Optimizing Thin Film Filters for Ultrashort Pulse Shaping

by

Sarah Dunning

(Under the direction of William M. Dennis)

Abstract

In this research I develop automated thin film filter design techniques for ultrashort laser pulse shaping applications. Genetic algorithms and simulated annealing techniques are used to search for these optimal filter designs. Both temporal and spectral domain methodologies are explored and, for the case of a double pulse reflection filter, comparisons are made between the two techniques. All filters in this work are constructed from two materials (not including the substrate) and consist of alternate high and low refractive index layers. The optimization algorithms search for an optimum filter design within the constraints of layer thickness, total number of layers, material choice, and computational run time. A double pulse filter, a pulse stretching mirror and a compression filter are some of the examples considered in this work.

Index words: Optical filters, Thin films, Dielectric stacks,
Finite difference time domain, Characteristic matrix method,
Genetic algorithms, Simulated annealing

Optimizing Thin Film Filters for Ultrashort Pulse Shaping

by

Sarah Dunning

B.Sc., The University of Auckland, New Zealand, 2002

B.S., The University of Georgia, 2003

A Dissertation Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2009

Optimizing Thin Film Filters for

Ultrashort Pulse Shaping

by

Sarah Dunning

Approved:

Major Professor: William M. Dennis

Committee: Uwe Happek
Heinz-Bernd Schuttler

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2009

Acknowledgments

If one way be better than another,

that you may be sure is nature's way.

- excerpt from *Aristotle*

These pages are a product of the relationships that have developed in the years since I began my graduate studies. There are many who contributed to my development as a researcher and teacher, and to them I give thanks.

Bill Dennis, my gracious advisor, taught me many things, among them the beauty of rigorous checking. Early on he introduced me to the finite-difference time domain technique. Over many coffee shop discussions we talked though the derivations, using up many napkins and sheets of paper. I began carrying pads of writing paper to the coffee shop for such discussions. The underlying algorithms of the finite-difference time domain technique and genetic algorithms remain a source interest to me. Bill provided key insights and direction for my research, and he exemplified patience, allowing me to learn from my own meanderings.

Thanks go to the members of my committee, Professors Uwe Happek, Heinz-Bernd Schuttler, and the late Bill Yen for their encouraging words and insightful feedback on this dissertation. Thanks also go to the current graduate coordinator, Loris Magnani. There is much to know in the administrative side of graduate school, and helped greatly in finding answers. To my professors, who through coursework showed me how to think critically, much credit is due: Bill Dennis, Mike Geller, Uwe Happek, David Landau, Howard Lee, Steve Lewis, Gary Love, Henning Meyer, and Kanzo Nakayama. Also, thanks to my colleagues, Mohua Bhattarcharya, Jeremy Gulley, Tom Lanier, and Sebastian Winkler who shared enthusiasm for and comments on this work.

Special gratitude goes to Jeff Deroshia and Mike Caplinger for their willingness to troubleshoot over hardware hiccups and generally entertain my various computational inquiries. Thanks to the department staff for helping with the administrative tasks necessary for completion of a doctoral program: Tom Barnello, Stephanie Crowe, Amy Fielder, Linda Lee, Sherri McElroy, Dale Richardson, and Mica Turner.

My family, Greg, Juliet, Jane and James Dunning, have greatly influenced my direction in life. You have been wonderful listeners, advocates, and proof-readers. Another great source of morale, Christian Hampton was there for me throughout much of the research stages. His encouragement and support is greatly appreciated. Thanks go my friend Tiffany Suk, who is a key cheer-leader, voice-of-reason, and correspondent. Finally, my gratitude and thanks go to Cal Clements, who makes many things possible. You kept me smiling though to this stage of completion. Thanks for the laughter and love.

Table of Contents

Chapter 1

Introduction

1.1 Historical Overview of Dielectric Stacks

The effects of interference due to single thin films have been observed since the 1600's, when Robert Boyle [1] and Robert Hooke [2] independently discovered the creation of light rings due to the interference in a single thin film of varying thickness. At the time, theories on the nature of light were unable to explain this effect, and it remained a mystery until 1801, when Thomas Young presented his theory on the interference of light [3, 4]. Young's theory was strongly challenged and recognition only came once Augustin Fresnel developed a complementary wave theory of light in 1816 [5]. Joseph Fraunhofer is credited with creating the first antireflection coatings in 1817 [6].

Developments were much more rapid in the 1930s, when evaporated zinc sulphide layers were being produced by Pfund for low-loss beam splitters [7]. Strong developed antireflection coatings by the evaporation of fluorite onto glass [8], and Geffcken created the first thin-film metal-dielectric interference filter [9, 10]. Macleod [6] attributes the flourishing of this field to sputtering and vacuum evaporation methods developed in the 1930's, which were boosted by redoubled efforts to improve optics following the first World War. Once reliable vacuum pumps were developed, the above methods became effective ways to produce a dielectric stack. Currently, dielectric stacks play an integral role in many optical devices, performing well in both high-reflection and antireflection applications [11]. Specific wavelengths can either be blocked or transmitted, the glare can be reduced from a camera lens, or the reflection of an ultrashort laser pulse can be tailored with the addition of a dielectric stack. As manufacturing capability develops, more intricate dielectric stacks are being realized, with both thinner and greater numbers of layers. Manufacturers can now deposit layers of only a few nanometers thickness, allowing dielectric stacks to match detailed spectral profiles with great accuracy, see for example, [12]. The limits and applications for optical thin-film dielectric stacks are still being explored, and this is one of the reasons the field remains an interesting one.

1.2 Designing a Dielectric Stack

While manufacturing technology has enhanced our ability to realize more sophisticated filters, computer simulations have greatly improved our ability to perform detailed modelling of the behavior of these multilayer stacks, which can in turn lead to the development of novel designs.

Given that the materials, substrate, layer number and thicknesses known, one can calculate the expected reflection or transmission profile of that stack using the characteristic matrix method [13]. The characteristic matrix method, which is detailed in Appendix A, essentially expresses the electric and magnetic fields on both sides of a thin film layer in (2×2) matrix form. Calculating the fields on either side of an arbitrary multilayer dielectric stack then reduces to an appropriate series of matrix multiplications and is extremely efficient. In practice, the fields are normalized, and the admittances, $Y = H/E$, on both sides of the stack are calculated. Once the admittances are known the reflectances can be calculated using,

$$R = \left| \frac{Y_0 - Y}{Y_0 + Y} \right|^2 \tag{1.1}$$

where Y_0 is the admittance of free space. As an example, should one want a high-reflecting dielectric stack for a prescribed wavelength range, thin-film theory directs one toward a filter constructed of a series of layers with one quarter wave optical thickness. The beams reflected from each interface in the quarter-wave stack have the same phase at the front of the dielectric stack, allowing them to combine constructively, thus producing a highly reflective device. A quarter-wave stack design is shown in the upper pane of Fig. 1.1. The lower pane shows the spectrum that was calculated using the characteristic matrix method. This particular stack is designed to maximally reflect the central wavelengths of a Ti:Sapphire laser, *i.e.,* $\lambda_L =$ 800 nm.

It is worth noting that the properties of a given quarter-wave stack can be modified in a variety of ways. To broaden the central reflectance band, a higher ratio between refractive indices of the two layers is used. If one requires a dielectric stack to reflect over a different wavelength range, then assuming the material parameters are still valid for the new wavelengths, one needs only to adjust the layers to be quarter-wave thicknesses for the new central wavelength.

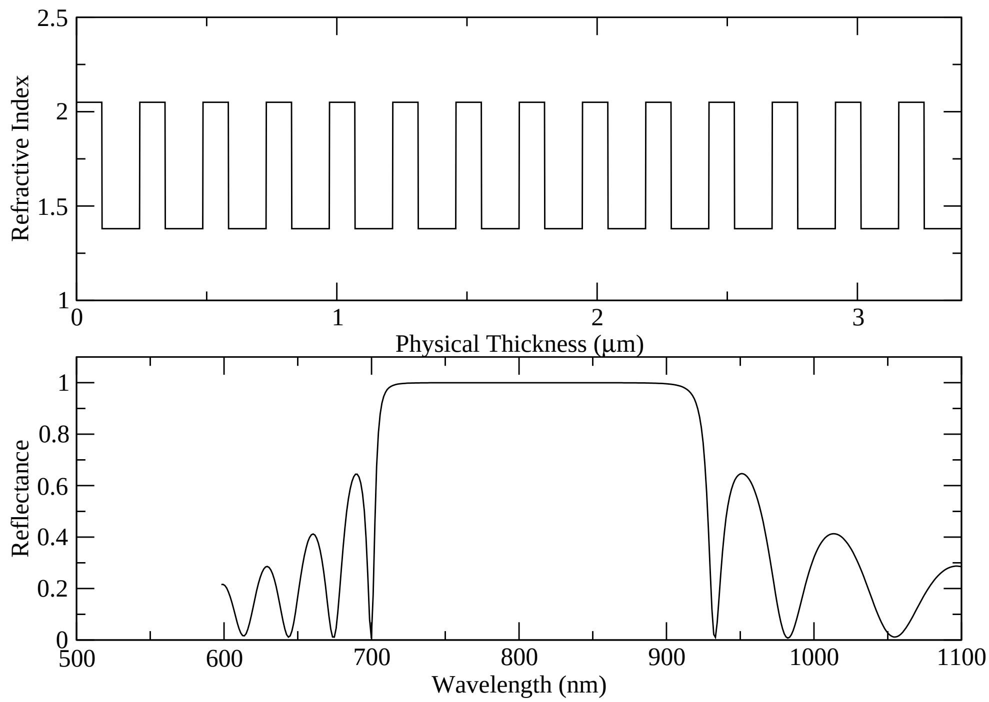

Figure 1.1: A quarter-wave stack design (upper pane) and its reflectance spectrum (lower pane). The resultant spectral profile contains a high-reflectance band at the design wavelength of 800 nm and shows characteristic side lobes.

The purpose of this dissertation is to develop dielectric stacks for ultrashort pulse shaping applications. The problem is this: given a desired reflected or transmitted pulse, calculate the appropriate dielectric stack. This is essentially the inverse of the problem of calculating the spectrum of a given filter, and it is harder because of the indirect method needed to achieve this. One approach is to use an iterative scheme whereby a calculation of the desired properties is incorporated into an optimization algorithm. The calculation of the desired properties can either be performed in the frequency domain, *i.e.* if a particular reflectance spectrum is desired, the characteristic matrix method can be used, or in the time domain, if a given pulse shape is desired, the time-dependent electric field can be calculated numerically by integrating either the wave equation or Maxwell's equations directly.

In this work, the finite-difference time domain (FDTD) method [14] is used to numerically integrate the Maxwell curl equations [15] for the electric and magnetic fields. The equations are converted to finite-difference equations on spatially-offset grids. Due to the interdependent nature of the electric and magnetic fields, the finite-difference time domain method uses a leapfrog algorithm, *i.e.* the electric field at the current time step is used to calculate the magnetic field at the next (half) time step, and so on. Thus the electric and magnetic fields can be propagated forward in time. The exceptional versatility of this method sees application in fields as diverse as tumor detection and stealth aircraft technology [16, 17, 18, 19, 20]. Additional details of the finite-difference time domain method can be found in Appendix B.

1.3 Optimization Methods

In order to decide upon an optimization method, one must consider the nature of the search space. Gradient-based methods excel when one has a good starting point, near the global minimum. If one does not know the search space well, a random starting position can lead to the algorithm converging at the nearest local minima – a problem that often affects gradient-based methods. Because of the complex search space for the particular filter designs in this current work, gradient designs were passed over in favor a method that could work independent of the starting position and therefore should not become trapped in local maxima/minima. One such non-gradient method, a genetic algorithm, is used in this dissertation because of its ability to leave a local maximum/minimum in search of the global maximum/minimum. Also, due to the computationally intensive nature of the finite-difference time domain calculations, it was advantageous to use an algorithm that could be parallelized.

Computational optimization has been integral to optical filter design since Baumeister's work in 1958 [21] where he described a method of relaxing the thicknesses of a multilayer filter design to produce a "close approximation to a given transmission curve in a limited spectral region." Indeed, within the various optimization methods available today, many are incarnations of this original work. Aguilera *et al.* compare the performance of several optimization methods for designing infrared filters [22], which shows how evolutionary algorithms fare against gradient optimization methods for a particular filter design problem. Dobrowolski and Kemp have also performed such comparisons and provide a useful overview of the various techniques [23]. Yang and Kao compare their family competition evolutionary algorithm (FCEA) to the standard genetic algorithm [24] and simulated annealing methods [25]. Their work, which is a special-case genetic algorithm, was useful for comparison with the frequency domain genetic algorithm in this dissertation.

Table 1.1 shows some of the popular gradient (damped least squares, Hooke-Jeeves) and non-gradient (family competition evolutionary algorithm, Monte Carlo simulated annealing, needle method) methods that have been applied to thin-film filter design to date. From the table, one can see the variation in the thin film filter thickness for the beam splitter and antireflection coating examples. All the solutions produced are of a reasonable thickness and layer number, such that they are all realistic options for constructing. The design problems chosen for comparison did not single out any method as being significantly better than the others.

Table 1.1: A comparison of several optimization methods as applied in the literature to a particular thin-film filter design problems. Optimization methods have been abbreviated as follows: damped least squares (DLS), family competition evolutionary algorithm (FCEA), Hooke-Jeeves (HJ), Monte Carlo simulated annealing (MCSA), needle method (NM). The optical thickness was calculated by $\sigma = \sum_{i=1} n_i d_i$.

Design Type	Optimization Method	Number of Layers	Optical Thickness	Fitness (when specified)
Beam splitter	FCEA[25]	15	1.65 μm	
	NM[25]	14	1.51 μm	
	DLS[23]	11	1.56 μm	1.13
	MCSA[23]	11	2.12 μm	1.62
Antireflection coating	DLS[22]	11	24.73 μm	1.38
	HJ[22]	21	31.62 μm	0.94

Basics of the Genetic Algorithm

The genetic algorithm takes its inspiration from the principle of natural selection and evolution. A standard version begins with a single population that is usually a random assignment of individuals. Each individual may be a single chromosome or set of chromosomes, depending on how many parameters are to be optimized. A single filter design is known as an individual. The fitness of each individual is evaluated, and the best performing one from each generation is stored. Individuals with better attributes have a higher chance of being selected to mate. In computational terms, mating amounts to passing the binary chromosome strings (which correspond to layer thicknesses) onto the next round of calculations. Thus, after successive iterations, the population, a set of filter designs, evolves to an optimized set of individuals. In this work, the thickness of each filter layer forms the variable parameters.

The algorithm acts on each individual through three common genetic operators,

1. Crossover: two binary strings are selected, and a randomly-chosen portion of their strings are swapped.

2. Mutation: a bit on the individual's string is chosen at random to be changed from a 1 to a 0 or from a 0 to a 1.

3. Selection: the method by which an individual is chosen for crossover and mutation. The total fitness of a population, S is summed. A random number n between 0 and S is chosen, and the individual with the fitness closest to or just above n is selected. In the case of crossover, this process is repeated twice to choose both the parents.

The genetic algorithm in this work is the "little genetic algorithm," which is described in Ref. [26]. The algorithm generates a population of individuals, with each individual representing a possible solution for the pulse shaping dielectric stack design. Each dielectric stack is of the same arrangement of $(\mathrm{HL})_{25}$, with H being the higher refractive index layer and L being the lower refractive index layer. Dielectric stacks are identical in their layer number and material type; the layers of each dielectric stack will be of differing thicknesses. At each generation of the algorithm, the algorithm evaluates the ability of each stack to produce the requisite pulse shape. The algorithm assigns fitness to each individual using a difference-squared match between the target and resultant intensities. A higher value of $\mathcal{F}$ indicates a better dielectric stack. When the next generation begins, the better dielectric stacks have a higher selection probability for the next round of crossover (mating). In this way, a survival of the fittest mechanism helps promote the better solutions while the poorer solutions receive a much lower probability of propagating their genes to the next generation. Parameters such as the total dielectric stack thickness and the individual layer thickness varied over the course of the optimization process. The total stack thickness and corresponding reflectance for each of the optimized designs are shown in the tables throughout this dissertation.

Appropriate selection of user-specified parameters (*e.g.* string length, crossover probability) enables the genetic algorithm to reach its optimum solution in a consistent and accurate manner. Deciding what values are appropriate for each variable, however, is not a straightforward task. The lack of available problem-dependent information often compounds this issue. To get an indication of the sensitivity of the algorithm's performance to each of these user-secified parameters, the performance of the algorithm for a variety of parameter choices and a variety of optimizing functions was examined.

1.4 The Layout of This Dissertation

The remainder of this dissertation is structured as follows: Chapter 2 outlines the dielectric stack design used in this project. The filters described in this dissertation are the product of two independent methodologies: optimizing in the time domain and optimizing in the frequency domain. Chapter 2 also introduces the time and frequency domain aspects of the optimization. The time domain approach is particularly novel for this type for filter design. To the best of my knowledge, this is the first time filters have been optimized with temporal pulse shaping as the target function for an ultra-short laser pulse. Chapter 3 discusses the time domain designs. Results for double pulse reflection, double pulse transmission, triple pulse reflection, and stretched pulse reflection filters are presented in Chapter 3. The possibility of a frequency domain approach is explored in Chapter 4. Both genetic algorithms and simulated annealing algorithms are used in conjunction with the characteristic matrix method in this chapter. Chapter 5 covers some aspects of the manufacturability of a pulse stretching filter. An additional application – a pulse stretcher-compressor pair – is also considered. Conclusions are presented in Chapter 6.

Chapter 2

Preliminary Considerations for Dielectric Stack Design

When we design a dielectric stack, we seek an arrangement of layers (a solution) that best delivers a specified performance. This layer arrangement needs specification of both the material type and layer thicknesses. While excellent solutions can be found by allowing continuous variation of the refractive indices or the layers, this can lead to refractive indices for which no material exists. In this case a post-refinement may be performed on these designs to bring them into an equivalent two-material solution. However, an optimization with two-materials produces a realistic, construction-ready design directly. Furthermore, the smaller search space required by the inclusion of only two materials requires fewer computational resources. For this reason, all designs presented in this dissertation involve a two-material solution.

The task of finding a dielectric stack to produce a desired spectral or temporal function is more difficult than simply calculating the properties of a multilayer stack. For example, one can calculate the properties of a known stack by using the characteristic matrix method in the frequency domain or by the FDTD method in the time domain. But when we wish to design a dielectric stack of a given performance, we are asking the inverse question. A designer typically employs automated methods [27, 28]. These begin with a template structure, *e.g* a 50 layer stack composed of two alternating materials, and refine it to an optimal design within the given constraints. Automated designs are at the same time both convenient and restrictive. By initially narrowing the field of possibilities, the algorithm potentially excludes excellent designs that lie outside the specifics of the input design template. As an example, if a genetic algorithm uses a two-material template, an initial population of 10 and runs for

500 generations, the optimal solution may be very good, but the solution found after 1000 generations might be significantly better. An important part of testing the applicability of a genetic algorithm to a given optimization problem is performing preliminary simulations with a variety of generations and population sizes to ensure the algorithm will produce acceptable results within a given time frame. These preliminary simulations influenced the choice of stack layers in the optimizations that followed. This was particularly noticible in the pulse stretching filters of Chapter 5, which needed more layers in the filter to achieve the necessary higher reflectance of the 100 fs incident pulse. The 100 layer templates produced as good (or better) results than the 150 layer templates. However, both were stronger performers than the 50 layer versions, meaning these filter designs produced reflected intensities that better matched the target function. Thus, 100 layer designs were used for this application. I used a two-material template throughout, since it has been shown elsewhere [29, 30] that two materials can produce dielectric stacks of comparable performance to stacks that use more than two materials.

2.1 Common Materials

The dielectric stacks are composed of alternating layers of high-index and a low-index material, adjacent to a silicon dioxide substrate. Both the high and low index materials, as well as the substrate, were chosen for their widespread use in optical filters and the ease with which they can be deposited as exceptionally thin layers. In the time domain simulations of Chapter 3, the stacks are composed of alternating layers of tantalum pentoxide, Ta_2O_5, and magnesium fluoride, MgF_2 on a silicon dioxide, SiO_2, substrate. Fig. 2.1 shows a schematic of this design. The MgF_2 layer is next to the substrate. For the frequency domain simulations of Chapter 4 and the time domain simulations of Chapter 5, all oxide layers were used, *i.e.* silicon dioxide, SiO_2, and niobium pentoxide, Nb_2O_5, with SiO_2 as the substrate. Table 2.1 gives the refractive indices for the materials used in the dielectric stack designs described in this dissertation.

Table 2.1: Optical properties for tantalum pentoxide, Ta_2O_5, niobium pentoxide, Nb_2O_5, silicon dioxide, SiO_2, and magnesium fluoride, MgF_2.

Material	Refractive Index, n_0
Ta_2O_5	2.05
Nb_2O_5	2.26
SiO_2	1.52
MgF_2	1.38

2.2 Fitness Functions for the Time and Frequency Domains

Chapter 1 introduced the concept of a fitness function, $\mathcal{F}$, as a way for the optimization to evaluate the performance of a filter. The fitness function may use different optical properties to characterize solutions in the time or frequency domain. In the time domain, the reflected and transmitted fields are monitored [31], and the resultant (reflected or transmitted) intensity profile, $I(t)$, is compared to the desired intensity profile, $I_{\text{ideal}}(t)$, using

$$\mathcal{F} = \frac{1}{\sum\limits_i [I_{\text{ideal}}(t_i) - I(t_i)]^2}. \tag{2.1}$$

The reciprocal in Eq. 2.1 is used because the genetic algorithm employed in this dissertation maximizes $\mathcal{F}$. If the desired intensity profile is an N-pulse train with each pulse centered at τ_j and with pulsewidth σ_j, then the target function is

$$I_{\text{ideal}}(t) = \frac{1}{N} \sum_{j=1}^{N} e^{-((t-\tau_j)/\sigma_j)^2}. \tag{2.2}$$

If one combines pulses of width σ_j at time intervals of $\tau_j = \sigma_j$, one can create a single stretched (flat-topped) pulse. This forms the target function for the stretched pulse profiles of Chapters 3 and 5,

$$I_{\text{ideal}}(t) = \frac{1}{N} \sum_{j=1}^{N} e^{-((t-\sigma_j)/\sigma_j)^2}. \tag{2.3}$$

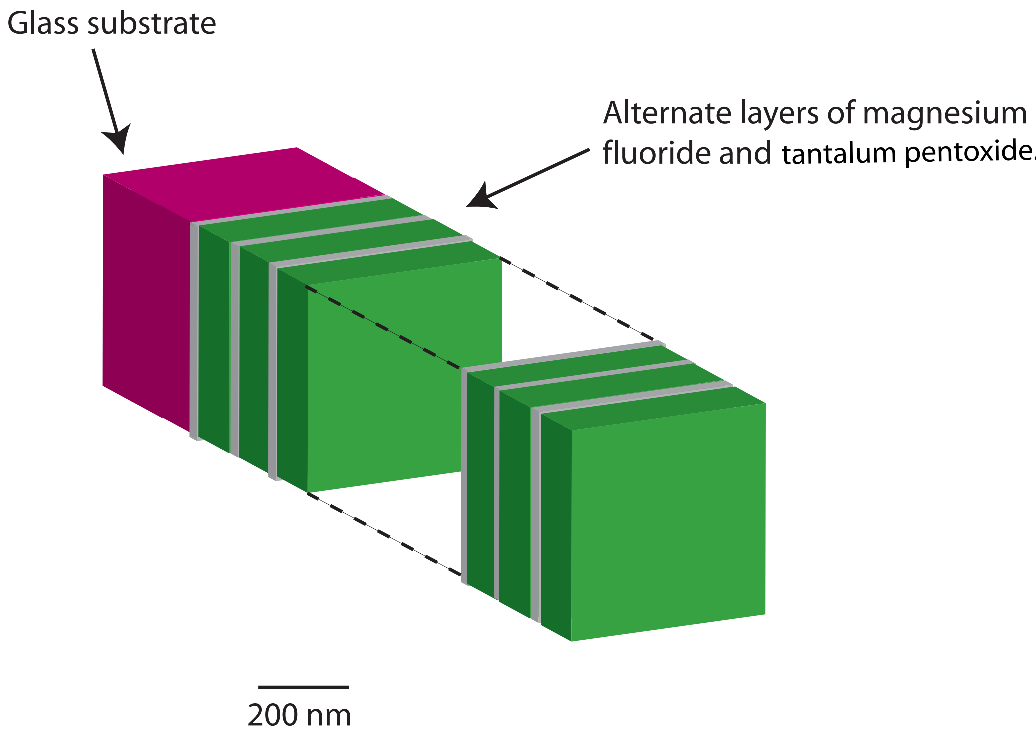

Figure 2.1: Schematic of the two material design for the dielectric stack as described in the dissertation. While the figure shows even thickness layers, the actual designs were of varied layer thickness.

As $\mathcal{F}$ is maximized, the difference between $I(t)$ and $I_{\text{ideal}}(t)$ is minimized and a dielectric stack is generated that will reflect fields close to the specified pulse shape.

There was a deliberate decision to fit for an entire pulse shape rather than to a set of metrics, *e.g.* reflectance, pulse width, *etc.* Since the former is a more stringent requirement that minimizes the multiplicity of a particular solution set. Furthermore, a judicious choice of

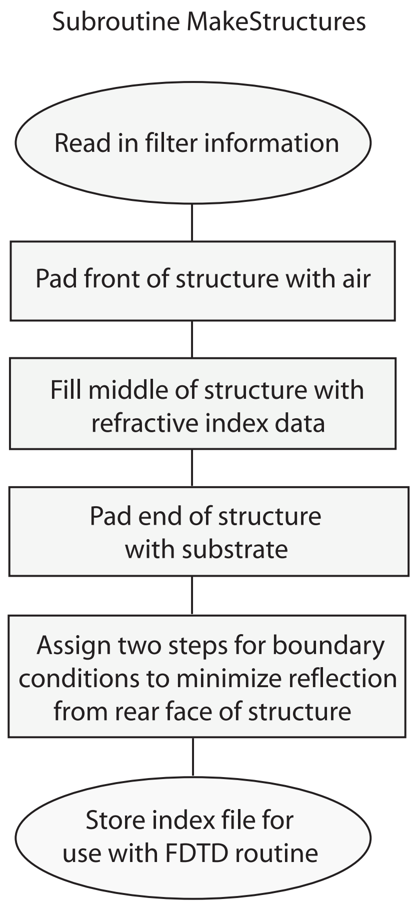

Figure 2.2: Flowchart showing the algorithm for padding the filter prior to its use in the FDTD subroutine.

the amplitudes A_j naturally incorporates the reflectance, which does not appear explicitly in Eq. 2.1. In this dissertation $I(t)$ is calculated using the finite-difference time domain method, which is described in more detail in both Chapter 3 and Appendix B. The GA-FDTD hybrid algorithm results that use Eqs. 2.1, 2.2, and 2.3 appear in Chapters 3 and 5.

The most computationally expensive part of the GA-FDTD algorithm is the finite-difference time domain calculation. Because of this, openMPI [32] was used to parallelize

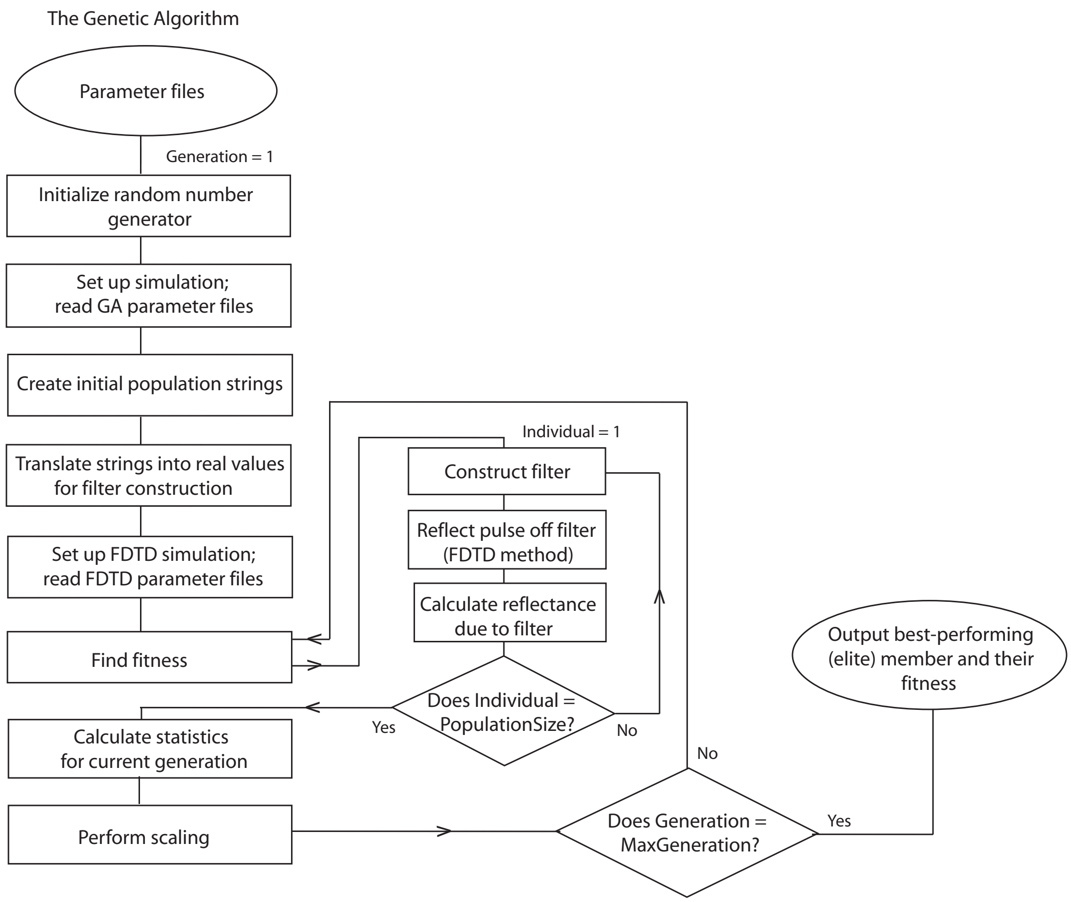

Figure 2.3: Flowchart for the dielectric filter optimization algorithm.

the algorithm by simultaneously calculating the reflected and transmitted pulses for each filter design in the population. A total of eleven processors were used: one as the master processor, and ten slave processors corresponding to the ten individuals in the population.The parallelization greatly reduced by a factor of 10 the overall run-time of the optimization. For the frequency domain optimizations, the finite-difference time domain (FDTD) step is replaced by the characteristic matrix (CM) method, which is used to calculate the reflection and transmission spectra for the thin-film structures. Appendix A gives an overview of the characteristic matrix calculations. The GA-CM algorithm executes much faster than the GA-FDTD algorithm because of the computational speed of the CM step.

For the frequency domain optimization, the reflectance spectrum of a given dielectric stack is compared to the required reflectance spectrum (that can be calculated by Fourier transforming the required time-domain pulse shape). The reciprocal of the sum of the differences squared between the two spectral functions squared gives the fitness. For some of the frequency domain optimizations, it was found useful to use a supergaussian to more heavily weight the central passband of the reflectance spectra, *i.e.*,

$$\mathcal{F} = \frac{1}{\sum_i e^{-\frac{(\omega_{mid}-\omega_i)}{\sigma}^{2m}} \left[I_{\text{ideal}}(\omega_i) - I(\omega_i) \right]^2}, \tag{2.4}$$

where ω_{mid} and m are parameters that control the location and shape (steepness) of the supergaussian within the spectral domain. Varying σ adjusts the width of the supergaussian. Results for the GA-CM hybrid appear in Chapter 4.

