Enhanced Features for Design of Traveling Wave Tubes Using CHRISTINE-1D

John A. David[1], Carol L. Kory[2], Hien T. Tran[1], R. Lawrence Ives[2], and David Chernin[3]

ABSTRACT

Traveling wave tubes (TWT's) are vacuum devices invented in the early 1940's for amplification of radio frequency (RF) power. These devices are critical for radar, communications and electronic warfare missions in the military, as well as in commercial applications. The physics-based design and simulation code, CHRISTINE-1D, was used in the past to explore different TWT circuit designs and to automate the process of parameter estimation. However, the current capability of CHRISTINE-1D allows optimization of only helix TWT designs, and includes a limited number of optimization goal functions. In addition, the current optimizer in CHRISTINE-1D employs a modified steepest descent method to carry out the optimization process. The objectives of our work are three-fold: (i) to investigate optimization techniques that may be better suited for this problem (for example, simplex type methods such as Nelder-Mead and DIRECT), (ii) allow optimization of non-helix TWTs, (iii) and implement new optimization goal functions. Finally, to show the feasibility of our approach, we apply our optimization algorithms to the problem of designing a folded waveguide slow-wave circuit.

Keywords: Traveling Wave Tubes, Folded waveguide slow-wave circuit, Optimization, Goal functions, DIRECT, Nelder-Mead

[1] John A. David and Hien T. Tran are with the Department of Mathematics and Center for Research in Scientific Computation, Box 8205, North Carolina State University, Raleigh, North Carolina 27695-8205.
[2] Carol L. Kory (correspondence author, Carol@calcreek.com) and R. Lawrence Ives are with Calabazas Creek Research Inc., Saratoga, California 95070.
[3] David Chernin is with Science Applications International Corporation, McLean, Virginia 22102.

I. INTRODUCTION

Traveling Wave Tubes (TWTs) are vacuum devices invented in the early 1940's [11942 #2, 2] for amplification of radio frequency (RF) power. They are critical for radar, communications and electronic warfare missions of all Armed Services, as well as in certain commercial applications. Because of their high power, broad-bandwidth, compact size, and high efficiency, TWTs are used for satellite communications, airborne, ship borne, and ground-based radars, jamming, and decoy applications. Commercial applications include satellite communications, radar, and materials processing.

The TWT amplifies by converting the kinetic energy of an electron beam to the electromagnetic energy of an RF electromagnetic wave. The TWT amplifies a low power RF signal for transmission at significantly higher power. To maximize the energy transfer between the electron beam and the RF signal, the beam must flow in near synchronism with the RF wave, i.e., the electrons in the beam must travel at approximately the same velocity as the RF signal. Towards the output section of the TWT, the beam decelerates as energy is transferred to the RF wave. By decreasing the phase velocity of the RF signal as the beam decelerates, the RF wave will remain in synchronism for a longer distance, and efficiency will increase. This is typically achieved using a phase velocity taper, which varies the period (or other geometric parameter) of the slow-wave circuit in the output section of the TWT. In general, the phase velocity of the RF signal is proportional to the circuit period.

Phase velocity tapers are also used to improve TWT linearity, and thus reduce distortion, to enable high-data rate throughput [3] and [4]. For the specific application we describe here, several conditions must be met over a 3 GHz bandwidth centered at W-band (83.5 GHz). These are minimum small signal gain variation with frequency, minimum phase shift between small signal and saturated conditions, at least 10 W of output power, minimum variation in input power required for saturation at frequencies in the band, and maximum efficiency. There is a tradeoff between parameters, i.e., one must sacrifice phase shift to increase efficiency, for example. Thus, it is very useful for the designer to assign different relative weights to the various optimization goal functions, so that optimizations can focus on the most important parameters for the specific application.

Most current design tools require manual input from the designer. Based on the

simulation results, the designer makes judgements on parametric changes to more closely achieve the desired performance. These input modifications are made, and the simulation repeated. The process continues until the required performance is achieved or time and funding are exhausted. The primary cost is the intensive labor for the designer.

To avoid this tedious design procedure, optimization was incorporated into the physics-based helix TWT design code CHRISTINE-1D [4], as well as other TWT, large signal codes [5]. To maximize the electronic efficiency, CHRISTINE-1D implements a phase velocity taper by varying the phase velocity of the slow-wave circuit at user-specified locations along the circuit. When the slow-wave circuit is physically altered to vary the phase velocity (such as by varying the period), the interaction impedance and attenuation will also change. The current version of CHRISTINE-1D assumes that the TWT incorporates a helix slow-wave circuit. With a helix, the phase velocity varies directly with the period, or pitch. Thus, if one reduces the pitch to 95% of its standard value, one can expect the phase velocity to be (approximately) 95% of its original value. With the current optimizer, the impedance is varied using the well-known sheath or tape helix model. These models use analytical functions to calculate the impedance as a function of helix geometry. Thus, the impedance is automatically calculated for a given helix geometry.

