0.2*pi*shs.kk); % grating efficiency in arm, b
shs.num_samples = 100;                    % number of sample averages

% Exit Optics:
shs.transmission2 = 0.8*exp(-(... % exit transmission function
    (shs.kk+shs.kmax-shs.kmin)...
    /(8*shs.kmax-shs.kmin)).^2);
shs.T2 = 280;                     % exit optics temperature (K)
shs.self_emission2 = Planck(...
    shs.meas_wavenumber, shs.T2); % exit emission (W/m^2/sr/cm^-1)

% Detector (assumed to be MCT):
shs.tot_eff = mean(...        % total system efficiency
    shs.transmission1.*shs.transmission2.*...
    (shs.Nga+shs.Ngb)/2);
shs.spectral_bandwidth = ...  % spectral bandwidth (cm^-1)
    shs.meas_dk;
shs.resolving_power = 4*...   % Resolving power derived by Harlander [1]
    shs.w*shs.lk*sind(shs.ltheta);
shs.FOVmax = 2*pi/(4*shs.w*...% Limiting FOV derived by Harlander [1]
    shs.lk*sind(shs.ltheta));

shs.pixel_pitch = 40*10^(-6); % pixel pitch (m)
shs.pixel_array = ...         % detector width (cm)
    shs.meas_N*shs.pixel_pitch*10^2;
shs.Adu = shs.pixel_pitch^2;  % detector element area (m^2)
shs.Aa = 60*21/10^6;          % aperature area (m^2)
shs.tint = 1/2000;            % integration time (s)
shs.frame_rate = 114.9;       % frame rate (frame / sec)
shs.QE = 0.70*exp(-(...       % quantum efficiency
    1-shs.meas_wavenumber/shs.lk)/2);
shs.Dstar = 2*10^10;          % D*  (W/cm/sqrt(Hz))
shs.R = 100;                  % Responsivity (V/W)
shs.NPSD = 91.6*10^(-9);      % Noise power spectral density (V/sqrt(Hz))
shs.bits = 17;                % number of bits

shs.p0 = 0.999;               % peak FPA sensitivity
shs.pc = 1/0.15*shs.pixel_array;% width of sensitivity Gaussian
shs.pe = 2;                   % exponential coefficient
shs.cp = shs.p0*exp(-(abs...  % sensitivity across FPA
    (shs.meas_x*1/shs.pc)...
    .^shs.pe)) + 0.001*cos(shs.meas_x*200*pi)...
    .*sin(shs.meas_x*300*pi);

shs.C1 = ...                  % conversion factor for spectral radiance to
    (shs.spectral_bandwidth...% number of electrons
        *shs.Adu*shs.FOV*shs.tint*shs.QE...
    ./(shs.h*shs.c*shs.meas_wavenumber));
shs.C2 = shs.Adu.*shs.tint... % conversion factor for irradiance to # e-
    ./(shs.h.*shs.c.*mean(shs.meas_wavenumber))*mean(shs.QE);
shs.C3 = sqrt(2/shs.meas_N)*2;% conversion factor from k noise to x noise

shs.saturation = 32*10^6;     % saturation level (e-)
shs.Lmax = shs.saturation...  % saturation level (W/m^2/sr/cm^-1)
    ./shs.C1;
shs.throughput = shs.Aa*shs...% optical throughput (sr*cm^2)
    .FOVmax*10^4;
% *******************************************

% Calibrating Blackbodies:
shs.hot_bb = 380;       % hot blackbody temperature (K)
shs.cold_bb = 290;      % cold blackbody temperature (K)
shs.emissivity = 0.994;  % emissivity of blackbodies

% [1] J. M. Harlander, "Spatial heterdyne spectroscopy: Interferometric
%      performance at any wavelength without scanning," Ph.D.
%      dissertation, The University of Wisconsin - Madison, Madison, WI,
%      1991.

end
```

B.3.9 shs_c_noise_model.m

```
function [ output shs ] = shs_c_noise_model( input, shs )

% This function calculates the various noise sources and adds them to the
% measured radiance spectrum and outputs the noisy signal and new "shs"
% terms that describe the noise figures of merit.

new_N = numel(input);
```

```
% Convert the measured sample radiance to #e-:
signal_e = input.*shs.C2;

% Noise sources in #e- (standard deviations):

% IN THE INTERFEROGRAM DOMAIN:

% Shot noise:
shot_noise = mean(sqrt(signal_e));