Pseudorandom Number Generation

One of the key components of the genetic algorithm's success comes from its ability to jump away from suboptimal solutions in particularly complex search spaces. In a large part this is due to the randomization of the initial population, crossover, and mutation sites of each individual's binary strings. The pseudorandom number generator for the algorithm produces a sequence of numbers that approximate the properties of random number ones. The generator was tested in several ways. The first test optimized the same stack, *e.g.* a 50 layer double pulse reflection dielectric stack, with a different pseudorandom number generator each time and compared the various results. It was important to check for consistency amongst several pseudorandom number generators to know that the eventual filter design would be the result of unbiased shuffling. For this I tested the Mersenne twister, the linear congruent method, and the "minimal" random number generator of Park and Miller. Further details about these generators can be found in Appendix C.

Table 2.2: Mean and standard deviation in the elite fitness for a genetic algorithm due to the choice of PNG seed. Three different pseudorandom number generators were tested.

Pseudorandom Number Generator	Mean Elite Fitness ($\bar{\mathcal{F}}$)	Standard Deviation ($\sigma_{\mathcal{F}}$)
Linear congruent method [33]	0.7537	0.1565
Mersenne Twister [34]	0.8558	0.1652
Park and Miller [35]	0.8607	0.1608

Since all of the pseudorandom number generators are initialized by a seed that starts the recursion, I also ran the genetic algorithm with the same random number generator – the minimal generator – and changed the seed with each run. Variation of the output was calculated to give an idea of fluctuation in the performance due to seed choice. Results for the seed variations are shown in Table 2.2 and give the average and standard deviation due to 10 different seed initializations per pseudorandom number generator.

Chapter 3

Time Domain Optimization

Tailored femtosecond laser pulses have found application in the precise control of microscopic processes. Pulse shaping techniques are already common practice in the areas of laser-matter interaction and materials processing. In chemistry, where coherent control of reactions is desired, the exact shaping of a laser pulse to the local mode of a specific chemical bond enables bond-cleaving reactions to take place [36]. Currently, the control of femtosecond laser pulse shapes is of fundamental importance in fields such as spectroscopy [37, 38], micromachining [39, 40], and the quantum control of wavepackets [41, 42]. Consistency and repeatability of the shaped pulses remain key issues within the experimental design.

Pulse shaping with genetic algorithms, often using a real-time feedback loop, has been demonstrated by Assion *et al.* [36] and, more recently, in the work of Nuernberger *et al.* [43]. Both groups used a grating compressor and a liquid-crystal spatial light modulator to create the spectral-phase modulations on the incident pulse. This experimental method works well for feedback-aided pulse shaping, but the set up can be somewhat time-consuming. While much work has already been done with genetic algorithms to optimize thin film filters for non-temporal applications ([44, 27, 45, 46]), to my knowledge this chapter presents the first examination of genetic algorithms for the design of pulse shaping dielectric stacks. In this chapter, I present the time domain results for the thin film dielectric stacks, thereby demonstrating an alternative means of femtosecond pulse shaping. The principle advantages over the pulse shaping methods used in the aforementioned applications are simplicity and ruggedness of the pulse shaping device.

3.1 Reflective Double Pulse Dielectric Stacks

To demonstrate the ability of the genetic algorithm in designing dielectric stacks for ultra-short pulse applications, the first series of designs were for stacks that produced double pulses upon reflection. Ten different dielectric stacks are described in this section. These devices are designed to convert a single incident pulse into two pulses spaced by Δt = 30 - 300 fs upon reflection.

The fitness function $\mathcal{F}$ for these double pulse dielectric stacks is

$$\mathcal{F} = \frac{1}{\sum_i [I_{\text{ideal}}(t_i) - I(t_i)]^2}. \tag{3.1}$$

where the intensity function, $I_{\text{ideal}}(t)$, is constructed from two σ = 30 fs Gaussian pulses, each centered at τ_j,

$$I_{\text{ideal}}(t) = \frac{1}{2}\sum_{j=1}^{2} e^{-((t-\tau_j)/\sigma)^2}. \tag{3.2}$$

The intensity, $I(t_i)$, is calculated using the finite-difference time domain method.

Performance

The double pulse dielectric stacks perform quite well. While some of the reflectances can likely be improved with further optimization, the algorithm does a good job of generating two pulses of the same intensity separated by the desired spacing. Fig. 3.1 shows the results from the FDTD calculation of the pulse-mirror interaction, with the reflected pulses (blue), the incident pulse (black), and the desired pulse train (dashed red). The calculation allowed sufficient time for the reflected pulse to return to its approximate starting location. Initial simulations used a 1D grid of 2009 steps, which corresponded to a $\lambda/20$ resolution for the 800 nm input pulse. While the Δt = 30 fs and Δt = 60 fs double pulses are too close together to be resolved completely, they are clearly of unequal height. Dielectric stacks that optimize for temporal separations greater than 60 fs begin to show clear double pulse shapes

Table 3.1: Principal design metrics of the double pulse reflection filters. Refractive index profiles are shown in Fig. 3.2. The physical thickness for the filter excludes that of the substrate.

Time Between Double Pulses (fs)	Reflectance (%)	Filter Layers	Filter Thickness (μm)
30	98.04	50	38.84
60	89.90	50	39.80
90	75.57	50	41.20
120	87.83	50	37.24
150	88.74	50	38.48
180	82.19	50	42.56
210	86.29	50	37.52
240	81.15	50	40.64
270	85.54	50	34.12
300	93.60	50	39.64

in their reflected intensity profile. In all figures, intensities are normalized in the sense that the intensity of the input pulse was 1 Wm^{-1}.

In Table 3.1 the main design parameters for the double pulse dielectric stacks are shown. All dielectric stacks use a 50 layer template, with the filter thicknesses ranging from 34.12 μm for the $\Delta t = 270$ fs stack through to 42.56 μm for the $\Delta t = 180$ fs stack. There is also a wide range in the reflectances produced by the different filters, suggesting that some of the filters would gain from further optimization. All optimizations ran for 500 generations and started with the same seed in the random number generator. These ten designs are end-padded with a substrate[1] so that the FDTD algorithm works with filters of a constant total thickness. Within this fixed total thickness of stack plus substrate, the filter thickness itself is able to vary. Individual filter thicknesses are shown in Table 3.1, while the associated designs are shown in Fig. 3.2.

[1] A MgF_2 substrate was used with the total filter + substrate thickness equalling 2009 grid spaces for a 50-layer filter.

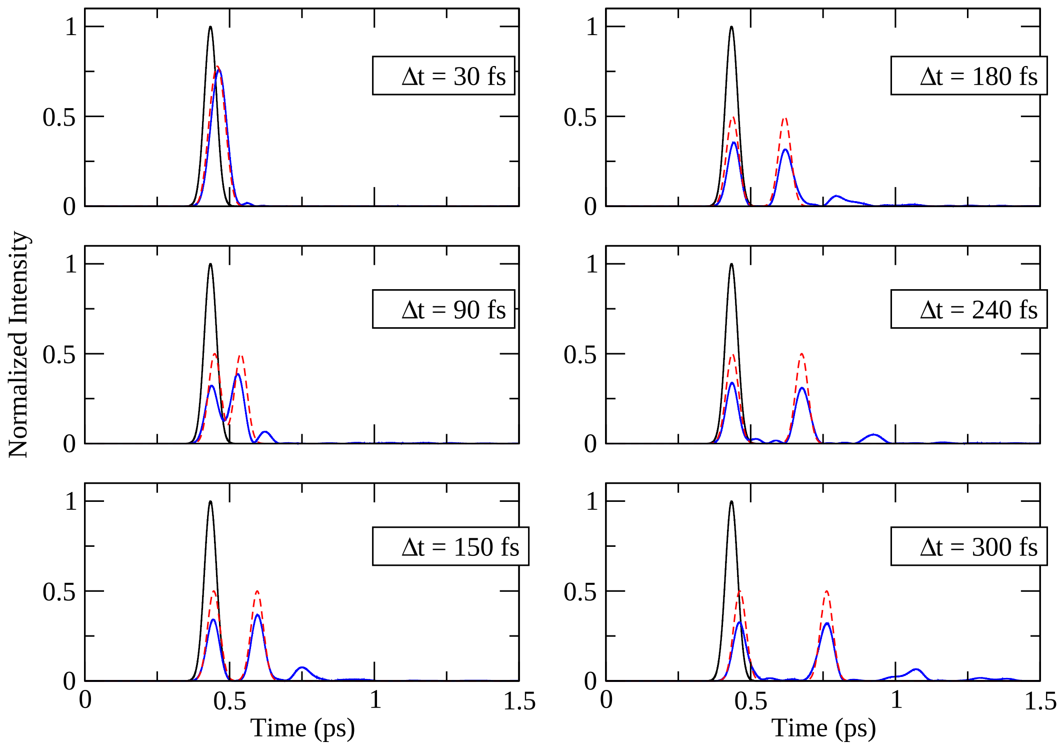

Figure 3.1: The above plots are simulations for which the target reflected double pulse was separated by increasing time intervals. The incident pulse is shown in black, while the reflected pulse is blue. The target pulse shape is shown as a dashed red line.

Fig. 3.3 shows the evolution of a double pulse reflection filter's intensity profile. The best-performing filter at the various generations from 1 through to 500 was used to generate the reflected pulse shapes shown. Note that the double pulse shape becomes more defined as the algorithm begins to converge upon better-performing solutions.

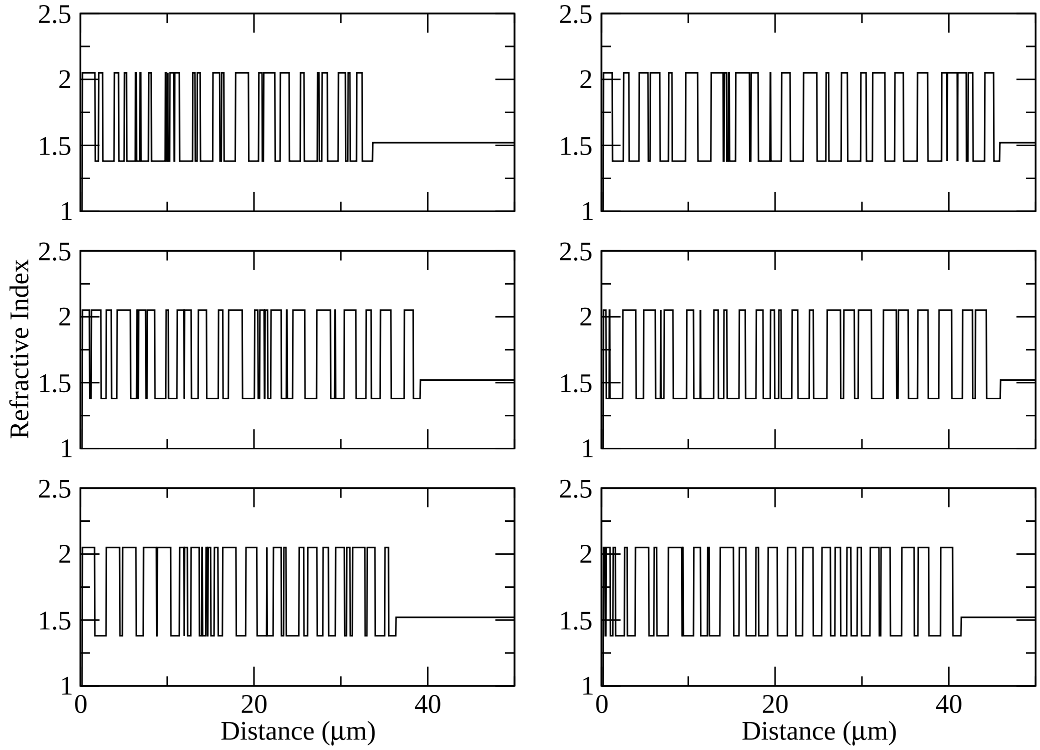

Figure 3.2: The corresponding dielectric stacks that generated the intensities in Fig. 3.1. The ordinate axis labels refractive index: $n_{\mathrm{air}} = 1.00$, $n_{\mathrm{Ta_2O_5}} = 2.05$, $n_{\mathrm{MgF_2}} = 1.38$, and $n_{\mathrm{SiO_2}} = 1.52$.

Conversion to Higher Resolution

After obtaining an optimized filter at $\lambda/20$ resolution, the design is fine-tuned for accuracy at the same central input wavelength of $\lambda = 800$ nm but at a higher grid resolution of $\lambda/200$. The three post-processing steps are illustrated in Fig. 3.4, which shows the results from a $\Delta t = 120$ fs double pulse mirror. I first rescaled the $\lambda/20$ filter generated by the genetic algorithm (upper pane) into a $\lambda/200$ version (middle pane). This amounts to adding 10

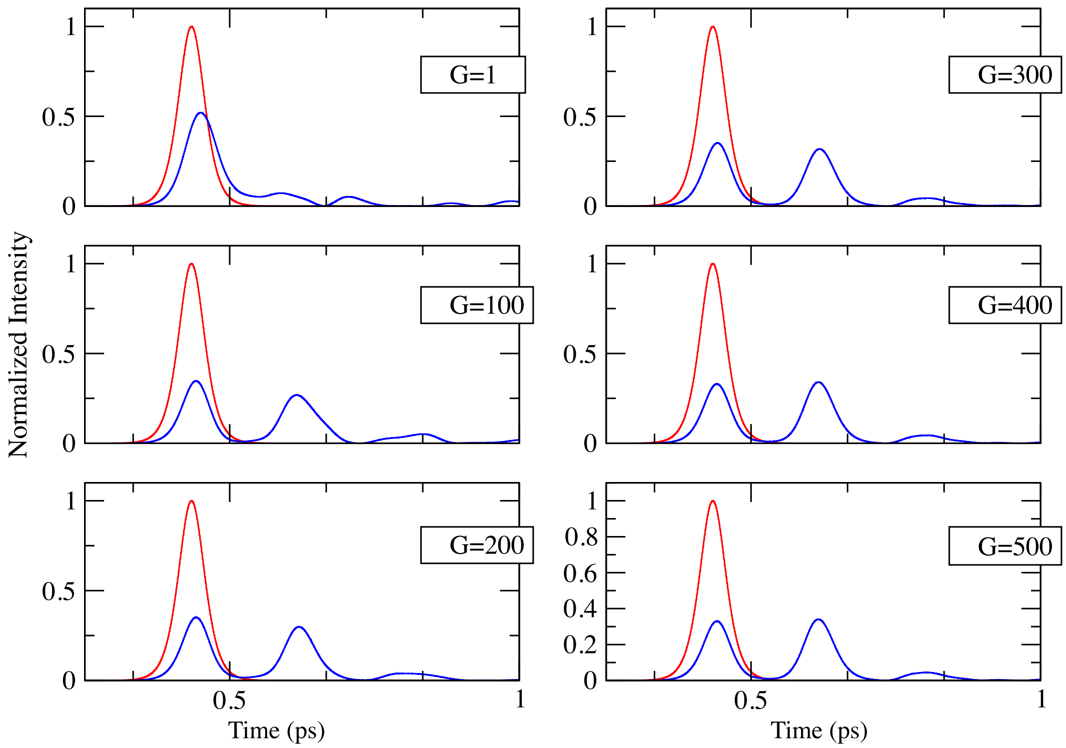

Figure 3.3: Plots showing the development of the reflected pulse for generations 1 through 500. The filter was designed to generate a double pulse from a 30 fs incident pulse. The incident pulse appears in red, while the evolving reflected pulse appears in blue.

data points on the $\lambda/200$ grid in place of one data point on the $\lambda/20$ grid. Multiple FDTD calculations, each with a different input wavelength, ran on the rescaled filter. The same metric equation (Eq. 2.1) gauged the fitness of this filter with respect to the various FDTD simulations (middle pane, inset). Using the optimized wavelength, the filter was then scaled to $\lambda/200$ resolution, by a a 1.7 % expansion of the grid, which was used to compensate for the offset in the reflectance spectra peak (786.6 nm) of the spectrum of the $\lambda/200$ resolution structure and the desired reflectance peak at 800 nm. This final step produced a filter that was optimized at high resolution for the desired 800 nm input pulse (bottom pane)

For additional validation, I calculated the reflectance spectra from both the FDTD-produced fields and independently using the characteristic matrix method [47], as shown in Fig 3.5. Despite a limit placed on the FDTD calculations by the grid resolution, the close match of reflectance properties in both the FDTD and characteristic matrix spectra of Fig. 3.5 suggests a robustness of the structure to small imperfections in its fabrication. In fact, a rough upper bound can placed on the sensitivity to construction errors: simulations with a $\lambda/20$ resolution showed strong deterioration in the features of the reflectance spectum while simulations with a $\lambda/200$ resolution produced spectra that matched rather well. Therefore, in the manufacture of such filters, inaccuracies below the order of $\lambda/200$ should still produce the desired pulse shaping effects.

The conversion of the optimized filter from a $\lambda/20$ grid to a $\lambda/200$ grid was an important problem overcome in this work. The underlying issue came from the offset between the dielectric stacks on the two different grids. Essentially, the higher resolution dielectric stack is able to make sharper transitions between the media. Overlaying a dielectric stack from the 40 nm grid with its 4 nm counterpart (scaled up by a factor of 10 for comparison) showed the mismatch in the two stacks. The reflectance spectrum from a $\lambda/20$ grid shows this mismatch when compared with that of a $\lambda/200$ grid. Essentially, the $\lambda/20$ spectrum is accurate to within $\pm\lambda/20$, while the $\lambda/200$ spectrum is accurate to within $\pm\lambda/200$. Fig. 3.5 shows this clearly. The upper pane has the reflectance spectra for a $\lambda/20$ filter calculated by both the characteristic matrix method and the $\lambda/20$ FDTD-produced spectrum. The two spectra differ by about 36 nm, which is the difference between their two grid resolutions in nm. The lower pane shows the resulting reflectance spectra when both calculations are performed at $\lambda/200$ resolution.

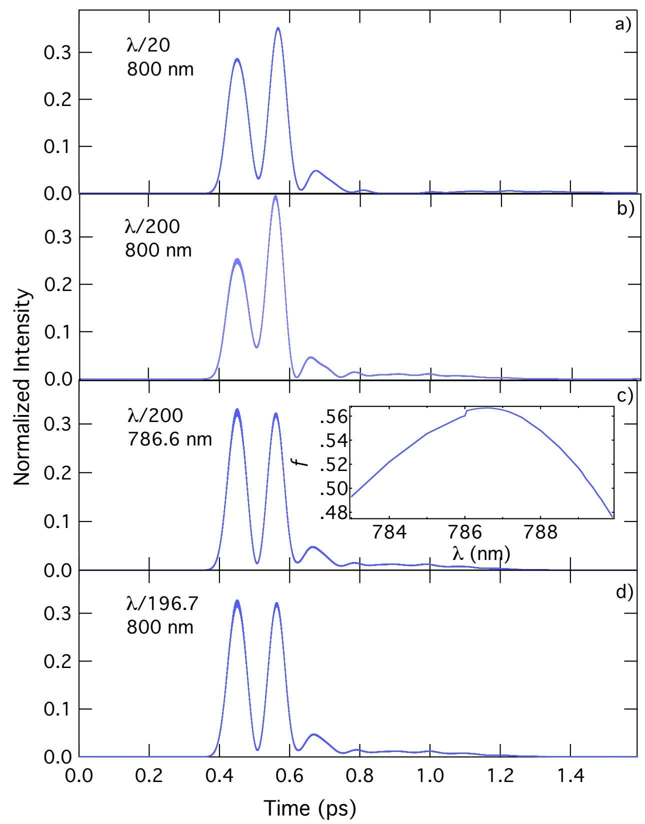

Figure 3.4: Converting the filter from a $\lambda/20$ resolution to $\lambda/200$ resolution. Pane a) shows the reflected intensities due to a filter at the $\lambda/20$ resolution. After scaling to $\lambda/200$ (pane b), wavelength tuning to 786.6 nm produced the best fitness for the filter (pane c, inset). The result for the incident wavelength at 786.6 nm appears in pane c). Pane d) displays the final result of tuning to $\lambda/200$ resolution.

Mechanisms

To better understand the pulse shaping mechanisms, I performed a case study of the $\Delta t =$ 300 fs, double pulse reflection filter. The reflected fields were calculated using the finite-difference time domain technique, as outlined above. As shown in Fig. 3.5, the reflectance spectrum comprises many sharp features, with no particular symmetry about the central 800 nm wavelength. The spectrum lacks a dominant central reflectance band – often seen

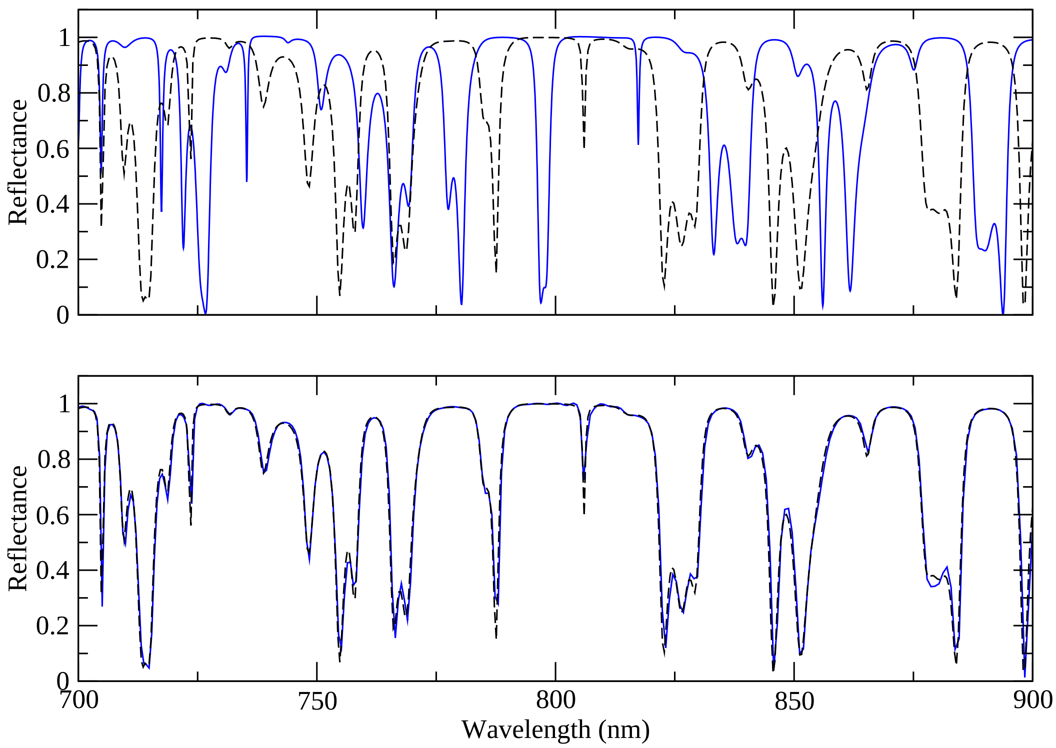

Figure 3.5: Reflectance spectrum for the $\Delta t = 300$ fs, double pulse reflection filter. The dashed line shows the spectrum calculated by the characteristic matrix method, while the solid blue line comes from the reflected field that was produced in the FDTD simulation. The upper pane shows the mismatch between the two methods when calculated at $\lambda/20$ resolution. The lower pane shows the same calculations performed at a $\lambda/200$ resolution.

with dielectric stack spectra – but it does show strong reflection in the region of 800 nm as specified in the optimization of this particular filter. While the reflectance spectra provide a useful method of characterizing the dielectric stacks, they do not explain the mechanism responsible for the dielectric stacks' pulse shaping ability. One current strategy for ultrashort pulse shaping mirrors is to design a mirror with a large group delay dispersion (GDD). A large group delay dispersion converts an ultrashort pulse into a longer pulse with a large chirp. To see if group delay dispersion played an important role in the operation of in these filter designs, I calculated the group delay dispersion of these devices.

The group delay dispersion at wavelength λ, D_λ, of an optical element is proportional to the second derivative of the spectral phase, ϕ, with respect to frequency, ω, is given by [48],

$$D_\lambda = \frac{\partial^2 \phi}{\partial \omega^2}. \tag{3.3}$$

Eq. 3.3 was used to calculate the group delay dispersion over the central frequencies of the incident pulse, while the corresponding result of the group delay dispersion as a function of wavelength for the case study filter is shown in Fig. 3.6 a). Note that the group delay dispersion is particularly small in the vicinity of the filter's central wavelength. In Fig. 3.6 the GDD spans the range $\pm$ 5 fs over the incident pulse frequencies of 760 - 840 nm. The third order dispersion (TOD) over the same spectral region is also shown in Fig. 3.6 b). The third order dispersion, T_λ, results from the frequency dependence of group delay dispersion [48],

$$T_\lambda = \frac{\partial^3 \phi}{\partial \omega^3} \tag{3.4}$$

and, if of a large enough magnitude, it can also contribute to pulse broadening. One can see from Fig. 3.6 that the values of the third order dispersion are on the order of $\pm$ 10 fs. For comparison one can consider fused silica intra-cavity prism pairs, which are used to create ultra-short pulses. These have group delay dispersion of order -862 fs^2 and third order

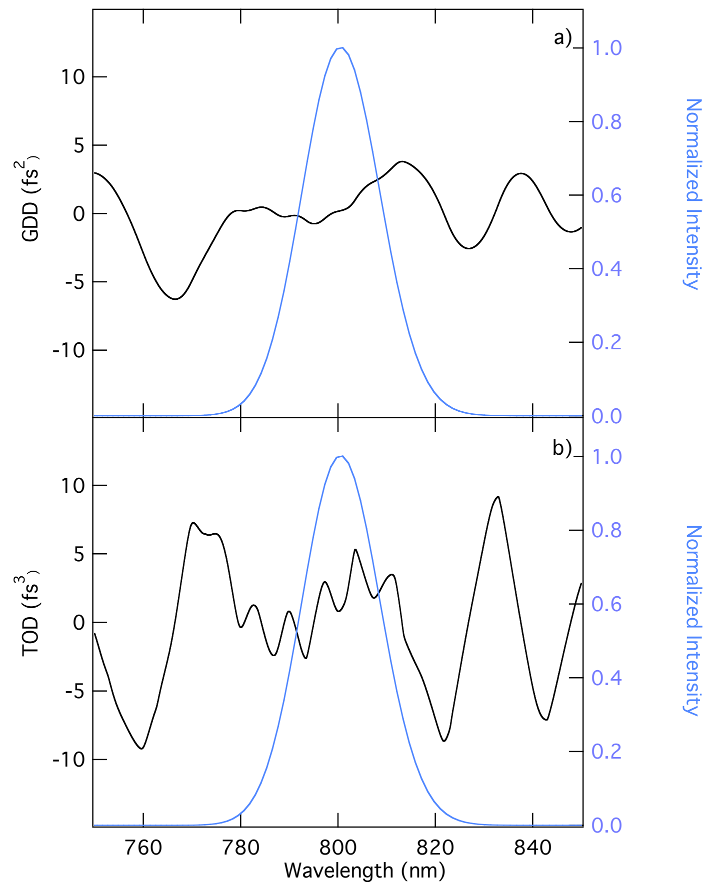

Figure 3.6: Group delay dispersion (pane a) and third order derivative (pane b) as functions of wavelength for the case study filter – a double pulse reflection filter with $\Delta t = 300$ fs. Both are plotted against the 30 fs Gaussian pulse bandwidth, to show the region of interest.

dispersion on the order of -970 fs^3, so from the above it appears that neither GDD nor TOD contribute in a major way to the pulse shaping mechanism employed by these devices [49].

As additional confirmation that GDD was not responsible for pulse shaping, spectrograms were calculated from the reflected pulses. A Gabor transform provides the measure of the pulse's frequency and phase content for local sections of the pulse over time,

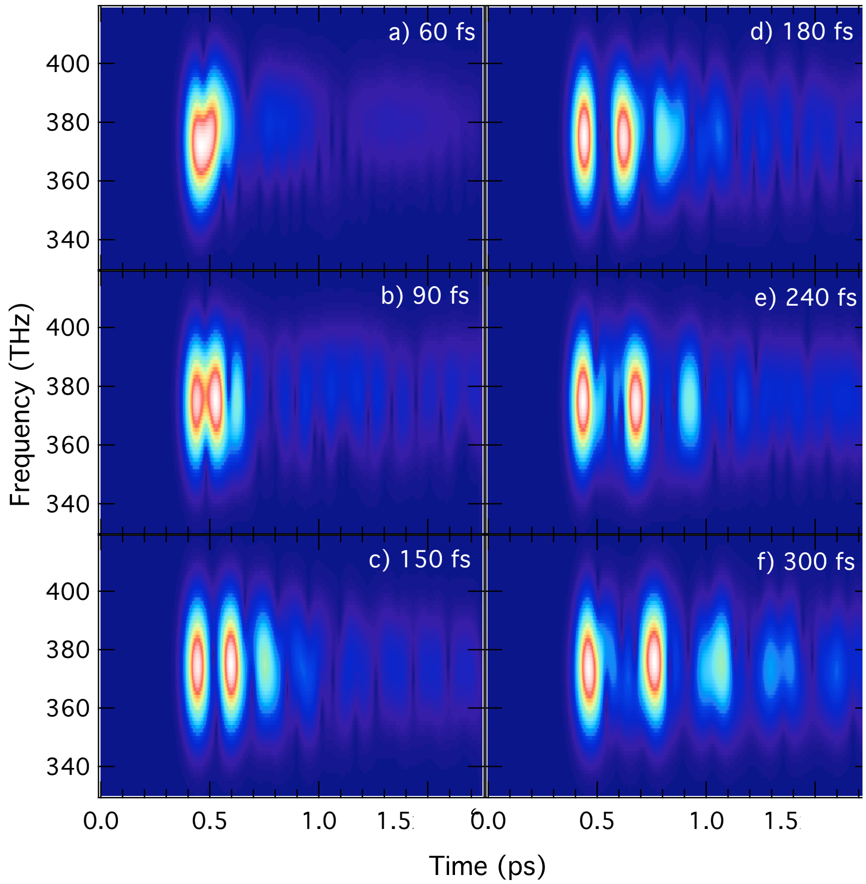

Figure 3.7: Spectrograms showing the resultant frequency distribution of the incident 30 fs Gaussian pulse after the pulse shaping effects of the double-pulse mirrors.

$$G_x(t, f) = \int_{-\infty}^{\infty} e^{-\pi(\tau - t)^2} e^{-2\pi i f \tau} x(\tau) d\tau. \tag{3.5}$$

τ is the time parameter of the original function, $x(\tau)$, and f is the frequency. A Gabor transform [50] represents a special case of the short-time Fourier transform, in which a Gaussian windowing function is used. A Gabor transform as implemented in the commercial software package Igor Pro [51] was used to calculate the spectrograms in Figs. 3.7, 3.13, and 3.17.

If group delay dispersion was responsible for the pulse shaping, then a spectrogram would show pulses with dramatically different central frequencies. The spectrograms in Fig. 3.7 show no major frequency shift between the leading and trailing sections of the double pulse, and so GDD is ruled out as the major pulse shaping mechanism.