The first task of our work was to explore different optimization routines for optimizing TWT circuit taper designs. CHRISTINE-1D uses a modified steepest descent calculation in which the direction of travel in parameter space alternates between the steepest descent path and the average direction of travel during the recent cycles of minimization. It is well known that this type of gradient-based method, in general, performs very well when the optimization landscape is relatively smooth, and when a good ‘first guess’ for the solution can be made. However, for many important applications, including high power RF devices, the discrete nature of variables embedded in these problems coupled with the nonlinear interaction physics give rise to nonsmoothness and nonconvexity in the landscapes, which can defeat gradient-based methods. We propose using sampling methods, Nelder-Mead and DIRECT, which do not require gradient information but rather sample the objective (goal) function on a stencil or pattern to determine the progress of the iteration and appropriately change the size, but not the shape, of the stencil. DIRECT is a global optimization routine with the ability to define a set of feasible design parameters. The routine will search this space for the global minimum. Since DIRECT is a global search method, it requires many iterations for high precision when the number of

parameters is large. See Appendix 1 for a further introduction to these optimization methods. In this paper, we explore using either DIRECT or Nelder-Mead algorithms, and DIRECT together with Nelder-Mead in a hybrid approach. That is, we initially apply the DIRECT method to generate a small set of iterates, and then use Nelder-Mead to explore the smaller parameter space, thus accelerating convergence. We previously demonstrated this hybrid approach for designing Brillouin-focused electron guns by geometrically altering cathode and control surfaces to achieve specified beam characteristics [6].

The second stage of our work was to apply the optimization methods with new goal functions to design a folded waveguide, slow-wave circuit. CHRISTINE-1D cannot model this geometry for optimization in its current form. More specifically, we cannot use the sheath or tape helix models to calculate variation of the impedance as a function of the phase velocity. Instead, it is computed using curve-fitting routines. In addition, since our design objectives for the folded waveguide slow-wave circuit are different than those built into CHRISTINE-1D, we propose several new goal functions.

Section II describes the application of sampling based optimization methods to a C-Band, helix TWT design problem, and their performance in comparison with the optimization algorithm in CHRISTINE-1D. Section III and IV contain our applications of the sampling methods for the optimal design of a folded waveguide, slow-wave circuit including the introduction of several new optimization goal functions. Finally, concluding remarks are given in Section V.

II. Design of a Linear, C-Band, Helix TWT

To compare these algorithms with the optimization routine in CHRISTINE-1D, we first optimized the same test case using CHRISTINE-1D and the Nelder-Mead algorithm. For our test case, we used the C-Band helix TWT design previously used by Abe et al. [4]. The helix circuit parameters were designed around an existing, well-characterized, electron gun. This gun, designed by Northrop Grumman using a methodology described in [14], produces a highly laminar beam with excellent transport properties, negligible helix interception at zero RF drive, and low harmonic growth. The nominal electron gun parameters are summarized in Table I. The vacuum barrel radius, the helix radius, the helix tape width, and the rectangular cross-section BeO support rods were held invariant with respect to the different optimized designs. In this

optimal design problem, only the helix pitch was modified according to a specified goal (or objective) function. The design was based on a two-stage circuit separated by a sever, located between $4.064 \le z \le 4.2164$ [cm]. Table II summarizes the common (invariant) parameters.

TABLE I

NOMINAL ELECTRON GUN PARAMETERS

Parameter	Nominal Value
Cathode voltage	-4.80 [kV]
Beam Current	146 [mA]
Min. Beam Radius	0.158 [cm]

TABLE II

HELIX PARAMETERS COMMON TO ALL OPTIMIZATIONS

Parameter	Value
Vacuum barrel radius	0.2794 [cm]
Helix radius	0.12446 [cm]
Helix tape width	0.03556 [cm]
Length of first stage	4.064 [cm]
Length of second stage	8.0776 [cm]

We wanted to optimize for peak power added efficiency, η_{eff}, which is defined as the difference in RF output power (P_{out}) and the RF input power (P_{in}) divided by the beam power. This is given by the following goal function,

$$\eta_{eff}(\lambda, P_{in}) = \frac{P_{out} - P_{in}}{I_b V_b}, \tag{1}$$

where λ represents the helix pitch taper, and I_b and V_b are the beam current and voltage, respectively. The optimal design problem is to vary the helix pitch as a function of distance to maximize the efficiency as defined by equation (1). Both optimization algorithms, CHRISTINE-

1D optimization and the Nelder-Mead algorithm, used the same initial guess for the helix pitch, which was a constant pitch of 0.1 [cm]/turn for all z values.

