A Modified Genetic Algorithm for Design Peak Pulse Power in Nonlinear Transmission Lines

by

Hai Lu

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 9, 2011

Keywords: Nonlinear transmission lines; peak pulse power;
Genetic Algorithm; interface; circuit optimization;

Approved by

Michael Baginski, Chair, Associate Professor of Electrical and Computer Engineering
Hulya Kirkici, Associate Professor of Electrical and Computer Engineering
Stuart Wentworth, Associate Professor of Electrical and Computer Engineering

Abstract

Non-linear transmission lines (NLTL) have become increasingly popular as a way to develop very high frequency, wide band waveforms that offer extremely fast rise times usually unobtainable by any other method. The applications of non-linear transmission lines range from very wideband telecommunications systems to high energy physics. An integrated optimization algorithm was built using between Matlab and LTspice to optimize the performance of the NLTL. A bidirectional interface was also created that allowed data to be easily transferred between the circuit simulator and the optimization algorithm batch command. A genetic algorithm was used for the optimization and designed for relative rapid convergence. Using this integrated optimization method, the NLTL circuit was designed that showed a peak pulse power enhancement of 570 % without the tedious trial and error process usually required.

Acknowledgments

I would like to thank my advisor Dr. Michael Baginski for his help and inspiration over the past two years of this project. It was he who helped me overcome several obstacles during the research process. Also, I would like to thank my committee members Dr. Kirkici and Dr. Wentworth for their help on the theoretical problem description and experimental evidence. I would especially like thank my parents for their tremendous support during my entire life.

Table of Contents

List of Figures

List of Abbreviations

NLTL	Nonlinear Transmission Lines
GA	Genetic Algorithm
GUI	Graphical User Interface

1. Introduction

1.1. Overview

Since 1940s non-linear materials have been used in transmission lines to generate wide-band signals having an extraordinary peak pulse powers. These types of waveforms are made possible by taking advantage of the nonlinear material properties to create solitons that propagate down the periodic structure. Over the last six decades a significant amount of research has focused on developing applicable, non-linear materials as well as designing novel transmission line housings. In 1950's linear capacitors were combined with nonlinear inductors as the basis for lumped-element Non-Linear Transmission Line's (NLTL's) that produced sub-us pulses. By the 1960's high resistivity ferrite cores were added to NLTL's that allowed even higher frequency pulses (pulse pumps). The Star Wars program of the 1980's created a need for extremely high power – short duration sources. This need lead to further advancements in NLTL designs that included the addition of periodically separated blocks of nonlinear ceramics in NLTL's or "shocklines". By the 1990's, fast responding Schottky diodes were used as nonlinear capacitance elements increasing the NLTL bandwidth further.

1.2. Soliton Theory

A soliton is defined as a wave that preserves its shape and propagates at a constant velocity. The technology of generating solitons [1] using nonlinear materials has been researched since 1940s and is widely used for many industrial applications. For instances, ultra-short duration pulse lasers rely on optical solitons to deliver the repetitive, ultra short duration pulses. This was first observed when researchers showed that the nonlinear and dispersive nature of nonlinear optical materials could reshape a series of relatively long duration pulses into a series of much shorter duration optical solitons [1]. Using the same concept, research in the microwave area has investigated different types of dielectric and magnetic materials to generate similar solitons.

A NLTL creates solitons or quasi-solitons by using a high voltage pulse pump to generate a series of pulses. The nonlinearity of the periodical lumped materials in the circuit will sharpen the rising and falling edges of the pulse. At the same time, the dispersive and nonlinear nature of the nonlinear transmission will break the relative long input pulse into an array of short duration solitons as the pulse is propagating through the transmission line.

The fundamental reason nonlinear materials allow the creation of solitons is that the wave velocity is a function of the electric and magnetic field strength. Specifically, the wave velocity in a transmission line is equal to $v = \frac{1}{\sqrt{\mu\varepsilon}}$, where ε is the permittivity and μ the permeability of the material. Both the permeability and permittivity decrease with increasing field strengths. Therefore the portion of the wave having a relatively large field strength propagates rapidly and the lower field strength portion at a reduced velocity thereby sharpening broad pulses into shorter duration pulses that usually having a larger peak value.

This phenomenon can be observed by analyzing a simple NLTL. The NLTL can be modeled as a typical L-C ladder network [1] where the series inductors are connected to ground by a shunt capacitor. Both the capacitors and inductors are made from nonlinear materials. The nonlinear capacitors are simply reverse biased diodes and the capacitance will decrease with increasing voltage, so that the crest of the pulse has a larger voltage value and will travel faster than lower voltage portions. In this way, the peak of the pulse will overtake the rising edge and a very short duration shockwave is generated. The same type of pulse compression is observed in nonlinear inductors. The inductance values will decrease as current increases causing the peak of the current pulse to travel faster thereby sharpening the wave.

Selection of the nonlinear materials for each of the respective components for pulse sharpening is critical in the design of NLTL’s. Typically reverse biased diodes are used for the nonlinear capacitors and nonlinear magnetic materials receive the most consideration for nonlinear inductors.

1.3. Magnetic Materials

Many types of magnetic materials have been characterized and considered for their ability to sharpen a propagating voltage pulse [1, 2]. The most promising material to date was discovered by BAE and is a patented highly resistive ferrite.

Generally, most nonlinear magnetic materials used for NLTL's or shocklines operate near the Curie temperature. They behave as a ferromagnetic material below the Curie temperature and a paramagnetic material above the Curie temperature. Since operation in the ferromagnetic region increases losses due to the magnetic hysteresis, NLTL's are normally operated above the Curie temperature, where the ferromagnetic material becomes paramagnetic. Eddy current losses are also of concern and minimized by laminating the magnetic material [1] and adding insulation layers.

A majority of nonlinear magnetic materials used today are a type of soft ferrites which are typically characterized at low frequencies (f ~ 10 KHz). They have a wide range of permeability; saturation flux and resistivity values and are relatively inexpensive [1]. This is a major problem when someone attempts to use Commercially-Off-The-Shelf (COTS) ferrites in NLTL design. It is therefore important that the constitutive parameters are measured over the frequency range of interest.

1.4. New Techniques

1.4.1. Hybrid NLTL

A majority of the NLTL's have been designed with invariant capacitance and inductance values along the line that are based on two separate sets of inductors and capacitors. These lines have the disadvantage of the characteristic impedance having discrete variations along the line. This can be avoided by integrating all materials in a single housing structure. A transmission line of this type is termed a hybrid line. Research by Smith investigated a type of hybrid line created by filling a planar conducting line with alternating ceramic and ferrite sections [3]. The nonlinear dielectric and ferrite materials tend to rapidly sharpen the pulse and create solitons. This type of structure has impedance that is invariant along the line length but operates near the relaxation time of the nonlinear material dielectric.

1.4.2. Gradually Scaled NLTL

A major problem that arises when a relatively wide voltage pulse is compressed into sharper and narrower pulse is that it tends to degenerate into multiple pulses. A technique that has been shown to reduce the likelihood of this occurring is to gradually scale the lumped parameter inductance and capacitance values along the direction of signal propagation. This technique has been shown to sharpen both the leading and trailing edges of the pulse [4]. Generally, it is difficult to reduce the pulse width when the parameters remain invariant in the NTLT. What may happen is that the input voltage pulse, having a relative long time duration, will be broken into a pulse sequence of shorter duration solitons. The problem of a wide pulse degenerating into multiple pulses rather than a single pulse is solved by using a gradually scaled NLTL. By gradual changing these parameters it is possible to achieve a series of very narrow pulses without pulse degeneration.

1.4.3. Optimal Design of NLTL

All the aforementioned NLTL designs were created by designers using a common sense approach based on trial and error. The goal of this research is to optimally design a NLTL using a genetic algorithm (GA). GAs are a type of global optimization that converge to a global minimum in a weakly probabilistic sense. They have been used in the design of many highly complicated systems with a remarkable degree of success.

In the research presented here, the controlling GA was coded in MATLAB and interfaced with the software LTspice to exchange information and run LTspice. The fitness function required that the circuit maximize the peak output pulse power. Generally this type of design is quite complicated and the circuit structure and types of materials used could span a large range of values. Therefore care must be taken when selecting materials knowing that they'll operate on the cusp of the ferrite B-H and dielectric P-E nonlinear saturation curve.

The GA algorithm emulates the natural evolutionary process by creating chromosomes, mutation rates, population sizes, etc. that are related to fundamental electrical parameters of the circuit. The parameter values in the circuit gradually evolve to the best solution of the fitness function without being limited by local minimum trapping. Each of the variable components' values (capacitors and inductors) are converted to a floating point number or "string" and used as the chromosomes of each child. The total number of chromosomes determines the genetic sequence of each child and the total number of children is the population size. Each child is simulated in LTspice and the results evaluated and ranked by the fitness function. The next step is the defining phase of the GA that allows a new generation to be created by crossover and mutation and the subsequent solutions ranked and the process repeated.

1.5. Benefit of the Method

Since the NLTL has been widely employed in many different areas including radar signal compression and high power microwave generation, the requirements for the NLTL will vary a great deal. After determining the system's operating parameters, including the generator frequency and nonlinear component characteristics, the task of optimizing of the NLTL to achieve a maximum output pulse power can be done using the integrated optimization technique described earlier. This will circumvent the usual trial and error approach that relies on human insight into the problem and directly solves for the best set of parameter values based on the system constraints.

2. Theory

2.1. The NLTL Implementation in LTspice

2.1.1. The Basic Circuit for NLTL

The basic building blocks of the NLTL are L-C sections comprised of nonlinear inductors and reverse biased diodes acting as nonlinear capacitors. The L-C sections act as a low-pass filter below frequencies specified by the Bragg frequency ($f_B = \frac{1}{\pi\sqrt{LC}}$) [5]. Figure 2.2 shows the equivalent lumped parameter model of the NTLT, where C_d and R_d represent the nonlinear capacitance and parasitic resistance of the reverse biased diode and L and C represent the nonlinear inductance and linear capacitance of the nonlinear transmission line respectively.