% Calibration noise:
calibration_noise = shs.cal_e*(shot_noise).^2;

% IN THE SPECTRAL DOMAIN:

% Detector noise:
det_noise = mean(2/(shs.Aa*shs.FOV*shs.meas_dk*shs.Dstar)*...
    sqrt(shs.Adu*10^4*shs.frame_rate)/shs.tot_eff.*shs.C1);

% Electronics noise:
electronics_noise = mean(4*shs.xmax*shs.NPSD*...
    sqrt(shs.frame_rate)/(shs.Aa*shs.FOV*shs.R)/shs.tot_eff.*shs.C1);

% Quantization noise:
quantization_noise = mean(shs.Lmax/(sqrt(12)*2^(shs.bits) - 1).*shs.C1);

% Bit error noise:
bit_noise = 0;
for q = 0:shs.bits-1;
    bit_noise = bit_noise + (2^q*(shs.Lmax)./(2^shs.bits - 1)).^2;
end
bit_noise = mean(sqrt(shs.bit_e/shs.bits*bit_noise).*shs.C1);


% Calculate total noise in #e-:

noise1 = (ones(1,new_N).*shot_noise) +...
    (ones(1,new_N).*calibration_noise);
noise2 = (ones(1,new_N).*det_noise) +...
    (ones(1,new_N).*electronics_noise) +...
    (ones(1,new_N).*quantization_noise) +...
    (ones(1,new_N).*bit_noise);

% Convert spectral noise to interferogram domain and add:
noise = mean(noise1,1) + real(fftshift(fft(fftshift(mean(noise2,1)))));

% Add noise and convert back to interferogram values:
output = (signal_e + noise)./shs.C2;
```

```
% Calculate noise-equivalent change in temperature:

% First, get the total noise standard deviation (#e-):
total_noise = sqrt((shot_noise*shs.C3).^2 + (det_noise).^2 +...
    (electronics_noise).^2 +...
    (quantization_noise).^2 + (calibration_noise*shs.C3).^2 ...
    + (bit_noise).^2);


% Then, get the noise-equivalent spectral radiance (W/cm&2/sr/cm^-1):
shs.NESR = (total_noise).*shs.C1.^(-1);
avg_k = shs.meas_wavenumber;
avg_T = 300;

% Get the derivative of the Planck equation for a given temperature:
dLBdT = 100^2*(2*shs.h^2*shs.c^3*avg_k.^4.*exp(shs.h*shs.c*avg_k./...
    (shs.kb*avg_T)))/(shs.kb*avg_T^2*(exp(shs.h*shs.c.*avg_k./...
    (shs.kb*avg_T)) - 1).^2); % derivative of Planck eq. w.r.t T

shs.NEDT = (shs.NESR./dLBdT); % noise-equivalent change in temperature (K)

end
```

B.3.10 shs_c_shift.m

```
function [ output ] = shs_c_shift( input, shs )

% We do not shift during characterization because it corrects the
% position-x error we are trying to study.

% Take magnitude.
output = real(input);

end

% [1] A. Ben-David and A. Ifarraguerri, "Computation of a spectrum from a
% single-beam fourier-transform infared interferogram," Optical Society of
% America, vol. 41, no. 6, February 2002.
```

B.4 SHS Optimization

The SHS Optimization folder contents are identical to those in the SHS folder except instead of shs_system we have shs_o_system.m and instead of shs_model_description.m we have shs_o_model_description.m.

B.4.1 shs_o_system.m

```
% This is a program to run spectral data through the SHS model and
% interferogram processing algorithm for design optimization studies.
%
% We study the following system tradeoffs:
% — spectral range vs spectral resolution
% — diffraction grating width (number of samples) vs spectral resolution
% — throughput (aperture area) vs SNR
%
% The user must define the SHS operational parameters in Step 1.
%
% The spectral data must be specified in Step 5. of the program. The
% data should be saved in a ".mat" file. The ".mat" file should contain one
% variable called "spectrum" with the spectral data" and another called
% "wavenumber" with the corresponding wavenumber values. Currently, the
% units are set to spectral radiance values of [W/m^2/sr/cm^−1] and
% wavenumber values of [cm^−1].




clear all
close all
clc
nicePlots;

% Define the parameter ranges:
% Minimum wavenumber:
min_v = 770:2:950;
% Maximum wavenumber:
max_v = 1250:2:1430;
% Number of interferogram samples:
N_v = 64:2:512;
% Aperture area:
aa_v = 20*20/10^6:64/10^6:60*60/10^6;

num_variables = 3;