CHRISTINE-1D was executed using its native gradient based algorithm for 400 iterations, at which point all parameters were well converged, as shown in Fig. 1. This required approximately 110 minutes of CPU time, but as shown in Fig. 1, after about 200 iterations the change in the helix pitch is negligible. Running CHRISTINE-1D using the Nelder-Mead algorithm required 686 iterations and 1051 function evaluations, which took approximately 26 minutes of CPU time. Figure 1 shows the convergence of the helix pitch values for the Nelder-Mead algorithm as a function of function evaluations. Note that the convergence for CHRISTINE-1D is with respect to iteration count while the convergence for Nelder-Mead is with respect to function evaluation. Nelder-Mead requires far less CPU time per iteration as it only requires a small number of function evaluations per iteration. See Appendix 1 for specific information on how an iteration of Nelder-Mead proceeds. The gradient information used by CHRISTINE-1D at each iteration generally requires more computational time to obtain than an iteration of Nelder-Mead. However, in the case of smooth objective functions gradient based algorithms can be far more effective per iteration. This implementation of Nelder-Mead uses the convergence criteria of small simplex size and small difference between function values on the simplex and function values of the current minimizer. CHRISTINE-1D uses a combination of user specified iterations and small step sizes in the parameter space. However, as this is a design problem computational time is not nearly as important as the ability to find a more efficient design, as long as the computational time is not prohibitive.

CHRISTINE-1D's gradient optimization routine and the Nelder-Mead algorithm attained efficiencies of approximately 30% and 32% respectively.

Figure 2 plots the electronic efficiency as a function of input power for the CHRISTINE-1D and Nelder-Mead optimized cases. From this test it appears that given a decent initial iterate both optimization routines return similar results. Figure 3 compares the pitch profiles for both the CHRISTINE-1D and Nelder-Mead optimized cases.

Another issue one must consider when working with an optimization routine is its robustness, i.e., how close must the initial iterate be to the minimum for the routine to converge to it. In the previous example the initial iterate was close to optimum and both routines converged to a similar result. Now we would like to consider the case when the initial iterate is

further from the minimum. We perturbed the initial helix values by multiplying each of the fifteen helix pitch values by 0.3*ν where ν is normally distributed with mean 1 and standard deviation 1, and ran both of the optimization routines. This experiment was repeated six times. The CHRISTINE-1D algorithm was run for 400 iterations for each optimization, which was sufficient for convergence in each case. The final peak efficiency is listed in Table III. The average peak efficiency using the Nelder-Mead algorithm was 25.3% while the average peak efficiency for the CHRISTINE-1D algorithm is 23.8%. Thus, in conditions where a good initial iterate is available either routine may be successful. However, if finding a good initial iterate is difficult, then the Nelder-Mead algorithm may be a useful alternative. This may be particularly useful in the optimization of TWT multistage depressed collector stages where a good initial iterate is not always available.

TABLE III

COMPARISON OF NELDER-MEAD AND STEEPEST DESCENT FOR RANDOM INITIAL ITERATES

Optimization Routine	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Nelder-Mead	30.2%	26.0%	20.6%	20.0%	23.2%	31.9%
CHRISTINE-1D	30.0%	17.4%	25.3%	18.7%	21.7%	29.8%

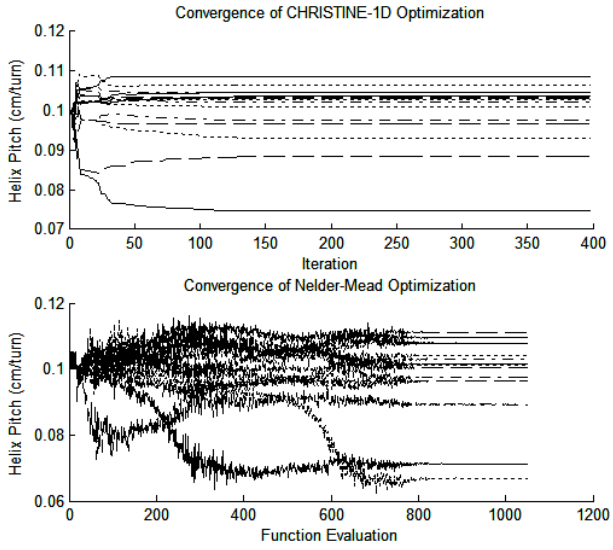

Fig. 1. (Top) Convergence of the helix pitch at fourteen axial grid locations from CHRISTINE-1D optimization as a function of iteration (each line represents pitch at separate axial location).

(Bottom) Convergence of the helix pitch at fourteen axial grid locations from the Nelder-Mead algorithm as a function of function evaluation (each line represents pitch at separate axial location).

III. Designing Folded Waveguide Slow-Wave Circuits

After performing this feasibility study on a known TWT design, we applied our optimization algorithms to the design of a folded waveguide, slow-wave circuit. This is a W-band, 9 kV, 28 mA TWT [15]. The customer provided specifications are based on the program application, and require optimization with custom goal functions as described in more detail below. CHRISTINE-1D was previously used to model a folded waveguide TWT, and the code gave good agreement with three-dimensional large signal analyses. This validated CHRISTINE-1D for modeling folded waveguide TWTs [16]. Since we are using a folded waveguide circuit, not a helix circuit, we cannot use the CHRISTINE-1D internal sheath or tape helix models to calculate the impedance variation with phase velocity. Alternatively, we used the three-dimensional electromagnetic code Microwave Studio [17] to calculate the phase velocity and impedance as a function of the folded waveguide, slow-wave, circuit geometry. Using these results, we varied the impedance with phase velocity in the CHRISTINE-1D code, using a polynomial curve fit of the form

$$fz(i) = \alpha + \beta\, fb(i) + \gamma\, fb(i)^2 + \delta\, fb(i)^3 + \varepsilon\, fb(i)^4, \tag{2}$$

where fz is the normalized impedance, and fb is the normalized phase velocity at the i^{th} axial location along the circuit. The CHRISTINE-1D code does not currently account for the change in attenuation due to a change in helix pitch in the built-in optimizer.