Multiple Mirrors

Another possible pulse shaping mechanism is the existence of two optimally-spaced mirror substructures within the filter that each reflect a pulse of the appropriate intensity. For example, a two mirror structure with $R_1 = 0.326$ ($T_1 = 0.674$) and $R_2 = 0.718$ ($T_2 = 0.282$) would produce a pulse train with the first five pulses having the following intensities:

$$\begin{aligned}
I_1 &= R_1 = 0.326 \\
I_2 &= T_1 R_2 T_1 = 0.326 \\
I_3 &= T_1 R_2 R_1 R_2 T_1 = 0.076 \\
I_4 &= T_1 R_2 R_1 R_2 R_1 R_2 T_1 = 0.018 \\
I_5 &= T_1 R_2 R_1 R_2 R_1 R_2 R_1 R_2 T_1 = 0.004 \qquad (3.6)
\end{aligned}$$

This is in reasonable agreement with the five pulses that are observed in Fig. 3.8 d), *i.e.*,

$$\begin{aligned}
I_1 &= 0.326 \\
I_2 &= 0.321 \\
I_3 &= 0.066 \\
I_4 &= 0.017 \\
I_5 &= 0.009, \qquad (3.7)
\end{aligned}$$

strongly indicating that the mechanism found by the optimization routine involves two mirrors spaced by an optical path difference of $\Lambda = 45\ \mu$m. Upon closer examination of filter sublayers, the two mirrors corresponding to the double pulse peaks are identified, as indicated in Fig. 3.8 b) and Fig. 3.8 c) with reflections $R_1 = 0.326$ and $R_2 = 0.321$. The optical path difference between these two mirrors as calculated from the stack structure in Fig. 3.8 a) is 37 μm. If the remainder of the stack from 65 μm to the substrate is included in the second mirror, the peak intensity is increased, as is the width of the pulse. Furthermore, the effective optical path difference between the two mirrors will also be increased. From the above we can infer that the optimization algorithm has produced a device that to zeroth order is based upon a two mirror structure in which the intermediate and trailing layers of the full dielectric stack play a role in delaying and reshaping the second pulse.

3.2 Transmissive Double Pulse Dielectric Stacks

Having seen the results of the reflected double pulse simulations, the next step was to see how well the genetic algorithm could design a filter for transmitted pulse shaping. Conventional transmission coatings originate from a single quarter-wave film with refractive index equal to that of the square root of the substrate. It is not always easy to find a suitable material that gives the needed refractive index, which is why multilayer coatings are more appropriate for many transmission cases. Cid *et al.* [52] showed that multilayer coatings are necessary for general transmissive applications because of the inherently narrow bandwidth provided by a single-layer coating.

The transmissive dielectric stacks used the same number of layers and materials as the double pulse reflection filters. For comparison with the double pulse reflection filters, the genetic algorithm used the same population (10 individuals) and optimized for the same time (500 generations). As shown in Fig. 3.9, the transmission filters exhibited rather poor transmission (37.30% to 61.48%) and extremely poor pulse shape. Fig. 3.9 shows the resultant dielectric stacks generated by the genetic algorithm. The multiple mirror scenario used to describe the double pulse reflection filters, may not apply to transmission filters. It is also possible that many more layers or different material choices are required to achieve pulse

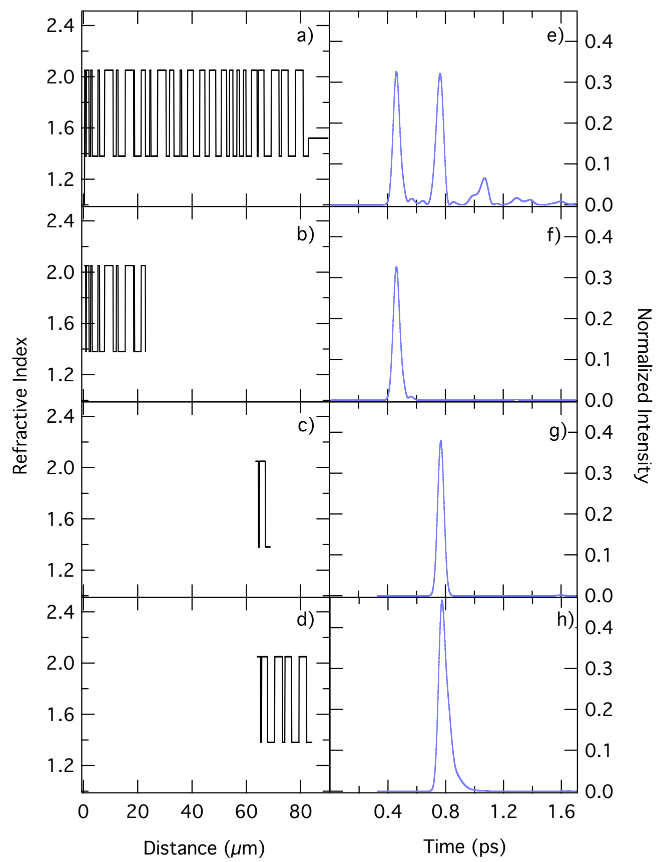

Figure 3.8: Identification of two mirrors within the case study filter. Pane a) is the full filter. Panes b) shows the section of the filter responsible for the first mirror. Panes c) and d) show the sections of the filter that contribute to the second mirror. Panes e)–h) show the reflected pulses generated by the filters shown in panes a) – d). Note that the time delay of the pulses shown in panes f) and h) have been artificially shifted in time to show their correspondence with the second pulse in pane e).

shaping in transmission. However, this search space was left unexplored in favor of pursuing the reflectance dielectric stacks in more detail. The remainder of this dissertation focuses on improving and further testing the performance of the reflectance dielectric stacks.

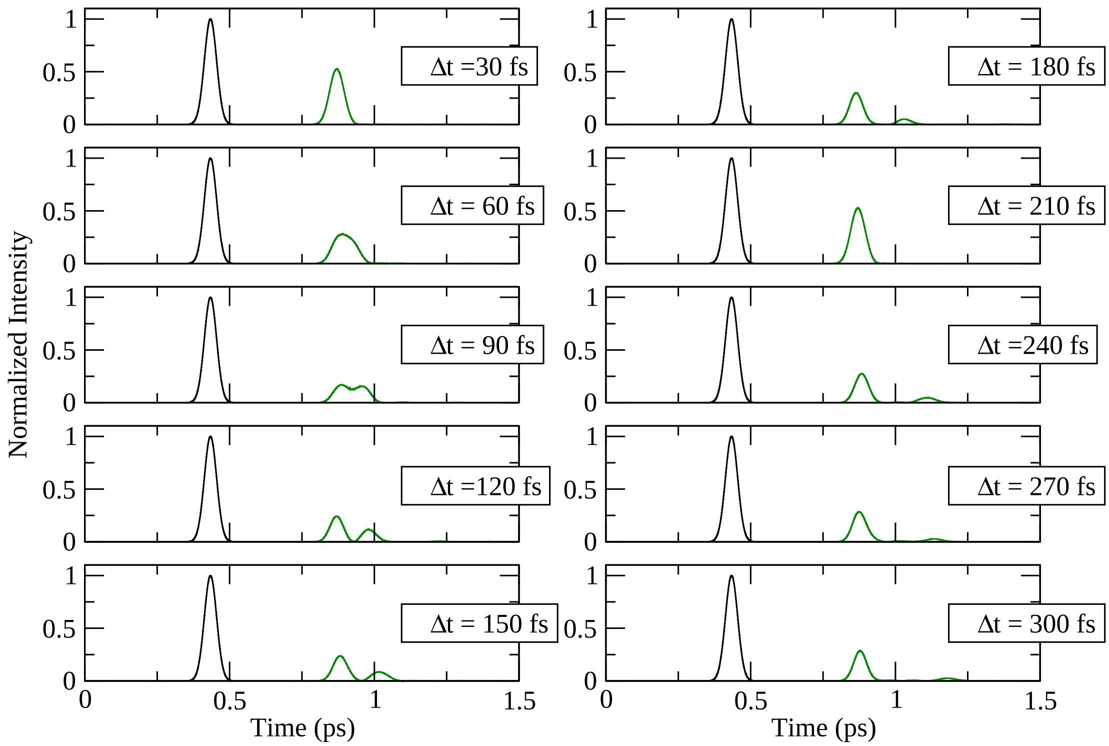

Figure 3.9: These simulations optimized for a double pulse produced in transmission. The incident pulse is shown in black while the transmitted pulse is shown in green.

3.3 Other Reflective Multipulse Dielectric Stacks

Reflective Triple Pulse Dielectric Stacks

Following the success of the double pulse mirrors, I explored the performance of the dielectric stacks that were designed to reflect a triple pulse train. The initial triple pulse designs were based upon 50 layer templates. It was found that the reflectivity of these devices degraded for large Δt since the dielectric stack needs to have a sufficient thickness to accommodate

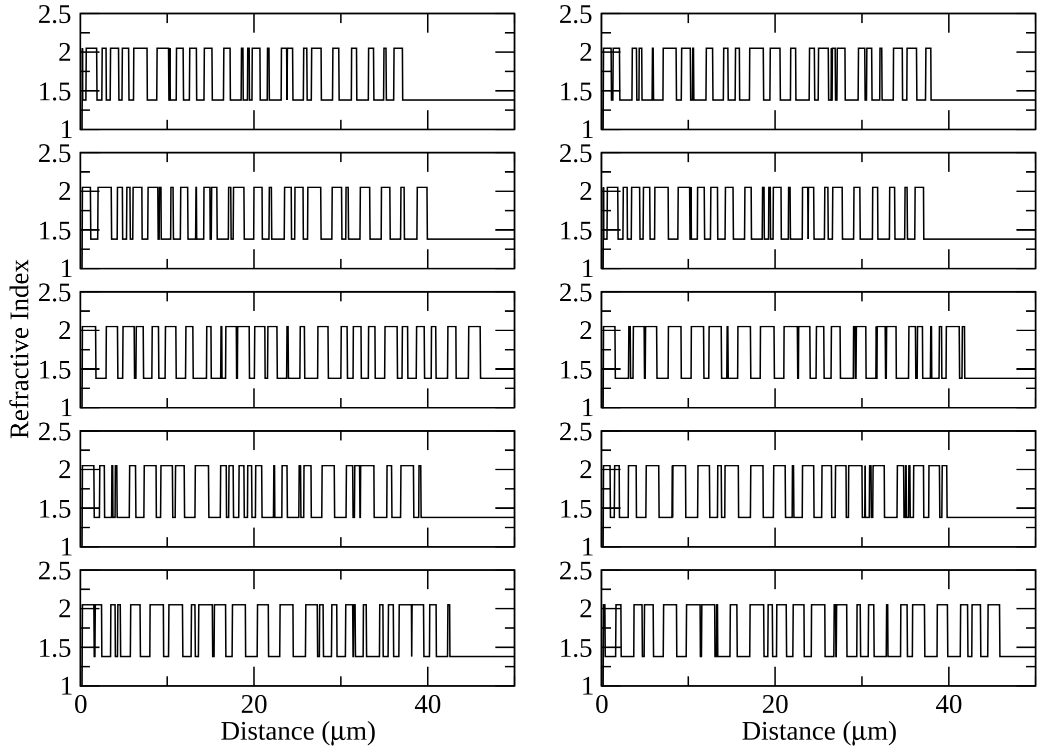

Figure 3.10: The corresponding filters that generated the intensities in Fig. 3.9. The ordinate axis labels refractive index: $n_{\text{air}} = 1.00$, $n_{\text{Ta}_2\text{O}_5} = 2.05$, $n_{\text{MgF}_2} = 1.38$, and $n_{\text{SiO}_2} = 1.52$.

the optical path distance required to delay the third reflected pulse. For that reason, all the simulations presented in this section use 100 layer templates. While 100 layers may not be the optimal choice, it provided a sufficient thickness for triple pulses to be generated with comparable reflectance. The filter parameters for this section are summarized in Table 3.2 while the results themselves are given in Fig. 3.11.

The fitness function $\mathcal{F}$ for the triple pulse dielectric stacks is again of the form

Table 3.2: Principal design metrics of the triple pulse reflection filters. Refractive index profiles are shown in Fig. 3.12. The physical thickness for the filter excludes that of the substrate.

Time Between Triple Pulses (fs)	Reflectance %	Filter Layers	Filter Thickness (μm)
60	91.25	100	86.78
90	95.12	100	77.44
120	94.75	100	78.20
180	91.52	100	77.16
210	82.83	100	84.24
240	88.78	100	81.20

$$\mathcal{F} = \frac{1}{\sum_i [I_{\text{ideal}}(t_i) - I(t_i)]^2} \tag{3.8}$$

where the intensity function, I_{ideal}, is constructed from three $\sigma = 30$ fs Gaussian pulses, each centered at τ_j,

$$I_{\text{ideal}}(t) = \frac{1}{3}\sum_{j=1}^{3} e^{-((t-\tau_j)/\sigma)^2}. \tag{3.9}$$

Performance

Fig. 3.11 shows the typical reflected pulse (blue) and the incident pulse (red) for the triple pulse simulations. Again, intensities are normalized to that of the input pulse. The data is obtained with an FDTD calculation that allows sufficient time for the reflected pulse to return to its approximate starting location. Since a 100 layer thickness is modeled for the triple pulses, the 1D grid is increased to 4009 steps (*c.f.* 2009 steps for the 50 layer filters). The grid resolution remains at $\lambda/20$ for the 800 nm input pulse. Filter thicknesses range from 77.44 μm for the $\Delta t = 90$ fs mirror to 86.78 μm for the $\Delta t = 60$ fs mirror. On average the reflectances for the 100 layer triple pulse mirrors are higher than those of the 50 layer double pulse mirrors.

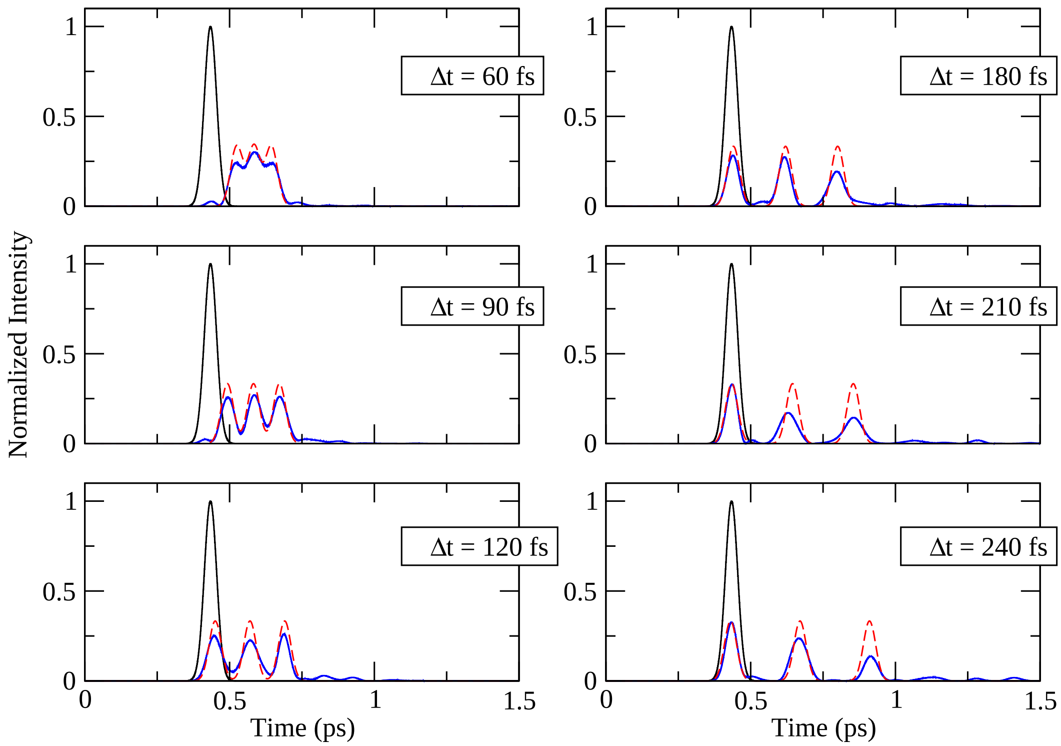

Figure 3.11: The incident (black) and reflected (blue) intensities for triple pulse mirrors. The consecutive pulse separation of the target function (dashed red line) ranges from 30 to 120 fs.

Pulse Stretching Mirrors

The motivation here was to see how well the algorithm could design a filter capable of significantly stretching a pulse upon reflection. The algorithm and filter parameters were kept the same (population = 10, generations = 500, and layers = 50) for comparison with the double pulse filters. In all simulations the incident medium is air, the substrate is glass and the filter materials are tantalum pentoxide and magnesium fluoride, as before.

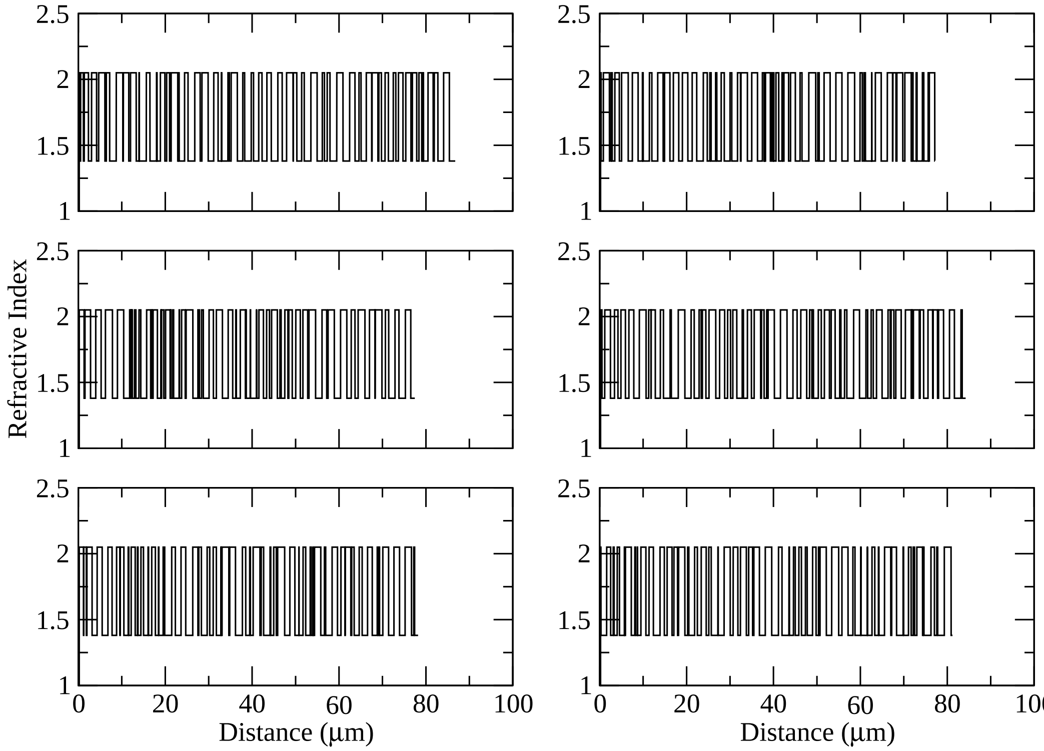

Figure 3.12: The corresponding filters that generated the intensities in Fig. 3.11. The ordinate axis labels refractive index: $n_{\text{air}} = 1.00$, $n_{\text{Ta}_2\text{O}_5} = 2.05$, $n_{\text{MgF}_2} = 1.38$, and $n_{\text{SiO}_2} = 1.52$.

The fitness function $\mathcal{F}$ for the pulse-stretching dielectric stacks again took the form

$$\mathcal{F} = \frac{1}{\sum_i [I_{\text{ideal}}(t_i) - I(t_i)]^2} \tag{3.10}$$

where the intensity function, I_{ideal}, was now constructed from multiple $\sigma = 30$ fs Gaussian pulses, each centered at τ_j,

$$I_{\text{ideal}}(t) = \frac{1}{N} \sum_{j=1}^{N} A_j e^{-((t-\tau_j)/\sigma)^2}. \tag{3.11}$$

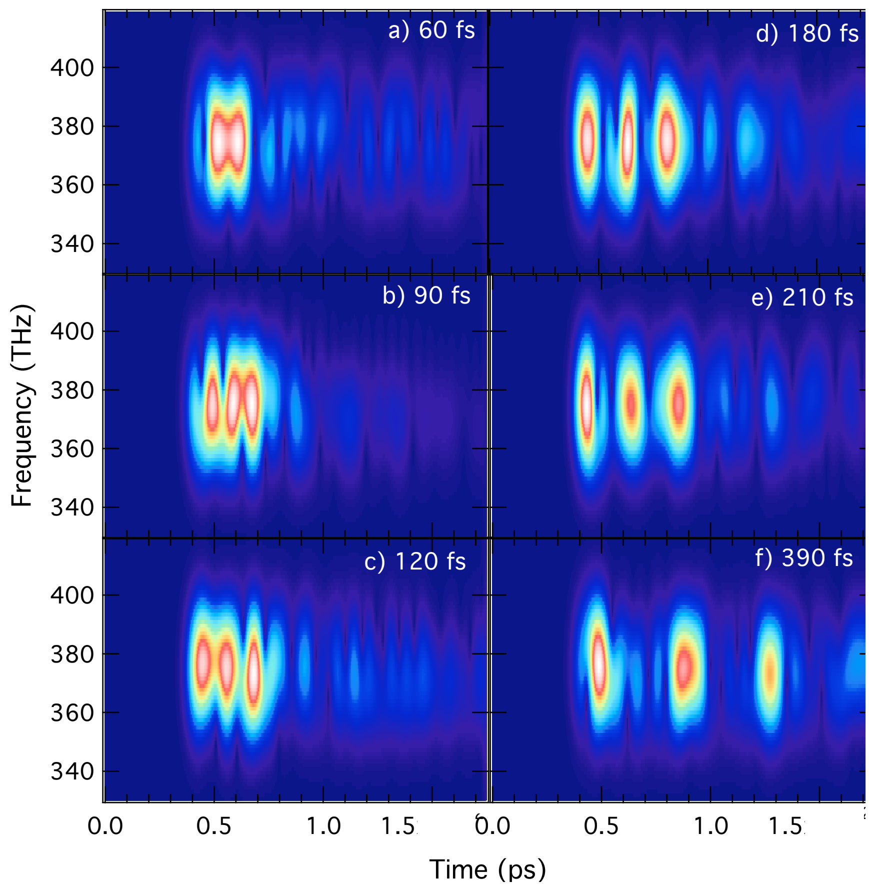

Figure 3.13: Spectrograms corresponding to the triple pulses shown in Fig. 3.11.

Performance

Table 3.3 shows the main design parameters of the dielectric stack for the pulse stretching mirrors. The associated designs are in Fig. 3.16. All ten mirrors are end-padded with a substrate to a fixed total thickness, and within that total thickness the optimization algorithm allowes for variation of the filter thickness itself, as shown in Table 3.3. It is interesting to note the effect of the 50 layer template on the longer target pulse shapes. In Fig. 3.15, panes a), b), and c) show the reflected pulse matching the width of the target function, while panes

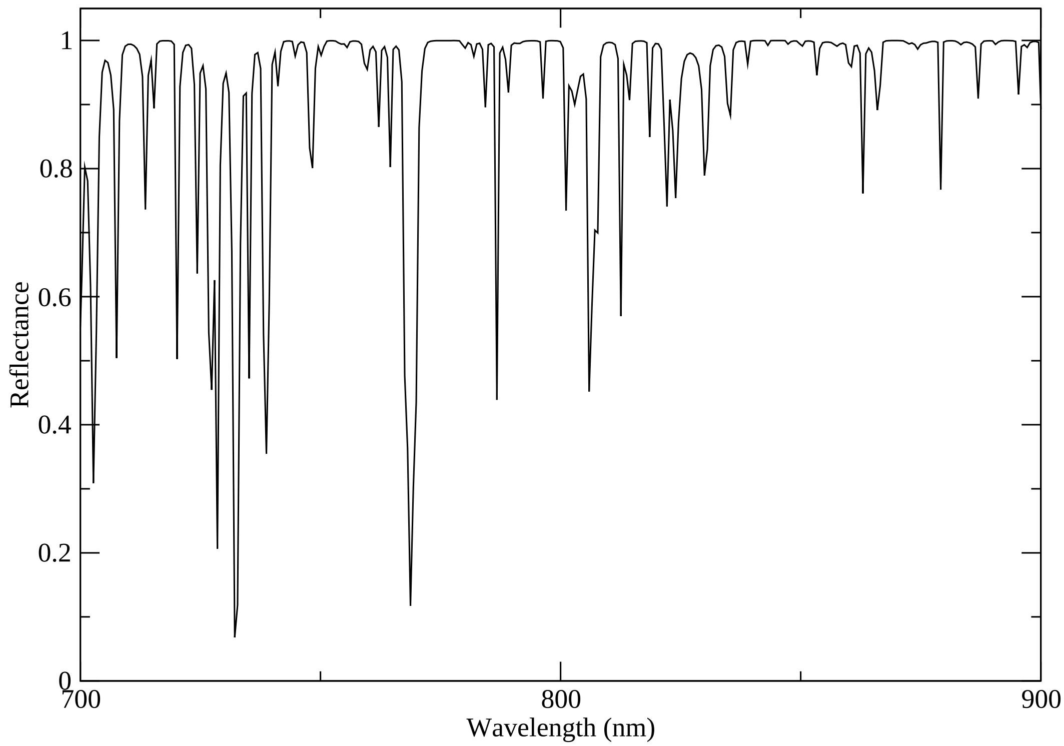

Figure 3.14: Reflectance spectrum for the $\Delta t = 120$ fs triple pulse filter shown in Fig. 3.11.

d), e), and f) show that the 50 layer dielectric stack (due to its short optical thickness) is unable to stretch the pulse to the requisite length. As a consequence, only the reflectance of the $N = 5$, 10, and 20 dielectric stacks maintain comparably high reflectances. Table 3.3 shows a decrease in the reflectance for increased target length, as expected from the discussion above.

Table 3.3: Principal design metrics of the higher-pulsed reflection filters. Refractive index profiles are shown in Fig. 3.16. The physical thickness of the filter excludes that of the substrate.

Number of Overlapping Pulses	Reflectance %	Filter Layers	Filter Thickness (μm)
3	94.25	50	37.96
5	91.39	50	37.60
8	96.05	50	36. 48
10	89.98	50	43.84
15	89.52	50	40.24
20	84.57	50	45.48
30	78.94	50	43.00
40	55.49	50	45.60
50	50.17	50	44.12
60	62.09	50	44.12

Optimization Enhancement Strategies

In an effort to improve the reflectances for the pulse stretching dielectric mirrors, a 25 layer and, separately, a 40 layer quarter-wave stack were incorporated into the filter template. Initially a pulse reflected from a pre-optimized structure with an added 25 layer quarter-wave stack was modelled, but the reflected pulse shapes deteriorated, due to the quarter-wave stack reflecting extra light back through the filter. This extra back-reflected light canceled some of the constructive, pulse shaping fields. To avoid this problem, the filter was optimized with a fixed quarter-wave stack in place.

The results of the combined filter and quarter-wave stack were marginally higher in reflectance than the stand-alone filter design. The addition of a 25 layer stack to the optimization template created a 1% improvement in reflectance over the solo filter, while the 40 layer stack produced a 2% improvement. In both cases the reflectance of the combination was between that of the quarter-wave stack and the stand-alone filter. These results are summarized in Table 3.4.

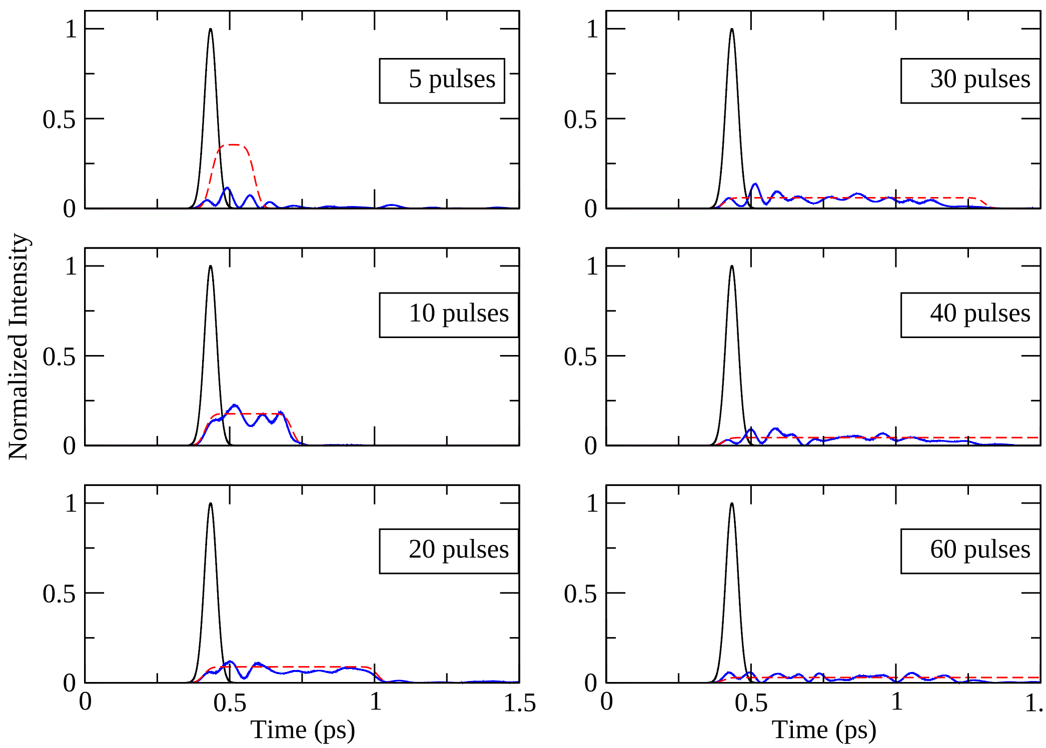

Figure 3.15: These simulations optimized for pulses that were stretched upon reflection. The incident pulse is shown in black while the reflected pulse is blue. The target function (dashed red line) was constructed from overlapping N Gaussian pulses, where N ranges from 3 to 60.

Table 3.4: Reflectivities for multipulsed reflection filters with quarter-wave stacks incorporated into the design template.