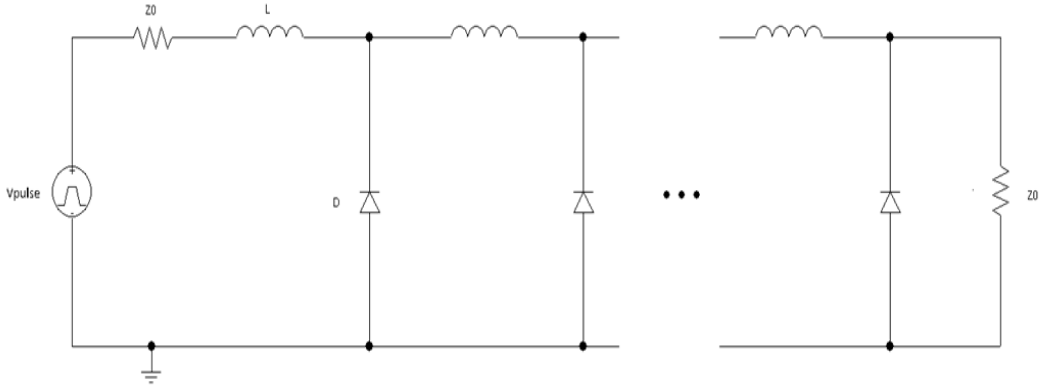


Figure 2.1: Basic NLTL circuit

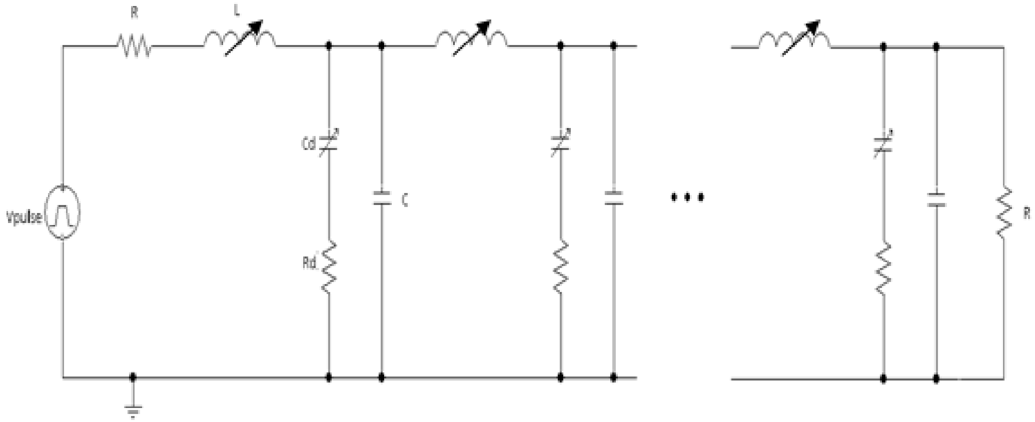


Figure 2.2: Equivalent circuit of NLTL

2.1.2. Nonlinear Inductor

Since the permeability for any magnetic material is realistically a function of both the magnetic field intensity and, to a lesser extent, the operational frequency, an empirical model must approximate the B-H curve of realistic ferrites in the frequency domain of interest. Also, the inductor's L-I curve must be directly proportional to the ferrite's B-H curve and since the magnetic field will be proportional to I. The inductor's L-I curve was designed to be representative of many commonly used nonlinear magnetic materials and based on the measurements taken at a frequency of 77 kHz shown below in Figure 2.3. The mathematically model used to approximate this behavior is shown in equation 2.1 and the corresponding curve in Figure 2.4. Several different nonlinear models have been presented in the research but were not used due to computational difficulties. The parameter values for $L_{\max}, L_0$ and ***mag*** were found from a least squares curve fit of the measured data and modeled in LTspice. It should be noted that this model may easily be adapted to describe other nonlinear inductor behavior.

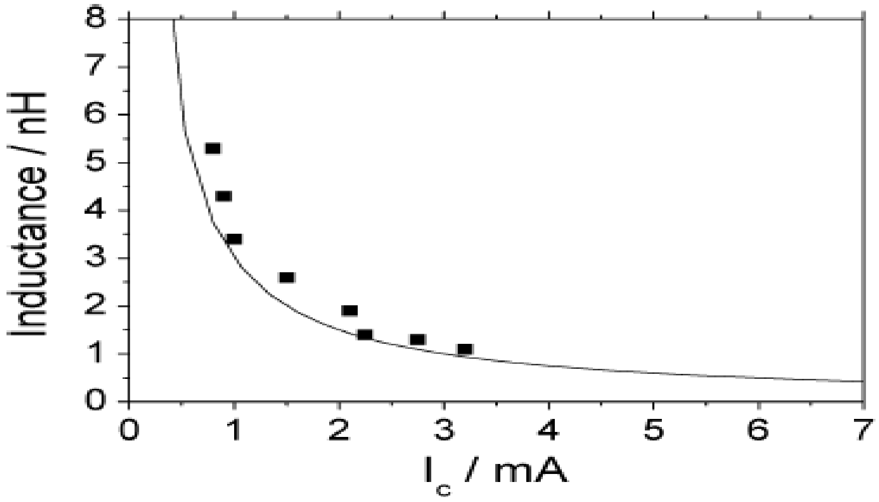


Figure 2.3 Inductance versus current curve

The nonlinear behavior of the inductor is modeled in this research using hyperbolic tangent function as shown below

$$\mathrm{L}=\mathrm{L}_{\max}\left[1-\tanh\left(\frac{abs(I)-I_{sat}}{mag}\right)\right]+L_0 \tag{2.1}$$

Equation 2.1 is plotted in Fig 2.4 with the parameters specified as $L_{\max}=7\ nH, I_{sat}=1\ mA, L_0=1\ nH, mag=1e-3$.

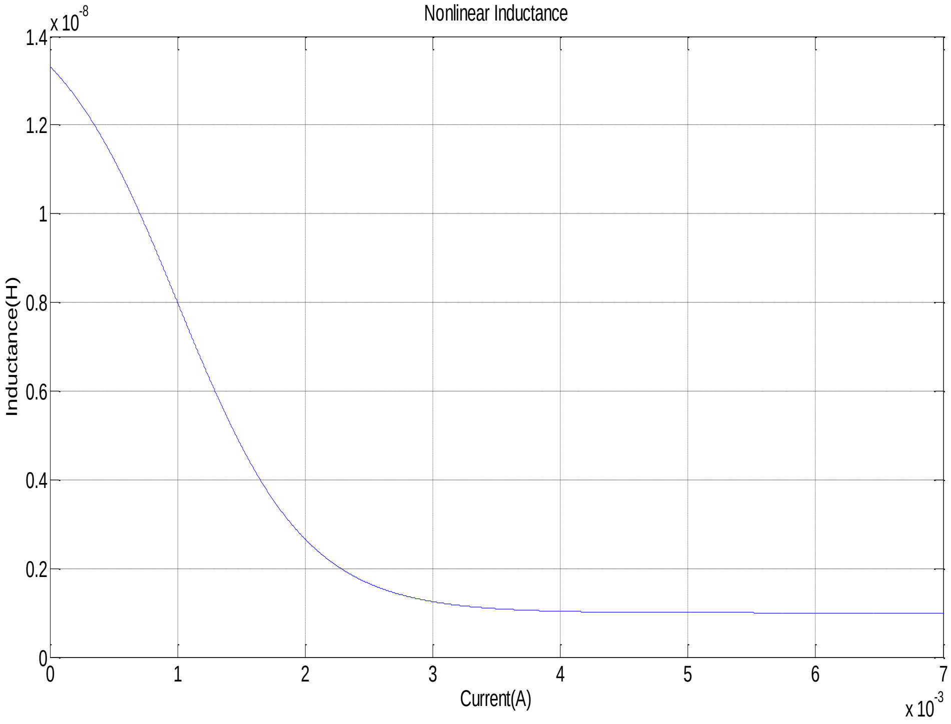


Figure 2.4: The inductance versus current curve for the nonlinear inductor

The voltage across the non-linear inductor is given by equation 2.2 and $\frac{dL}{dI}$ given in equations 2.2 and 2.3 respectively.

$$v = L\frac{dI}{dt} + \frac{dL}{dI}\frac{dI}{dt} = (L + \frac{dL}{dI})\frac{dI}{dt} \tag{2.2}$$

$$\frac{dL}{dI} = -\frac{\mathrm{L}_{\max}}{mag}\frac{I}{abs(I)}\left[1 - \tanh^2\left(\frac{abs(I) - I_{sat}}{mag}\right)\right] \tag{2.3}$$

By inserting equation 2.1 and 2.3 into equation 2.2 the final form of the equation used in LTspice is shown in equation 2.4.

$$v = \left\{ \mathrm{L}_{\max}\left[1-\tanh\left(\frac{abs(I)-I_{sat}}{mag}\right)\right] + L_0 - \frac{\mathrm{L}_{\max}}{mag}\frac{I}{abs(I)}\left[1-\tanh^2\left(\frac{abs(I)-I_{sat}}{mag}\right)\right]\right\}\frac{dI}{dt} \tag{2.4}$$

The nonlinear inductor behavior is described first using a subcircuit in LTspice and the netlist shown below.

```
.subckt nonind n2 n3

B1 n1 n3 V=(({Lmax}*(1-tanh((abs(I(B1)-{Isat})/{mag})))+{L0}-
{Lmax}*(1-pow(tanh((abs(I(B1)-
{Isat})/{mag})),2))*(I(B1)/({mag}*abs(I(B1)))))*ddt(I(B1)))

R2 n2 n1 .0001

.param Lmax=300n Isat=2.5 L0=1n mag=3

.ends
```

The netlist calls the subcircuit definition according to the requirement of LTspice. The subcircuit parameter Lmax defines the maximum value of the nonlinear inductance, Isat the current where the magnetic material begins to saturate, and ***mag*** is the variance. The resistance accounts for the magnetic material’s parasitic losses.

2.1.3. Nonlinear Capacitance

The value of the NLTL capacitance selected must have a relatively large breakdown voltage and operate with nonlinear behavior for range of voltages developed in the line. After some experimentation, a diode was selected that had a reverse breakdown voltage of several hundred volts and labeled MMSD4148 [6]. Figure 2.5 depicts the capacitance versus reverse voltage curve at the lower end of the reverse voltage range.

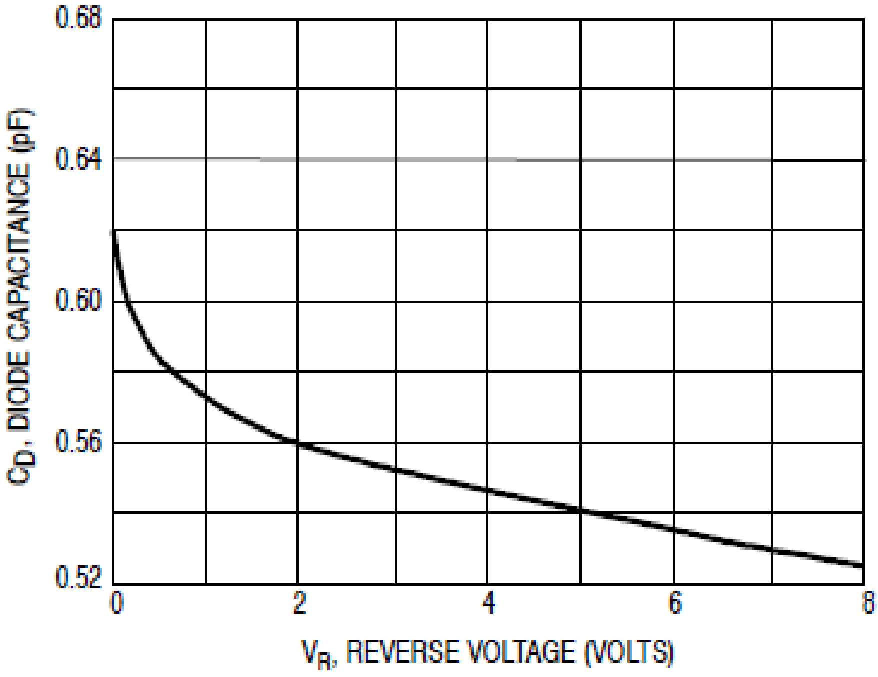


Figure 2.5: Diode capacitance versus reverse voltage curve

2.1.4. Simulation Time Step

For transmission line analysis, the time step is required to be equal to or smaller than the period associated with the Bragg frequency [2], which is defined in equation 2.5. Since the value of the per unit length capacitance and inductance vary, an effective capacitance and inductance must first be calculated according to equations 2.6-7.

$$f_B = \frac{1}{\pi\sqrt{LC}} \tag{2.5}$$

$$C_{D\_ave} = \frac{1}{V_{\max} - V_{\min}} \int_{V_{\min}}^{V_{\max}} C_D(v)dv \tag{2.6}$$

$$L_{ave} = \frac{1}{I_{\max} - I_{\min}} \int_{I_{\min}}^{I_{\max}} L(i)di \tag{2.7}$$

The Bragg cutoff frequency for the components used is $2.9300 \times 10^{8}\,\text{Hz}$ and associated time step $\square t = \frac{1}{f_B} = 3.4130\text{e} \times 10^{-9}$ seconds.