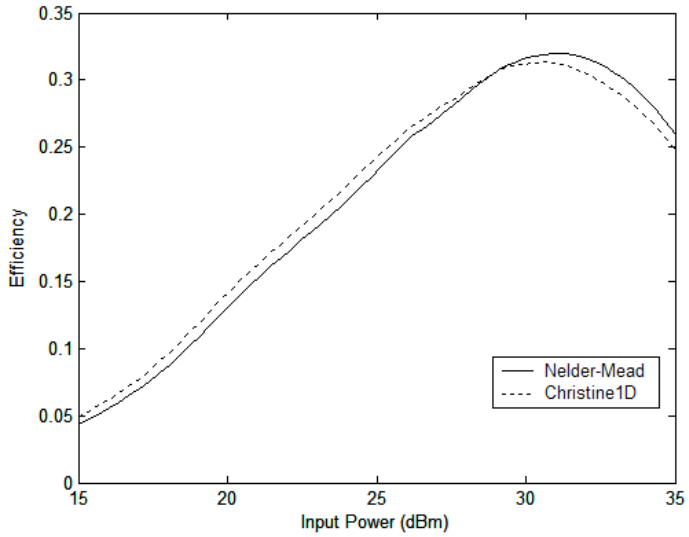

Fig. 2. Comparison of CHRISTINE-1D's native gradient-based optimization algorithm and the Nelder-Mead algorithm for maximizing the electronic efficiency.

The goal functions we use require us to compute the performance of the folded waveguide TWT over ranges of both input power and frequency. The range of input powers was -25 [dBm] to 15 [dBm] with 30 total input powers where the model was evaluated. The frequency range was 82 [GHz] to 85 [GHz] with 5 total frequencies (82, 82.75, 83.5, 84.25, 85) where the model was evaluated. The inputs for our optimization were 25 values for the phase velocity. Using these values of phase velocity, the impedance was computed using the polynomial curve fitting function (2) and written to a CHRISTINE-1D input file by a MATLAB (The MathWorks, Inc., Natick, Massachusetts) function. We note that because our optimization routines, Nelder-Mead and DIRECT, are written in MATLAB, the interface between CHRISTINE-1D and the optimization algorithms are handled through programs written in MATLAB. CHRISTINE-1D was then run in the scanning mode with the input file computed as described above. Next, the output files from CHRISTINE-1D were read by MATLAB to compute cost functions based on certain design specifications. We summarize this process in the flowchart depicted in Figure 4.

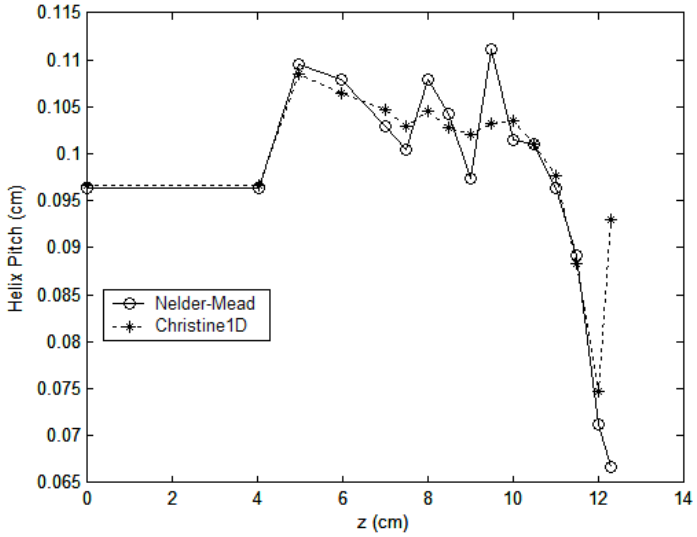

Fig. 3. Comparison of optimal helix pitches found by CHRISTINE-1D and Nelder-Mead.

The optimal design criteria are defined by goal functions. Our approach to treat the simulation and optimization codes as "black boxes" allows us to easily implement different goal functions. We initially wanted to minimize the variation of the small-signal gain magnitude (the ratio of the output power to the input power at -25 dBm) with frequency. Therefore, we defined the following goal function

$$\text{gaincost} = \sum_{i=\,frequencies \neq 83.5GHz} (G_{mag}(-25dBm, i) - G_{mag}(-25dBm, 83.5Ghz))^2, \tag{3}$$

where the gain magnitude G_{mag} is a function of the input power and frequency. Next, we wanted to keep the phase change small, so our second goal function was

$$\text{phasediff} = \sum_{k=\,frequencies} \sum_{j=inputpowers} (phase(j,k) - phase(-25dBm,k))^2, \tag{4}$$

where the phase is a function of input power and frequency. We also wanted to make sure we achieved at least 10 Watts of output power at saturation at each frequency. However, no bonus was given for achieving more than 10 Watts of output power at saturation. Therefore, the desired goal function is

$$\text{poutcost} = \sum_{P_{out,i}<10W} (10W - P_{out}(P_{in}(i))), \tag{5}$$

where the value used is the output power at saturation for the i^{th} frequency and is only added if the output power falls below 10W. Note that if 10W of output power is achieved at all frequencies, this cost will be zero.