Layers in additional quarter-wave stack	Reflectance of filter and stack (%)	Reflectance of stack only (%)
0	96.23	0.00
25	97.76	99.98
40	98.47	99.99

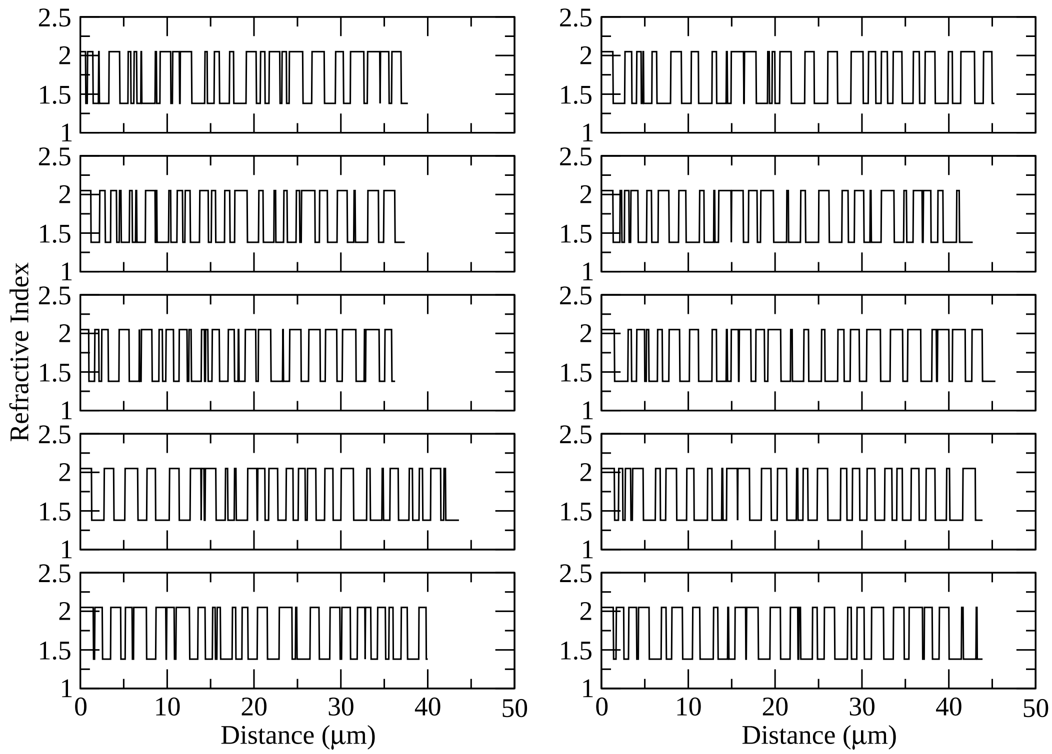

Figure 3.16: The corresponding filters that generated the intensities in Fig. 3.15. The ordinate axis labels the refractive index: $n_{\mathrm{air}} = 1.0$, $n_{\mathrm{Ta_2O_5}} = 2.05$, and $n_{\mathrm{MgF_2}} = 1.38$.

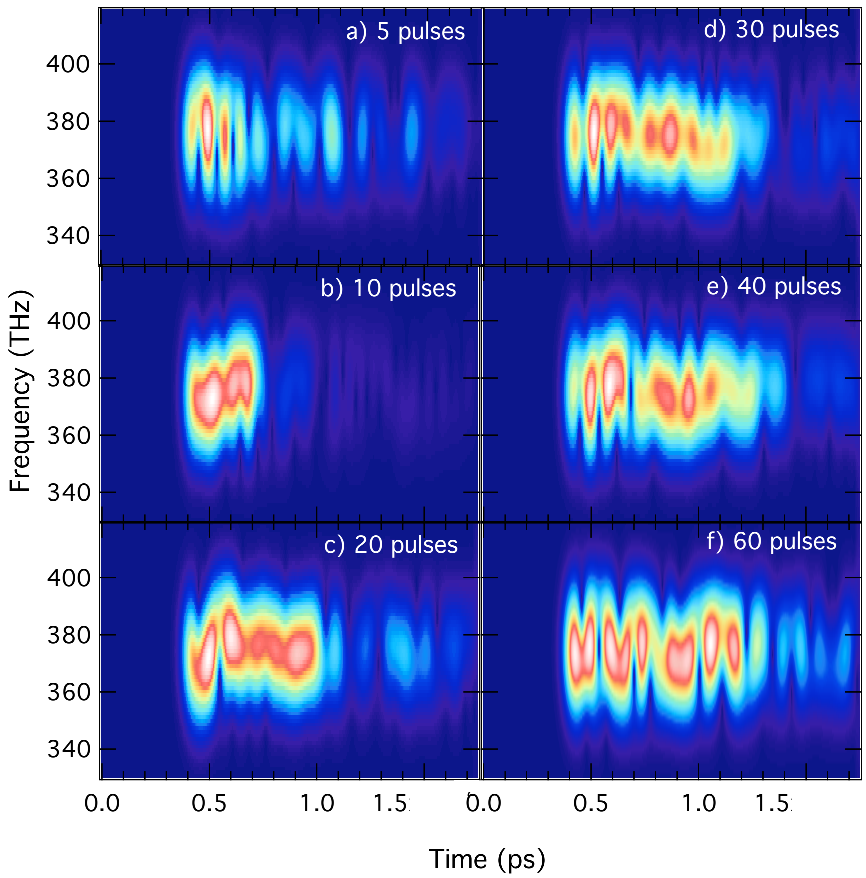

Figure 3.17: Spectrograms corresponding to the pulses shown in Fig. 3.15.

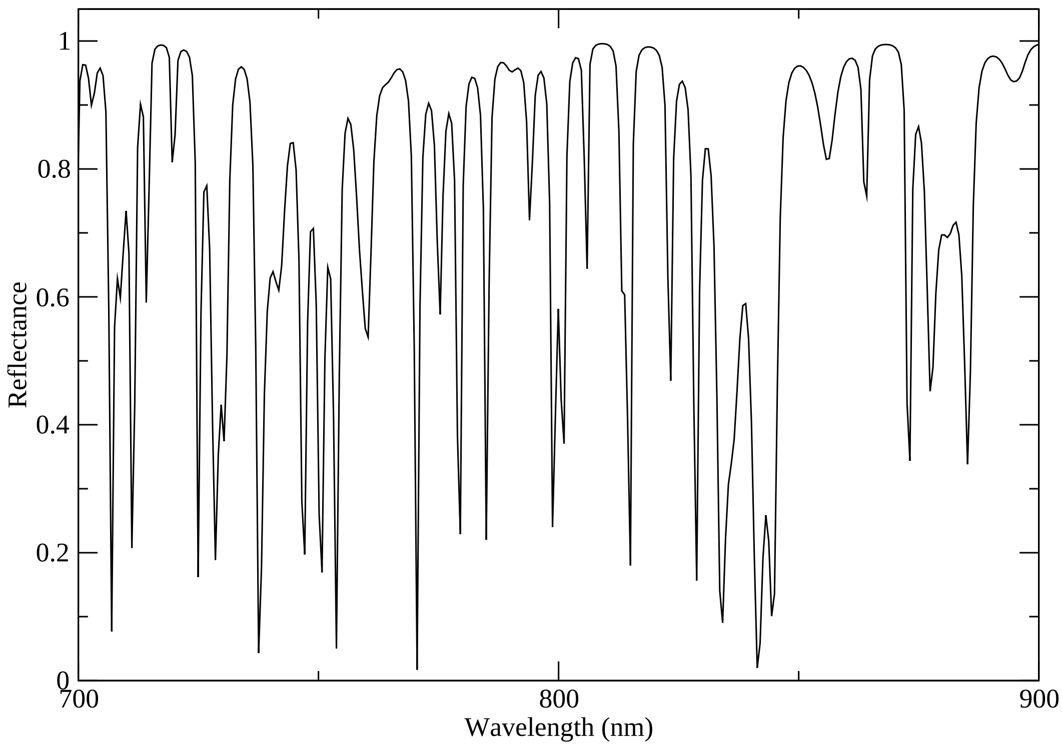

Figure 3.18: Reflectance spectrum for the N = 20 pulse stretching filter shown in Fig. 3.15.

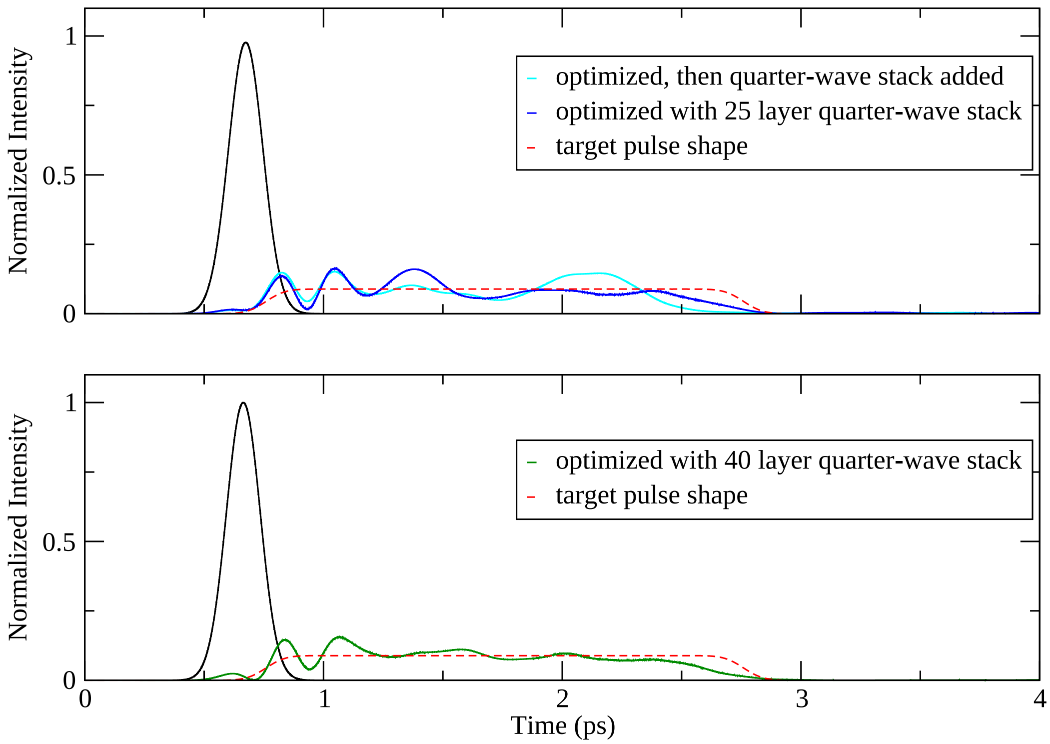

Figure 3.19: Resultant pulses from the incorporation of a quarter-wave stack to the optimized filter design. In both cases the input pulse is black and the target function is shown by a dashed red line. In the upper plot, the reflected pulse due to a filter optimized with a 25 layer quarter-wave stack addition appears in blue. The result from an already-optimized filter that then had a 25 layer quarter-wave stack affixed to it is shown in cyan. In the lower plot, the green line shows the the reflected pulse due to a filter with a 40 layer stack attached.

Chapter 4

Frequency Domain Optimization

Since pulse shaping calculations in the time domain are computationally expensive and the underlying optical problem is linear, an attractive alternative to the approach described in Chapter 3 is to calculate the reflectance spectrum for these pulse shaping filters. One can then optimize in the frequency domain using the computationally efficient characteristic matrix method to calculate the reflectance spectra from the individual filter designs. The only difference between the two methods used is the replacement of the finite-difference time domain method by the characteristic matrix method. The genetic algorithm uses the characteristic matrix method to calculate the fitness for each filter in frequency space. Apart from running on a single processor, the genetic algorithm itself remains unchanged for this section.

The first effort in the frequency domain was to design filters according to the spectral profiles of the double pulse filter described in Chapter 3. The genetic algorithm struggled to provide a good match to the target function, so a second optimization method – simulated annealing – was introduced for comparison with the genetic algorithm. An introduction to this method is provided in this chapter as well as a comparison of the double pulse spectral results. Since neither optimization strategy in the frequency domain performed as well as had been expected, much of this chapter is devoted to understanding the limitations of the frequency domain optimization.

However, before describing these investigations, this following section provides a summary of the double pulse train results in the frequency domain.

4.1 An Overview of Some Frequency Domain Results

The literature often describes the spectral properties of filters in terms of reflectance or transmittance spectra, and it is here that one can classify filters as notch, broadband, lowpass and highpass. Frequency domain genetic algorithms began appearing in filter applications in the 1990's with a work by Michielssen, Ranjithan, and Mittra [53]. In that work, a real-coded algorithm was used to design lowpass and high pass filters of alternating lead telluride (PbTe, $n_H = 5.10$) and zinc sulfide (ZnS, $n_L = 2.20$) layers with a germanium substrate (Ge, $n_{sub} = 4.00$). The filters matched their target spectral functions with some success and the performance of the real-coded genetic algorithm compared favorably with other then-contemporary methods. From that time onward, the literature quickly expanded to test the performance of more elaborate genetic algorithms, using more involved real-coded versions, family competition evolutionary algorithms [54], and genetic algorithms that incorporated post-processing gradient refinement methods.

Often the research that sustained binary-coded genetic algorithms began to design filters in which the refractive index of the layers could assume any value within a range. This effectively expanded the space in which the algorithm could search. In fact, both the use of real-coded algorithms and the progression toward the possibility of inhomogeneous layers allowed the dielectric stack design to move toward continuous parameterization of the layers, which is considered by some to be a more natural way to design a dielectric stack. The disadvantage of designing a filter with inhomogeneous layers is that while it permits more sophisticated spectra to be matched, the filters themselves are rather delicate to manufacture. Careful monitoring is required in the deposition stages to control and change the refractive index as a layer.

4.2 Double Pulse Filters in the Frequency Domain: A Comparison with the Time Domain Optimization Strategy

The only difference between the frequency domain optimization code and the time domain optimization code used in this dissertation is the replacement of the finite-difference time domain calculation for the reflected pulse profile with a characteristic matrix calculation of the reflectance within the fitness function. For comparison with the results of Chapter 3, the same materials, layer numbers, population size and maximum generation are used by the algorithm, *i.e.* 50 layer filters constructed from MgF_2 and SiO_2. A population of 10 evolves for 500 generations of the genetic algorithm. It is worth noting that while the time domain optimizations of Chapter 3 run for about an hour using 11 processors, these frequency domain optimizations are dramatically faster, taking approximately 1.5 minutes on a single processor. The speed of calculation in the frequency domain makes this approach very appealing for the design of pulse shaping dielectric stacks.

To produce a double pulse target function in the frequency domain, I selected as my starting point the target function from the $\Delta t = 150$ fs double pulse reflection filter. I Fourier transformed both the incident and target pulses and took the ratio of the magnitude squared to produce a reflectance spectrum. Fig. 4.2 shows the results from the frequency domain simulations that used this target function. As can be seen from the figures, the optimization was a complete failure. The frequency optimization code was then modified to use the simulated annealing method to see if this optimization algorithm would be more effective.

The Simulated Annealing Method

Like the genetic algorithm, simulated annealing is a stochastic optimization method. For the class of problems that contain multi-dimentional, nonlinear, or discontinuous functions stochastic algorithms are often a successful way of finding global minima [55]. Simulated annealing works well when the search space is large, since it has the ability to explore

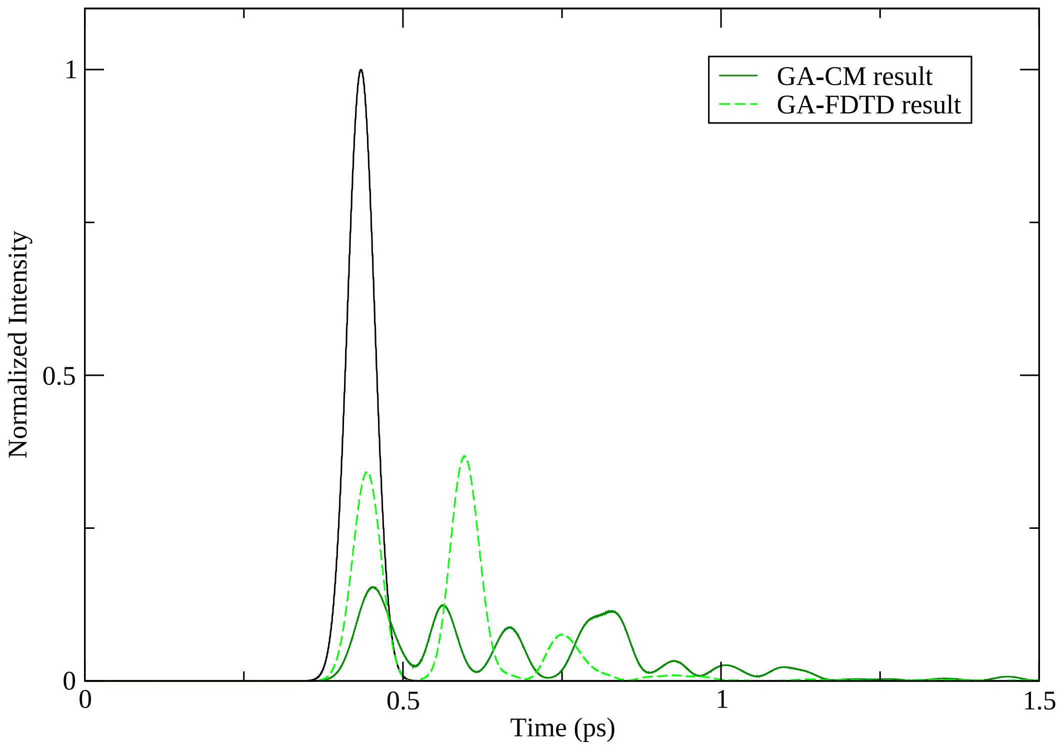

Figure 4.1: The frequency domain genetic algorithm result (green solid line) compared with the time domain genetic algorithm (light green dashed line) for the case of a double pulse filter. The incident pulse for both simulations is shown in black.

big sections of the space with its step sizes. The name refers to its heuristic analogy with metallurgic annealing, a technique that involves the heating and controlled cooling of a material to help the constituent atoms reach a lower energy configuration. The heating step allows the atoms to move from their initial positions and explore more energetic states in the internal energy of the system. The cooling process allows the atoms to come to rest in configurations of lower internal energy, permitting a more relaxed configuration to be found.

By analogy with this physical process, simulated annealing uses an artificial temperature to control the probability of accepting an uphill (more energetic) step within the search space. As the system is "heated," it accepts more energetic solutions with a higher probability than before. This enables the optimization to step outside of local minima and search farther in the space for a better solution. As the system is "cooled," smaller steps are taken and solutions closer to the previous step are considered. The possibility of the algorithm accepting a more energetic solution decreases. The allowance for uphill moves prevents the system from becoming stuck at local minima – which is the bane of methods that accept downhill moves only. One can fine-tune these optimizations by the duration of each heating cycle as well as the temperature change in each loop.

To adapt a simulated annealing algorithm for dielectric stack design, the fitness at each step, F_i, is calculated from the stack's reflectance spectrum from an idealized spectrum. The algorithm works with one filter design, evaluating the fitness, and then assigning small, random changes to all design parameters. The algorithm evaluates the fitness of the next design, F_{i+1}. If the small perturbations in the filter's design parameters result in an improved fitness, the changes will be accepted. If the resulting fitness is worse than before, this is accepted with the probability [23],

$$P = \begin{cases} 1 & F_{i+1} < F_i \\ e^{-[\beta(F_{i+1}-F_i)]} & F_{i+1} > F_i \end{cases} \tag{4.1}$$

where β is a positive, inverse temperature parameter that scales the proportion of worse-performing designs that are accepted. All simulated annealing results in this chapter are the product of 30,000 iterations. For this work, I adopted the simulated annealing algorithm of Tsallis and Stariolo [56] because the global optimum is guaranteed for appropriate choices of the cooling schedule [57].

A Comparison of Optimization Methods for the Double Pulse Spectral Target Function

The reflected pulse profile from the simulated annealing optimization can be seen as the blue line in Fig. 4.2, while the reflected pulse profile result produced by the frequency domain genetic algorithm is shown in green. The genetic algorithm time domain result is shown as a dashed green line for comparison. One can see that the time domain run produced two evenly-weighted peaks, while the frequency domain simulations matched one peak well but barely produced a second peak. The frequency domain genetic algorithm and simulated annealing algorithm results were not promising.

Despite their ability to hill-climb and thereby avoid local minima, simulated annealing algorithms are sensitive to their starting configuration. While these methods take larger steps at the beginning of the optimization, if the starting configuration is far from the global maximum then the algorithm may not sample the whole parameter space. To explore this effect, I ran the simulated annealing algorithm and used as input a randomly-assigned filter, an elite filter from a 500 generation genetic algorithm, and the worst filter from the initial generation of a genetic algorithm. To see if the simulated annealing would benefit from the work done by the genetic algorithm. Results from this investigation are shown in Fig. 4.3, where the random-filter (upper panes) and fittest-filter (lower panes) initializations are shown. Both the temporal pulse profiles (left panes) and reflectance spectra (right panes) are shown. In all cases, the red dashed line is the target function, the incident pulse is shown in black, and the temporal/spectral match of the best SA-designed filter is shown in blue. While the spectral match is better in the fittest-filter initialization, but the temporal match is still very unimpressive. Initializing with a randomly-generated filter to the SA produced a filter that matched to the one temporal peak well. It is important to note that the target reflectance function oscillated rapidly in wavelength and that the optimized structure captured many, but far from all, of these oscillations. Neither version was able to produce a clear double pulse

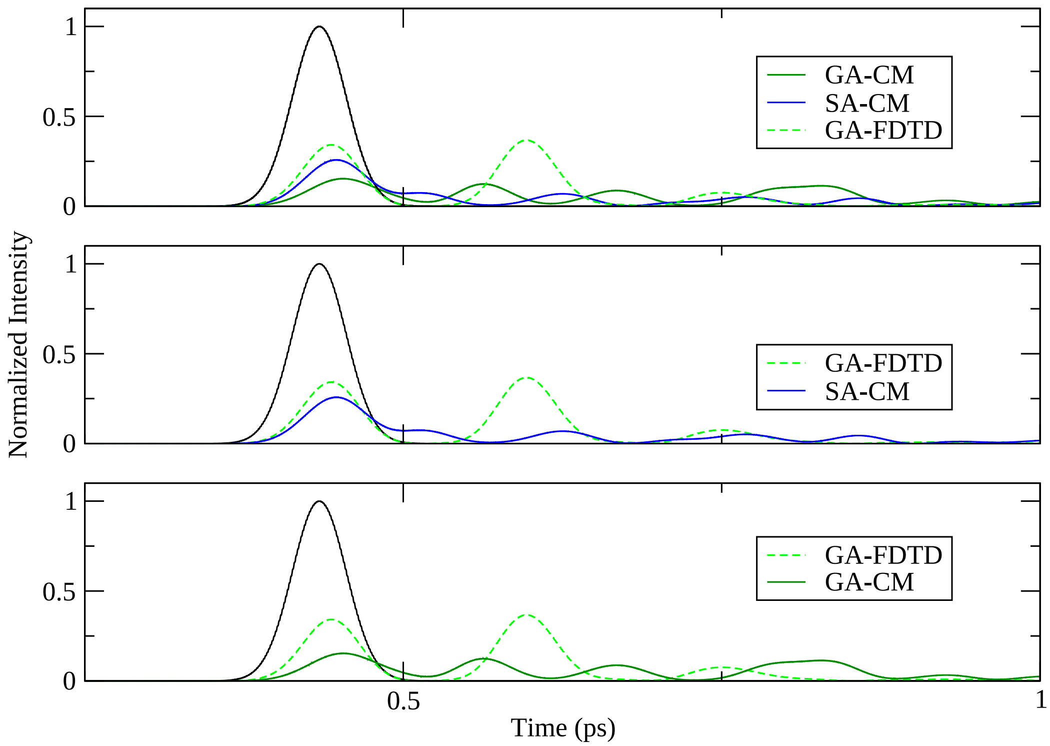

Figure 4.2: Comparison of the frequency domain genetic algorithm result (green line) and simulated annealing (blue line) algorithms compared with the best performance of the time domain genetic algorithm (dashed green line) for the case of a double pulse filter. The incident pulse for all simulations is shown in black.

shape in the time domain, nor were they able to match all of the peaks in the reflectance spectrum.

From the above discussion it appears that the optimization problem in the frequency domain may be harder than in the time domain. In order to gain a better understanding of this issue, several other frequency domain problems were attempted using the genetic algorithm.

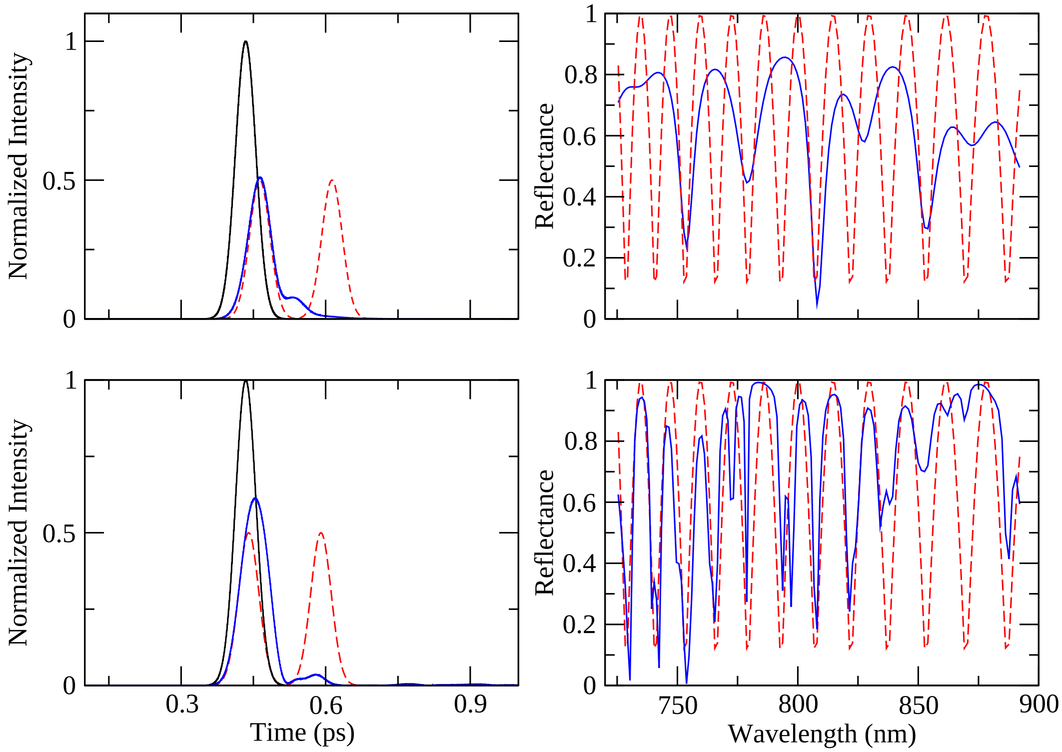

Figure 4.3: Initializing the simulated annealing in two different ways. The upper panes show the result from initializing the simulated annealing algorithm with a randomly-generated filter design, while the lower panes show the result from passing the fittest filter from a 500 generation genetic algorithm run. In all cases, the incident pulse is black, the reflected pulse or spectrum is blue, and the red dashed lines represent the target pulse shape or the target spectrum.

4.3 Tests with Quarter-Wave Stack Design

The first of these was to see if the frequency domain genetic algorithm could design a filter based upon the reflectance spectrum of a quarter-wave stack. The target function was generated by the design of a 28 layer quarter-wave stack out of magnesium fluoride

(MgF_2, $n_{\mathrm{L}} = 1.38$) and silicon dioxide (SiO_2, $n_{\mathrm{H}} = 1.52$) on a silicon dioxide substrate. The characteristic matrix method was used to calculate the reflectance spectrum over the 600 - 1200 nm range. The genetic algorithm's initial designs were composed of the same materials, number of layers, and layer order as the quarter-wave stack from which the target function had been generated. With random assignments of the layer thicknesses, the filter population then evolved, and each filter's performance was evaluated against the target function. The algorithm was varied in several ways, using different maximum generations and different weights on the target function. The simulations consisted of 10 individuals in a population and was allowed to evolve through 500 - 1600 generations. In order to maintain a direct comparison, again the same values of 10 individuals and 500 generations were chosen initially. Fig. 4.4 shows the simulation results for two representative variations of the algorithm. The upper pane shows the results from a 500 generation optimization that followed the steps outlined above. It took just under two minutes to produce a filter of fitness $\mathcal{F} = 0.0818$. The spectral profile of this filter is shown in black compared with the dashed red target function. The lower pane displays the same optimization problem, but this time a supergaussian weight function was used,

$$\mathcal{S}(\omega) = e^{[-[(w-w_0)/\sigma]^{20}]} \tag{4.2}$$

to emphasize the central reflectance band of the target function. Since the earlier trials had taken such a small amount of time, this version was allowed to run for 1600 generations. The runtime was 85 minutes and the final filter produced a fitness of $\mathcal{F} = 1.567$. Note that the central reflectance better matches the steep sides of the target function. Both filters produced reasonable single-pulse shapes when examined in the time domain. Again, the results of this optimization were unimpressive.

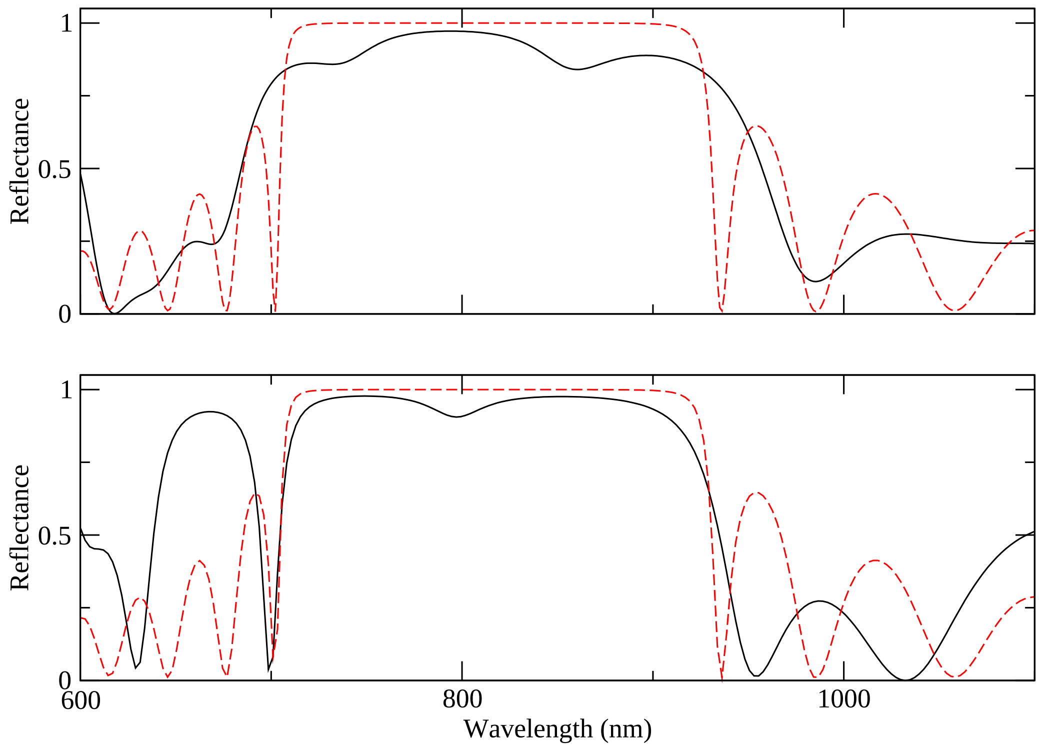

Figure 4.4: Two different quarter-wave stack optimization simulations are compared here. The upper pane shows a 500 generation run that used a standard target function. The lower pane shows the same optimization problem, but this time used a supergaussian weighted target function, to emphasize the central passband. It ran for 1600 generations. In both cases, the spectral profile of the filters are shown in black compared with the dashed red target function.

4.4 Comparison with the Literature

To find further functions that the frequency domain genetic algorithm could replicate, a literature search was carried out. Two similar papers were located, but both contained key differences from this work in the methodology. One incorporated inhomogeneous layers, which, as mentioned earlier, requires difficult manufacturing control. The second work used homogeneous layers, however, their method differed from this work in that they used additional selection methods in their genetic algorithm. These papers were used as a test cases for the genetic algorithm, with the idea to match the genetic parameters that these two papers specified yet maintain the key elements of this original genetic algorithm and filter template.