% Loop through the SHS system to study all the parameters:
for k = 1:num_variables

    % Determine which parameter we are changing:
    switch k
        case 1
            parameter = min_v;
        case 2
            parameter = N_v;
        case 3
            parameter = aa_v;
    end



    % Loop through the entire variable range:
    for j = 1:numel(parameter)



        % 1. **USER-DEFINED** SHS OPERATIONAL PARAMETERS:

        % wavenumber range:
        shs.kmin = 950; % (1/cm)
        shs.kmax = 1250; % (1/cm) --> Littrow wavenumber
        %
        % diffraction grating characterisitics:
        shs.m = 1; % order number
        shs.d = 1/143; % grating groove separation (cm)

        % number of interferogram samples:
        shs.meas_N = 128;
        % *******************************************


        % *******************************************
        % Define the aperture area:
        shs.Aa = 60*21/10^6;

        % Re-define the varying parameter:
        if k == 1
            shs.kmin = min_v(1, end + 1 - j);
            shs.kmax = max_v(1, j);
        elseif k == 2
            shs.meas_N = N_v(1,j);
        elseif k == 3
            shs.Aa = aa_v(1,j);

        end
        % *******************************************


        % 2. PARAMETERS CALCULATED FROM OPERATIONAL PARAMETERS:

        % Littrow settings of the gratings:
        shs.lk = shs.kmax; % Littrow wavenumber (1/cm)
        shs.ltheta = asind( shs.m / (2*shs.d*shs.lk)); % Littrow angle (degrees)
```

```
% number of high-resolution samples:
shs.N = 2^(nextpow2(2*(shs.kmax-shs.kmin)));

% grating width:
shs.w = shs.meas_N / (2*4*(shs.kmax - shs.kmin)*sind(shs.ltheta)); % (cm)

% maximum position, x, along the detector:
shs.xmax = 1/2*shs.w*cosd(shs.ltheta); % (cm)

% high-res interferogram sample spacing:
shs.dx = 2*shs.xmax / shs.N; % (cm)

% high-res spectral resolution:
shs.dk = 2*(shs.kmax-shs.kmin) / shs.N; % (1/cm)

% high-res position x and difference wavenumber (k - l.k) arrays:
shs.x = -shs.xmax:shs.dx:shs.xmax-shs.dx; % (cm)
shs.k = 0:shs.dk:shs.kmax-shs.kmin; % (1/cm)

% high-res spectral range array:
shs.wavenumber = shs.kmin:(shs.kmax-shs.kmin)*2/(shs.N):shs.kmax;
% *******************************************



% 3. DETECTION PARAMETERS:

% spectral resolution:
shs.meas_dk = 1 / (2*4*tand(shs.ltheta)*shs.xmax); % (1/cm)

% interferogram sample spacing:
shs.meas_dx = 1 / (4*tand(shs.ltheta)*shs.meas_dk*shs.meas_N); % (cm)

% position x and difference wavenumber (k - l.k) arrays
shs.meas_x = -shs.xmax:shs.meas_dx:shs.xmax-shs.meas_dx; % (cm)
shs.meas_k = 0:shs.meas_dk:(shs.kmax-shs.kmin-shs.meas_dk); % (1/cm)

% measured spectral range array:
shs.meas_wavenumber = shs.kmin:(shs.kmax-shs.kmin)*2/(shs.meas_N):(shs.kmax-shs.meas_dk);
% *******************************************



% 4. RUN DATA THROUGH SYSTEM:

% First, get the full SHS model description:
shs = shs_o_model_description(shs);

% Get the SHS Spectral Data folder directory.
toolName = 'SHS Optimization';
```

```
toolDirectory = mfilename('fullpath');
toolDirectory(strfind(toolDirectory, toolName)...
    + length(toolName) :  end) = [];
dataDirectory = fullfile(toolDirectory, 'SPECTRAL DATA');
addpath(genpath(dataDirectory));
data = what(dataDirectory);
num_files = numel(data.mat);
% *******************************************

start = 2;
% Loop through all the files to average the results:


% We only need to run the spectral data through the system if
% we're observing throughput vs. SN
if k == 3
    for i = start:num_files


        % a. CALIBRATION MEASUREMENTS:

        % Perform flat—fielding measurements:
        shs.ff = get_ff(shs);
        % Calculate system gain and responsivity:
        [shs.r shs.g] = shs_calibration(shs);
        % *******************************************