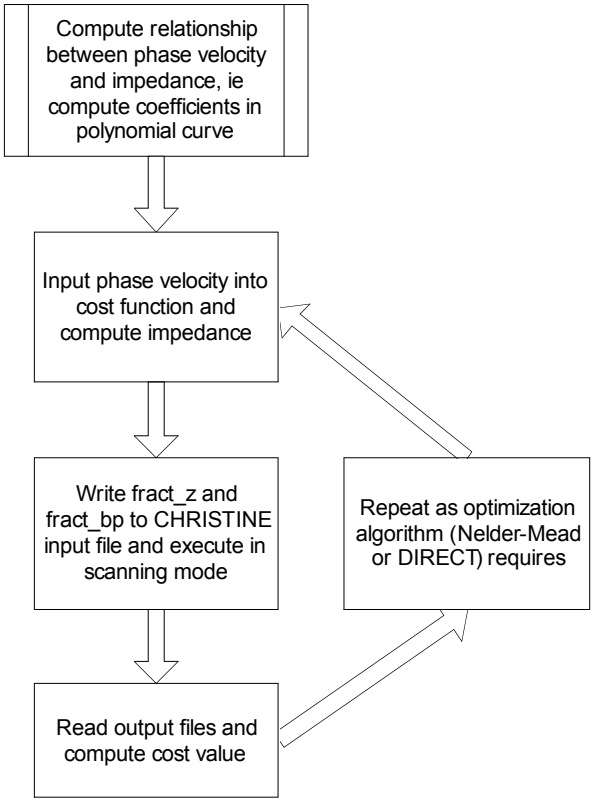

Fig. 4. Flowchart of folded waveguide optimization

Next, we wanted saturation to occur at the same input power for each frequency. Let $pout_j$ denote the vector containing the 30 output powers corresponding to the 30 input powers with the j^{th} frequency. We then define $I(j)$ to be the index in $pout_j$ where saturation occurs. The corresponding goal function for this design criterion is

$$\text{diffsat} = \sum_{k=frequencies \neq 83.5GHz} |I(k) - I(83.5GHz)|. \tag{6}$$

Finally, to maximize the efficiency, we define the following goal function

$$\text{effcost} = \frac{1}{\sum_{i=\,frequencies} \sum_{j=\,power} \eta_{eff}(j,i)}, \tag{7}$$

where the efficiency, η_{eff}, is a function of the input power and frequency. Note that instead of optimizing for peak efficiency we chose to optimize for a sum of the efficiencies over both power and frequency ranges. The idea behind choosing this goal function is that it looks for a design that is generally efficient across a wide range of frequencies and powers instead of only attempting to maximize the efficiency at power saturation, a power value at which the device is not generally operated. However, the sum over powers in (7) will be dominated by values near saturation. Each of the goal functions (3)-(7) were equally weighted to magnitude one when all phase velocities took their nominal, untapered values. The total combined goal function is a weighted sum of each of these goal functions as

$$\text{Cost} = a \times \text{gaincost} + b \times \text{phasediff} + c \times \text{poutcost} + d \times \text{diffsat} + e \times \text{effcost}, \tag{8}$$

where (a,b,c,d,e) are user specified constants weighting the relative significance of each goal function (3)-(7).

IV. Optimal Design Results

First, we chose various weightings of the goal function and executed DIRECT for 40 iterations. Table IV presents the most promising results using various combinations of weighting coefficients. If a goal function weighting is not listed, it is assumed to be zero. The goal function value is a percentage of the goal function when it is computed with a constant phase velocity. More specifically, if gaincost is 0.5 then the value of the gaincost goal function is half that of what it was for the non-optimized constant phase velocity. For each case, DIRECT was given a budget of 10000 function evaluations, though, to complete the 40 iterations, it required approximately 7000 function evaluations.

We found that if we applied a substantial weighting (*e*) for the effcost, no weighting for poutcost was required. In other words, appropriately weighting the efficiency goal function achieved 10 W of output power over the operating frequency band. Also, note that the 0 value

for poutcost indicates we achieved the minimum power specification for each of the optimizations.