The first paper [58] constructed 40 layer dielectric stacks from two materials of refractive indices $n_{\mathrm{L}} = 1.35$ and $n_{\mathrm{H}} = 2.20$. The authors used a binary-coded genetic algorithm with 100 individuals, a maximum generation of 1600, and a crossover probability of 0.6. Fig. 4.5, shows the results from the genetic algorithm in black against the target function in red. The central reflectance band reached 0.86 compared with the desired 0.90; the side passbands averaged 0.1 reflectance compared with the target 0.0 reflectance.

While my genetic algorithm struggled to reach the high central reflectance of the target function, the differences were most likely due to the continuously-varying refractive index design used by Martin's group [58]. My genetic algorithm used the more practical two material design. Nonetheless, an attempt was made to replicate the Martin work closer in a separate run by specifying all 40 refractive indices used. However, this increases the material choices effectively increased the search space 20-fold. The algorithm still had to find the best layer thicknesses to match the target function, but by changing from 2 to 40 materials there were many more possibilities to check. However, runtimes were not allowed to increase proportionally for consistency with the maximum generations and population count of Martin's study. Again, the match was poor.

The second paper [25] used 33 layers with the refractive indices 1.35 and 2.35. They used a binary-coded genetic algorithm with 50 individuals, a maximum generation of 1000, and a crossover probability of 0.6. For comparison, I replicated the target function from pane 1b) of the paper. Fig. 4.6 shows the reconstructed target function (red dashed line) compared

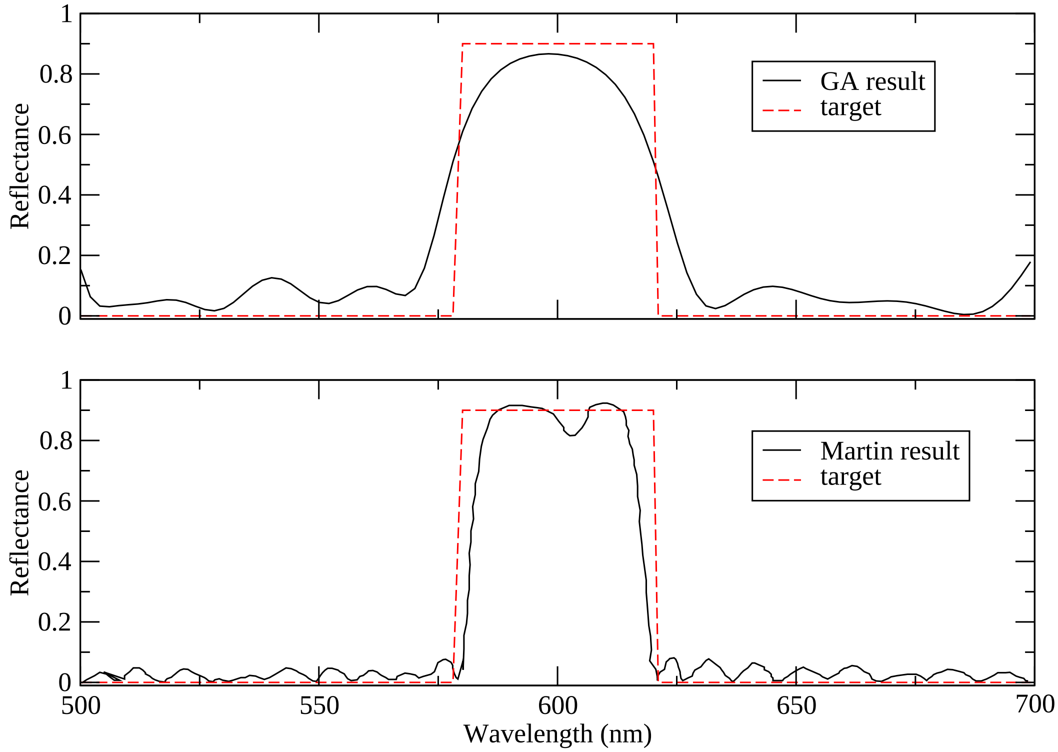

Figure 4.5: Optimization of this dissertation's genetic algorithm against a target function from [58] (upper pane). The lower pane shows the result from the reference work compared to the same target.

with the results of my genetic algorithm (black line) and simulated annealing algorithm (blue line).

Both algorithms did a reasonable job of providing smooth reflectance in the 500 - 550 nm range and in the 700 - 750 nm range. However, they both had trouble matching the flat line for lower reflectances in the range 400 - 450 nm and 600 - 650 nm. Overall, the Yang-designed dielectric stack matched the target better than the genetic algorithm or simulated

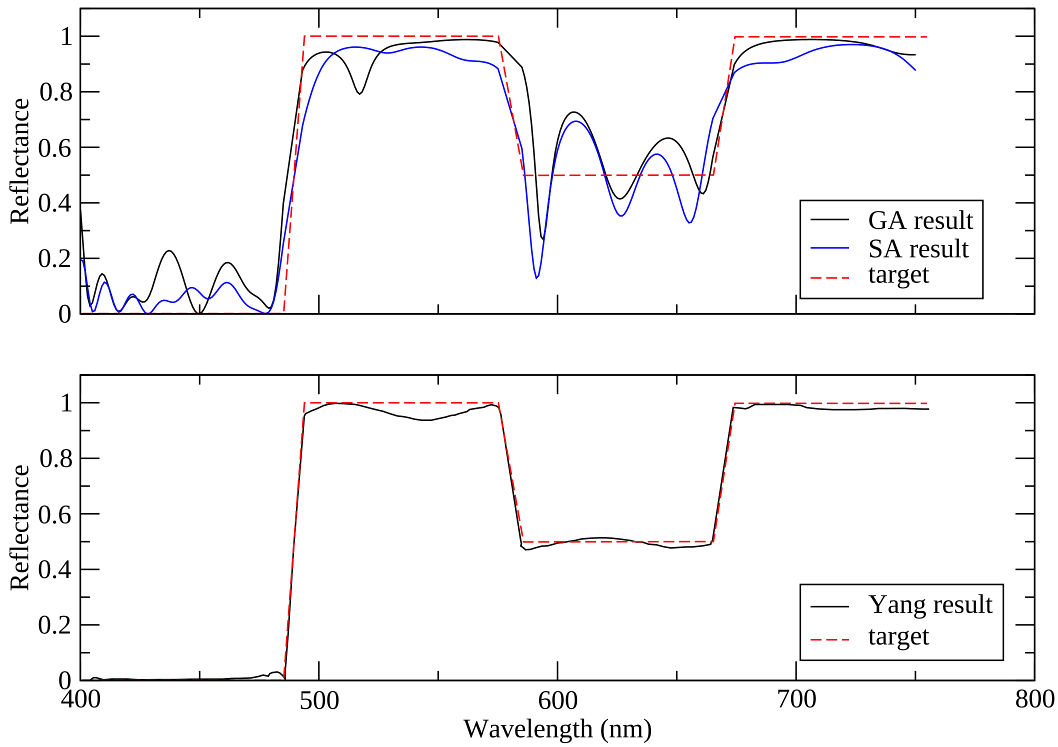

Figure 4.6: The upper pane compares the genetic algorithm and simulated annealing attempts to match a target function from Ref. ??. The lower pane shows results from the reference work compared to the same target.

annealing-produced stacks. Yet, the results from the Yang paper and those generated by my genetic algorithm are not directly comparable; the Yang paper describes results from a family competition evolutionary algorithm, which allows groups of individuals to compete amongst themselves for promotion to the next generation. This advantage is not present in the genetic algorithm used in this dissertation. The family competition evolutionary algorithm also uses an adaptive step-size, starting out with larger steps to better survey the search space and decreasing the step length as the algorithm converges. While this functionality is present in the simulated annealing method, the genetic algorithm does not have that capability. These differences may explain the poorer performance of the genetic algorithm when compared with simulated annealing simulations.

In relation to both the Martin and the Yang papers, a complete match with their data could not be made. Their studies either allowed more freedom in the design parameters, or they used a modified genetic algorithm. It appears that the basic genetic algorithm is insufficient for the task of designing a dielectric stack with spectral specifications. This could explain the lack of papers that use a plain binary-coded genetic algorithm to design a two-material dielectric stack in the frequency domain. Real-coding the strings, encouraging family competition within the generations, or expanding the two-material format to allow a continuous range of refractive indices are all possible ways to improve the performance of the frequency domain algorithm.

Having established that steep-sided and rapidly oscillating target functions are harder for the genetic algorithm, some (hopefully) easier target functions were investigated. Since the algorithm consistently rounds out spectral features, it seemed reasonable to create a smoother version of the target function from one of the earlier simulations in this section. A supergaussian target function (red), which was inspired by the Martin function, as well as the optimization result (black) is shown in Fig. 4.7. It shows a good agreement between the two functions. As was the case before, the filter that produced this spectral result comprises 40 alternating layers of magnesium fluoride ($n_{\mathrm{L}} = 1.35$) and zinc sulphide ($n_{\mathrm{H}} = 2.2$).

Inspired by the time domain double pulse optimization, I also investigated the ability of the algorithm to match a double Gaussian pulse in the frequency domain. In Fig. 4.8, these reflectance spectra (black) are compared with the target function (red). 16,000 generations of the genetic algorithm were simulated here. Note that the asymmetric shape of the reflectance peaks is due to using a wavelength rather than a frequency scale.

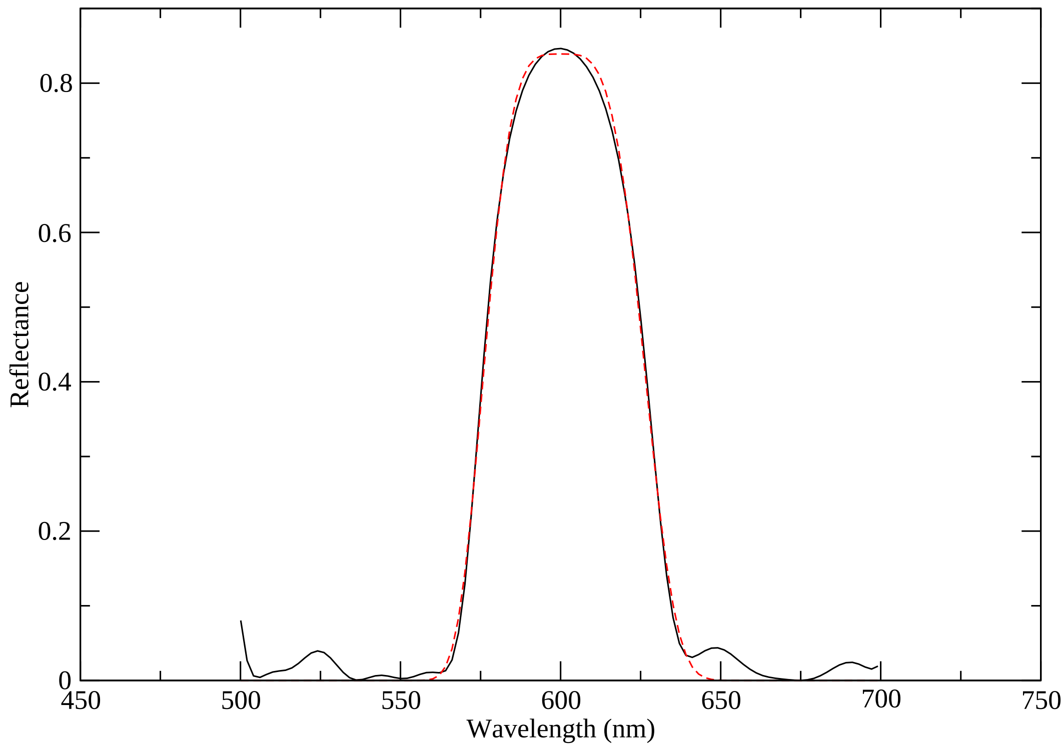

Figure 4.7: Results of an optimization using a supergaussian target function (red). The reflectance spectrum of the optimized filter (black) is in good agreement with the target function.

From the investigations above, there appear to be problems that are easy to optimize in the time domain but that are difficult in the frequency domain [59, 60, 61, 62, 63]. Simple, continuous functions appear to be easy problems for both domains. The optimization of two Gaussian features in the spectral domain was comparable to results in the time domain. I conclude that despite the linearity of the optical problem, the optimization of a pulse shaping

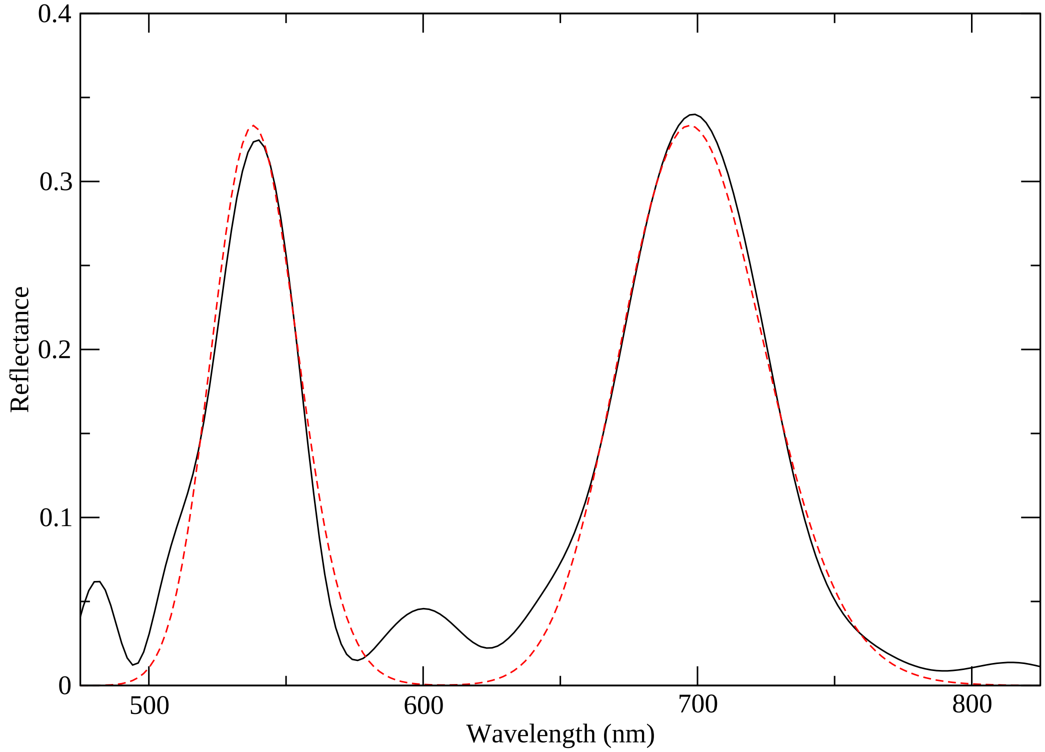

Figure 4.8: The result from a double Gaussian target in the frequency domain. The target function appears in red while the reflectance spectrum is shown in black.

filter is a problem more appropriately solved in the time domain, and this strategy is used in the remainder of the dissertation.

Chapter 5

The Design of Manufacturable Pulse-Stretching Filters

In this chapter I discuss some of the efforts required to develop one of the concepts described in Chapter 3, the pulse-stretching filter, to a stage where it can be constructed by an optical component manufacturer.

5.1 Improved Utilization of the Binary Encoding

As a result of rewriting and optimizing the finite-difference time domain routine, I was able to use the algorithm at a spatial resolution of $\lambda/64$, rather than the $\lambda/20$ resolution described in Chapter 3. This improvement reduces the thinnest filter layers from 40 nm to 12.5 nm. Furthermore, using a $\lambda/64$ resolution takes better advantage of the binary encoding ability of the genetic string.

In the work described in the previous chapters of this dissertation, each layer in the filter was allowed to range from 0 - 2λ in thickness. At a $\lambda/20$ resolution, it is necessary to use a 6-bit encryption, which can specify any non-negative integer from 0 to 63 and only requires 40 sublayers of 40 nm thickness to construct a 2λ layer. Thus the additional 41 to 63 bits that could be used for encoding in the binary string are redundant. If one uses a $\lambda/64$ grid and 7-bit binary encoding, the 0-64 sublayers can specify a layer thickness between 0 and 2λ.

5.2 Conversion to Higher Resolution

Once the results from a $\lambda/64$ resolution optimization are produced, the design needs conversion to at least $\lambda/200$ resolution. At this higher resolution, the reflectance spectrum produced by an FDTD calculation closely matches that of the spectrum produced by the characteristic matrix method. In practice it is easier to scale a $\lambda/64$ resolution structure to a $\lambda/256$ resolution structure rather than to a $\lambda/200$ resolution structure. At $\lambda/256$ the grid resolution increases such that the grid spacing is reduced to $\Delta x = 3.125$ nm. As described in Chapter 3, the dielectric stacks often drop slightly in their fitness or optical performance upon conversion to the higher grid resolution. Again, tuning is employed to find a grid spacing that produces slightly better performance than the exact $\Delta x = 3.125$ nm grid for the $\lambda/256$ resolution.

In contrast with the higher resolution conversions described in Chapter 3, the filters described in this chapter were tuned by directly varying the grid spacing rather than the wavelength. This reduced the number of post-processing FDTD calculations by one. Fig. 5.1 illustrates the steps involved in converting of a 50 layer dielectric stack from a $\lambda/64$ to a $\lambda/256$ resolution. First, I converted the genetic algorithm's resultant $\lambda/64$ resolution dielectric stack into a $\lambda/256$ resolution design by replacing each $\lambda/64$ sublayer with four $\lambda/256$ sublayers. The associated reflected intensity profiles for the $\lambda/64$ and $\lambda/256$ resolution structure are shown in Fig. 5.1. Multiple FDTD calculations, each with a different grid spacing, were performed for the rescaled dielectric stack. Again, Eq. 2.1 was used to determine the fitness of this dielectric stack with various grid spacings, and the results of this tuning generate the plot that is inset in the middle pane of Fig. 5.1. Once the optimized grid spacing was found, the dielectric stack tuning was complete at the higher resolution (bottom pane). In this case, $\Delta x = 3.135$ nm produced the best result for the optimized filter. All the filter layers are then scaled as though they were on a 3.135 nm grid, giving the final dielectric stack blueprint.

A Modification to the Dielectric Stack Design

Material Considerations

For all of the structures described in this chapter, I use niobium pentoxide, Nb_2O_5, and silicon dioxide, SiO_2, because it is easier to deposit structures composed of all oxide layers rather

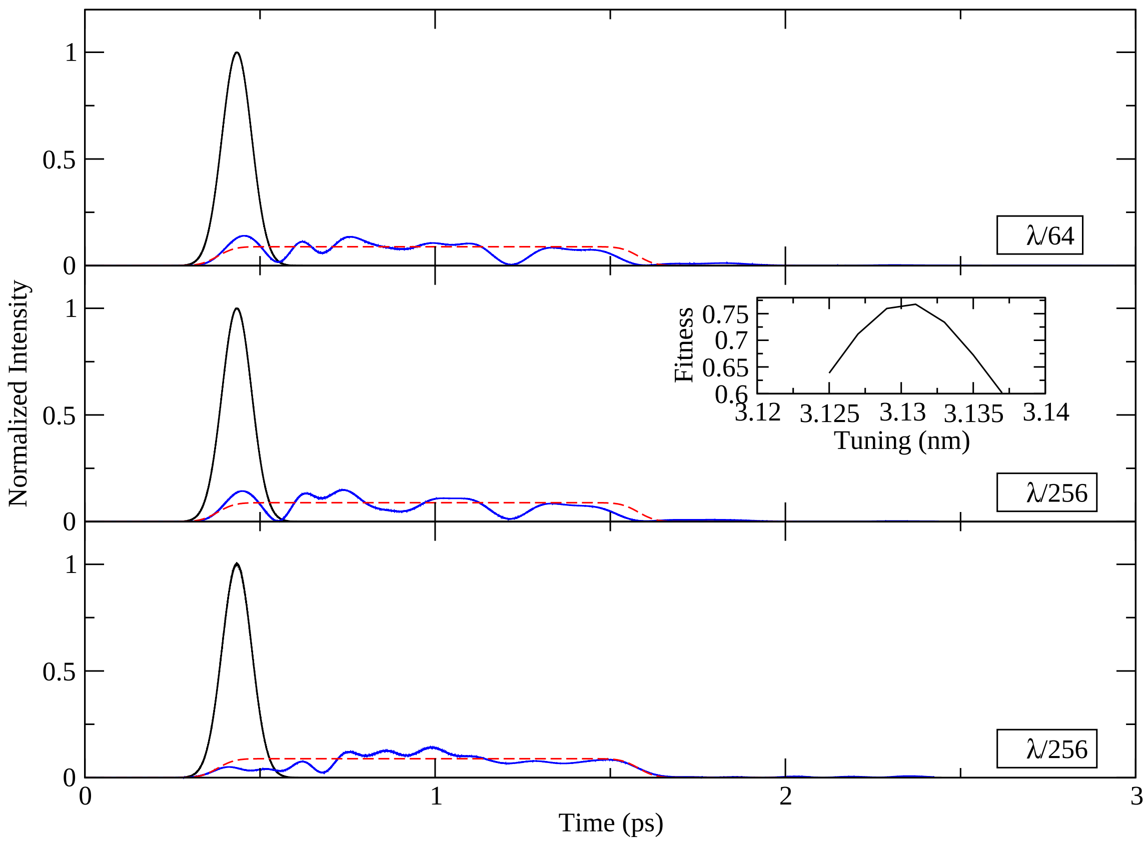

Figure 5.1: Converting the filter from a $\lambda/64$ resolution to a more realistic $\lambda/256$ resolution. The results of the optimization appears in the top pane, while the middle pane shows the result of directly mapping that filter onto a higher resolution grid without tuning. The inset in the middle pane shows the results from tuning the grid spacing. The bottom pane shows the reflected pulse profile from the tuned $\lambda/256$ structure.

than from oxide/fluoride combinations. Niobium pentoxide was chosen as a pair for silicon dioxide because its refractive index is slightly higher than that of tantalum pentoxide. The low index material was changed from magnesium fluoride to silicon dioxide, which increased the refractive index of the L layers. Therefore, I used a higher-refractive index oxide for the H layers, in an effort to maintain a high ratio between the low and high refractive indices, *i.e.*, $n_{\mathrm{H}}/n_{\mathrm{L}}$.

5.3 Pulse Stretching Filters for 100 fs Pulses

These devices represent two major design changes from the earlier pulse stretching filters of Chapter 3. Firstly, they are designed for 100 fs input pulses with a central wavelength of 800 nm rather than the 30 fs input pulses that were used earlier. Secondly, these filters make use of all oxide layers, *i.e.* Nb_2O_5 and SiO_2, to make fabrication of these mirrors easier. The final product should be a blueprint for a readily-manufacturable and laboratory-testable dielectric stack.

Performance

Using the new design parameters, the first step is explore the performance of a pulse-stretching mirror. In particular, a target function that comprises of 20 overlapping Gaussians was used to stretch the pulse by a factor of 10 on reflection. Several layer templates are used. Fig. 5.2 shows the results from the 50, 100, and 150 layer dielectric stacks. One can see that there is only modest improvement in increasing the number of layers from 50 to 100 to 150. Since a 50 layer structure is manufacturable using today's technology (100 layers being tomorrow's technology), only the 50 and 100 layer structures were explored further.

Robustness

If these devices are to find practical application, it is necessary to see how a particular dielectric stack will perform in sub-optimal conditions. In particular, if the central wavelength

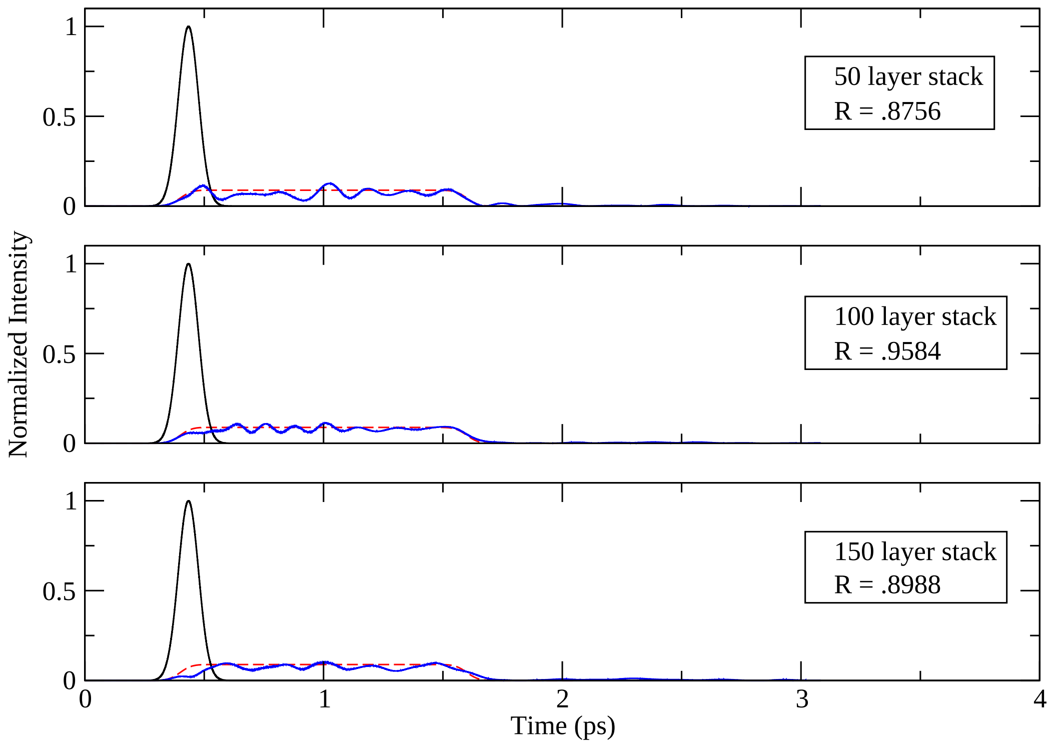

Figure 5.2: Comparison of filters optimized for the same target pulse shape (red dashed line) but with differing layer numbers in the design template. Shown here are 50, 100, and 150 layer filters that were asked to spread a 100 fs input pulse (black line) over the distance of 20 overlapping Gaussian pulses. The reflection due to each filter is shown in blue.

of the laser drifts or if its pulsewidth changes, to what degree does it impact the filter's ability to produce the desired reflected pulse? To investigate this, the reflected pulses with different central wavelengths and pulsewidths were calculated. Results for the robustness tests are shown in Figs. 5.3, 5.4, 5.6, 5.7, and 5.5 while the corresponding reflectance data appears in Tables 5.1 and 5.2. All of the FDTD calculations here were run using $\lambda/256$ resolution.

It is gratifying to note the lack of variation in the reflected pulse shape due to wavelength and pulse width detunings. For the 50 layer filter, the optimized reflectance occurs at the standard central wavelength of 800 nm and intensity FWHM of 100 fs. The filter produces a very similar reflection shape for $\lambda = 790$ nm and only drops the reflectance by 0.09 %, from 91.15 % to 91.04 %. With the pulse width variations, the drop in reflectance was 0.25 % for an incident pulse width shift of 15 fs (from 100 fs to 85 fs).

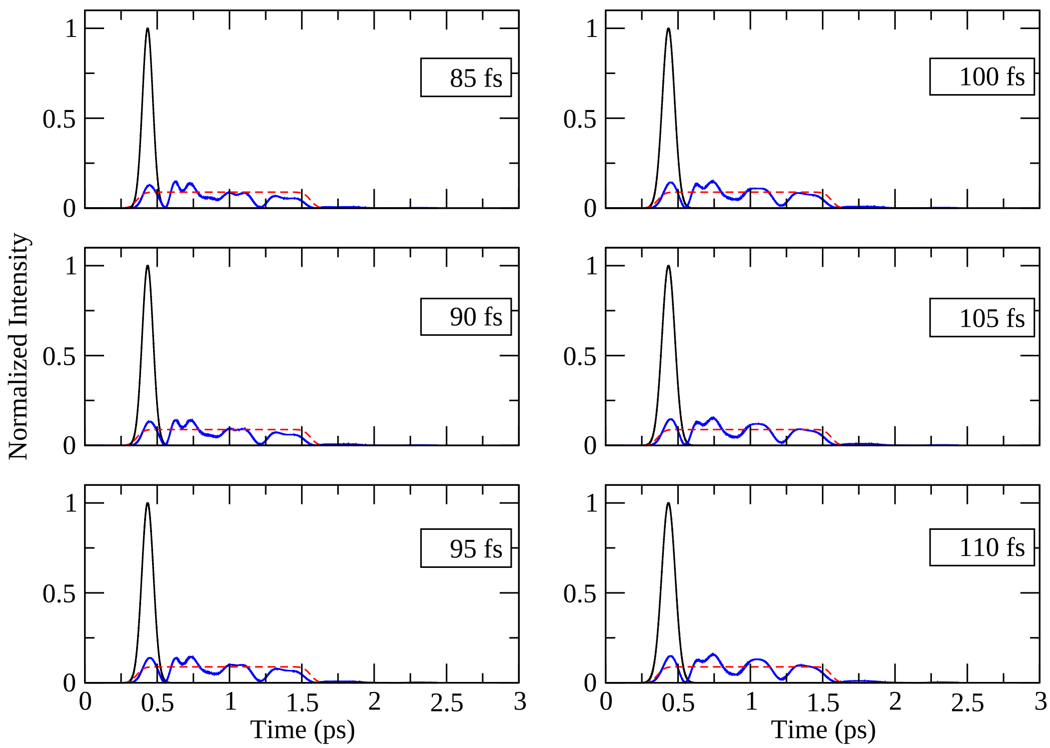

Figure 5.3: Testing the 50 layer pulse-stretching filter performance with sub-optimal incident pulse widths. Corresponding reflectance data appears in Table 5.1. The incident pulse is black, the reflected pulse is blue and the target function is a dashed red line.

Table 5.1: Results for the robustness tests of the 50 and 100 layer pulse stretching filters with respect to incident intensity pulse width. Reflected pulse profiles appear in Figs. 5.3, 5.4 and 5.5.

Number of Filter Layers	Pulsewidth (fs)	Reflectance (%)
50	85	90.90
50	90	91.25
50	95	91.63
50	100	91.15
50	105	89.36
50	110	90.78
100	70	93.00
100	85	94.09
100	90	94.37
100	95	94.12
100	100	94.50
100	105	94.32
100	110	93.85
100	130	93.45

Table 5.2: Results for the robustness tests of the 50 and 100 layer pulse stretching filters with respect to incident wavelength.