2.1.5. Voltage Source

Another parameter that determines the NLTL design is the source voltage magnitude and frequency. The voltage level was arbitrarily set to values of ±50 volts and at a pulse period of 15 ns. These parameters are typical for pulsed power sources and the entire pulse described below [3,4].

Voltage:	±50 V
Rise time:	1.5ns
Fall time:	1.5ns
Pulse width:	5ns
Pulse period:	15ns

2.1.6. Matched Load

The load impedance and source resistance was set to 50 ohms and the peak power enhancement that occurs at the load is calculated according equation 2.8. The denominator in this equation is equal to average power supplied to the source without the NLTL present and attaches a figure of merit to the pulse power enhancement.

$$PowerRatio = \frac{V_0{}^2(t)/R_L}{V_{in}{}^2/2R_S}$$

where:

$V_0(t)$ = instantaneous output voltage

V_{in} = peak input voltage (2.8)

2.2. Function of NLTL

2.2.1. Pulse Narrowing

The specific nonlinear and dispersive qualities of the NLTL are the key to the pulse sharpening that occurs. The NLTL allows velocity of the frequency components of the pulse to be proportional to pulse voltage and current. This is due to the decrease in capacitance and inductance with increasing pulse voltage and current as shown in equation 2.9 below.

$$
\begin{aligned}
&v_p = \frac{1}{\sqrt{L(I)C(V)}} \\
&\text{where:} \\
&v_p = \text{phase velocity} \\
&L(I) = \text{ inductance per unit length} \\
&C(V) = \text{ capacitance per unit length}
\end{aligned}
\qquad (2.9)
$$

Thus the leading edge of the voltage pulse will travel slower than the crest of the pulse voltage and pulse magnitude will increase and width of the pulse decrease. Also, dispersive effects of the NLTL will be effectively canceled out if the NLTL is designed properly. Ideally, if the NLTL is designed or “tuned” correctly, controllable solitons and quaisi-soliton pulses will be produced as shown in Figure 2.6 and the pulse will gradually sharpen as it propagates down the line.

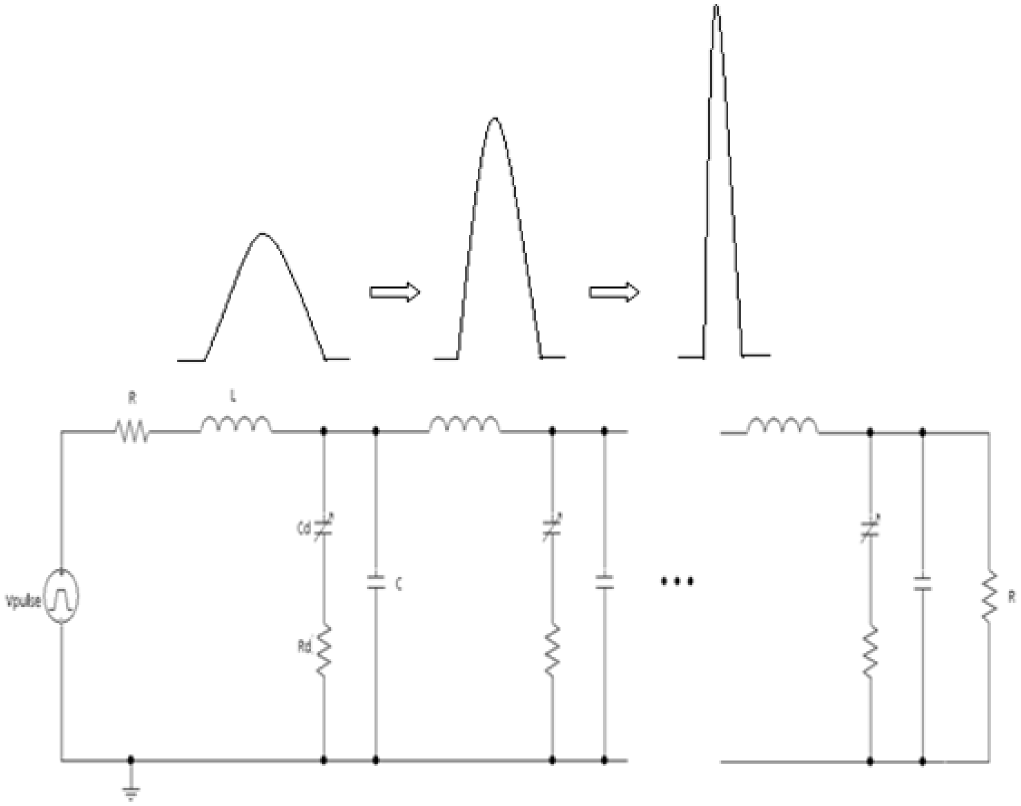


Figure 2.6: Improvement of the pulse through the NLTL

2.3. GA

2.3.1. Basic Theory and Implementation

Genetic algorithms (GAs) are one of the most popular global optimization methods used today for solving the complex optimization problems. They are based on the biological sciences and have been used very effectively on many “noisy” or highly nonlinear problems not well suited for gradient based solvers (Quasi-Newton or conjugant gradient) [7]. Since NLTL are inherently nonlinear, the GA is ideal for optimizing peak pulse output power for any combination of material properties and input voltage characteristics [7].

GAs are relatively easy to understand since they attempt to replicate the Darwnian concepts of natural selection and evolution. The parameters solved for in the research are encoded as floating point numbers and become the chromosome of each child of a single generation represented as a [1×18] vector. Each of the children represents a specific set of capacitance and inductance values for a NLTL and each child is ranked based on its performance by evaluating its strength using a fitness function. The fitness function determines a numerical figure indicative of the relative ability of each child to obtain a desired solution. After the children are ranked, the elite children are copied into the next generation and the remaining population is used in a roulette wheel selection process for crossover and mutation. The crossover process is done by taking a portion of the chromosomes from two children and combining them to form two new children. The rate that this takes place is called the crossover rate. Mutation also takes place where a certain number of children have a small percentage of their chromosomes changed in a random way. The crossover and mutation process is used to create the next generation while holding the total population of each generation fixed. This process will create the new generation and will continue until either a certain value of the fitness function is attained or a specified number of generations completed.

When designing a GA to solve any problem there are a number of initial parameters that must be set. The first parameter is an initial guess at the starting values for all unknowns. In our case we begin by initializing the values of the linear capacitance and inductance by assigning a random value within a specified range to all linear capacitive and nonlinear inductive elements. The total time duration and period of the initial pulse voltage also needs to be determined. This is somewhat arbitrary and was found to be highly dependent on the number of sections used.

a. Chromosome encoding

As mentioned earlier, either binary coding or floating point coding could be used for the chromosomes. It was found that using floating point numbers allowed a more rapid convergence to a solution and thus used. Also, converting the parameter values to binary numbers and then back to floating point numbers would add an additional processing step.

b. Population size

The population size needs to be determined before proceeding. There is however a tradeoff between the population size and rate of new generation creation. A larger population will likely yield a larger number of solutions with relatively “good” fitness values but will slow the overall creation rate of the new generation. If the population is too small many “good” solutions may be overlooked. Therefore, it is recommended that the population be of medium size suitable for the specific NLTL problem. In this work the population size was set to 200.

c. Selection scheme

All the children have their relative fitness evaluated and ranked. The best 12.5% of the children (largest fitness value) will be copied directly into the next generation to protect good chromosomes (solutions). A Roulette Wheel selection process is then used on the remaining 87.5% population to determine the children that will be used to create the remainder of the next generation. The probability of being selected for crossover is calculated using equation 2.10:

$$P_i = \frac{F_i}{\sum_{i=1}^{n} F_i}$$

where
P_i=possibilty of being selected for i_{th} child
F_i=normalized fitness value of i_{th} child
n=number of children (2.10)

The value calculated by equation 2.10 is assigned to each chromosome and then selection takes place in the following manner. A uniformly distributed random number between 0 and 1 is generated and designated *a*. Then selection (*selected_child*) takes place according to the following pseudo code:

```
$for\ i = 1 : n$
    $a = rand;$
    $p = 0; j = 1;$
    $while\ p < a$
        $p = p + P_j;$
        $j = j + 1;$
    $end$                                                    (2.11)
$child\_selected(i) = child\_previous\ generation(j-1);$
$end$
```

where P_j is the selection probabilty of the j_{th}child
n is the number of the children (175)
a is the random number generated for selection process

The greater the value of the fitness function the greater the possibility of being selected for crossover.

d. Mutation and crossover

The last 5% of the children selected experience only mutation and are placed in the new generation. Mutation takes place based by adding a small random number as follows:

child mutated = original child's chromosome + random value

and the random value given by equation 2.12.

$$\begin{array}{l} random\ value = erfinv[(2 \times rand) \text{-} 1] \\ \text{where:} \\ rand \text{ is a random number between 0 and 1} \\ erfinv \text{ is the inverse error function} \end{array} \qquad (2.12)$$

The function *erfinv* creates random values that will be added to each entry of the chromosome. These values are limited to the range of each component as shown below:

```
$for\ \ i = 1:n$
     $if\ \ child_i < lower\_bound$
             $child_i = lower\_bound;$
     $elseif\ \ child_i > upper\_bound$
             $child_i = upper\_bound;$                    (2.13)
     $end$
     $end$
$end$
where
$child_i$ is the value of the $i_{th}$ child in mutation
$n$ is the number of the children in the mutation
```

The remaining 82.5% of selected children are divided into two equally sized subgroups. Each of the parents of the first subgroup experience crossover with members of the second subgroup and create two new children according to equation 2.14. The technique used for the crossover process uses the first and last parents of each group to create the first two new children, then the second and second to last parents are used to create the next two children and so on. The entire process is repeated until the new generation is complete.

$$\begin{aligned} &child1 = rand \times parent1 + (1 - rand) \times parent2 \\ &child2 = rand \times parent2 + (1 - rand) \times parent1 \\ &\text{where rand is a random number between 0 and 1} \end{aligned} \tag{2.14}$$

e. Fitness Evaluation

The fitness value assigned to each chromosome was set equal to the peak output voltage generated by the chromosome. This condition enforces the desire to achieve a peak pulse output power since the output resistance was fixed. If a certain minimum pulse width is required the fitness function could be modified to include the any width speciation.