TABLE IV

RESULTS OF DIRECT OPTIMIZATION FOR VARIOUS COMBINATIONS OF WEIGHTING COEFFICIENTS

Weighting	gaincost	phasediff	poutcost	diffsat	effcost	cost
$e = 1$	1.23	1.99	0	1.20	.85	.85
$b = 1, e = 10$	.61	.44	0	1.18	.94	9.84
$b = 1, e = 5$	.48	.23	0	1.18	.97	1.2
$a = b = 1, e = 10$	.45	.33	0	1.21	.95	10.28
$a = b = 1, e = 5$	.29	.13	0	1.21	1.02	5.52

DIRECT is a global sampling algorithm; however, global convergence usually requires a large and exhaustive search over the problem domain. In particular, DIRECT performs many functional evaluations without improvement to the minimal value of the objective function [12]. To overcome this shortcoming in the local behavior of the DIRECT algorithm, we explored the implementation of DIRECT algorithm in a hybrid approach. That is, we first used the DIRECT method as a starting-point generator. We then used the extremum values found by DIRECT as initial iterates for a local optimization method, such as the Nelder-Mead algorithm. Table V summarizes our results for this hybrid approach. Even though some better results were observed, the improvements were not significant, indicating that DIRECT method already achieved a good minimum. In general, we found that minimizing the combined goal function given by equation (8) required compromises between the various design components. Based on feedback from the TWT customer, for the specific application of interest the case where $a = b = 1, e = 5$ provided the best compromise. Please note, specifications for different applications may differ, and one of the strengths of this approach is in the flexibility of the optimization design process. Since applications will be different, it is beneficial for the designer to have the flexibility to weight the optimization accordingly.

Table VI summarizes the results for the non-optimized case, where the circuit period is uniform as a function of axial distance (no taper) and the optimal design corresponding to

$a = b = 1, e = 5$ case. As a function of frequency across the operating band, the table compares the range of saturated RF efficiencies, the small signal gain variation, saturated output power and the phase difference between small-signal and saturation. We note the large improvement in performance compared to the circuit geometry before optimization.

TABLE V

OPTIMIZATION RESULTS USING THE HYBRID APPROACH (DIRECT FOLLOWED BY NELDER-MEAD)

Weighting	gaincost	phasediff	poutcost	diffsat	effcost
$e = 1$	1.23	2.00	0	1.20	.84
$b = 1, e = 10$	0.60	0.43	0	1.18	0.94
$b = 1, e = 5$	0.48	0.22	0	1.18	0.97
$a = b = 1, e = 10$	0.44	0.32	0	1.24	0.95
$a = b = 1, e = 5$	0.31	0.14	0	1.20	1.00

TABLE VI

FREQUENCY DEPENDENT RANGE OF SATURATED RF EFFICIENCIES, SMALL SIGNAL GAIN VARIATION, RANGE OF SATURATED OUTPUT POWER AND RANGE OF PHASE DIFFERENCES BETWEEN SMALL-SIGNAL AND SATURATION. THE NON-OPTIMIZED CASE, WHERE THE CIRCUIT PERIOD IS UNIFORM AS A FUNCTION AXIAL DISTANCE (NO TAPER), AND THE OPTIMIZED CASE, $a = b = 1, e = 5$, ARE COMPARED.

	Saturated RF Efficiency Range [%]	Small signal gain variation [dB]	Saturated Output Power [W]	Phase Difference Range [degs]
No Taper	3.8-4.1	9.1	7-9	36-41
$a = b = 1, e = 5$	5.4-6.4	7.3	10-11.3	22-35

V. CONCLUSIONS

In this work, we described additional tools to effectively explore different circuit designs and automate parameter variation and optimization. In particular, we demonstrated that accurate simulation codes, such as the physics-based large-signal code CHRISTINE-1D, can be seamlessly integrated with state-of-the-art optimization codes, such as DIRECT and/or Nelder-Mead algorithms. By treating simulation and optimization codes as "black boxes", one can explore different circuit designs using combinations of optimization criteria formulated through the goal (objective) functions. We demonstrated that our proposed optimization tool, the Nelder-Mead algorithm, appears to be more robust when compared to the CHRISTINE-1D built-in efficiency optimization algorithm for this circuit design. This may be particularly useful in the optimization of TWT multistage depressed collector stages where a good initial iterate is not always available. In addition, we also applied our optimization tools to the design of a folded waveguide slow-wave circuit. We demonstrated that CHRISTINE-1D with DIRECT and Nelder-Mead in a hybrid optimization approach can improve the folded wave guide slow-wave circuit performance based on several customer supplied, frequency dependent, optimization criteria. These criteria include saturated RF efficiency, small signal gain and phase difference minimization, and desired output power at saturation. One of the strengths of this approach is in the flexibility of the optimization design process. Specifications for different applications will differ, and thus it is beneficial for the designer to have the flexibility to weight the optimization accordingly.

APPENDIX: SAMPLING OPTIMIZATION METHODS

In this appendix we provide an overview of the sampling optimization algorithms that we use in the optimal design of the TWT circuits. Basically, in an optimal design problem, we begin by formulating a cost (or goal) function that characterizes our design goals. Our task is then to minimize or maximize this function, and thus obtain a design that meets our desired criteria. Mathematically speaking, the problem is given a function $f:\mathbb{R}^N \to \mathbb{R}$ find $\lambda^* \in \mathbb{R}^N$ such that $f(\lambda^*) \le f(\lambda)$ for all λ of interest. If the λ's of interest are only those near λ^*, then it is a local optimization problem. On the other hand, if the λ of interest belong to a subset $\Omega \subset \mathbb{R}^N$, then it is a global optimization problem. The optimization routines we used in our optimal circuit design, Nelder-Mead and DIRECT, are known as deterministic sampling methods [7]. By a

sampling method we mean that they use only function values to find the minimum or maximum and do not use gradient information as in CHRISTINE-1D. For a discussion on the advantages and disadvantages of sampling based methods versus gradient based methods, we refer the interested reader to [7].