        % b. PREPARE DATA FOR SHS MODEL:

        char(data.mat(i)) % print the file name
        spectrum = prepare_data(char(data.mat(i)), shs);
        % *******************************************



        % c. GENERATE THE INTERFEROGRAM:

        [interferogram shs] = shs_model(spectrum, shs);
        % *******************************************



        % d. SHIFT THE INTERFEROGRAM CENTER—BURST TO x=0cm:

        shifted = shs_shift(interferogram, shs);
        % *******************************************
        % e. CONVERT THE RAW INTERFEROGRAM TO CALIBRATED
        %    SPECTRAL RADIANCE:

        % Convert back to the radiance domain:
        [converted shs] = shs_conversion(shifted, shs);
        % Quantize:
        recovered = shs_quantization(converted, shs);
        % **********************



        % f. CALCULATE NOISE FIGURES OF MERIT:

        SNR = recovered ./ shs.NESR;
        NEDT = shs.NEDT;
        % **********************



        % g. CALCULATE PERFORMANCE METRICS:

        spectrum1 = interp1(shs.wavenumber, spectrum,...
            shs.meas_wavenumber);
        % SNR
        SNR1(i-start+1,j) = mean(SNR);
        % **********************

      end

    end


  % h. PLOT RESULTS:

    % Determine parameter variables for plotting:
    if k==1
      spectral_range(1,j) = shs.kmax - shs.kmin;
      spectral_resolution(1,j) = shs.meas_dk;
    elseif k==2
      w(1,j) = shs.w;
      spectral_resolution(1,j) = shs.meas_dk;
    elseif k==3
      throughput(1,j) = shs.Aa*shs.FOVmax*10^4;
    end
  end

if k ==1
  figure, plot(spectral_range, spectral_resolution);
  xlabel('Spectral Range (cm^-^1)');
  ylabel('Spectral Resolution (cm^-^1)');
```

```
    title('Spectral Range vs. Spectral Resolution');
elseif k==2
    figure, plot(w, spectral_resolution);
    xlabel('Diffraction Grating Width (cm)');
    ylabel('Spectral Resolution (cm^-^1)');
    title('Diffraction Grating Width vs. Spectral Resolution');
elseif k==3
    figure, plot(throughput, mean(SNR1,1));
    xlabel('Throughput (cm^2sr)');
    ylabel('SNR');
    title('Throughput vs. Average SNR');
end



end
```

B.4.2 shs_o_model_description.m

```
function [ shs ] = shs_o_model_description( shs )

% This is a complete description of the SHS model. All necessary parameters
% are defined, here. This includes constants, system error rates,
% parameters describing the Entrance Optics, Interferometer, Exit Optics,
% and Detection blocks, and the built-in calibrating blackbodies'
% description.
%
% The function input is the struct called "shs". shs already has the field
% parameters defined by the user and calculated by the model to describe
% the SHS basics. The output is the same struct, "shs", but with additional
% parameters to create a complete description.

% Constants:
shs.q = 1.602*10^(-19); % Coulomb (C = A/s)
shs.h = 6.26*10^(-34);  % Planck's constant (Js)
shs.c = 3*10^(10);      % speed of light (cm/s)
shs.kb = 1.38*10^(-23); % Boltzman constant (J/K)

% System Error Rates:
shs.cal_e = 0.0001;         % calibration error rate
shs.bit_e = 10^(-9);        % bit error rate
shs.vib_x = 10^(-6);        % position, x, error rate (cm)
shs.littrow_error = 0.001;  % Littrow angle error rate (degrees)
shs.shift = 4;              % center-burst phase shift (number of dx's)
shs.phase_error = 0.0367;   % k-dependent phase error rate (radians)

% Create a double-sided differential wavenumber array:
shs.kk = [-fliplr(shs.k) shs.k(1,2:end-1)];
```

```
% Entrance Optics:
shs.transmission1 = 0.85*exp(-(... % entrance transmission function
    (shs.kk+shs.kmax-shs.kmin)...
    /(2*shs.kmax-shs.kmin)).^2);
shs.T1 = 290;                      % entrance optics temperature (K)
shs.self_emission1 = Planck(...    % entrance emission (W/m^2/sr/cm^-1)
    shs.wavenumber, shs.T1);
shs.f_num = 2.4;                   % system f/#
shs.FOV = pi/(4*shs.f_num^2+1);    % system FOV (radians)