2.3.2. GA Flowchart

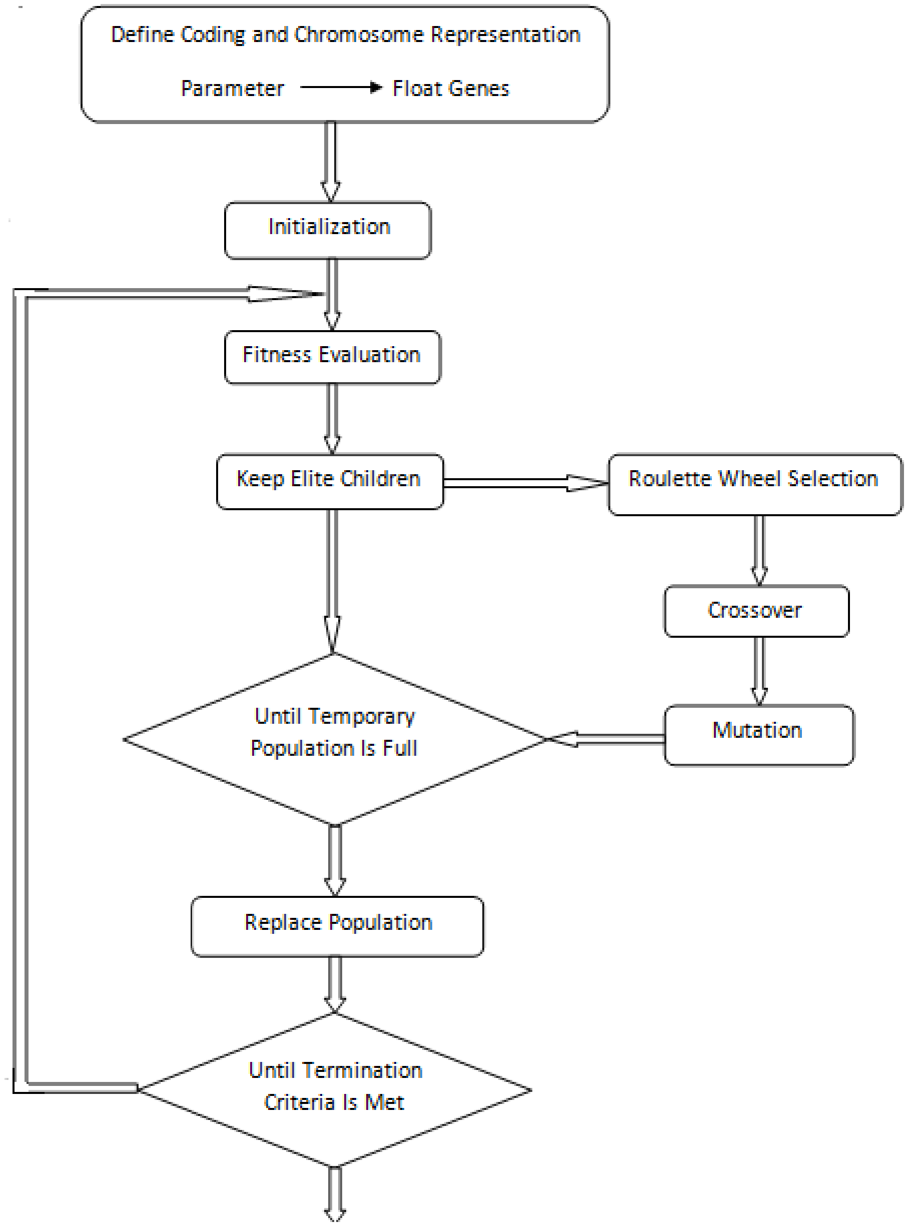


Figure 2.7: Flowchart of the GA used in this thesis

The flowchart shown in Figure 2.7 indicates the order that each operation occurs in the GA. The first step is the creation of initial values for the inductances and capacitances using a random number generator in MATLAB. These numbers are then used to create a netlist in LTspice and the circuit's output voltage calculated. This is followed by determining the fitness function each child. The top 12.5% of the chromosomes will be directly copied to the next generation. The remaining children will be selected using a roulette wheel selection process and then used for crossover and mutation processes. The crossover and mutation processes are repeated until the new population is complete. This process of selection, crossover and mutation will be repeated until the termination criteria is met.

2.3.3. Parameters of the GA

The goal of this research is to optimize the output voltage of a NLTL using a GA and thus provide proof of concept. To limit the amount of computer usage, only six L-C sections will be cascaded together for the design. This could be extended to any number of sections depending on the specific design requirements. The parameters selected for use in the GA include population and generation size, crossover and mutation rates. All these parameters were adjusted as necessary to increase the algorithm's performance. The specific NLTL design undertaken and has the parameters listed below:

a. 18 component values are optimized
b. Selection scheme: roulette wheel
c. Population size:200
d. Mutation rate:0.05
e. Number of generations:170

2.4. Involved Software

After choosing the GA as the optimization algorithm for NLTL, it's important to decide what software is most appropriate to realize the optimization process. The software must be efficient and able to provide accurate nonlinear circuit simulations. LTspice was chosen since it is both efficient and contains an extensive library of standard electronic components in addition to being free to use. The GA and the bi-directional interface to LTspice was written in Matlab. The main function of the interface is to forward the LTspice simulation data to Matlab for later processing and transfer the updated circuit netlist to LTspice without the need for any mouse and keyboard operations.

2.5. Interface Implementation

2.5.1. Function of the Interface

The interface between Matlab and LTspice is critical to proper operation of the algorithm and must be the compatible with both Matlab and LTspice. Since this is a nonlinear optimization problem, there is the possibility of non realizable or artifact solutions occurring. Therefore, appropriate measures must be incorporated into the interface to prevent this from occurring. The interface should also be efficient to minimize the iteration time of the optimization process. Finally, the interface needs to allow information to flow bi-directionally between Matlab and LTspice in a closed loop so that the optimization will be continue automatically without any further commands. The interface was written in Matlab and performed the following tasks:

a. Create netlist for LTspice with component values using Matlab

b. Access LTspice using Matlab and simulate circuit behavior

c. Read output voltage from LTspice and calculate fitness function

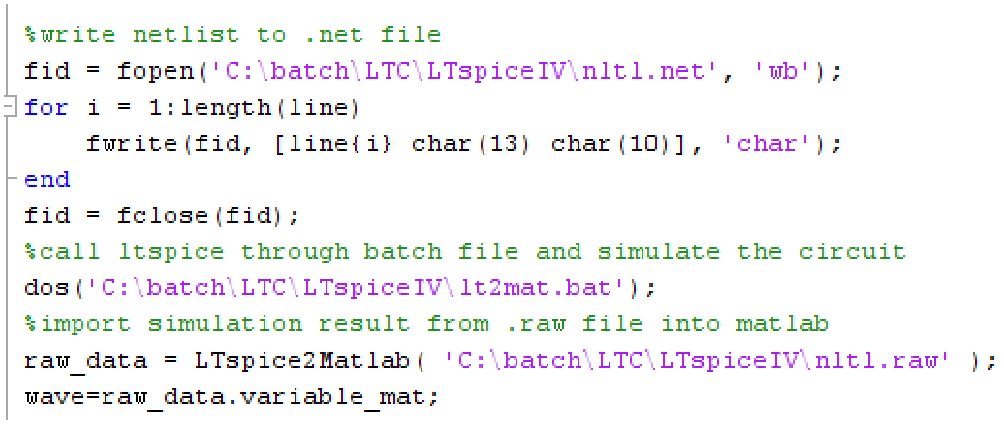
d. Use GA to calculate new component values and repeat from step a.

2.5.2. LTspice Batch Command

The biggest problem experienced when creating the interface was finding a method to automatically start the simulation in LTspice from Matlab. It's not possible to directly access the GUI interface radio button on the LTspice control panel using Matlab. However, every time a new combination of components' values is obtained from the GA, LTspice must be rerun. Fortunately LTspice provides batch command for circuit simulation. This is done using the batch command " –b". This command will run and leave the results in a file named ".raw". It's important that the windows batch file be written and saved in the same directory as LTspice to allow easy access by Matlab. The batch command used is shown below [8]:

```
C:\batch\LTC\LTspiceIV\scad3.exe-b C:\batch\LTC\LTspiceIV\nltl.net
```

The first half of the command specifies the location of the executable file, command "-b" for automatic simulation and data storing, and the second half refers to the location of the netlist file of the NLTL that is optimized. Also, the batch command should be saved as a windows executable batch file. The commands used to access the interface are given below that will allow the batch job to be executed in Matlab.



```
%write netlist to .net file
fid = fopen('C:\batch\LTC\LTspiceIV\nltl.net', 'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) char(10)], 'char');
end
fid = fclose(fid);
%call ltspice through batch file and simulate the circuit
dos('C:\batch\LTC\LTspiceIV\lt2mat.bat');
%import simulation result from .raw file into matlab
raw_data = LTspice2Matlab( 'C:\batch\LTC\LTspiceIV\nltl.raw' );
wave=raw_data.variable_mat;
```

Figure 2.8: Key words in Matlab codes building the interface

2.5.3. LTspice to Matlab

The output of LTspice in the ".raw" file is imported into a Matlab as a data structure that contains the variable components' name, values and the time axis vector. By processing these results the peak voltage is found and set equal to the fitness value for the specific parameters simulated. Figure 2.9 is a plot of the output voltage versus time for an example LTspice simulation.