(a) Nelder-Mead Algorithm

The Nelder-Mead algorithm is a simplex-based method described by Nelder-Mead [8]. Here the word simplex is not to be confused with the simplex method of linear programming [9]. The Nelder-Mead method or downhill simplex method is a numerical method for optimizing multi-dimensional and nonlinear unconstrained problems. It uses the concept of a simplex, which is a polytope of $N+1$ vertices in N dimensions. It is a line segment in one-dimension, a triangle on a plane, a tetrahedron in three-dimensions, and so forth.

In one-dimensional minimization, the strategy is to bracket a minimum so that subsequent isolation can be guaranteed. The Nelder-Mead algorithm extends this one-dimensional minimization algorithm to the multi-dimensional case. In essence, the algorithm starts with N+1 simplex of points in $\mathbb{R}^N$, $\{\lambda_1, \lambda_2, ..., \lambda_{N+1}\}$; it first sorts the simplex points such that $f(\lambda_1) \le f(\lambda_2) \le ... \le f(\lambda_{N+1})$. It then attempts to minimize the function by replacing the simplex point with the highest function value (i.e., the worst point) with a new point with a lower function value. It first computes the centroid of the simplex, not including the worst points $\overline{\lambda} = \frac{1}{N}\sum_{i=1}^{N} \lambda_i$. It then attempts to replace λ_{N+1} with $\lambda_{NEW} = (1+\beta)\overline{\lambda} - \beta\lambda_{N+1}$, where $\beta = \{\beta_R, \beta_E, \beta_{OC}, \beta_{IC}\}$, and the values for β stand for a reflection, expansion, outward contraction and inward contraction of the simplex, respectively. The values for β are taken from $-1 < \beta_{IC} < 0 < \beta_{OC} < \beta_R < \beta_E$. If none of these values are better than the previous worst point, the algorithm shrinks the simplex towards the best point, i.e., it replaces the points with $\overline{\lambda_i} = (\lambda_i + \lambda_0)/2$. The algorithm then resorts the points and iterates. To better illustrate this method consider the case where $N = 2$. In this case our parameter space is $\mathbb{R}^2$, and our simplex is a triangle in the plane. Figure 5 illustrates the Nelder-Mead process for this case. Note λ_3 is the "worst parameter", and the algorithm attempts to replace this point with a reflection (r), expansion (e), outward contraction (oc), or inward contraction (ic). There are various stopping

criteria for this algorithm, including termination of the algorithm when the iteration number exceeds a user prescribed value, when the vector distance moved in that step is fractionally smaller than some user given tolerance, or when the difference between the function at the best and worst points are sufficiently small. For a more detailed treatment of the Nelder-Mead algorithm see [7, 10].

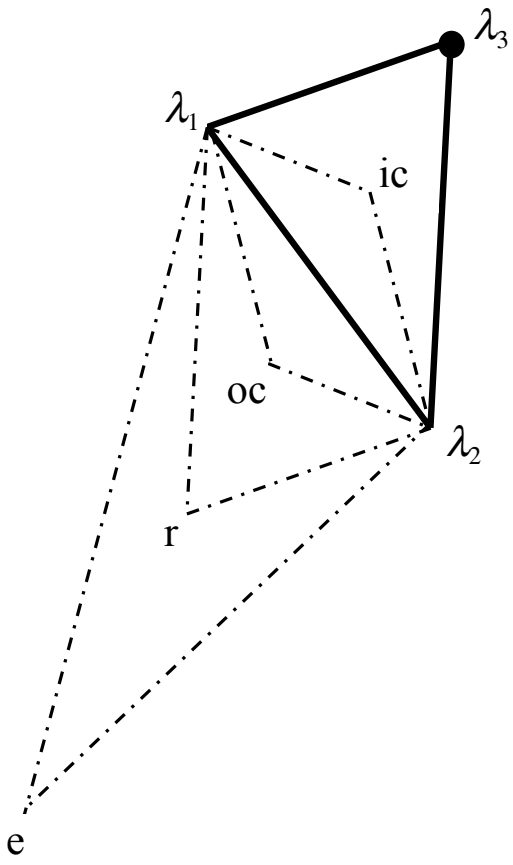

Fig. 5. Nelder-Mead in $\mathbb{R}^2$.