Number of Filter Layers	Wavelength (nm)	Reflectance (%)
50	790	91.04
50	795	90.81
50	800	91.15
50	805	91.34
50	810	90.99
50	815	90.55
100	790	94.37
100	795	94.58
100	800	94.70
100	805	94.41
100	810	94.36
100	815	94.50

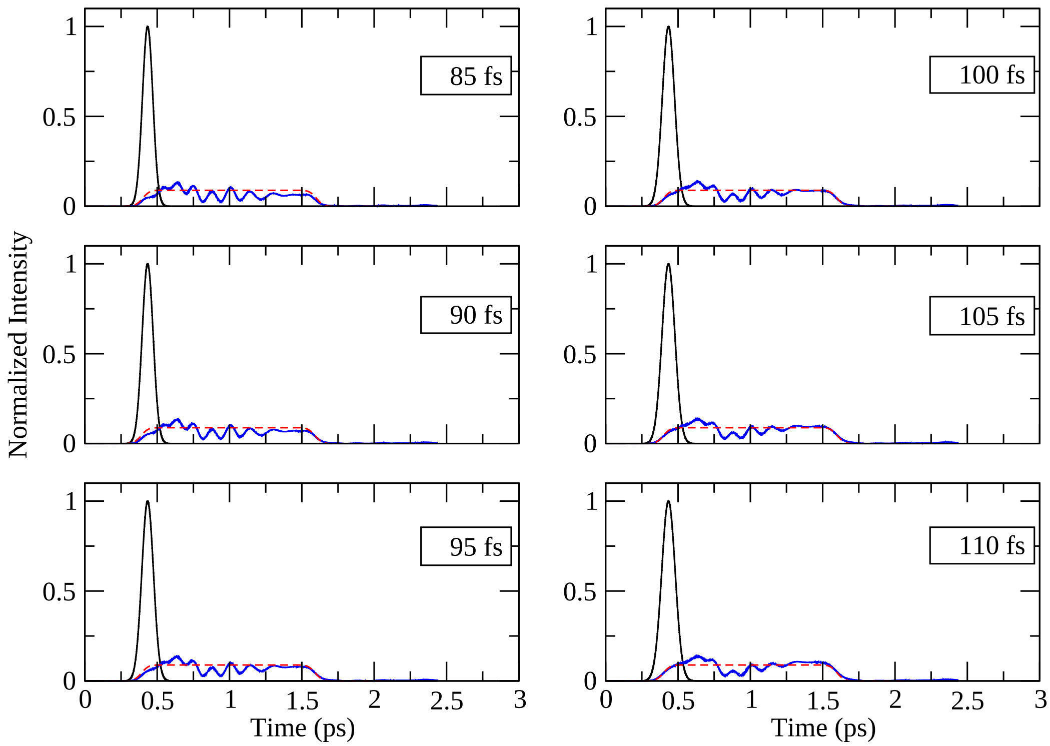

Figure 5.4: Testing the 100 layer pulse-stretching filter performance with sub-optimal incident pulse widths. Corresponding reflectance data appears in Table 5.1. The incident pulse is black, the reflected pulse is blue and the target function is a dashed red line.

Resolution Comparison

This series of simulations show the gains that can be made when optimizing directly on a higher grid resolution. Fig. 5.8 shows the outcomes from three separate optimizations. All simulations had the same goal of designing a pulse stretching filter with 50 layers of Nb_2O_5 and SiO_2, but used resolutions of $\lambda/20$, $\lambda/64$, and $\lambda/128$. The results from the grids of $\lambda/20$, $\lambda/64$, and, $\lambda/128$ optimizations are shown respectively in the upper, middle, and lower

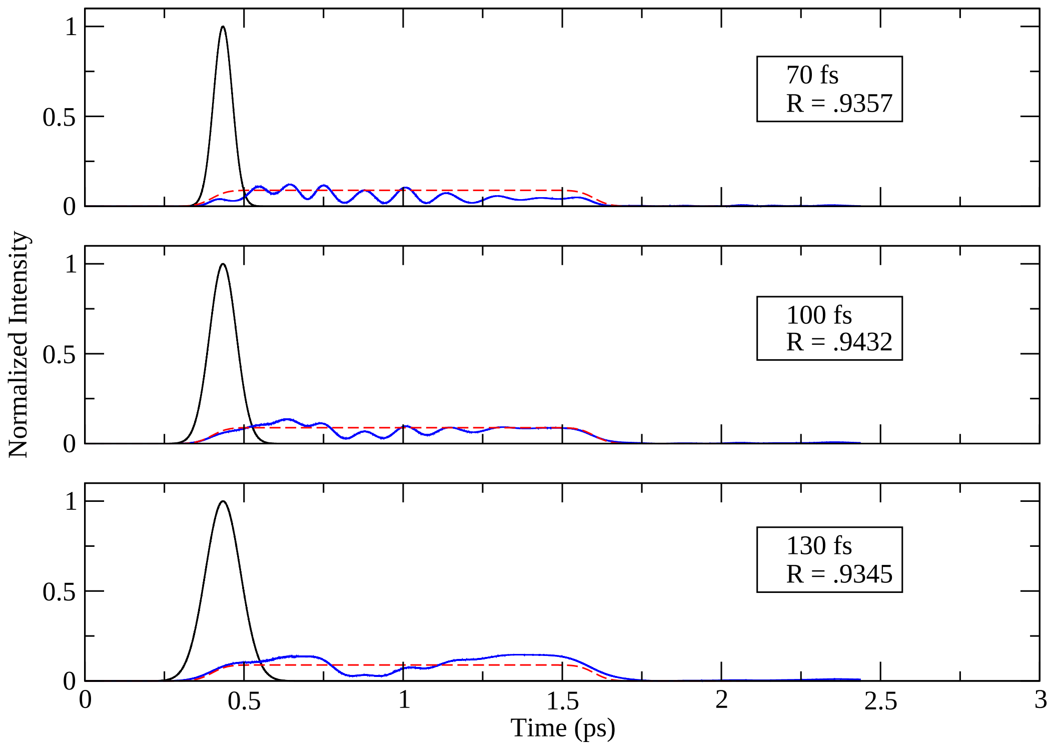

Figure 5.5: A direct comparison of the intensity profiles for both optimal and non-optimal pulses. 70, 100, and 130 fs incident pulses were reflected off the 100 layer pulse stretching dielectric stack. Sub-optimal incident pulses are shown in the top and bottom panes, while the middle pane shows the incident pulse for which the filter was designed. In all cases the target reflectance is shown as a red dashed line, while the incident pulse is in black and the reflected pulse appears in blue.

panes of Fig 5.8. The $\lambda/64$ grid provided sufficient resolution, with its resultant best filter's spectrum matching the length of the pulse stretching target well. The $\lambda/128$ optimization produces a good filter design with better fitness than the $\lambda/64$ optimization. However, the reflectance is not as high. The corresponding reflectances and fitnesses are given in Table 5.3.

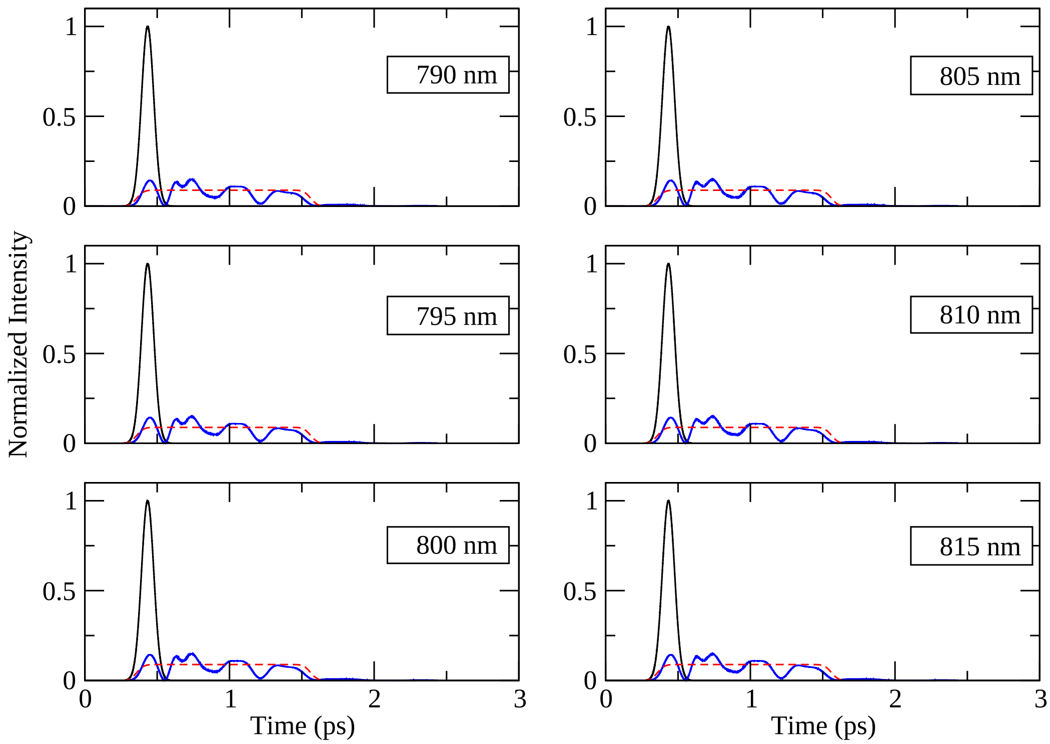

Figure 5.6: Testing the 50 layer pulse stretching filter for performance with detuned incident wavelengths. The incident wavelegnth (black) varied from 790 nm through to 815 nm. The reflected pulse shape appears in blue while the target function is shown in red for comparison. Corresponding data appears in Table 5.2.

5.4 Application: A Matched Pulse Stretcher/Compressor Pair

Having spent much of this dissertation discussing the optimization of pulse-stretching filters, I was interested to determine if the stretched pulse could be re-compressed. One current practice in ultrafast optics is to stretch and compress laser pulses using a pair of diffraction gratings [64, 65]. Diffraction grating arrangements have limited robustness and energy

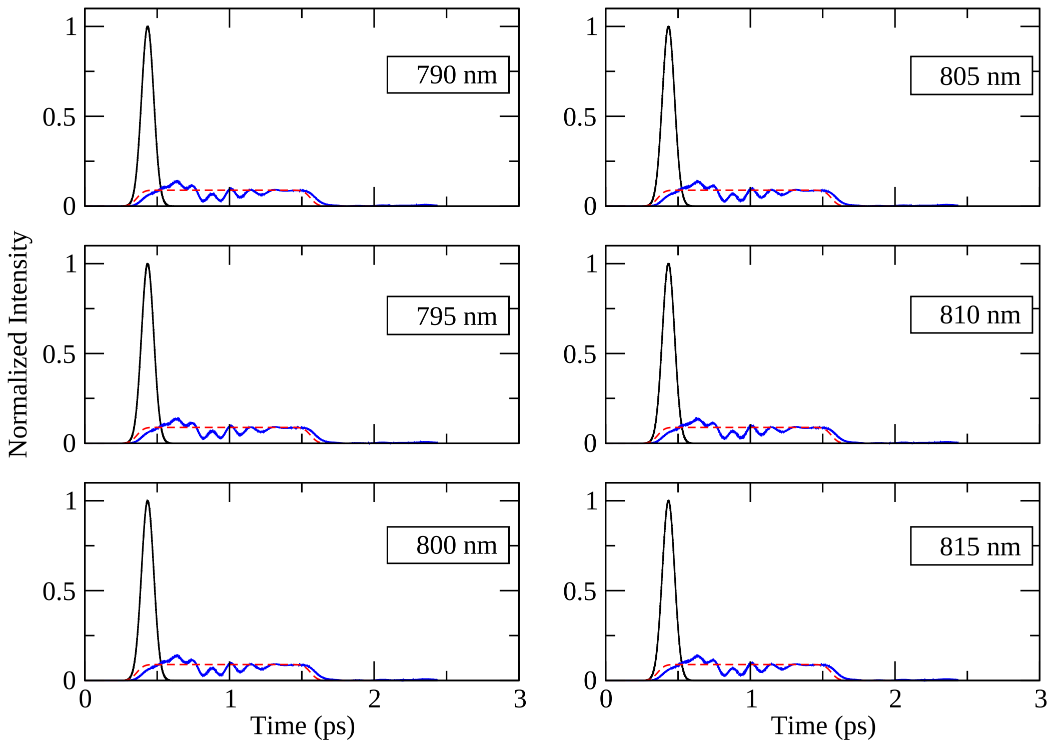

Figure 5.7: Testing the 100 layer pulse stretching filter for performance with detuned incident wavelengths. The incident wavelegnth (black) varied from 790 nm through to 815 nm. The reflected pulse shape appears in blue while the target function is shown in red for comparison. Corresponding data appears in Table 5.2.

throughput due to the metallic coatings, and have been known to distort the output beam profile. These features are undesirable, hence the attractiveness of circumventing these problems by using a fixed filter-based device.

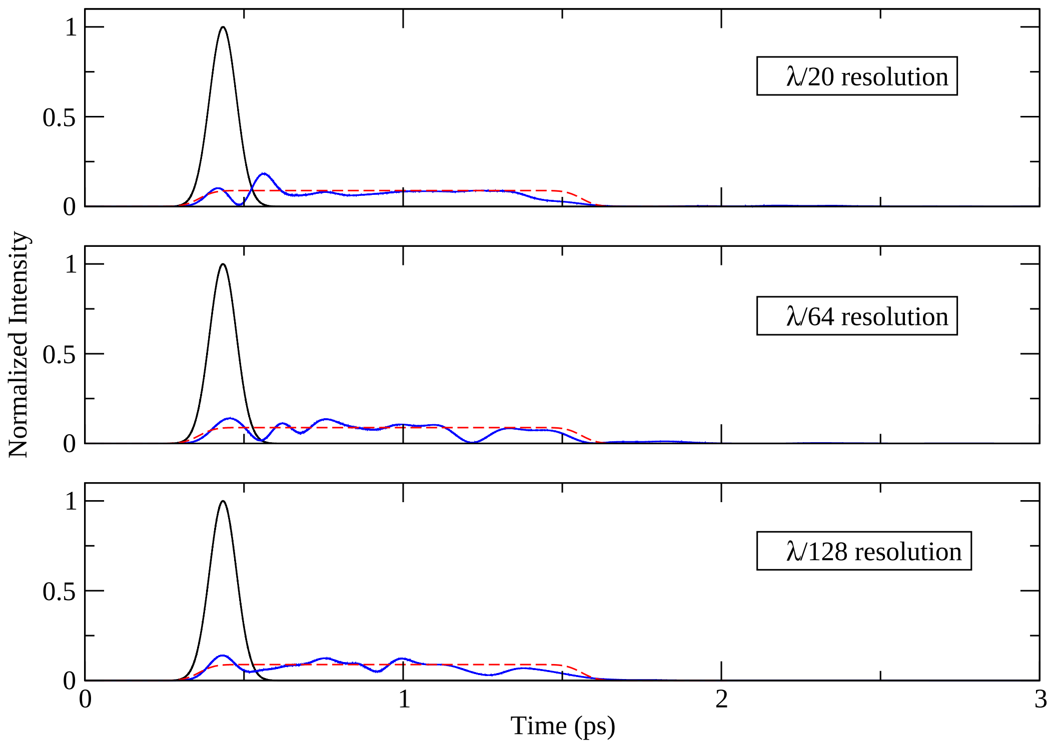

Figure 5.8: Reflected pulse shapes (blue) for the 50 layer pulse-stretching filters designed using spatial resolutions of $\lambda/20$, $\lambda/64$, and $\lambda/128$, respectively. The dashed red curve shows the target function in each case, and the incident pulse is shown in black. Table 5.3 displays the reflectances and grid sizes for these cases.

Table 5.3: Comparison of simulations for the 20 overlapping pulse-stretching design. All simulations use a 50 layer filter template, but are run on different grid resolutions. Fig. 5.8 shows the corresponding pulse shapes.

Spatial Resolution	Δx (nm)	Fitness	Reflectance (%)
$\lambda/20$	40	1.155	87.81
$\lambda/64$	12.5	1.170	91.54
$\lambda/128$	3.125	1.312	88.11

Reading in an Arbitrary Incident Pulse

In order to develop a compressor stack, the stretched pulse must be read into the optimizing algorithm. The GA-FDTD program was converted to read from a file rather than generate the incident pulse from a Gaussian function, as was done in the previous dielectric stack simulations. This modification also has the advantage that it allows any pulse shape, say from experimental data or from the results of the pulse-stretching simulations to be read into the optimization algorithm. It can also be used to input the incident pulses for the different grid tunings when converting the dielectric stack to a higher resolution.

With the ability to read in arbitrary pulse shapes, the algorithm could now be used to design a pulse compressing dielectric stack. The optimization created a dielectric stack that could best restore the original Gaussian shape from a reflected pulse produced by either the 20 or 40 overlapping pulse stretching filter. The pulse compression results are shown in the upper and lower panes of Fig. 5.9, respectively. The genetic algorithms was run for 500 generations with 10 individuals.

In both cases, the simulation ran on a $\lambda/64$ grid and used a dielectric stack composed of 100 alternating niobium pentoxide and silicon dioxide layers. Both versions worked well in producing Gaussian shapes upon reflection. The peak intensity of the 40 overlapping pulse version is somewhat poorer than the 20 overlapping pulse version. It is possible that the 40 overlapping pulse version requires a thicker dielectric stack to produce the necessary submirrors within the dielectric stack.

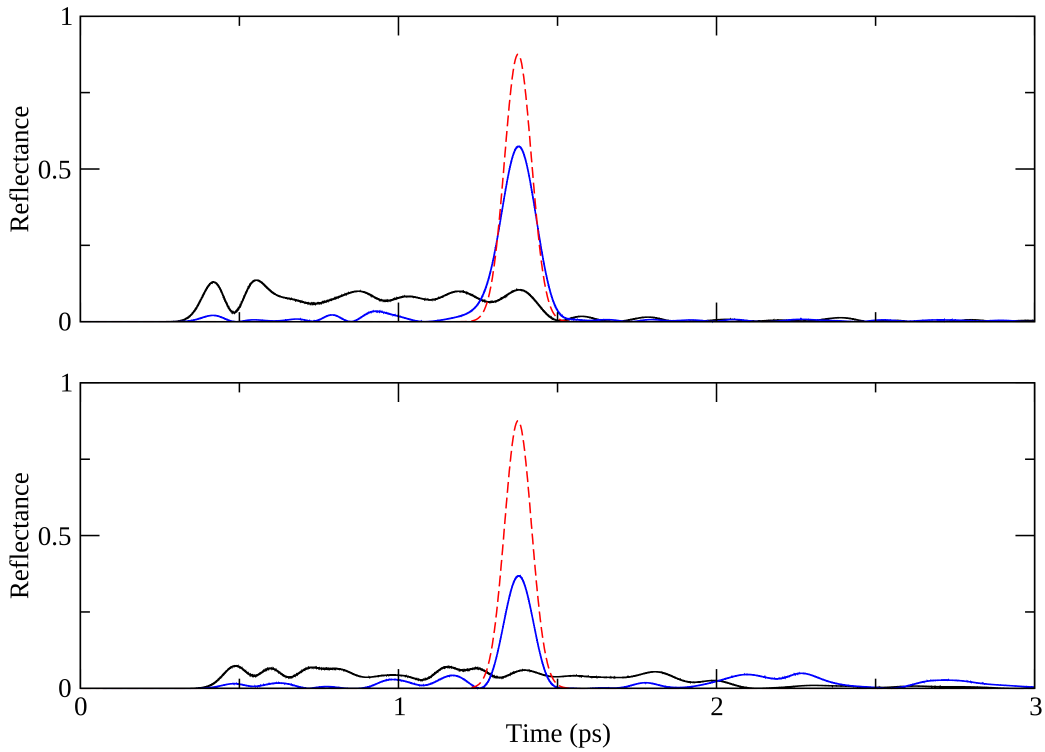

Figure 5.9: Reflected pulse shapes for the 50 layer compression dielectric stack designed using a spatial resolution of $\lambda/64$. Results from the 20 and 40 overlapping pulse stretching dielectric stack were used as the incident pulse (black). In both cases the reflected pulse is shown in blue and the target function appears in red.

Chapter 6

Conclusions

The main contribution of this dissertation is the exploration of thin-film filter design for ultra-short pulse shaping applications. The filters described in this dissertation are the product of two independent methodologies: optimizing in the time domain and optimizing in the frequency domain. The time domain approach is particularly novel for this type for filter design. To the best of my knowledge, this is the first time filters have been optimized with temporal pulse shaping as the target function for an ultra-short laser pulse. With the combination of the finite-difference time domain technique and a genetic algorithm, various filter designs were created that produce double, triple, and stretched pulses upon reflection. The spacing between the double peaks was investigated for a range of 30 - 300 fs, while the spacing between consecutive peaks for the triple pulse reflections was investigated for the 30 - 120 fs range. To understand the mechanisms of the pulse shaping filters, a case study was performed. By investigating the group delay dispersion, spectrograms for the reflected pulses and isolating particular layers of the filter, it became apparent that the double pulses were created by two effective mirrors within the stack.

The possibility of using a frequency domain technique was also explored. The problem here was to design a filter for a desired spectral profile. The target reflectance or transmittance function was fed into the algorithm and then used as a means of identifying better-performing filters. For the spectral domain optimizations, both genetic algorithms and simulated annealing algorithms were used in conjunction with the characteristic matrix method. To check the efficacy of this optimization strategy, the frequency domain algorithm was used to produce a filter with a reflectance function that matched the Fourier transform of a double-pulse (time domain) filter's reflected pulse. In the case of the double, triple, and stretched time-domain pulses, the target functions for these calculations became much more difficult to match. Where a target function is straight-forward for a genetic algorithm in one domain, its Fourier transform can be quite a difficult structure for the genetic algorithm to solve. The main contribution from this section was the further characterization of target functions that are easy and target functions that are hard for the genetic algorithm.

Both the filter template and the genetic algorithm used in this dissertation had several modifications made to improve the realizability of the final filter design. The materials were converted to all-oxide layers for ease of manufacture. Changes to the genetic algorithm's binary encoding made better use of the computational resources, improving the grid resolution without slowing the algorithm noticeably. The algorithm was also modified to read in arbitrary stretched pulse shapes so that a stretcher/compressor filter pair could be designed. Chapter 5 detailed the results of filters designed for 100 fs input pulses. I am pleased to note that a detailed manufacturability study of one of my pulse stretching designs is being performed by an optical coating company at the time of writing.

Appendix A

The Characteristic Matrix Method

The characteristic matrix method provides an elegant way of handling Maxwell's equations subject to the boundary conditions in a multilayer stack. An outline is given here, following the development given by Born and Wolf [47] and Macleod [6]. Our frequency domain algorithms use the resulting equations exactly.

Consider the stack of l films illustrated in Fig. A.1. The layers run from 1 to l while their interfaces are labeled 1 to l-1. We choose an orientation of the coordinate system so that the materials change in only one dimension, enabling Maxwell's equations to reduce to independent sets of equations for the transverse electric (TE) and transverse magnetic (TM) modes. We assume the incident light propagates in the $\hat{y}$-direction. The angle of incidence is denoted υ.

We treat the sum of the positive and negative traveling waves at each interface in the multilayer. A positive traveling wave is distinguished from the total field by use of the notation E^{+}. Likewise, we use E^{-} to denote negative propagating waves. We consider the tangential components, which are continuous across the interfaces. At a particular interface, such as the boundary between film 1 and film 2 in Fig. A.2, we can relate the tangential components at this interface, E_b and H_b, as

$$
\begin{aligned}
E_b &= E_{1b}^{+} + E_{1b}^{-} \\
H_b &= \eta_1 E_{1b}^{+} - \eta_1 E_{1b}^{-}
\end{aligned}
\tag{A.1}
$$

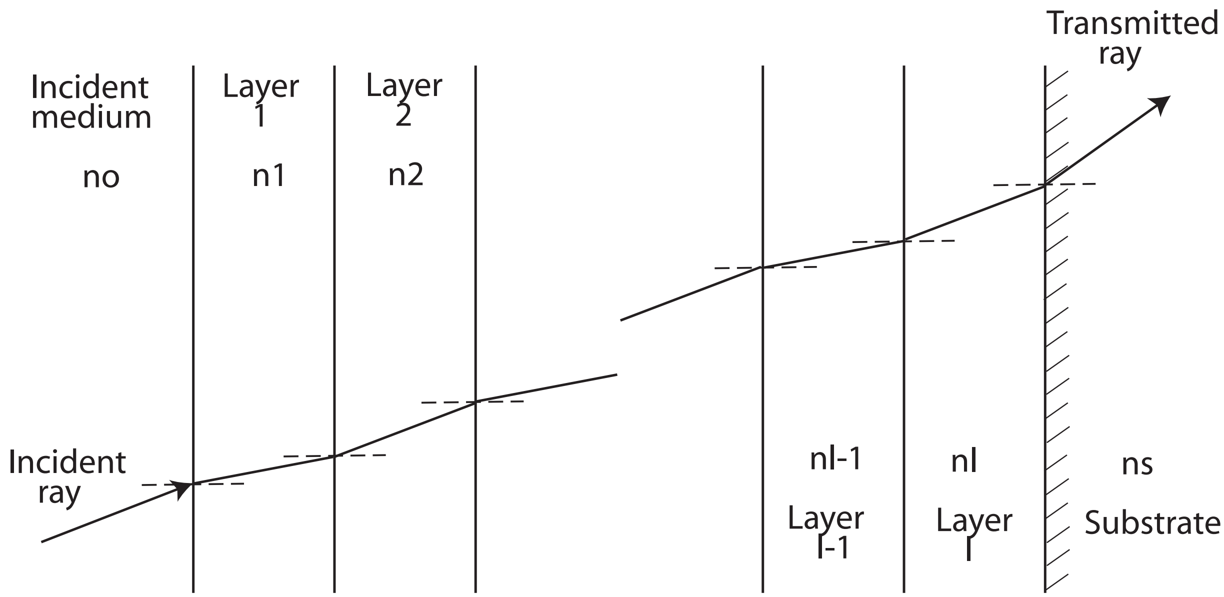

Figure A.1: A multilayer stack consisting of air, l homogeneous planar films, and a semi-infinite substrate.

Note that there is no negative traveling wave in the substrate. We calculate the path difference δ as a fraction of the wavelength.

$$\delta = 2\pi N_1 d \cos\theta_i/\lambda \tag{A.2}$$

This allows us to infer the fields, denoted with a, at the boundary between the surface and film 1 of Fig. A.2, using the phase factor $e^{i\delta}$ for forward traveling waves and it's complex conjugate, $e^{-i\delta}$ for backward traveling waves,

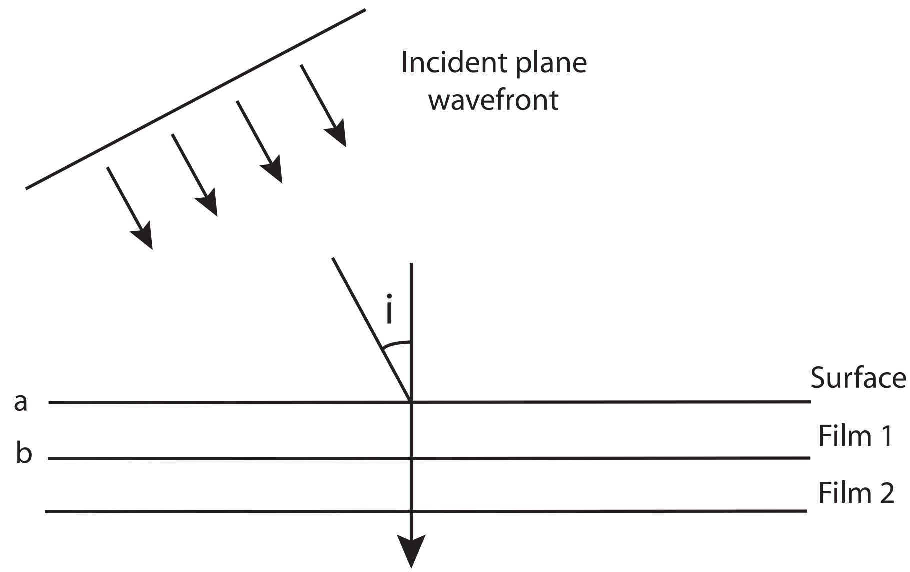

Figure A.2: Light incident on two thin films with a substrate.

$$\begin{aligned} E_{1a}^{+} &= E_{1b}^{+} e^{i\delta} \\ E_{1a}^{-} &= E_{1b}^{-} e^{-i\delta} \\ H_{1a}^{+} &= H_{1b}^{+} e^{i\delta} \\ H_{1a}^{-} &= H_{1b}^{-} e^{-i\delta}. \end{aligned} \tag{A.3}$$

Substitution of the explicit forms for $E_{1b}^{\pm}$ and $H_{1b}^{\pm}$ gives us a system of two equations that relate the tangential fields at the surface-film 1 boundary, E_a and H_a, to the tangential fields at the film 1-film 2 boundary, E_b and H_b. Compiling these in matrix form we have

$$\begin{pmatrix} E_a \\ H_a \end{pmatrix} = \begin{pmatrix} \cos\delta & \frac{i}{\eta_1}\sin\delta \\ i\eta_1\sin\delta & cos\delta \end{pmatrix} \begin{pmatrix} E_b \\ H_b \end{pmatrix}. \tag{A.4}$$

The final form used in our optimization algorithm was a normalized version. Most properties of thin film systems are extracted from the following transfer matrix,

$$\begin{pmatrix} E_a/E_b \\ H_a/H_b \end{pmatrix} = \begin{pmatrix} B \\ C \end{pmatrix} = \begin{pmatrix} \cos\delta & \frac{i}{\eta_1}\sin\delta \\ i\eta_1\sin\delta & \cos\delta \end{pmatrix} \begin{pmatrix} 1 \\ \eta_2 \end{pmatrix}. \tag{A.5}$$

In particular, we can extend this to the general case of an l-layer stack. The characteristic matrix for this assembly is simply the product of the matrices for the individual layers, multiplied in the correct order:

$$\begin{pmatrix} B \\ C \end{pmatrix} = \prod_{i=1}^{l} \begin{pmatrix} \cos\delta_i & \frac{i}{\eta_i} sin\delta \\ i\eta_i\sin\delta & \cos\delta_i \end{pmatrix} \begin{pmatrix} 1 \\ \eta_m \end{pmatrix}. \tag{A.6}$$

Appendix B

The Finite-Difference Time Domain Method

The basic FDTD space grid and time stepping routine was developed in the seminal 1966 paper by Kane Yee [14]. In this paper he converted Maxwell's equations into a set of finite difference equations that could calculate the propagation of an electromagnetic wave for boundaries involving a perfectly conducting medium. Here, we follow Yee's method to cast Maxwell's equations into a form suitable for our one-dimensional FDTD algorithm.

We begin with Maxwell's curl equations for an isotropic medium:

$$\begin{aligned} \nabla \times \mathbf{H} &= \mathbf{J}_f + \frac{\partial \mathbf{D}}{\partial t} \\ \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \end{aligned} \tag{B.1}$$

The first of these is the Maxwell-Ampere law, which relates the motion of a magnetic field in a closed loop to the electric field passing through the loop. The second equation, Faraday's law, describes the induced electromotive force (EMF) caused by a changing magnetic flux. The motion of one field induces the presence of the other, so as a pair, this creates an electromagnetic wave. For the purposes of the filter design calculations, we ignore any source terms, such as $\mathbf{J}_f$. We also work with E and H rather than H, D, E, and B, so we use $D = \epsilon E$ and $B = \mu H$ to convert the equations into the desired form. μ is the permeability of the medium, and ϵ is the medium's permittivity.

In Cartesian coordinates the rearranged Maxwell curl equations can be written as six coupled scalar equations

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y}\right)$$

$$\begin{aligned}
\frac{\partial H_y}{\partial t} &= \frac{1}{\mu}\left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z}\right) \\
\frac{\partial H_z}{\partial t} &= \frac{1}{\mu}\left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x}\right) \\
\frac{\partial E_x}{\partial t} &= \frac{1}{\epsilon}\left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z}\right) \\
\frac{\partial E_y}{\partial t} &= \frac{1}{\epsilon}\left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x}\right) \\
\frac{\partial E_z}{\partial t} &= \frac{1}{\epsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}\right).
\end{aligned} \tag{B.2}$$

Yee defined the grid coordinates (i, j, k) as:

$$(i, j, k) = (i\Delta x, j\Delta y, k\Delta z) \tag{B.3}$$

where $\Delta x, \Delta y$ and Δz are increments between successive grid steps.