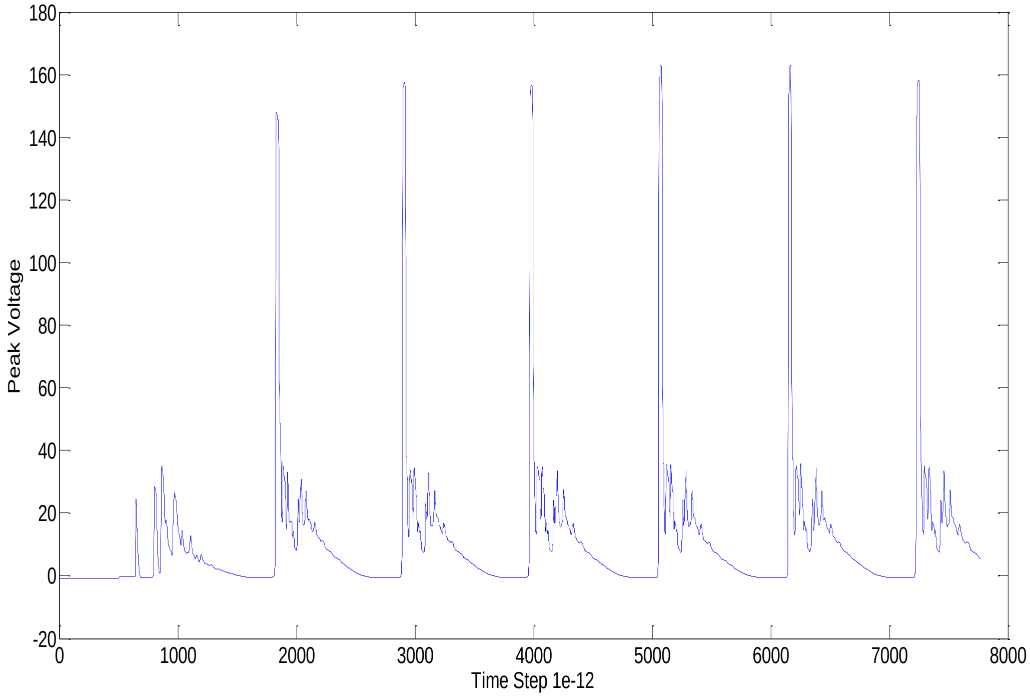


Figure 2.9: Data imported into Matlab

2.5.4. Matlab to LTspice

a. Netlist creation

Each new generation created by the GA in MATLAB generates a new netlist file that must be evaluated by LTspice. The following set of Matlab commands is used for the process.

i. fid = fopen(‘filename’)

fopen will open a file and requires the filename argument to be in single quotation marks

ii. count = fwrite (fid, A)

After obtaining the identification of the target file, fwrite is used to write the matrix of input information into the file.

iii. Status = fclose(fid)

The fclose command will close the file specified by the argument and return the status after closing.

b. Sub-circuit and main circuit netlist

It is important that the correct data format is used by Matlab to create the netlist used by LTspice or an error will occur. The Matlab executable file used to generate the subcircuit netlist for the nonlinear inductors is shown below.

```
line{01} = ['.subckt nonind n2 n3'];
line{02} = ['B1 n1 n3 V=(({Lmax}*(1-tanh((abs(I(B1)-
{Isat})/{mag})))+{L0}-{Lmax}*(1-pow(tanh((abs(I(B1)-
{Isat})/{mag})),2))*(I(B1)/({mag}*abs(I(B1)))))*ddt(I(B1)))'];
line{03} = ['R2 n2 n1 .0001'];
line{04} = ['.param Lmax=' num2str(x(1)) 'n Isat=2.5 L0='
num2str(x(7)) 'n mag=3'];
line{05} = ['.ends'];
fid = fopen('C:\batch\LTC\LTspiceIV\nonind.lib', 'wb');
for i = 1:length(line)
   fwrite(fid, [line{i} char(13) char(10)], 'char');
end
fid = fclose(fid);
```

There are six sections in the NLTL simulated and the main program netlist is given below.

Main circuit for NLTL

```
line{01} = ['* C:\batch\LTC\LTspiceIV\nltl.asc'];
line{02} = ['XU1 N004 N003 nonind'];
line{03} = ['V1 N001 0 PULSE(-50 50 0 1.5n 1.5n 5n 13n)'];
line{04} = ['R1 N002 N001 50'];
line{05} = ['D1 0 N004 MMSD4148'];
line{06} = ['R2 N014 0 50'];
line{07} = ['C1 N004 0 ' num2str(x(13)) 'f'];
line{08} = ['D2 0 N006 MMSD4148'];
line{09} = ['C2 N006 0 ' num2str(x(14)) 'f'];
line{10} = ['D3 0 N008 MMSD4148'];
line{11} = ['C3 N008 0 ' num2str(x(15)) 'f'];
line{12} = ['D4 0 N010 MMSD4148'];
line{13} = ['C4 N010 0 ' num2str(x(16)) 'f'];
line{14} = ['XU3 N008 N007 nonind3'];
line{15} = ['XU2 N006 N005 nonind2'];
line{16} = ['XU4 N010 N009 nonind4'];
line{17} = ['R3 N004 N005 100µ'];
line{18} = ['R4 N007 N006 100µ'];
line{19} = ['R5 N003 N002 100µ'];
line{20} = ['R6 N009 N008 100µ'];
```

```
line{21} = ['D5 0 N012 MMSD4148'];

line{22} = ['C5 N012 0 ' num2str(x(17)) 'f'];

line{23} = ['D6 0 N014 MMSD4148'];

line{24} = ['C6 N014 0 ' num2str(x(18)) 'f'];

line{25} = ['R7 N011 N010 10µ'];

line{26} = ['R8 N013 N012 10µ'];

line{27} = ['XU5 N012 N011 nonind5'];

line{28} = ['XU6 N014 N013 nonind6'];

line{29} = ['.model D D'];

line{30} = ['.lib C:\batch\LTC\LTspiceIV\lib\cmp\standard.dio'];

line{31} = ['.tran 0 100n 0 1e-11'];

line{32} = ['.lib C:\batch\LTC\LTspiceIV\nonind4.lib'];

line{33} = ['.lib C:\batch\LTC\LTspiceIV\nonind3.lib'];

line{34} = ['.lib C:\batch\LTC\LTspiceIV\nonind2.lib'];

line{35} = ['.lib C:\batch\LTC\LTspiceIV\nonind.lib'];

line{36} = ['.lib C:\batch\LTC\LTspiceIV\nonind6.lib'];

line{37} = ['.lib C:\batch\LTC\LTspiceIV\nonind5.lib'];

line{38} = ['.backanno'];

line{39} = ['.end'];

fid = fopen('C:\batch\LTC\LTspiceIV\nltl.net', 'wb');

for i = 1:length(line)

    fwrite(fid, [line{i} char(13) char(10)], 'char');

end
```

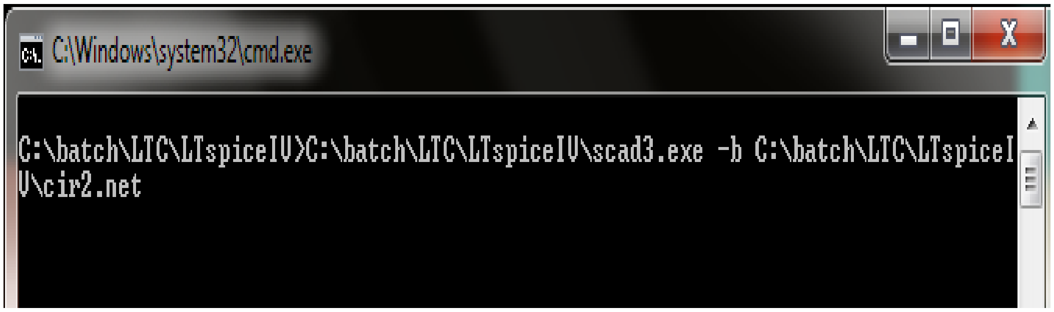
fid = fclose(fid);

The list includes six nonlinear inductors each having separate L_{max} and L_0 values and a six capacitance that are optimized. Each of the nodes for the nonlinear inductor must be defined six times in the netlist (a simple call statement is insufficient). The vector x[n] called above represents the component values for the inductance and capacitance in the netlist file created by Matlab and the num2str command is used to convert the floating point value into a string which is used by Matlab.

c. Call batch file to simulate the circuit

After creating the netlist described above, the batch file is executed in LTspice from Matlab by calling the DOS shell script and running it in Microsoft Windows. The Matlab command used for this is shown below along with the corresponding DOS batch file [8].

dos('C:\batch\LTC\LTspiceIV\lt2mat.bat');



C:\Windows\system32\cmd.exe

```
C:\batch\LTC\LTspiceIV>C:\batch\LTC\LTspiceIV\scad3.exe -b C:\batch\LTC\LTspiceI
V\cir2.net
```

Figure 2.10: Screen shoot on the dos batch file

Every time this command is called the batch file will be displayed in the Matlab command window and a completion status will be returned.

d. Obtain the peak voltage

The process of obtaining the maximum output voltage for a given ".raw" file was done by sorting the time dependant voltage data in descending order and keeping the maximum value.

e. Artificial situation handling

Since the inductance and capacitance used are nonlinear, LTspice sometimes fails to converge and gives non-realizable results which cause an error to occur when Matlab attempts to accesses the peak voltage. A Matlab command was added to the interface to prevent this error from occurring. When an artificially large voltage occurs, the unreasonable result is ignored by Matlab and the component values causing this behavior returned to their pre-defined settings that would result in a relatively low output voltage occurring. This will allow the combination of the components' values to be treated as a child with low fitness value and will not affect the optimization process negatively.

2.5.5. Packet the Interface

The entire bi-directional interface between Matlab and LTspice was written as a single Matlab function called ***lt2mat*** that was called by the GA. The inputs of the function are the component values obtained from the GA that are used to create the LTspice netlist file. The return value of this function is the peak voltage of the single circuit simulation which will be treated as fitness evaluation in the genetic algorithm.

2.5.6. GA and interface operation

The entire optimization scheme needs to operate efficiently and the specific working of all the operators is described below. Following the creation of a new generation, each child's fitness must be evaluated. This is done by calling the interface described previously where the child's chromosome was used to create the netlist. LTspice will then be executed and the resulting peak output voltage of the circuit returned to the GA using the interface previously described. The peak value of the output voltage will be assigned as the child's fitness value.

2.5.7. Computation complexity and iteration time

The population size, cross over rate, and mutation rates all affect the performance of the algorithm. Following a thorough literature review that provided many helpful suggestions, a trial and error process was used to determine the parameter combination that would provide the best overall algorithm performance. The parameters selected provided a sufficient convergence rate while still yielding optimal results in real-time. It should be noted all Matlab ***for*** loops were vectorized to reduce runtime when possible.

Besides adjusting parameters used in the Matlab GA, a minimal number of sections used for the simulation needed to be set. The number of sections was required to be sufficient to allow true transmission line behavior to occur. It could not be made too large or the simulations would not be completed in a reasonable time. This is somewhat dependant on the computer resources available and after some experimentation the final number of sections was set to six and the results found in a reasonable time (approximately 1 day to complete a single simulation).