% Interferometer:
shs.Nga = 0.8 + 0.01*cos(0.2*pi*shs.kk);  % grating efficiency in arm, a
shs.Ngb = 0.85 + 0.01*cos(0.2*pi*shs.kk); % grating efficiency in arm, b
shs.num_samples = 10;                    % number of sample averages

% Exit Optics:
shs.transmission2 = 0.8*exp(-(... % exit transmission function
    (shs.kk+shs.kmax-shs.kmin)...
    /(8*shs.kmax-shs.kmin)).^2);
shs.T2 = 280;                     % exit optics temperature (K)
shs.self_emission2 = Planck(...
    shs.meas_wavenumber, shs.T2); % exit emission (W/m^2/sr/cm^-1)

% Detector (assumed to be MCT):
shs.tot_eff = mean(...        % total system efficiency
    shs.transmission1.*shs.transmission2.*...
    (shs.Nga+shs.Ngb)/2);
shs.spectral_bandwidth = ...  % spectral bandwidth (cm^-1)
    shs.meas_dk;
shs.resolving_power = 4*...   % Resolving power derived by Harlander [1]
    shs.w*shs.lk*sind(shs.ltheta);
shs.FOVmax = 2*pi/(4*shs.w*...% Limiting FOV derived by Harlander [1]
    shs.lk*sind(shs.ltheta));

shs.pixel_pitch = 40*10^(-6); % pixel pitch (m)
shs.pixel_array = ...         % detector width (cm)
    shs.meas_N*shs.pixel_pitch*10^4;
shs.Adu = shs.pixel_pitch^2;  % detector element area (m^2)
% shs.Aa = 60*21/10^6;          % aperture area (m^2)
shs.tint = 1/2000;            % integration time (s)
shs.frame_rate = 114.9;       % frame rate (frame / sec)
shs.QE = 0.70*exp(-(...       % quantum efficiency
    1-shs.meas_wavenumber/shs.lk)/2);
shs.Dstar = 2*10^10;          % D* (W/cm/sqrt(Hz))
shs.R = 100;                  % Responsivity (V/W)
shs.NPSD = 91.6*10^(-9);      % Noise power spectral density (V/sqrt(Hz))
shs.bits = 17;                % number of bits

shs.p0 = 0.999;               % peak FPA sensitivity
shs.pc = 1/15*shs.pixel_array;% width of sensitivity Gaussian
shs.pe = 2;                   % exponential coefficient
shs.cp = shs.p0*exp(-(abs...  % sensitivity across FPA
    (shs.meas_x*1/shs.pc)...
    .^shs.pe)) + 0.001*cos(shs.meas_x*200*pi)...
    .*sin(shs.meas_x*300*pi);

shs.C1 = ...                  % conversion factor for spectral radiance to
    (shs.spectral_bandwidth...% number of electrons
        *shs.Adu*shs.FOV*shs.tint*shs.QE...
    ./(shs.h*shs.c*shs.meas_wavenumber));
shs.C2 = shs.Adu.*shs.tint... % conversion factor for irradiance to # e-
    ./(shs.h.*shs.c.*mean(shs.meas_wavenumber))*mean(shs.QE);
shs.C3 = sqrt(2/shs.meas_N)*2;% conversion factor from k noise to x noise

shs.saturation = 32*10^6;     % saturation level (e-)
shs.Lmax = shs.saturation...  % saturation level (W/m^2/sr/cm^-1)
    ./shs.C1;
% shs.throughput = shs.Aa*shs...% optical throughput (sr*cm^2)
%     .FOVmax*10^4;
% *******************************************

% Calibrating Blackbodies:
shs.hot_bb = 380;       % hot blackbody temperature (K)
shs.cold_bb = 290;      % cold blackbody temperature (K)
shs.emissivity = 0.994;  % emissivity of blackbodies