In one dimension this reduces to only two coupled partial differential equations to solve,

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon}\left(-\frac{\partial H_y}{\partial z}\right) \tag{B.4}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(-\frac{\partial E_x}{\partial z}\right) \tag{B.5}$$

We now consider the magnetic field at a time n to be the average of the previous half time-step and the successive half time-step. This semi-implicit approximation allows us to calculate $\mathbf{H}$ on a grid, one half time and one half spatial step removed from the $\mathbf{E}$ grid.

$$H_x|_{i,j,k}^{n} = \frac{H_x|_{i,j,k}^{n+1/2} + H_x|_{i,j,k}^{n-1/2}}{2}. \tag{B.6}$$

We can express all the magnetic fields about the Yee Cell in terms of this half step forward in time:

$$H_x|_{i,j,k}^{n+1/2} = \left(\frac{1 - \frac{\Delta t}{2\mu_{i,j,k}}}{1 + \frac{\Delta t}{2\mu_{i,j,k}}}\right) H_x|_{i,j,k}^{n-1/2} + \left(\frac{\frac{\Delta t}{\mu_{i,j,k}}}{1 + \frac{\Delta t}{2\mu_{i,j,k}}}\right) \left(\begin{array}{c} \frac{E_y|_{i,j,k+1/2}^{n} - E_y|_{i,j,k-1/2}^{n}}{\Delta z} - \\ \frac{E_z|_{i,j+1/2,k}^{n} - E_z|_{i,j-1/2,k}^{n}}{\Delta y} \end{array}\right) \tag{B.7}$$

$$H_y|_{i,j,k}^{n+1/2} = \left(\frac{1 - \frac{\Delta t}{2\mu_{i,j,k}}}{1 + \frac{\Delta t}{2\mu_{i,j,k}}}\right) H_y|_{i,j,k}^{n-1/2} + \left(\frac{\frac{\Delta t}{\mu_{i,j,k}}}{1 + \frac{\Delta t}{2\mu_{i,j,k}}}\right) \left(\begin{array}{c} \frac{E_z|_{i+1/2,j,k}^{n} - E_z|_{i-1/2,j,k}^{n}}{\Delta x} - \\ \frac{E_x|_{i,j,k+1/2}^{n} - E_x|_{i,j,k-1/2}^{n}}{\Delta z} \end{array}\right)$$

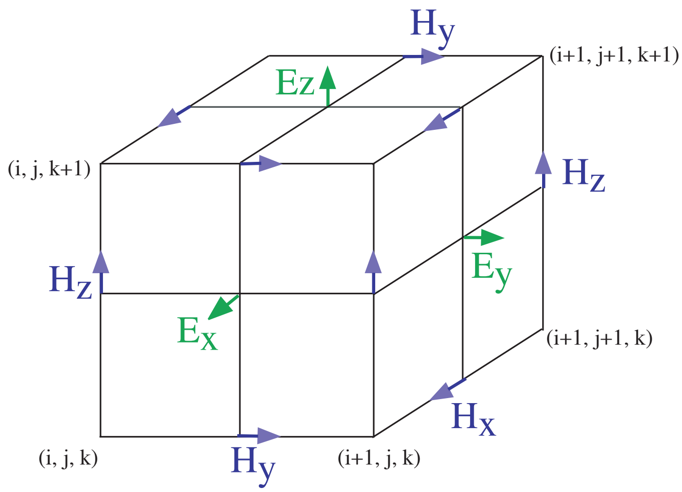

Figure B.1: The Yee Cell for the three dimensional FDTD algorithm. This schematic shows the orientation of all electric and magnetic field components for a single unit cube. Note the half-step offset of the **E** and **H** fields.

(B.8)

$$H_z\big|_{i,j,k}^{n+1/2} = \left(\frac{1 - \frac{\Delta t}{2\mu_{i,j,k}}}{1 + \frac{\Delta t}{2\mu_{i,j,k}}} \right) H_z\big|_{i,j,k}^{n-1/2} \quad + \quad \left(\frac{\frac{\Delta t}{\mu_{i,j,k}}}{1 + \frac{\Delta t}{2\mu_{i,j,k}}} \right) \left(\begin{array}{c} \frac{E_x|_{i,j+1/2,k}^{n} - E_x|_{i,j-1/2,k}^{n}}{\Delta y} - \\ \frac{E_y|_{i+1/2,j,k}^{n} - E_y|_{i-1/2,j,k}^{n}}{\Delta x} \end{array} \right) \tag{B.9}$$

$$E_x|_{i,j,k}^{n+1} = \left(\frac{1 - \frac{\sigma_{i,j,k}\Delta t}{2\epsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\epsilon_{i,j,k}}} \right) E_x|_{i,j,k}^{n} \quad + \quad \left(\frac{\frac{\Delta t}{\epsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\mu_{i,j,k}}} \right) \left(\begin{array}{c} \frac{H_z|_{i,j+1/2,k}^{n+1/2} - H_z|_{i,j,k-1/2}^{n+1/2}}{\Delta y} - \\ \frac{H_y|_{i,j,k+1/2}^{n+1/2} - H_y|_{i,j,k-1/2}^{n+1/2}}{\Delta z} \end{array} \right) \tag{B.10}$$

$$E_y|_{i,j,k}^{n+1} = \left(\frac{1 - \frac{\sigma_{i,j,k}\Delta t}{2\epsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\epsilon_{i,j,k}}} \right) E_y|_{i,j,k}^{n} \quad + \quad \left(\frac{\frac{\Delta t}{\epsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\mu_{i,j,k}}} \right) \left(\begin{array}{c} \frac{H_z|_{i,j,k+1/2}^{n+1/2} - H_z|_{i,j,k-1/2}^{n+1/2}}{\Delta z} - \\ \frac{H_z|_{i+1/2,j,k}^{n+1/2} - H_z|_{i-1/2,j,k}^{n+1/2}}{\Delta x} \end{array} \right) \tag{B.11}$$

$$E_x|_{i,j,k}^{n+1} = \left(\frac{1 - \frac{\sigma_{i,j,k}\Delta t}{2\epsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\epsilon_{i,j,k}}} \right) E_x|_{i,j,k}^{n} \quad + \quad \left(\frac{\frac{\Delta t}{\epsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\mu_{i,j,k}}} \right) \left(\begin{array}{c} \frac{H_y|_{i+1/2,j,k}^{n+1/2} - H_z|_{i-1/2,j,k}^{n+1/2}}{\Delta x} - \\ \frac{H_x|_{i,j+1/2,k}^{n+1/2} - H_x|_{i,j-1/2,k}^{n+1/2}}{\Delta y} \end{array} \right) \tag{B.12}$$

The last step is to set the grid spacings equal, $\Delta x = \Delta y = \Delta z = \Delta$, so that we are dealing with a uniform cubic grid. We group the material parameters together as a single coefficient.

$$\begin{aligned}
H_x|_{i,j,k}^{n+1/2} &= D_a(m)|_{i,j,k} H_x^{n+1/2} + D_b(m)|_{i,j,k}(E_y|_{i,j,k+1/2}^{n} - E_y|_{i,j,k-1/2}^{n} \\
&\quad + E_z|_{i,j-1/2,k}^{n} - E_z|_{i,j+1/2,k}^{n}) \\
H_y|_{i,j,k}^{n+1/2} &= D_a(m)|_{i,j,k} H_y^{n+1/2} + D_b(m)|_{i,j,k}(E_z|_{i+1/2,j,k}^{n} - E_z|_{i-1/2,j,k}^{n} \\
&\quad + E_x|_{i,j,k-1/2}^{n} - E_z|_{i+1/2,j,k}^{n}) \\
H_z|_{i,j,k}^{n+1/2} &= D_a(m)|_{i,j,k} H_z^{n+1/2} + D_b(m)|_{i,j,k}(E_x|_{i,j+1/2,k}^{n} - E_x|_{i,j-1/2,k}^{n} \\
&\quad + E_y|_{i-1/2,j,k}^{n} - E_y|_{i+1/2,j,k}^{n})
\end{aligned} \tag{B.13}$$

$$\begin{aligned}
E_x|_{i,j,k}^{n+1} &= C_a(m)|_{i,j,k} E_y|_{i,j,k}^{n} + C_b(m)|_{i,j,k}(H_z|_{i,j+1/2,k}^{n+1/2} - H_z|_{i,j-1/2,k}^{n+1/2} \\
&\quad + H_y|_{i,j,k-1/2}^{n-1/2} - H_y|_{i,j,k+1/2}^{n+1/2}) \\
E_y|_{i,j,k}^{n+1} &= C_a(m)|_{i,j,k} E_y|_{i,j,k}^{n} + C_b(m)|_{i,j,k}(H_x|_{i,j,k+1/2}^{n+1/2} - H_x|_{i,j,k-1/2}^{n+1/2} \\
&\quad + H_z|_{i-1/2,j,k}^{n+1/2} - H_z|_{i+1/2,j,k}^{n+1/2}) \\
E_z|_{i,j,k}^{n+1} &= C_a(m)|_{i,j,k} E_z|_{i,j,k}^{n} + C_b(m)|_{i,j,k}(H_y|_{i+1/2,j,k}^{n+1/2} - H_y|_{i-1/2,j,k}^{n+1/2}
\end{aligned}$$

$$+H_z|_{i,j-1/2,k}^{n+1/2} - H_x|_{i,j+1/2,k}^{n+1/2}) \tag{B.14}$$

The electric field coefficients used above are:

$$C_a(m)|_{i,j,k} \;=\; \frac{1-\frac{\sigma_{i,j,k}\Delta t}{2\epsilon_{i,j,k}}}{1+\frac{\sigma_{i,j,k}\Delta t}{2\epsilon_{i,j,k}}} \qquad \text{and} \qquad C_b(m)|_{i,j,k} \;=\; \frac{\frac{\Delta t}{\epsilon_{i,j,k}\Delta}}{1+\frac{\sigma_{i,j,k}\Delta t}{2\epsilon_{i,j,k}}},$$

and the magnetic field coefficients are:

$$D_a(m)|_{i,j,k} \;=\; \frac{1-\frac{\Delta t}{2\mu_{i,j,k}}}{1+\frac{\Delta t}{2\mu_{i,j,k}}} \qquad \text{and} \qquad D_b(m)|_{i,j,k} \;=\; \frac{\frac{\Delta t}{\mu_{i,j,k}\Delta}}{1+\frac{\Delta t}{2\mu_{i,j,k}}}.$$

So for one dimension, we have the following FDTD equations:

$$E_x|_k^{n+1} \;=\; C_a(m)|_k E_x|_k^n + C_b(m)|_k \left(H_y|_{k+1/2}^{n+1/2} - H_y|_{k-1/2}^{n+1/2}\right) \tag{B.15}$$

$$H_y|_{k-1/2}^{n+1/2} \;=\; D_a(m)|_k H_y|_{k-1/2}^{n-1/2} + D_b(m)|_k \left(E_x|_k^n - E_x|_{k-1}^n\right). \tag{B.16}$$

Algorithm Stability

Since an electromagnetic pulse cannot propagate faster than the speed of light, we ensure the one in our algorithm doesn't either. To do this, the time taken to propagate across one cell, Δx, is fixed to $\Delta t = \Delta x/c$. For one dimensional simulations this is absolutely convergent. For two dimensional simulations, the condition becomes $\Delta t = \Delta x/(\sqrt{2}c)$, allowing for motion diagonally across a cell. Three dimensions is a simple extension of this, and in general we write this Courant Condition for n-dimensions as $\Delta t \leq \Delta x/\sqrt{n}c$[66, 67].

Appendix C

Genetic Algorithms

Genetic algorithms are particularly intuitive because of the parallels they share with evolutionary processes. The general idea is to solve a problem (often an optimizing one) by initializing a *population* of guesses. These individual guesses are evaluated for how well they address the problem at hand. Accordingly, a *fitness* for each individual is assigned. Better-performing individuals have a better chance of passing on traits, by offspring solutions, much like the process in Nature that we know as survival of the fittest. Recombination and mutation, operators borrowed from the field of genetics, are applied to individuals in an effort to maintain the diversity of a population. The algorithm simulations until some pre-destined cutoff is reached: either a computational limit is reached or a satisfactory fitness is found.

There exist many types of genetic algorithms (messy, hybrid, order-based, *etc.*) and many methods of implementation [68, 69]. Certainly, this flexibility of design has enabled genetic algorithms to be used in a vast array of disciplines, and it should be noted that function optimization utilizes only a small fraction of a genetic algorithm's capabilities [60]. More details can be found in Refs. [26, 70] and [71]; the ICGA proceedings [72] provide a good source of results regarding genetic algorithm theory and applications.

This chapter is organized as follows. Section 2.1 introduces the main user-specified parameters. Section 2.2 verifies the sensitivity of the algorithm to changes in each of these parameters, while section 2.3 concludes with a summary of the algorithm's sensitivity to these parameters.

Individual	String
1	101100110111001
2	011010110001110
3	011110100011101
4	010110100111010
5	111011001101000
6	100110101111000
7	111000011001101
8	101101111001001
9	110011010011100
10	001010101101100

Table C.1: An example of initial population strings generated by the genetic algorithm.

C.1 Initializing the Algorithm

Creating an Initial Population

To begin a genetic algorithm, we need an initial population. Our initial population is generated by collecting a random series of 0's and 1's to generate the strings of each individual.

There is a trade-off between the number of individuals in a population and how many generations it takes to converge. Too small a population means the algorithm may take longer to find the right combinations before converging. Each generation will be quicker to evaluate, for there are fewer members from which to make selections. Widely scattered populations (more likely with a larger initial population size) tend to have a better chance of finding good solutions early on, although each generation may be slower to process.

Experience tells us that the ideal population size depends upon the function at hand. As with all parts of the genetic algorithm, one wishes to use the minimum precision necessary. Some functions perform well with $N = 10$ individuals but others need $N = 10000$.

Limiting the Search Space

When seeking an optimal solution one must always consider any computational limits at hand. To reduce the space in which the algorithm searches, restrictions are placed upon the range of variables (*unknowns*) that are to be optimized . For example, the following test function has maxima at $(1,2),(5,4)$ and $(6,6)$.

$$\begin{aligned} f(x,y) &= e^{-(x-2)^2}+1.5e^{-4(x-1)^2}e^{-4(y-2)^2} \\ &+ 2.5e^{-4(x-5)^2}e^{-4(y-4)^2}+1.5e^{-4(x-6)^2}e^{-4(y-6)^2} \end{aligned} \tag{C.1}$$

From Fig. C.1, one can see that the peak at $(5,4)$ is the global maximum. If the search space is restricted to $x\epsilon[0,4]$ two out of the three maxima are omitted.

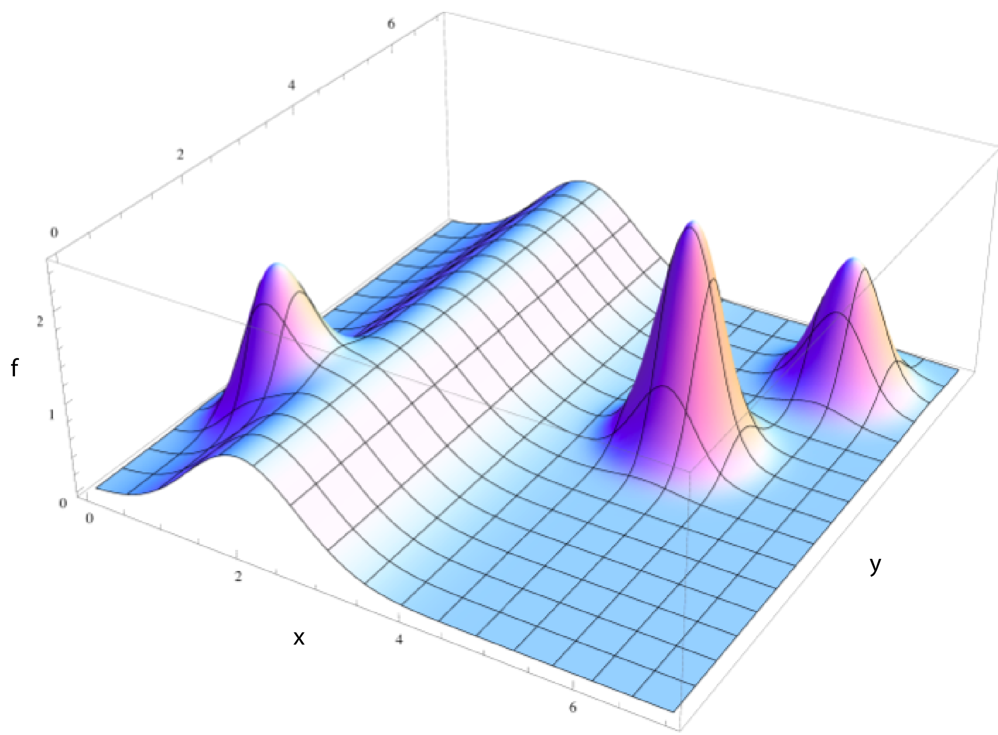

Figure C.1: One of the test functions used to validate the genetic algorithm.

Over-restricting the search space can therefore result in not finding the best solution, while not restricting the space at all can be computationally expensive.

C.2 Genetic Operators

Selection

In the algorithm used in this dissertation, "roulette wheel" selection was used, whereby the probability of selection is proportional to that individual's fitness. Better-performing individuals maintain a higher probability of reproduction. The selection mechanism is performed twice, so that a total of two individuals can be chosen to undergo, or to not undergo, crossover. Selection continues until N (the population size) individuals have been selected.

This weighted selection method allows all individuals the chance of continuing in the next generation and therefore maintains the diversity of the population [69]. Over a number of generations the traits of less fit parents disappear.

Table C.2: An example of roulette selection: choosing an individual for crossover.

Individual	Fitness	Cumulative total	Random number n	Individual chosen
1	7	7	23	3
2	2	9	49	7
3	17	26	6	3
4	12	38	7	1
5	7	45	15	3
6	3	48	27	4
7	2	50	45	5
8	14	64	13	3
9	1	65	61	8
10	3	68	9	2

For roulette selection, the algorithm calculates the sum fitness of a generation, S. A random integer n, which is between 0 and S is called, and the individual who has a cumulative fitness equal or just greater than n is selected. Notice in table C.2 that individual 3 gets chosen the most, in accordance with having the best fitness. Lesser performing individuals, such as individual 9 still get chosen but not as frequently.

Crossover & No Crossover

Crossover is also known as recombination. We all have a direct familiarity with crossover in genetics: you and your siblings are not identical to your parents, but you inherit one or more attributes, be it skin tone, a facial feature or your body type. This is how crossover works. Good traits are preserved from generation to generation while allowing some new combinations to appear, too.

The same applies to recombination in the genetic algorithm. The introduction of new traits by crossover allows the algorithm to explore the search space by providing opportunities for novel solutions (in the form of offspring) to appear in the population. The most common form of crossover is done at a single-point along the parent chromosomes. The locus of interest is chosen at random, and the two parents switch half their string with each other. This enables good traits to persist in subsequent generations.

Crossover works better than no crossover provided the population contains sufficiently diverse members, otherwise there is a high probability that individuals will be crossing the same strings with each other. Representative samples of different building blocks for good solutions need to be present in the population, and the evaluation function needs to reflect the contributions of these building blocks. Finally, to work well, crossover needs to be capable of putting good building blocks together, and not somehow splitting the feature up when recombining the chromosomes of different individuals.

Mutation

Where crossover creates new individuals by the mixing of two parents, mutation does so by modification of a single individual.

This process occurs frequently in nature; quite often it is a replication error. For whatever its cause, mutation beneficially diversifies the population: novel chromosomes can be passed onto the new generation. Mutations that correspond to poorer fitnesses die out, but sometimes a better fitness (a better solution) is introduced by mutation. Thus, in coding, it prevents a genetic algorithm from converging around a sub-optimal solution. When modeling functions with many local maxima, mutation enables the algorithm to generate more possible solutions before converging on a particular one. It also aids the effectiveness of crossover when the population's diversity is maintained.

In practice, mutation probabilities between 0.01 and 0.2 seem to work well. If the mutation rate is too low, it barely has an effect. Mutation rates that are too high permute too many of the good genes, slowing down the algorithm's progress. One's task in setting mutation rates is to balance caution against excess.

C.3 Encoding and Decoding

Chromosomes & String Length

Choice of encoding method and string length seem like an optimization problem in themselves. I used binary encryption since it lends itself to easy implementation of the genetic operators in the algorithm (crossover, selection, and mutation). String length, on the other hand, remains particular to the problem at hand, and so it was determined through test simulations what length would provide sufficient encoding accuracy without excessively slowing the algorithm.

A Note on Accuracy

If a parameter is continuous then it must be quantized for encoding purposes. The range is limited between the upper and lower bounds, and divided between these points evenly. Depending on the bit length for a particular parameter, not all numbers will be possible, and some rounding is involved, either up or down, to match the set of allowable values. Error will always be present in quantizing continuous parameters; the amount depends on the length of each gene representing that parameter.

Lower bounds on the filter thickness, our unknown, eliminated the possibility of the algorithm optimizing for negative thicknesses.

Table C.3: The initial population used for testing genetic algorithm parameters.

Individual	Initial x	Initial y
1	1.14432847499847412	3.14193534851074219
2	3.02259731292724609	6.90307140350341797
3	6.61785030364990234	6.37295627593994141
4	3.94131422042846680	6.11682844161987305
5	5.01945543289184570	5.62727975845336914
6	0.564872443675994873	5.17120981216430664
7	0.432738363742828369	5.86820507049560547
8	1.43496441841125488	0.489470124244689941
9	2.99558973312377930	2.30775785446166992
10	4.52329635620117188	2.69720053672790527

Decoding the Unknowns

Next we must translate the binary strings into numbers with physical meaning. In our current example, each string represents a guess of the coordinate for the global optimum. An excellent explanation of this can be found in David A. Coley's book [26].

The following transformation makes use of the upper and lower bounds, r_{max} and r_{min}, that we imposed on the search space in the beginning. The binary string is first converted into a base-10 integer, z, and from there into a real number, r. The number of bits in each string, ℓ, are used to divide the range between upper and lower bounds into even increments of all allowable real values that a solution to this problem can have. We reach the transformation:

$$r = \frac{r_{max} - r_{min}}{2^{\ell} - 1} z + r_{min}. \tag{C.2}$$

For the purposes of comparing values of each genetic operator, a fixed initial population was used (see Table C.3). Here, we list the actual initial guesses that the genetic algorithm used for test simulations. There is a fairly even distribution of guesses across the search space.

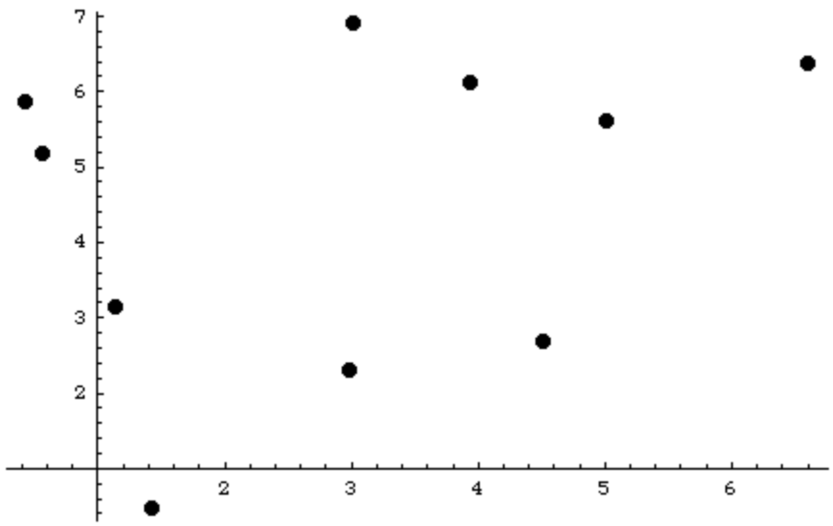

Figure C.2: Positions of the initial population, as randomly assigned by the genetic algorithm.

C.4 Assessing the Fitness

In assessing the fitness of each individual, some key decisions are made. These are specific to our algorithm, and may differ from how other genetic algorithms operate. We chose

1. a fixed objective (fitness) function – one that does not change over time. For example, possible damage of the optical filter due to the laser pulse is unaccounted for. We assume the filter appears the same for each successive generation, or we assume that each generation sees a new filter each time we run a simulation of the system.

2. binary encryption and fixed length strings. Many other coding methods are possible, but this lends itself well for our purposes.

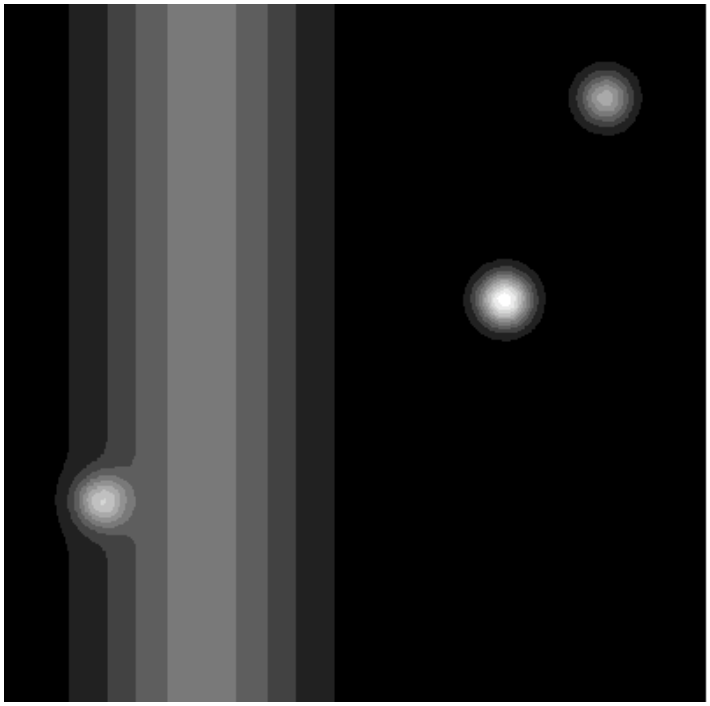

Figure C.3: A contour plot of the corresponding fitness function (below). White represents local maxima, while black represents minima.

3. a fixed size population.

This means that for each generation, the same number of individuals need to be evaluated for their ability. The algorithm creates as many filters as there are individuals, using individual 1's information for filter 1, *etc.* It performs an FDTD propagation of a laser pulse through each filter and records the reflected and transmitted fields. We chose to evaluate fitness based on a desired output pulse shape. Filters that generate a pulse shape and reflectance closer to that requested are assigned better finesses. All chromosomes are of equal size (the number of layers) and encoding accuracy.

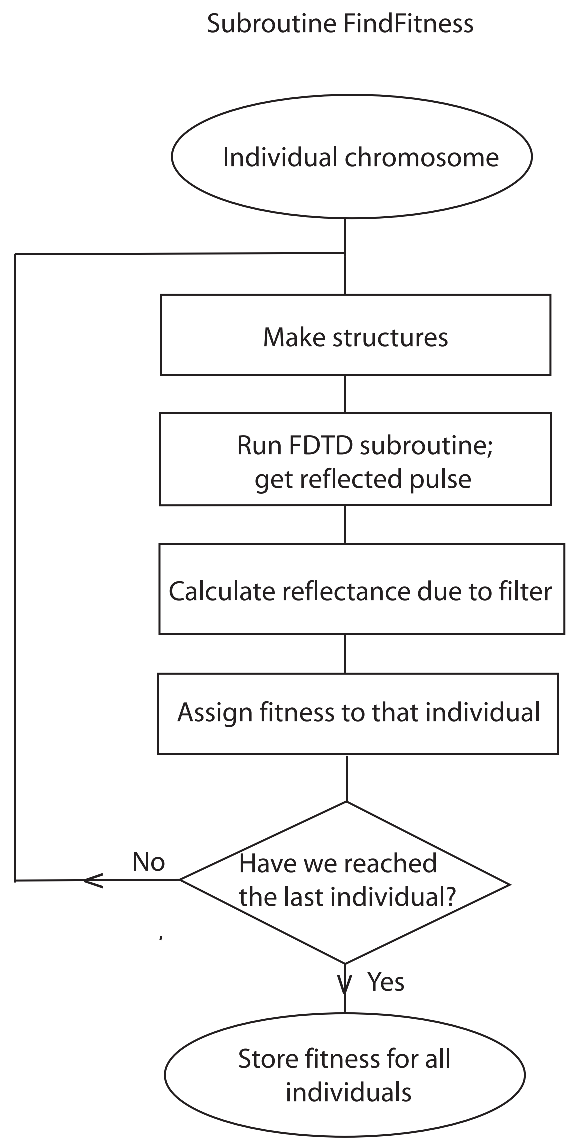

Figure C.4: Flowchart for fitness evaluation in the genetic algorithm.

C.5 Improving the Algorithm

Fitness Scaling

In some simulations, particularly fit individuals may appear early in the sequence. Fitness proportional selection allows these individuals to replicate in larger quantities, increasing

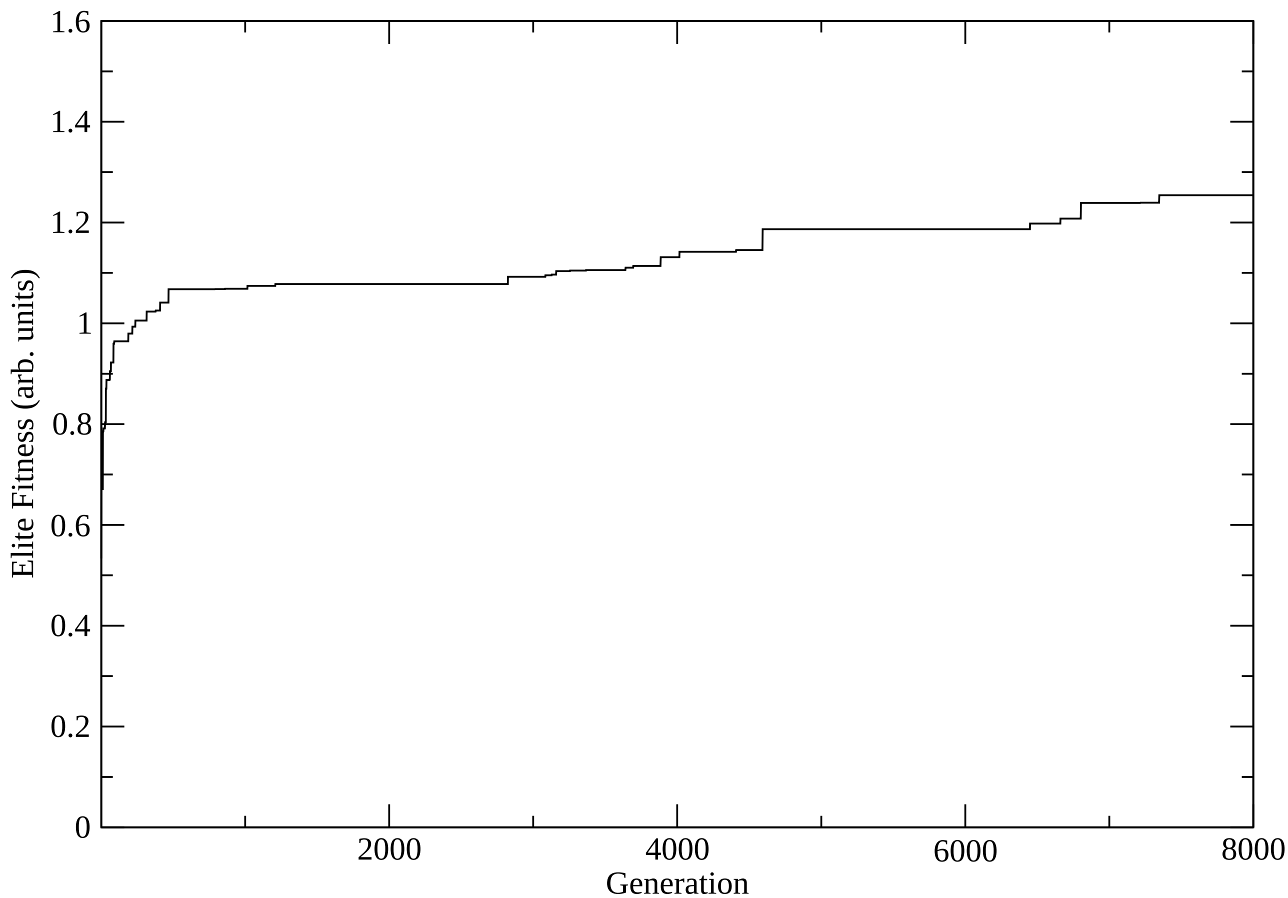

Figure C.5: The fitness of the elite member in the population as it evolved during the algorithm. This particular run had a population of ten individuals and a crossover probability of 0.6. As is often the case, progress is made rapidly and early on (steep increase in fitness), but it levels out once the individuals all reach a fairly good level of fitness.

their market share of their particular traits. On one hand rapid convergence is good, but only if it is convergence about the correct solution. To avoid convergence about a local (not global) optimum, fitness scaling is applied.