2.5.8. Optimization System

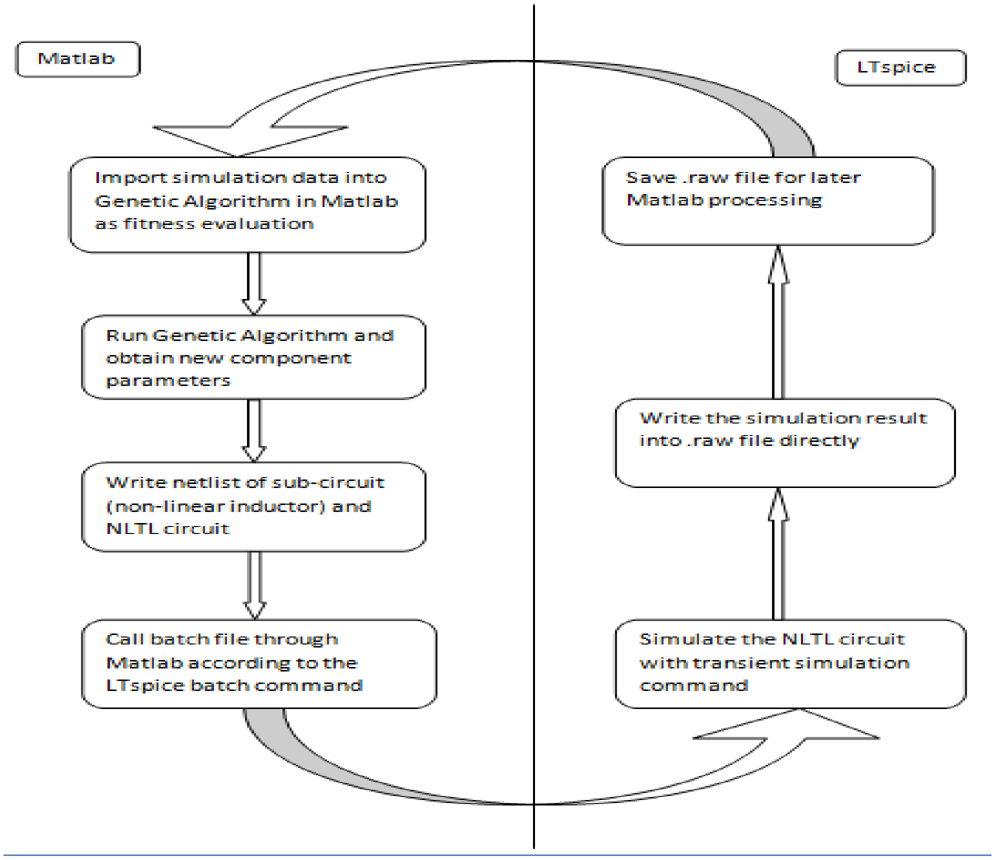

Matlab
LTspice
Import simulation data into Genetic Algorithm in Matlab as fitness evaluation
Run Genetic Algorithm and obtain new component parameters
Write netlist of sub-circuit (non-linear inductor) and NLTL circuit
Call batch file through Matlab according to the LTspice batch command
Simulate the NLTL circuit with transient simulation command
Write the simulation result into .raw file directly
Save .raw file for later Matlab processing

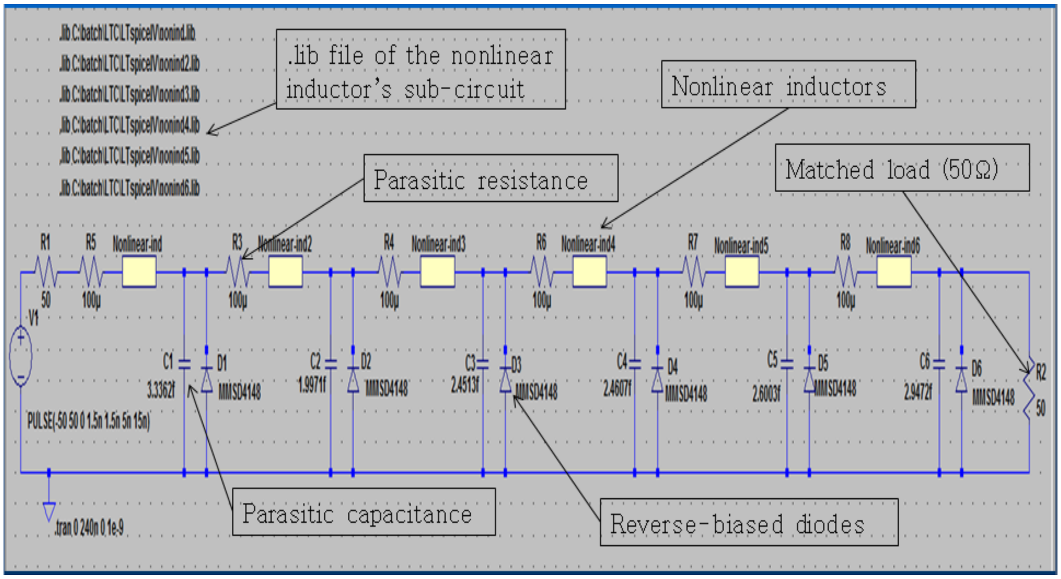
Figure 2.11: Flowchart of the optimization system

Figure 2.11 contains a flowchart of the entire optimization process. The process can basically be divided into two parts: The calculations done in Matlab and the simulations done by LTspice. Matlab is responsible for all the GA calculations and the data interface while LTspice is used to generate the transient output voltage waveform. To summarize, a single iteration starting at the LTspice output file is described: the ".raw" file is imported by Matlab and the peak output voltage is obtained. This value is treated as the chromosome's fitness value and a new combination of all the component values is obtained in the GA and written into the netlist file. Finally, the batch file is called (run) again, circuit simulated in LTspice and results saved in the ".raw" file.

3. Results

3.1. Simulation

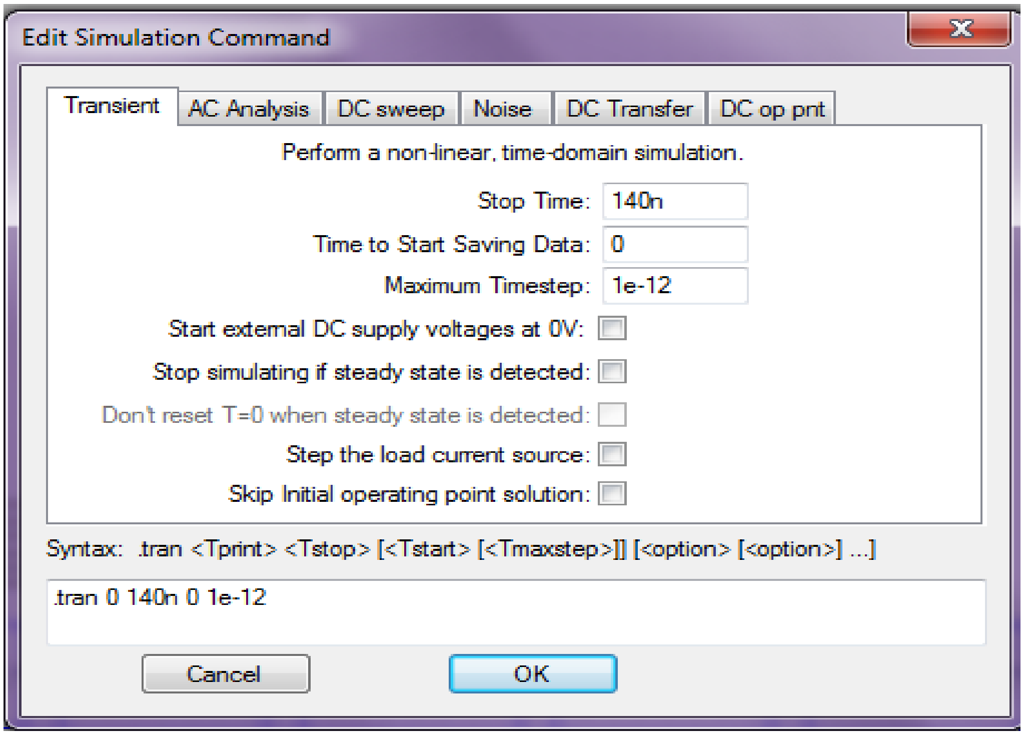
The NLTL circuit design contains six L-C sections. In each section, the nonlinear inductor is defined using a subcircuit that mathematically defines its nonlinear behavior and the nonlinear capacitance is realized using a linear capacitor in parallel with reverse-biased diode. The pulse voltage source was based on typical parameters found in the literature [9] and the load and source impedance arbitrarily set to 50 Ω. Figure 3.1 illustrates the structure of the final NLTL circuit optimized by the simulations. Figure 3.1 is a screen capture image of the actual circuit as it appears in the LTspice Graphical User Interface (GUI). The "lib." files define each nonlinear inductor and need to be included as part of the netlist that describes the main circuit.



```
.lib C:\batch\LTC\LTspiceIV\nonind.lib
.lib C:\batch\LTC\LTspiceIV\nonind2.lib
.lib C:\batch\LTC\LTspiceIV\nonind3.lib
.lib C:\batch\LTC\LTspiceIV\nonind4.lib
.lib C:\batch\LTC\LTspiceIV\nonind5.lib
.lib C:\batch\LTC\LTspiceIV\nonind6.lib
PULSE(-50 50 0 1.5n 1.5n 5n 15n)
.tran 0 240n 0 1e-9
```

Figure 3.1: Schematic of the main NLTL circuit

3.1.1. Simulation Settings



Perform a non-linear, time-domain simulation.

Stop Time: 140n

Time to Start Saving Data: 0

Maximum Timestep: 1e-12

Start external DC supply voltages at 0V: □

Stop simulating if steady state is detected: □

Don't reset T=0 when steady state is detected: □

Step the load current source: □

Skip Initial operating point solution: □

Syntax: .tran <Tprint> <Tstop> [<Tstart> [<Tmaxstep>]] [<option> [<option>] ...]

```
.tran 0 140n 0 1e-12
```

Figure 3.2: LTspice simulation command setting

Since the NLTL circuit contains many nonlinear components (nonlinear inductors and capacitors), the equivalent set of nonlinear differential equations must be solved using a relatively small time step to achieve an accurate solution. The initial time step was set to equal to the inverse of the Bragg frequency and steadily reduced until the resulting simulations were insensitive to further time step reductions. The final time step was equal to 1 ps. The LTspice simulation duration was set to 140 ns since output voltage waveform became approximately steady state.

3.1.2. Components optimized

When the simulations were completed the values for the nonlinear inductor and capacitors were determined. These include six values of $L_{\max}$ and L_0 and six capacitance values.