end

% [1] J. M. Harlander, "Spatial heterdyne spectroscopy: Interferometric
%      performance at any wavelength without scanning," Ph.D.
%      dissertation, The University of Wisconsin - Madison, Madison, WI,
%      1991.
```

Appendix C

Summary of Comparison between Our SHS Model and IRISHS

Table C.1: The comparison between our SHS model and the IRISHS model.

Model Characteristic	**IRISHS**	**Our Model**
Instrument self-emission terms	entrance and exit optics, field and cold-stops	entrance and exit optics
Optical transmission functions	entrance and exit optics	entrance and exit optics
Grating efficiencies	assumed equal	assumed unequal
FPA sensitivity	Yes	Yes
System noise	Yes	Yes
Built-in calibration	Yes	Yes
Phase error	Yes	Yes
Position, x, error	No	Yes
Littrow angle error	No	Yes

Appendix D

Model Parameters

Table D.1: The additional parameters incorporated in the SHS model used to generate the results in Chapter 7.

Entrance Optics	
τ_1	$0.85\exp -f(k)$
Self-emission temperature, T_1 $[K]$	290
$f\#$	2.4
Ω $[radians]$	0.1307
Interferometer	
η_A	$0.80\cos f(k,x)$
η_B	$0.85\cos f(k,x)$
# of sample averages	100
Littrow angle error rate, θ_e $[°]$	0.001
Phase error rate, $\phi(k,x)$ $[radians]$	0.01
Position error rate, x_e $[cm]$	10^{-6}
Exit Optics	
τ_2	$0.80\exp -f(k)$
Self-emission temperature, T_2 $[K]$	280
Detector	
$\delta k[cm^{-1}]$	4.6875
A_d $[\mu m^2]$	40x40
A_a $[mm^2]$	60x21
t_{int} $[ms]$	0.5
T $[frames/s]$	114.9
QE	$0.70\exp -f(k)$
D^* $[W/cm/\sqrt{Hz}]$	$2x10^{10}$

R $[V/W]$	100
$NPSD$ $[V/\sqrt{Hz}]$	$91.6x10^{-9}$
M	17
Saturation level [electrons]	$32x10^{6}$
Calibrating Blackbodies	
T_H $[K]$	380
T_C $[K]$	290
Emissivity	0.994
System Error Rates	
ϵ_C	0.001
ϵ_B	10^{-9}

Bibliography

[1] [Online]. Available: http://www.optra.com/images/420_Dispersive_Image1.gif

[2] [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/phyopt/imgpho/diffgrat.gif

[3] D. R. Hearn, "Fourier transform interferometry," Massachusetts Institute of Technology Lincoln Laboratory, Lexington, MA, Tech. Rep. 1053, October 1999.

[4] B. J. Cooke, B. W. Smith, B. E. Laubscher, P. V. Villeneuve, and S. Briles, "Analysis and system design framework for infrared for spatial heterdyne spectrometer," *Proc. SPIE, Infrared Imaging Systems: Design, Analysis, Modeling, and Testing X*, vol. 3701, no. 167, July 12 1999.

[5] H. Ma, "Grating Sagnac Fourier transform spectrometer and its applications," Ph.D. dissertation, University of Hawaii, December 2007.

[6] D. W. Ball, *Field Guide to Spectroscopy.* Bellingham, WA: SPIE Press, 2006.

[7] M. Pisani and M. Zucco, "Compact imaging spectrometer combining Fourier transform spectroscopy with a Fabry-Perot interferometer," *Optical Society of America, Opt. Express 17, 8319-8331*, no. 10, May 2009.

[8] J. M. Harlander, "Spatial heterdyne spectroscopy: interferometric performance at any wavelength without scanning," Ph.D. dissertation, The University of Wisconsin - Madison, Madison, WI, 1991.

[9] C. R. Englert, J. M. Harlander, J. C. Owrutsky, and J. T. Bays, "SHIM-Fire breadboard instrument design, integration, and first measurements," Naval Research Laboratory, Tech. Rep. 7640-05-8962, 2005.

[10] M. Chamberland, C. Belzile, V. Farley, J.-F. Legault, and K. Schwantes, "Advancements in field-portable imaging radiometric spectrometer technology for chemical detection," *Proc. SPIE 5416, Chemical and Biological Sensing V, 63*, 2004.

[11] M. R. Carter, C. L. Bennett, D. J. Fields, and F. D. Lee, "Livermore imaging Fourier transform infrared spectrometer (LIFTIRS)," *Proc. SPIE, Imaging Spectrometry*, vol. 2480, pp. 380 – 386, 1995.