What we need is a method that prevents good individuals from dominating the optimization process, while still maintaining a selection pressure throughout the run. Linear fitness scaling works by calibrating the population about the average population fitness. This enables a fair distribution of copies to be made, from both better performing and averagely performing individuals.

Typical values for the scaling constant, s_c, are in the range 1.0 to 2.0. $s_c = 2.0$ means approximately twice as many elite individuals will propagate to the next generation than will average individuals.

Scaling is implemented right before selection so that the scaled results will impact the subsequent selection round. In our genetic algorithm, a linear transformation was used:

$$f_i^s(g) = a(g)f_i(g) + b(g) \tag{C.3}$$

where $f_i(g)$ is the true fitness of an individual, i, and $f_i^s(g)$ denotes its scaled fitness.

We should note here that such scalings can produce a negative fitness for the weaker members of the population. This can be avoided by setting negative values to zero, effectively turning off the scaling on those individuals.

Benefits of scaling:

1. If the fitness function produces negative values, scaling removes the possibility of these being translated into negative selection probabilities.

2. Slows dominance of super individuals.

Three types of scaling are in common use: linear scaling, power law scaling, and exponential scaling.

Elitism

Elitism provides a mechanism for copying the best individuals from one generation into the subsequent generation. Without this system, offspring and other individuals evolve with little regard for the quality of the individual being replaced. By allowing the top individual another generation in which it may produce offspring, we ensure excellent solutions have another opportunity to express themselves as the final solution.

The Web Interface

The focus of this section is the development of the web interface for the GA-FDTD algorithm. While all of the simulations mentioned in this dissertation were run through a terminal window, we wanted to have the option of platform-independent remote access for other users. Jeff Deroshia was instrumental in setting up the web forms, and testing the php algorithm for accuracy. He also set up the password-protected accounts, so that each run is stored a file with a user name identifier.

The web interface comprises of a series of php and html forms that collect information from the user. Default values are offered in the boxes, and the information collected will depend on the type of run the user specifies at the outset. This information generates the parameter files for the run. Some secondary parameters rely on this primary information for their values. The interface takes care of these self-consistent calculations, both minimizing potential error in the input files and automating parts that the user needn't specify directly. Figs. C.6 - C.9 show images from the web interface.

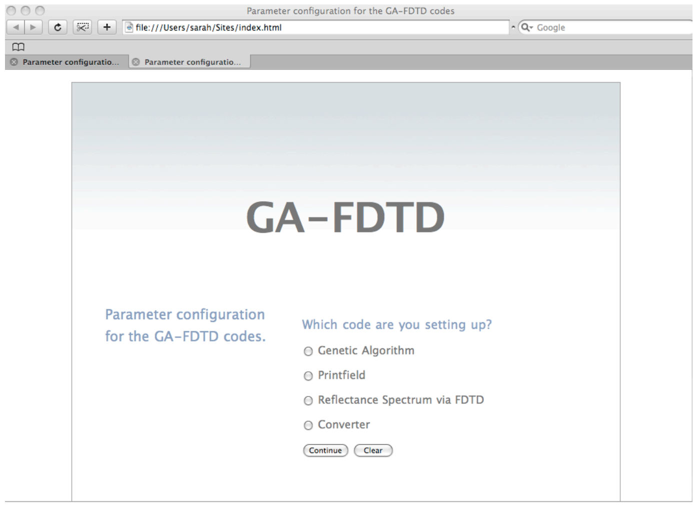

Figure C.6: The first page of the web interface asks the user what type of operation they want. Options are optimization of a new filter, calculating the reflected and transmitted fields due to a filter, calculating the reflectance spectrum of a filter, or converting a filter file to a higher resolution grid.

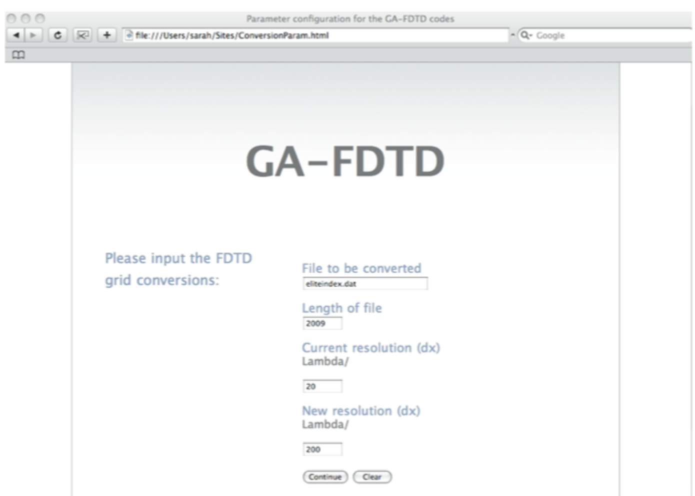

Figure C.7: Parameters for converting an index file to a different resolution. The algorithm needs the file name and length as well as the old and new resolution in terms of the central wavelength.

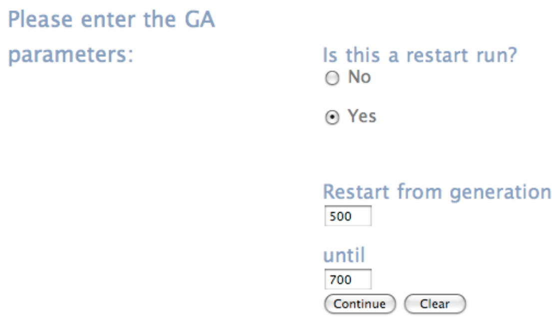

Figure C.8: Parameters for a restarted GA-FDTD run. The web form asks from what generation the algorithm will resume and also for the new maximum generation.

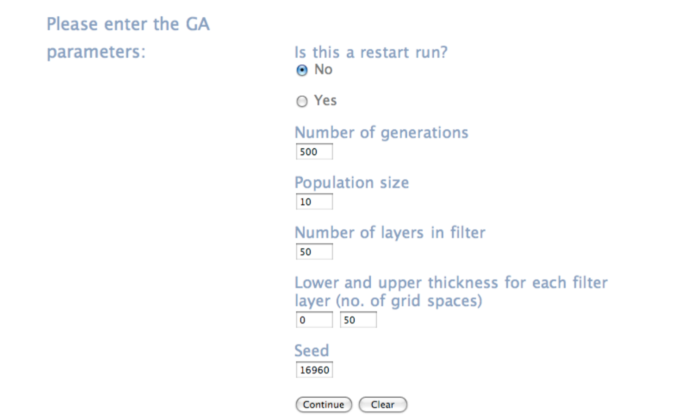

Figure C.9: Image of the web interface showing the parameters for a full GA-FDTD run. Both pulse and filter parameters are required.

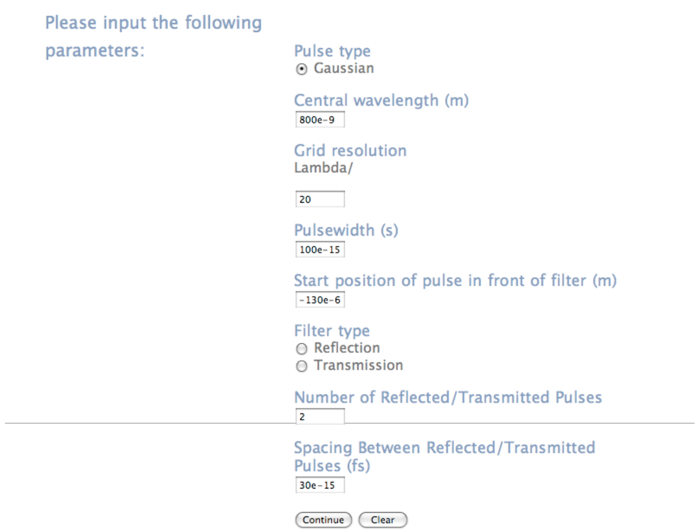

Figure C.10: Image of the web interface showing the parameters for the incident pulse and simulational resolution.

Appendix D

Pseudorandom Number Generators

Definitions

Linear feedback shift register: A shift register where the input bit is a function of the previous output state.
Mersenne prime: A prime number that is also one less than a power of two, $2^n - 1$. There are currently 46 known Mersenne Primes.
Period: The number of values a generator outputs prior to repetition of the sequence. A generator that produces values modulo some value m, is seen to have a full period if the period length is equal to m.
Shift register: A group of mathematical operations that shift the bits left or right by one or more positions.

The Mersenne Twister

This generator uses a Mersenne prime, $2^{19937} - 1$, in the parameters and a twisting algorithm that shifts the seed on a repeated basis. Essentially the algorithm is a very large linear-feedback shift register. The 19937 bit seed is stored in a 624 element array of 32 bit integers, which more than satisfies the needs of the genetic algorithm presented here.

The Mersenne Twister algorithm is available online [34], courtesy of its creators Makoto Matsumoto and Takuji Nishimura.

The Linear Congruent Method

Much simpler than the Mersenne twister, a linear congruent generator (LCG) begins with the recurrence relation

$$X_{n+1} = (aX_n + c) \bmod m. \tag{D.1}$$

Where X_0 initializes the sequence, a scales the sequence and c increments the sequence. m is the modulus. LCG's will generate satisfactory strings of pseudorandom numbers but only for careful parameter choice. RANDU, an infamous linear congruent generator of the 1960's was an example of nonrandom sequences due to poor parameter choice [73]. It was included in IBM's scientific computation package for the System/360 mainframe computers and removed once researchers realized many of the numbers fell into planar correlations within the three dimensional number space.

A careful choice of a, c, and m circumvents this [74]:

1. c and m are relatively prime,
2. $a - 1$ is divisible by all prime factors of m
3. $a - 1$ is a multiple of 4 if m is a multiple of 4

Relatively prime means they share no common positive factors (divisors) except the number1. George Marsaglia's random number generator, also available online [33], uses a congruent generator as its core. The algorithm includes a shift register that moves bits by ± 3 places and has a 2-bit multiply with carry generator, to further increase the generator's period. A subtract-with-borrow step, developed by Marsaglia, gives very long periods from a comparatively simple algorithm. A number in the cache is replaced by the difference of two others:

$$a_i = (a_{i+20} - a_{i+5} - b) \tag{D.2}$$

b is set to 0 or 1, the latter being used if computation of a_{i-1} caused overflow in the 32-bit algebra. Marsaglia's version was implemented as an alternative pseudorandom number generator for the genetic algorithm testing.

The Minimal Random Number Generator

A second good LCG algorithm comes is the minimal random number generator of Park and Miller [35]. The above-mentioned choices of a and m are incorporated into the algorithm, and c is set to zero. The algorithm prevents sequential correlations by shuffling the output with a Bays-Durham method. As an added safeguard, the algorithm manipulates the choice, should zero be the seed (apparently a common mistake made by users of this algorithm).

The version of the minimal random number generator used in our genetic algorithm testing came from Timothy Kaiser's Advanced Fortran 90 webpage [75].

Appendix E

Exploring Genetic Algorithm Parameter Variation in the Frequency Domain

In designing dielectric stacks with simulational methods, a balance must be struck between the complexity of the algorithm and the resources available for computation [68, 76, 77]. Although this is to some extent affected by processing power, generally it is a good idea to work with a manageable-sized algorithm so that one can gain feedback at greater intervals and thereby make design adjustments. To determine the best parameters for minimizing run time and maximizing dielectric stack fitness, I tested various sizes of genetic populations and compare that to the ability of the dielectric stack performance. Also, each of the simulations presented below were set with several different seeds, so that the results could be independent of the algorithm's starting position.

The number of generations controls the total run time of the algorithm. For the generation variations the population was held at 20 individuals. The run times for smaller generations took only a few seconds. Generations of 50, 100, and 200 were all under a few minutes run time, so I also simulated generations of 500, 1000, 2000, and 4000 to see if the payoff was much better for the longer run. Although these were very fast simulations, the larger populations did not yield much of an improvement in the dielectric stack fitness. It was a case of diminishing returns. The upper pane of Fig. E.1 shows the trailing off of fitness as a function of generation. A population of 500 yielded a good return for its computational load. For the generation variations, the population was held at 20 individuals. This made the shorter simulations (fewer generations) very fast.

The population size - the number of individual dielectric stacks compared at any one time by the algorithm - sets the demand for computational resources at any one time. The algorithm is in series for the frequency domain calculations, which means that each dielectric stack must have its spectrum and associated fitness calculated before the rest of the algorithm proceeds. Increasing the number dielectric stacks makes each generation of the algorithm take longer. More dielectric stacks can also be a good thing, as it allows exploration of wider computational space. To understand at what point the trade between speed and thoroughness ceased to benefit the algorithm's final answer, several preliminary simulations were performed. The lower pane of Fig. E.1 shows the results of this exploration for the frequency domain calculation of dielectric stacks. Populations of 10, 20, 50, 100, 200, 500, and 1000 dielectric stacks were simulated and plotted against the value of the best fitness reached during the run. As a result of these investigations, it appears that a population of 10 provides the best return for computational expense. Fig. E.1 shows that although the other populations of 20, 50 and 500 gave a slightly higher fitness by the end of their simulations, the improvement was small over the increase in population. In cases where the algorithm is compared to previous results in the literature, I used the same population as the comparison work. If no population was specified, then a population of 10 was used as it gave the best results *per capita*.

For this optimization, one would expect population increases to yield better returns, since the larger population maintained a larger gene pool. It turned out that a small population could find a good filter design just as well as the larger populations. This suggests that as long as the algorithm has one good answer, having a lot of average dielectric stacks doesn't add much to the algorithm's performance.

It is worth noting that although the total number of layers plays a role in the speed of computation, it does so to a lesser degree than the already-mentioned parameters. The layer number becomes more significant when one reproduces a filter that matches a sharper

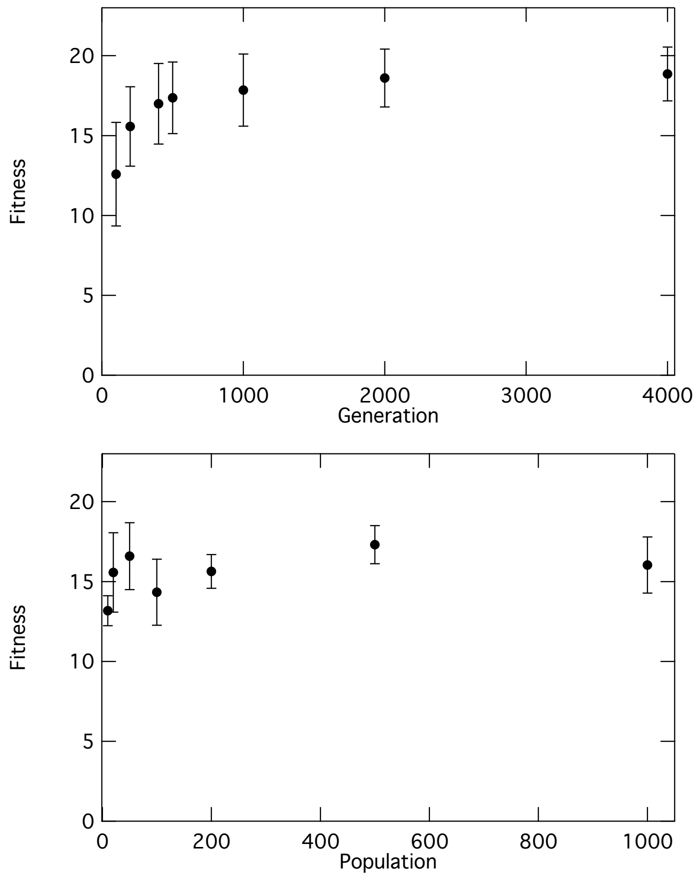

Figure E.1: Variation of the generation (upper) and population (lower) as functions of fitness for the algorithm's elite filter.

spectral function. For example, a bandpass filter can be specified by a step function. The sharper a spectral function is, the more Fourier components are required to specify the shape. Consequently, a smoother spectral profile will be accurately represented by fewer Fourier harmonics, which translates into fewer filter layers being needed to produce the target spectrum. In the case of the pulse-shaped Fourier transforms, the resultant spectral functions are relatively undulating, so the 50 - 150 layer templates are adequate for this type of filter design.

Appendix F

Symbols and Abbreviations

This section presents many of the symbols and abbreviations that appear repeatedly in the text. As much as possible, these symbols are consistent with generally-used versions, although there may be some discrepancies where the general literature is known to interchangeably use different symbols to represent the same physical quantity. Conversely, some symbols (such as the letter m) are used more than once, each time representing different quantities.

d_k	The physical thickness of the k^{th} layer in a thin film.
$\mathbf{E}$	The electric field.
$\mathcal{F}$	The fitness of a particular filter design.
$\mathbf{H}$	The magnetic field.
I	The pulse intensity.
k	The imaginary part of the refractive index. The presence k in the complex refractive index, $N = n - ik$, indicates absorption.
n	The real part of the refractive index.
N	The complex refractive index, $N = n - ik$.

R	Reflectance. The ratio of reflected intensity to incident intensity at a boundary.
T	Transmittance. The ratio of the transmitted intensity to incident intensity at a boundary.
TE	Transverse electric. The electric vector is polarized such that it propagates normal to the plane of incidence.
TM	Transverse magnetic. The magnetic vector is polarized such that it propagates normal to the plane of incidence.
$\mathcal{Y}$	Admittance.
$\mathcal{Y}_{\prime}$	Admittance of free space.
α	Absorption coefficient.
ϵ	The permittivity of the medium.
λ	The wavelength of light. Typically the wavelength in free space is given.
μ	The permeability of the medium.
ϕ	The phase of a wave.
σ	The incident or reflected pulsewidth.
θ	The incident angle of light on a medium.

Bibliography

[1] *http : //www.bbk.ac.uk/boyle/boyle_papers/boylepapers_index.htm.*

[2] R. G. Driggers, Ed., *Encyclopedia of optical engineering*, vol. 2, CRC Press (2003).

[3] T. Young, "The theory of light and colour," *Phil. Trans. Roy. Soc.* **92**(12) (1802).

[4] E. S. Barr, "Men and milestones in optics ii. thomas young," *Appl. Opt.* **2**(6), 639–647 (1963).

[5] R. H. Silliman, *Biography of Augustin Fresnel, in Dictionary of Scientific Biography* (1970).

[6] H. A. Macleod, *Thin Film Optical Filters*, Institute of Physics Publishing, Philadelphia, third ed. (2001).

[7] A. H. Pfund, "Highly reflecting films of zinc sulphide," *J. Opt. Soc. Am.* **24**, 99–102 (1934).

[8] J. Strong, "On a method of decreasing the reflection from non-metallic substances," *J. Opt. Soc. Am.* **26**, 73–74 (1936).

[9] W. Geffcken, "Interferenzlichtfilter," Tech. Rep. 716153, Germany Patent (1939).

[10] A. Thelen, *Thin Films on Glass*, ch. The pioneering contributions of W Geffcken, 227–239. Springer, Berlin (1997).

[11] O. S. Heavens, *Optical Properties of Thin Solid Films*, Academic Press (1955).

[12] B. T. S. et. al, "Manufacture of complex optical multilayer filters using an automated deposition system," *Vacuum* **51**, 647–654 (1998).

[13] K. A. Craig, "Method for analysis of the characteristic matrix in optical systems," *J. Opt. Soc. Am. A* **4**, 1092–1096 (1987).

[14] K. S. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *IEEE Trans. Antennas and Propagat.* **14**, 302–307 (1966).

[15] J. C. Maxwell, "On physical lines of force," *The London, Edinburgh, and Dublin Philosphical Magazine and Journal of Science* (1861).

[16] E. J. Bond, L. Xu, S. C. Hagness, and B. D. V. Veen, "Microwave imaging via space-time beamforming for early detection of breast cancer," *IEEE Transactions on Antennas and Propagation* **51**, 1690–1705 (1993).

[17] B. Chaudhury and S. Chaturvedi, "Three-dimensional computation of reduction in radar cross section using plasma shielding," *IEEE Transactions on Plasma Science* **33**, 2027 – 2034 (2005).

[18] B. Chaudhury and S. Chaturvedi, "3-d simulations for radar cross-section reduction using plasma absorbers," in *The 33rd IEEE International Conference on Plasma Science, 2006. ICOPS 2006. IEEE Conference Record - Abstracts.*, 24–24, IEEE, (Traverse City, MI) (2006).

[19] X. Li, S. C. Hagness, B. D. V. Veen, and D. van der Weide, "Experimental investigation of microwave imaging via space-time beamforming for breast cancer detection," in *IEEE MTT-S International Microwave Symposium Digest*, (2003).

[20] L. Xu, S. K. Davis, S. C. Hagness, D. W. van der Weide, and B. D. V. Veen, "Microwave imaging via space-time beamforming: experimental investigation of tumor detection in multilayer breast phantoms," *IEEE Transactions on Microwave Theory and Techniques* **52**, 1856 – 1865 (2004).

[21] P. Baumeister, "Design of multilayer filters by successive approximation," *J. Opt. Soc. Am.* **48**, 955–958 (1958).

[22] J. A. Aguilera, J. Aguilera, P. Bauermeister, A. Bloom, D. Coursen, J. A. Dobrowolski, F. T. Goldstein, D. E. Gustafson, and R. A. Kemp, "Antireflection coatings for germanium ir optics: a comparison of numerical design methods," *Appl. Opt.* **27**, 2832–2840 (1988).

[23] J. A. Dobrowolski and R. A. Kemp, "Refinement of optical multilayer systems with different optimization procedures," *Appl. Opt.* **29**, 2876–2893 (1990).

[24] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI (1975).

[25] J. M. Yang and C. Y. Kao, "An evolutionary algorithm for the synthesis of multilayer coatings at oblique light incidence," *J. Lightwave Tech.* **19**, 559–569 (2001).

[26] D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific, Singapore (1999).

[27] D. G. Li and A. C. Watson, "Optical thin film optimization design using genetic algorithms," in *IEEE International Conference on Intelligent Processing Systems*, IEEE (1997).

[28] A. V. Tikhonravov and J. A. Dobrowolski, "Quasi-optimal synthesis for antireflections coatings: a new method," *Appl. Opt.* **32**, 4265–4275 (1993).

[29] V. Pervak, A. V. Tikhonravov, M. K. Trubetskov, J. Pistner, F. Krausz, and A. Apolonski, "Band filters: two-material technology versus rugate," *Appl. Opt.* **46**(8), 1190–1193 (2007).

[30] A. V. Tikhonravov, "Some theoretical aspects of thin-film optics and their applications," *Appl. Opt.* **32**(28), 5417–5436 (1993).

[31] C. Liebig and W. M. Dennis, "Simulation of interactions of high-intensity ultrashort pulses with dielectric filters," *Opt. Eng.* **46**(023801) (2007).

[32] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall, "Open mpi: Goals, concept, and design of a next generation mpi implementation," in *11th European PVM/MPI Users' Group Meeting*, (Budapest, Hungary) (2004).

[33] *http://www.fortran.com/kiss.f90.*

[34] M. Matsumoto, "Mersenne twister home page, http://www.math.sci.hiroshima-u.ac.jp/ m-mat/mt/emt.html."

[35] S. K. Park and K. W. Miller, "Random number generators: good ones are hard to find," *Commun. ACM* **31**(10), 1192–1201 (1998).

[36] A. Assion, T. Baumert, M. Bergt, T. Brixner, B. Kiefer, V. Seyfried, M. Strehle, and G. Gerber, "Control of chemical reactions by feedback-optimized phase-shaped femtosecond laser pulses," *Science* **282**(5390), 919 – 922 (1998).

[37] J. Konradi, A. Scaria, V. Namboodiri, and A. Materny, "Application of feedback-controlled pulse shaping for control of cars spectra: the role of phase and amplitude modulation," *J. of Raman Spectro.* **38**(8), 1006 – 1021 (2007).

[38] R. R. Frontiera and R. A. Mathies, "Polarization dependence of vibrational coupling signals in femtosecond stimulated raman spectroscopy," *J. Chem. Phys.* **127**(12), 124501 (2007).

[39] T. Shih, R. R. Gattass, C. R. Mendonca, and E. Mazur, "Faraday rotation in femtosecond laser micromachined waveguides," *Opt. Express* **15**(9), 5809–5814 (2007).

[40] C. B. Schaffer, A. Brodeur, J. F. García, and E. Mazur, "Micromachining bulk glass by use of femtosecond laser pulses with nanojoule energy," *Opt. Lett.* **26**(2), 93–95 (2001).

[41] B. Kohler and *et al.*, "Quantum control of wave packet evolution with tailored femtosecond pulses," *Phys. Rev. Lett.* **74**(17), 3360 – 3363 (1995).

[42] A. M. Weiner, "Femtosecond pulse shaping using spatial light modulators," *Review of Scientific Instruments* **71**(5), 1979 (2000).

[43] P. Nuernberger, G. Vogt, T. Brixner, and G. Gerber, "Femtosecond quantum control of molecular dynamics in the condensed phase," *Phys. Chem. Chem. Phys.* **9**, 2470 – 2497 (2007).

[44] S. Martin, J. Rivory, and M. Schoenauer, "Synthesis of optical multilayer systems using genetic algorithms," *Appl. Opt.* **34**(13), 2247–2254 (1995).

[45] T. Eisenhammer, M. Lazarov, M. Leutbecher, U. Schöffel, and R. Sizmann, "Optimization of interference filters with genetic algorithms applied to silver-based heat mirrors," *Appl. Opt.* **32**(31), 6310–6315 (1993).

[46] H. Grenier, "Robust optical coating design with evolutionary strategies," *Appl. Opt.* **35**(28), 5477–5483 (1996).

[47] M. Born and E. Wolf, *Principles of Optics*, Pergamon Press, Oxford, sixth ed. (1980).

[48] *http://www.rp-photonics.com/encyclopedia.html* (2009).

[49] D. M. Finlayson and B. D. Sinclair, *Advances in lasers and applications*, Institute of Physics Publishing (1999).

[50] J. Ville, "Theorie et applications de la notion de signal analytique," *Cables et Transmission* **2A**, 61–74 (1948).

[51] Wavemetrics, "Igor pro 5.05b." Software.

[52] M. Cid, N. Stem, C. Brunetti, A. F. Beloto, and C. A. S. Ramos, "Improvements in anti-reflection coatings for high-efficiency silicon solar cells," *Surface and Coatings Technology* **106**, 117–120 (1998).

[53] E. Michielssen, S. Ranjithan, and R. Mittra, "Optimal multilayer filter design using real coded genetic algorithms," *J. IEE Proceedings* **139**, 413 – 420 (1992).

[54] J. M. Yang and C. Y. Kao, "Efficient evolutionary algorithm for the thin-film synthesis of inhomogenious optical coatings," *Appl. Opt.* **40**, 3256–3267 (2001).

[55] S. Kirkpatrick, C. D. Gellatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science* **220**, 671–680 (1983).

[56] C. Tsallis and D. A. Stariolo, "Generalized simulated annealing," *Physics A* **233**, 395–406 (1996).

[57] I. O. Bohachevsky, M. E. Johnson, and M. L. Stein, "Generalized simulated annealing for function optimization," *Technometrics* **28**, 209–217 (1986).

[58] S. Martin, J. Rivory, and M. Schoenauer, "Synthesis of optical multilayer systems using genetic algorithms," *Appl. Opt.* **34**, 2247–2254 (1995).

[59] M. Mitchell, S. Forrest, and J. H. Holland, "The royal road for genetic algorithms: Fitness landscapes and ga performance," in *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, 1991*, F. J. Varela and P. Bourgine, Eds., 245–254, MIT Press, (Paris) (1992).

[60] S. Forrest and M. Mitchell, "What makes a problem hard for a genetic algorithm? some anomalous results and their explanation.," in *Machine Learning.*, **13**, 285–319 (1993).

[61] K. de Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, Ann Arbor, MI (1975).

[62] M. Mitchell, "Genetic algorithms: An overview," *Complexity* **1**(1), 31–39 (1995).

[63] A. D. Bethke, *Genetic Algorithms as Function Optimizers.* PhD thesis, University of Michigan, Ann Arbor, MI (1980).

[64] E. B. Treacy, "Optical pulse compression with diffraction gratings," *EEE J. Quantum Electron.* **QE-5**(454) (1965).

[65] C. V. S. et al, "Compression of femtosecond optical pulses," *Appl. Phys. Lett.* **40**(761) (1982).

[66] K. S. Kunz and R. J. Luebbers, *The Finite Difference Time Domain Method for Electromagnetics*, CRC Press, Boca Raton, FL (1993).

[67] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time Domain*, Artech House, Boston, third ed. (2005).

[68] G. Harik, D. E. Goldberg, E. Cantu-Paz, and B. L. Miller, "The gambler's ruin problem, genetic algorithms, and the sizing of populations," *Evolutionary Computation* **7**, 231–253 (1999).

[69] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *The Clearinghouse for Genetic Algorithms*, Department of Engineering Mechanics, University of Alabama (1990).

[70] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, J. Wiley and Sons, Inc., New York, NY, USA (1998).

[71] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass. (1989).

[72] *http://www.isgec.org/books-hz.html*, International Society for Genetic and Evolutionary Computation - ISGEC.

[73] G. Marsaglia, "Random numbers fall mainly in the planes," *Proc. Natl. Acad. Sci.* **61**(1), 25–28 (1968).

[74] D. E. Knuth, *The Art of Computer Programming*, vol. 2, Addison-Wesley, Reading, Massachusetts, 3 ed. (1997).

[75] T. H. Kaiser, *Advanced Fortran 90, http://www.sdsc.edu/t̃kaiser/f90.html#Format%20for%20our%20t*

[76] L. D. Davis, Ed., *The Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, NY (1991).

[77] D. E. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Kluwer Academic Publishers, Norwell, MA, USA (2002).