3.1.3. Initial conditions

The initial values for $L_{\max}$ were determined using the random number generator in Matlab that ranged from 10 to 500 nH. All capacitor values were initial based on the inductance range and the source voltage period and the estimated range was found using the formula shown in equation 3.1

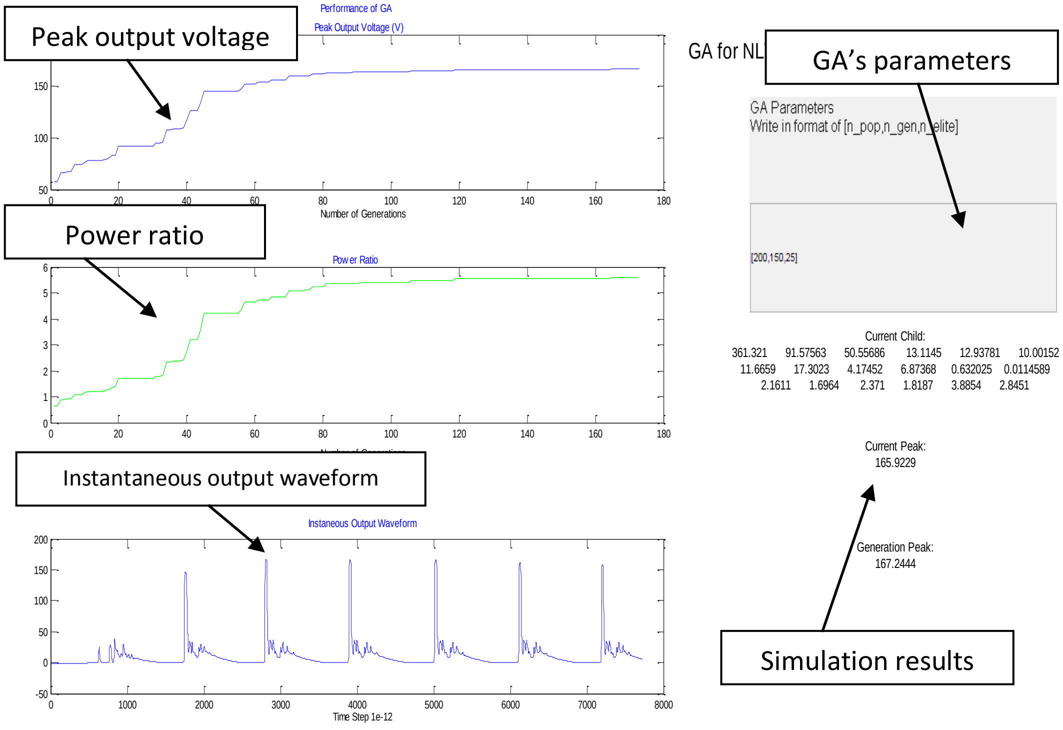
$$C = \frac{T^2}{(2\pi)^2 L} \tag{3.1}$$

The value of the total capacitance of each section is ~ 6.04 pF and the average value of the nonlinear inductors is ~ 268.23 nH . Since the parasitic capacitance is in parallel with the diodes, the estimated value of the parasitic capacitance is ~ 1.64 pF .

The initial value of the capacitance was arbitrarily set using a random number generator in Matlab that ranged from 1 fF to 600 fF and the final values calculated in the optimization.

3.1.4. Matlab GUI

Because it was impossible to predict the behavior of the GA, the process was monitored and the parameters adjusted if the rate of convergence was too slow. A GUI was developed in MATLAB to monitor the maximum voltage output and power ratio versus number of generations and output voltage waveform. It also allowed the population size, number of generations, and number of elite children to be set in the GUI as shown in Figure 3.3.



GA Parameters
Write in format of [n_pop,n_gen,n_elite]

[200,150,25]

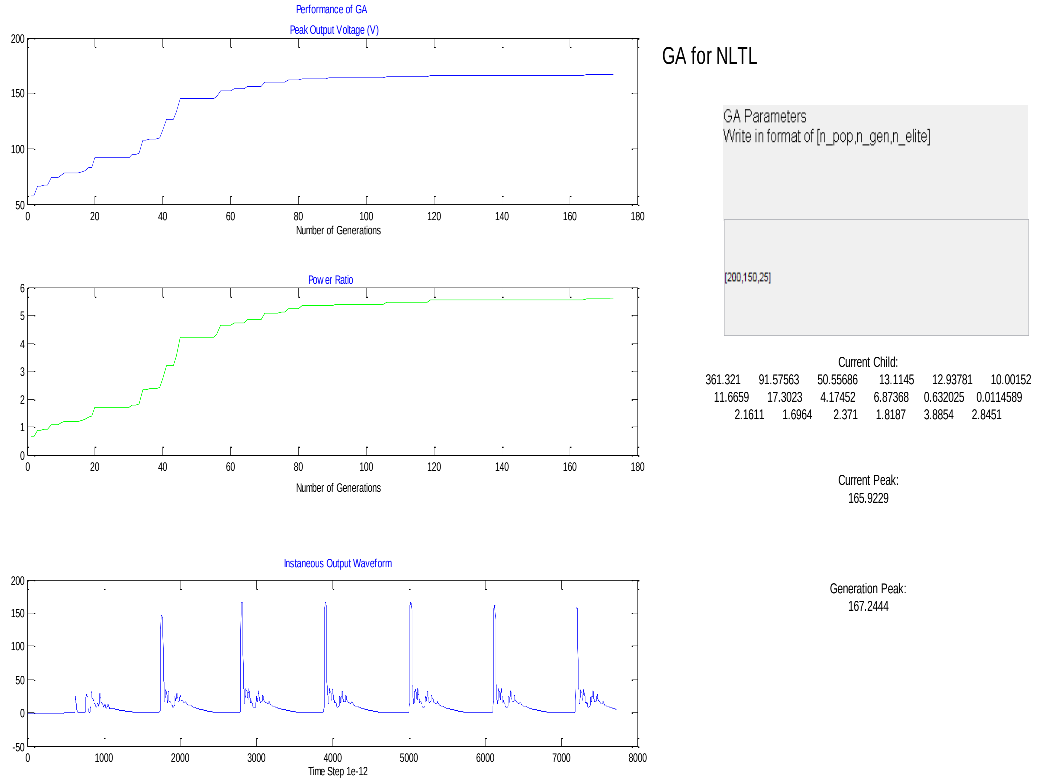
```
Current Child:
361.321    91.57563    50.55686    13.1145    12.93781    10.00152
11.6659    17.3023    4.17452    6.87368    0.632025    0.0114589
2.1611    1.6964    2.371    1.8187    3.8854    2.8451

Current Peak:
165.9229

Generation Peak:
167.2444
```

Figure 3.3: Layout of the Matlab GUI

If the process does not proceed at a reasonable rate the optimization process was terminated and the parameters modified.

3.2. Simulation Results

The interface was specifically designed to be transferable across Windows platforms providing that MATLAB and LTspice are available. In this research, the entire simulation process was done on a HP Pavilion dv7-4070 us laptop with a Windows 7 operation system for testing purpose.



GA for NLTL

GA Parameters
Write in format of [n_pop,n_gen,n_elite]

[200,150,25]

```
Current Child:
361.321    91.57563    50.55686    13.1145    12.93781    10.00152
11.6659    17.3023    4.17452    6.87368    0.632025    0.0114589
2.1611    1.6964    2.371    1.8187    3.8854    2.8451

Current Peak:
165.9229

Generation Peak:
167.2444
```

Figure 3.4: Simulation results showed in Matlab GUI

The simulation resulted in the peak voltage being increased from ± 50 V to 165.9229 V and the width of the pulse being narrowed appreciable to ~ 0.17 ns compared to the input pulse width of 8 ns. The power ratio, which indicates the degree of peak pulse power enhancement, was increased to a value of 5.7.

The final component values will appear in the Matlab GUI after the termination criteria has been met. It is obvious that the number of generations could be increased to attain an even greater output voltage. However, the goal of this research was to show the feasibility of using a GA to optimally design a NLTL. The final parameter values found are shown below. (The $L_{\max}$ and L_0 values of each nonlinear inductor and capacitance value of each parasitic capacitance are given)

The maximum values for the nonlinear inductors were found to be

$$L_{\max\_1} = 361.321\,nH \quad L_{\max\_2} = 91.576\,nH \quad L_{\max\_3} = 50.557\,nH$$
$$L_{\max\_4} = 13.115\,nH \quad L_{\max\_5} = 12.938\,nH \quad L_{\max\_6} = 10.002\,nH$$

and L_0 values for the nonlinear inductors are

$$L_{0\_1} = 11.666\,nH \quad L_{0\_2} = 17.302\,nH \quad L_{0\_3} = 4.175\,nH$$
$$L_{0\_4} = 6.874\,nH \quad L_{0\_5} = 0.632\,nH \quad L_{0\_6} = 0.012\,nH$$

The component values for the parasitic capacitors are

$$C_1 = 2.161\,fF \quad C_2 = 1.696\,fF \quad C_3 = 2.371\,fF$$
$$C_4 = 1.819\,fF \quad C_5 = 3.885\,fF \quad C_6 = 2.845\,fF$$

The resulting output voltage waveform is shown below

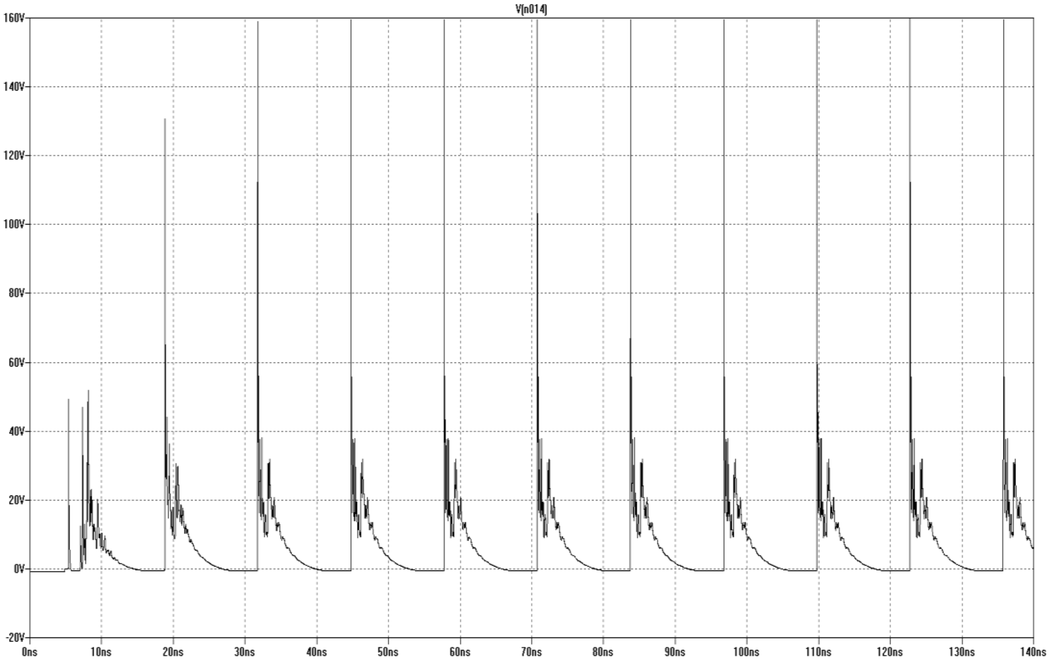


Figure 3.5: Confirmation of LTspice

3.3. Method Confirmation

The component values of the nonlinear inductors obtained from the optimization gradually decrease from section to section. This was also reported by Afshari [4] as a method of enhancing the NLTL performance. It is also noteworthy that by gradually scaling, the NLTL soliton degeneration will be minimized [4].

4. Conclusions

In this research, a novel integrated optimization method was presented that maximized the peak output voltage and power of a six section NLTL. A GA was chosen for the optimization method and the software code written in Matlab which created component values that were then used to simulate the circuit's behavior in LTspice. The results show that the peak output voltage steadily increased for each new generation proving that code was functioning correctly.