[12] R. Wright, P. Lucey, S. Crites, K. Horton, M. Wood, and H. Garbeil, "BBM/EM design of the thermal hyperspectral imager: An instrument for remote sensing of earth's surface, atmosphere and ocean, from a microsatellite platform," *Acta Astronautica*, vol. 87, no. 0, pp. 182–192, 2013/7// 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094576513000027

[13] C. M. Gittins, W. J. Marinelli, and J. O. Jensen, "Remote sensing and selective detection of chemical vapor plumes by LWIR imaging Fabry-Perot spectrometry," *Proc. SPIE 4574, Instrumentation for Air Pollution and Global Atmospheric Monitoring*, no. 63, February 12, 2002. [Online]. Available: +http://dx.doi.org/10.1117/12.455169

[14] C. R. Englert, D. D. Babcock, and J. M. Harlander, "Spatial heterodyne spectroscopy for long-wave infrared: first measurements of broadband spectra," *Optical Engineering*, vol. 48, no. 10, pp. 105 602–105 602, 10 2009. [Online]. Available: http://dx.doi.org/10.1117/1.3250194

[15] [Online]. Available: http://2.bp.blogspot.com/_9k1SiI_BnX0/S_b6Hgy7CAI/AAAAAAAABCE/wxzoX5okxBA/s1600/800px-Rectangular_function.svg.png

[16] [Online]. Available: http://en.wikipedia.org/wiki/File:SincFunction.png

[17] [Online]. Available: http://www.ericweisstein.com/research/thesis/img150.gif

[18] Y. Lin, G. Shepherd, B. Solheim, M. Shepherd, S. Brown, J. Harlander, and J. Whiteway, "Introduction to spatial heterodyne observations of water (SHOW) project and its instrument development," May 2005.

[19] A. R. Korb, P. Dybwad, W. Wadsworth, and J. W. Salisbury, "Portable Fourier transform infrared spectroradiometer for field measurements of radiance and emissivity," *Appl. Opt.*, vol. 35, no. 10, pp. 1679–1692, Apr 1996. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-35-10-1679

[20] V. Saptari, *Fourier-Transform Spectroscopy Instrumentation Engineering.* Bellingham, WA: SPIE Press, 2003.

[21] "Introduction to Fourier transfrom infrared spectrometry," Thermo Nicolet Corporation, 5225 Verona Road, Madison, WI 53711, Tech. Rep., 2001.

[22] P. Griffiths and J. deHaseth, *Fourier Transform Infrared Spectrometry.* Wiley & Sons, 1986.

[23] W. Harris, F. Roesler, L. Ben-Jaffel, E. Mierkiewicz, J. Corliss, R. Oliversen, and T. Neef, "Applications of spatial heterodyne spectroscopy for remote sensing of diffuse UV–VIS emission line sources in the solar system," *Journal of Electron Spectroscopy and Related Phenomena*, vol. 144–147, no. 0, pp. 973 – 977, 2005, proceeding of the Fourteenth International Conference on Vacuum Ultraviolet Radiation Physics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0368204805002793

[24] J. Harlander, R. J. Reynolds, and F. L. Roesler, "Spatial heterodyne spectroscopy for the exploration of diffuse interstellar emission lines at far-ultraviolet wavelengths," *The Astrophysical Journal*, vol. 396, no. 1, pp. 730–740, September 1992.

[25] F. L. Roesler, "An overview of the SHS technique and applications," *OSA Technical Digest Series (CD) (Optical Society of America, 2007)*, no. FTuC1.

[26] J. M. Harlander, F. L. Roesler, J. G. Cardon, C. R. Englert, and R. R. Conway, "SHIMMER: a spatial heterodyne spectrometer for remote sensing of earth's middle atmosphere," *Appl. Opt.*, vol. 41, no. 7, pp. 1343–1352, Mar 2002. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-41-7-1343

[27] J. M. Harlander, J. E. Lawler, F. L. Roesler, and Z. Labby, "A high resolution broad spectral range spatial heterodyne spectrometer for UV laboratory astrophysics." Optical Society of America, 2007, p. FWA3. [Online]. Available: http://www.opticsinfobase.org/abstract.cfm?URI=FTS-2007-FWA3

[28] W. Herres and J. Gronholz, "Understanding FT-IR data processing," Tech. Rep.

[29] P. G. Lucey and J. Akagi, "A Fabry-Perot interferometer with a spatially variable resonance gap employed as a Fourier transform spectrometer," *Proc. SPIE 8048, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVII*, pp. 80 480K–80 480K–9, (May 20, 2011). [Online]. Available: +http://dx.doi.org/10.1117/12.886534