A special emphasis was placed on trying to replicate NLTLs currently used. A standard NLTL circuit was modeled using a L-C ladder network. Each of the nonlinear inductor's L-I curves was created by scaling the permeability curve of realistic magnetic material to the current range of the inductor. A typical ± 50 V pulsed voltage with 50 Ω of internal resistance was used to simulate a realistic input pulse source.

The final optimization results showed that the width of input impulse was narrowed and the peak voltage had been increased. The power ratio indicated that the peak output peak power had been enhanced by 5.7 times the maximum possible peak power that would have occurred without a NLTL. Also, a somewhat gradually scaling of the nonlinear inductor's value from the source to load end was observed. The actual degree of scaling had similarities to the gradually scaled NTLT design currently used for enhancing the NLTL's performance.

It is also noteworthy to mention that the interface between Matlab and LTspice was reliable and worked very efficiently. Finally, the complicated problem of designing NTLT was simulated in a 24 hour time frame and clearly showed steady improvement. If a much faster computer was available many more sections could be added and the optimization allowed to run until the maximum output voltage settled to a constant value.

4.1. Suggestions for Future Research

The ultimate goal of this thesis is to demonstrate the feasibility of employing a modified GA in the optimization of the NLTL's peak pulse power. If a more complicated NLTL with known adjustable nonlinear inductor and capacitor values was optimized, it would be reasonable to expect that significantly better performance could be achieved using an extension of this method. Additionally, further research on improvements to the optimization speed should be considered. For instance, the entire system of nonlinear equations could be written directly in either FORTRAN or C and solved thus eliminating the need for an interface to the circuit simulation software. Also, parallel computing is another feasible method to speed up the optimization. A very large population could be used in the GA and shared by several computers. Parallel processing has been used to achieve remarkable speed improvements in many optimization processes.

References

[1]. J.Gaudet, E. Schamiloglu, J.O. Rossi, C.J. Buchenauer, and C. Frost, “Nonlinear transmission lines for high power microwave applications- a survey,” *IEEE International Power Modulators and High Voltage Conference, Proceedings of the 2008*, pp.131-138, 27-31 May 2008.

[2]. W. Sullivan III, J.Dickens and M. Kristiansen, “Shock wave simulation of ferrite-filled coaxial nonlinear transmission lines,” *IEEE International Power Modulators and High Voltage Conference, Proceedings of the 2008*, pp.517-520, 27-31 May 2008.

[3]. P.W. Smith, Transient Electronics – Pulsed Circuit Technology, West Sussex, England: John Wiley & Sons, 2002, pp. 249-255.

[4]. Afshari, E.; Hajimiri, A.; , "Non-linear transmission lines for pulse shaping in silicon," *Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE 2003* , vol., no., pp. 91- 94, 21-24 Sept. 2003

[5]. M.J.W. Rodwell, M. Kamegawa, R.Y. Yu, M.Case, E. Carman and K.S. Giboney, “GaAs nonlinear transmission lines for picosecond pulse generation and millimeter-wave sampling,” *Microwave Theory and Techniques, IEEE Transactions,* vol.39, pp.11994-1204, Jul 1991.

[6]. J. Rossi and P. Rizzo, “Study of hybrid nonlinear transmission lines for high power RF generation,” in *Pulsed Power Conference, 2009. PPC'09. IEEE*, pp. 46-50, 2009.

[7]. Y. Rahmat-Samii and E. Michielssen, Electromagnetic Optimization Genetic Algorithms, John Wiley & Sons, Inc., 1999.

[8]. LTspice help file.

[9]. Ming Li; Harrison, R.G.; Amaya, R.E.; Duchamp, J.-M.; Ferrari, P.; Tarr, N.G.; , "CMOS varactors in NLTL pulse-compression applications," *Microwave Conference, 2007. European* , vol., no., pp.1405-1408, 9-12 Oct. 2007

[10]. J.M. Duchamp, P. Ferrari, M.Fernandez, A.Jrad, X. Melique, J.W.Tao, S. Arscott, D.Lippens and R.G. Harrison, “Comparison of fully distributed and

periodically loaded nonlinear transmission lines," *Microwave Conference, 2001. 31st European* , pp.1-4, 24-26 Sept. 2001

[11]. N. Seddon, C.R. spiking and J.E. Dolan, "RF pulse formation in nonlinear transmission lines," *IEEE Pulsed Power Plasma Science Conference*, Albuquerque, NM, 18 June 2007, Institute of Electronics and Electrical Engineers (IEEE). IEEE, 2007.

[12]. U. Bhattacharya, S.T. Allen and M.J.W. Rodwell, "DC-725 GHz sampling circuits and subpicosecond nonlinear transmission lines using elevated coplanar waveguide," *Microwave and Guided Wave Letters, IEEE* , vol.5, no.2, pp.50-52, Feb 1995.

[13]. H. Ikezi, "Compression of a single electromagnetic pulse in a spatially modulated nonlinear dielectric," *Appl. Phys.* Vol. 64, pp. 3273-3276, 1998.

[14]. H. Ikezi, J.S. Degrassie and J. Drake, "Soliton generation at 10MW level in the very high frequency band," *Appl. Phys. Lett.*, Vol. 58, pp. 986-987, 1991.

[15]. P.M. Brown and P.W. Smith, "High power, pulsed soliton generation at radio and microwave frequenciese," *Proc. 11th International Pulsed Power Conference,* Baltimore, MD, pp. 346-354, 29 June 1997.

[16]. D.W. Van Der Weide, J.S Bostak, B.A. Auld and D.M. Bloom, "All-electronic free-space picosecond pulse generation," *Electron. Lett.*, Vol.27, p. 1412, 1991.

[17]. J.D. Darling and P.W. Smith, "High power RF generator from nonlinear delay lines," *Proc. 16th International Pulsed Power Conference,* Albuquerque, NM, pp. 472-475, June 2007.

[18]. M.J. Rodwell, S.T. Allen, R.Y. Yu, M.G. Case, U. Bhattacharya, M. Reddy, E. Carman, M. Kamegawa, Y. Konishi, J. Pusl and R. Pullel, "Active and nonlinear wave propagation in unltrafast electronic s and optoelectronics," *Proc. IEEE*, vol. 82, pp. 1037-1059, July 1994.

[19]. H. Shi, W.-M. Zhang, C.W. Domier, N.C. Luhman, Jr., L.B. Sjogren and H.-X.L. Liu, "Novel concepts for improved nonlinear transmission line performance*,"* *IEEE Trans. Microwave Theory Tech*., vol. 43, pp. 780-789, Apr. 1995.

[20]. M.J. Rodwell, "Picosecond electrical wavefront generation and picosecond optoelectronic instrument," Ph.D. dissertation, Stanford Univ. Stanford, CA, 1987.

[21]. A. Jrad, W. Thiel, P. Ferrari and J.W. Tao, "Comparison of SPICE and FDTD simulations for lossy and dispersive nonlinear transmission lines*," Electron. Lett.*, vol. 36, no. 9, pp. 797-798, May 2000.

[22]. M.F. Diego, "Non linear transmission line for frequency multiplier applications," Ph.D. dissertation, Dept. Electron., Univ. Lille, Lille, France, 2001.

[23]. J.E. Dolan and H.R.Bolton, "Shock front development in ferrite-loaded coaxial lines with axial bias,"*Proc. IEEE- Science, Measurement & Technology,* vol. 147, pp. 237-342, Sept, 2000.

[24]. J.E. Dolan and H.R.Bolton, "Length equation for ferrite-loaded high voltage pulse sharpening lines," *IEEE Electronics Letters,* vol. 34, issue 13, pp. 1299-1300, June 1998.

[25]. S. Furuya, H. Matsumoto, H. Fukuda, et al., "Simulation of nonlinear coaxial line using ferrite beads," *Jpn. J. Appl. Phys.*, vol. 41, pp. 6536-6540, Nov. 2002.

[26]. A.M. Belyantzev, A.I. Dubnev, S.I. Klimin, Yu.A. Kobelev and L.A. Ostrovskii, "Generation of radio pulses by an electromagnetic shock wave in a ferrite loaded transmission line," *Tech. Phys.* Vol 40(8), pp. 820-826, 1995.

[27]. D.W. Van der Weide, "Delta-doped Schottky diode nonlinear transmission lines for 480-fs 3.5 V transitions," *Appl. Phys. Lett.*, vol. 65, no. 7, Aug. 1994.

[28]. S.T. Allen, U. Bhaatacharya and M.J.W. Rodwell, "725 GHz sampling circuits integrated with nonlinear transmission lines," in *Proc. IEEE Device Res. Conf.,* Boulder, CO., June 1994.

[29]. T.W. Crowe, W.C.B. Peatman and R. Zimmermann, "Consideration of velocity saturation in the design of varactor diodes*," IEEE Micorwave and Guider Wave Lett.*, vol. 3, no. 6, pp. 161-163, June, 1993.

[30]. R.V. Khokhlov, "On the theory of shock radio waves in non-linear lines," *Radiotekh. Elektron.*, vol. 6, no. 6, pp. 917-925, 1961.

[31]. R.H. Freeman and A.E. Karbowiak, "An investigation of nonlinear transmission lines and shock waves*," J. Phys. D: Appl. Phys.*, vol. 10, pp. 633-643, 1977.

[32]. M.J.W. Rodwell, D.M. Bloom and B.A. Aul, "Nonlinear transmission-line for picosecond pulse compression and broadband phase modulation," Electron. Lett., vol. 23. p. 109, Jan. 29, 1987.

[33]. C.J. Madden, M.J.W. Rodwell, R.A. Marsland, Y.C. Pao and D.M. Bloom, “Generation of 3.5 ps fall time shock-waves on a monolithic GaAs nonlinear transmission line*,” IEEE Electron Device Lett*., vol. 9, pp. 303-305, June, 1998.

[34]. R. Landauer, “Shock waves in nonlinear transmission lines and their effect on parametric application,” *IBM J. Res. Develop*., vol. 4, no. 4, pp. 391-401, Oct. 1960

[35]. R. Landauer, “Phase transition waves: Solitons versus shock waves,” *J. Appl. Phys*., vol. 51, no.11, pp. 5594-5600, Nov. 1980.

[36]. Y. Rahmat-Samii and E. Michielssen, Electromagnetic Optimization Genetic Algorithms, John Wiley & Sons, Inc., 1999.

[37]. Zhou, T.G.; Fang, L.; Li, S.; Ji, L.; Song, F.B.; He, M.; Zhang, X.; Yan, S.L.; , "Nonlinear Inductance of Intrinsic Josephson Junction Arrays and Its Application to Tunable Filters," *Applied Superconductivity, IEEE Transactions on* , vol.17, no.2, pp.586-589, June 2007.

[38]. Ming Li; Harrison, R.G.; Amaya, R.E.; Duchamp, J.-M.; Ferrari, P.; Tarr, N.G.; “CMOS varactors in NLTL pulse-compression applications”