[30] F. L. Pedrotti, L. M. Pedrotti, and L. S. Pedrotti, *Introduction to Optics*, 3rd ed. Addison-Wesley, 2007.

[31] L.-J. Chen, T.-F. Kao, H.-H. Chang, J.-Y. Lu, and C.-K. Sun, "Terahertz Fourier transform spectrometer based on a low-reflectivity Fabry-Perot interferometer," in *Conference on Lasers and Electro-Optics/Quantum Electronics and Laser Science and Photonic Applications Systems Technologies.* Optical Society of America, 2005, p. QTuI2. [Online]. Available: http://www.opticsinfobase.org/abstract.cfm?URI=QELS-2005-QTuI2

[32] J. M. Harlander and F. L. Roesler, “Spatial heterodyne spectroscopy: a novel interferometric technique for ground-based and space astronomy,” *Proc. SPIE, Instrumentation in Astronomy VII*, vol. 1235, no. 622, July 1, 1990. [Online]. Available: +http://dx.doi.org/10.1117/12.19125

[33] J. Roger L. Easton, *Fourier Methods in Imaging.* Wiley & Sons, 2010.

[34] E. J. Ientilucci and J. R. Schott, “Radiometry and radiation propagation,” September 2011.

[35] W. J. Marinelli, C. M. Gittins, and T. E. Ustun, “AIRIS wide area detection system,” Physical Sciences, Inc., Andover, MA, Tech. Rep., July 2003.

[36] B. R. Cosofret, S. Chang, M. L. Finson, C. M. Gittins, T. E. Janov, D. Konno, W. J. Marinelli, M. J. Levreault, and R. K. Miyashiro, “AIRIS standoff multispectral sensor,” *Proc. SPIE, Chemical, Biological, Radiological, Nuclear, and Explosives (CBRNE) Sensing X*, vol. 7304, pp. 73 040Y–73 040Y–4, May 08, 2009. [Online]. Available: +http://dx.doi.org/10.1117/12.816647

[37] W. J. Marinelli, C. M. Gittins, A. H. Gelb, and B. D. Green, “Tunable Fabry-Perot etalon-based long-wavelength infrared imaging spectroradiometer,” *Appl. Opt.*, vol. 38, no. 12, pp. 2594–2604, Apr 1999. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-38-12-2594

[38] J. Harlander, H. T. Tran, F. L. Roesler, K. Jaehnig, S. M. Seo, W. Sanders, and R. J. Reynolds, “Field-widened spatial heterodyne spectroscopy: correcting for optical defects and new vacuum ultraviolet performance tests,” *Proc. SPIE 2280, EUV, X-Ray, and Gamma-Ray Instrumentation for Astronomy V*, vol. 310, no. doi:10.1117/12.186821, September 16, 1994.

[39] C. R. Englert, J. M. Harlander, J. G. Cardon, and F. L. Roesler, “Correction of phase distortion in spatial heterodyne spectroscopy,” *Appl. Opt. 43, 6680-6687*, vol. 43, no. 36, pp. 6680 – 6687, December 2004.

[40] J. R. Schott, *Remote Sensing*, 2nd ed. Oxford University Press, Inc., 2007.

[41] J. Schoukens and J. Renneboog, “Modeling the noise influence on the Fourier coefficients after a discrete fourier transform,” *IEEE Transactions on Instumentation and Measurement*, vol. IM-35, no. 3, September 1986.

[42] C. R. Englert and J. M. Harlander, “Flatfielding in spatial heterodyne spectroscopy,” *Appl. Opt.*, vol. 45, no. 19, pp. 4583 – 4590, July 2006.

[43] A. Ben-David and A. Ifarraguerri, “Computation of a spectrum from a single-beam Fourier-transform infared interferogram,” *Optical Society of America, Appl. Opt.*, vol. 41, no. 6, pp. 1181–1189, February 2002.

[44] M. L. Forman, W. H. Steel, and G. A. Vanasse, “Correction of asymmetric interferograms obtained in Fourier spectroscopy,” *J. Opt. Soc. Am.*, vol. 56, no. 1, pp. 59–61, January 1966.

[45] E. W. Weisstein, “Hamming function.” From MathWorld–A Wolfram Web Ressource. [Online]. Available: http://mathworld.wolfram.com/HammingFunction.html

[46] N. I. of Standards and Technology, “NIST Standard Reference Database 69: NIST Chemistry WebBook.” [Online]. Available: http://webbook.nist.gov/cgi/cbook.cgi?ID=C67561&Type=IR-SPEC&Index=1#IR-SPEC

[47] J. Melonakos, “Run MATLAB 10x-50x faster using GPUs and Jacket plug-in,” http://www.accelereyes.com/content/doc/061709-Tesla-Jacket-webinar.pdf.