(b) DIRECT Method

The second algorithm we used is DIRECT, which was first introduced in [11]. DIRECT was designed for simple, bounded, constrained, non-smooth problems, and, as the optimization progresses, it performs a global search. In the limit, DIRECT will sample a dense grid of points in the feasible set. The name DIRECT stands for DIviding RECTangles , which is very indicative of how the algorithm progresses towards finding the optimum. More specifically, we consider a bounded-constrained optimization problem of the form

$$\min_{x \in \Omega} f(x), \tag{1}$$

where $f : \mathbb{R}^N \to \mathbb{R}$ and is Lipschitz continuous on $\Omega = \{x \in \mathbb{R}^N \mid L_i \le x_i \le U_i\} \subset \mathbb{R}^N$. Since Ω has hard bounds, we refer to Ω as a hyper-rectangle in $\mathbb{R}^N$. DIRECT initiates its search by sampling the objective function at the center of Ω, c. At each iteration, DIRECT method balances the global and local searches by identifying potential optimal hyper-rectangles based on the value of the objective function at the center and the size of the hyper-rectangle. For example, at the first iteration when the hyper-rectangle Ω is potentially optimal, all coordinate directions of the hyper-rectangle are long, and DIRECT divides every direction (see Fig. 6). DIRECT employs a simple heuristic argument to determine the order in which long sides (or directions) of the hyper-rectangles are divided. In particular, it evaluates the objective function at $c \pm \delta e_i / 3$, $\omega_i = f(c \pm \delta e_i / 3)$, where δ is the maximal side length of the hyper-rectangle. It then divides the hyper-rectangle into smaller hyper-rectangles (thirds), starting with the dimension with smallest ω_i and continuing to the dimension with the largest ω_i. In subsequent iterations, it determines which of the hyper-rectangles is potentially optimal and divides them in a similar manner. Once again, to better illustrate this method, we consider the case where $N = 2$. Here hyper-rectangles are merely rectangles, and the division process is easy to visualize. Figure 6 illustrates this process. For a more detailed treatment of the DIRECT method, we refer the interested reader to [12, 13].

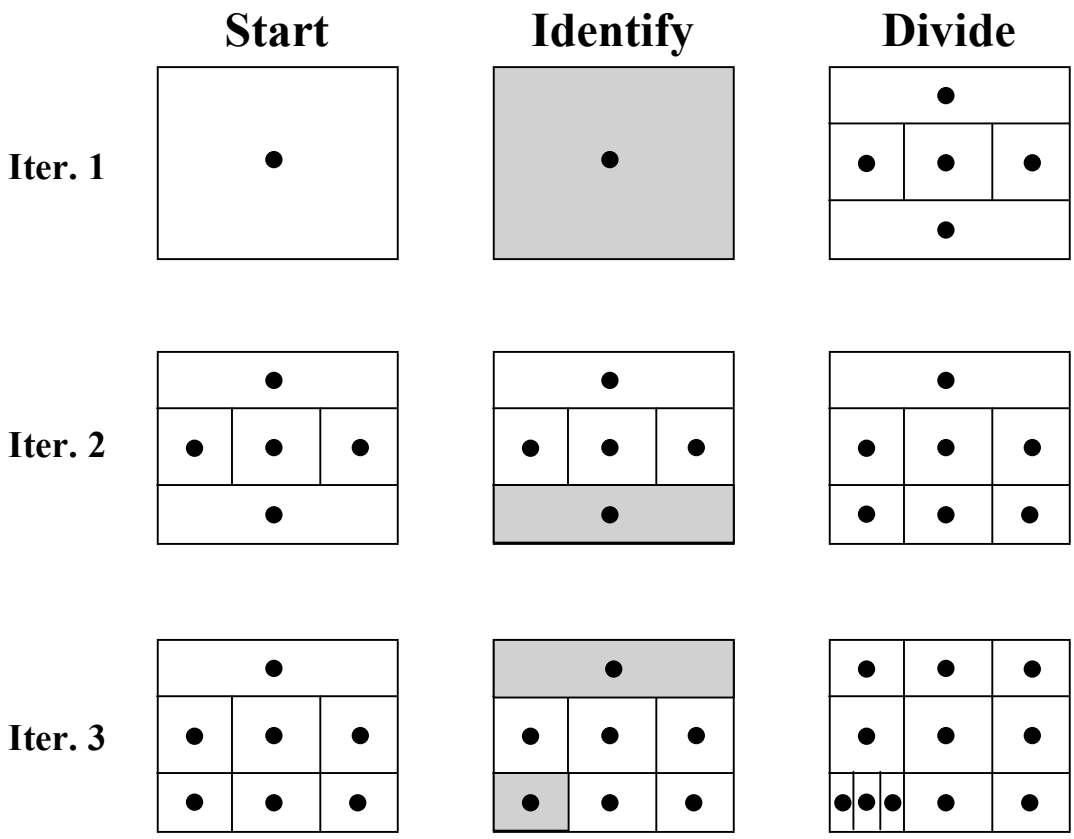

Fig. 6. An illustrative example of the iterative process of the DIRECT algorithm in a plane. The shaded rectangle is the region that is identified by the algorithm for division.

ACKNOWLEDGMENTS

This research was supported by the Naval Research Laboratory, contract number N00014-04-M-0170, and the Air Force Research Laboratory, contract number F29601-03-C-0049.

REFERENCES

1. Lindenblad, N., *U. S. Patent 2,3000,052*. Filed May 4, 1940 Issued October 27,1942: United States of America.
2. Kompfner, R., *The invention of the traveling-wave tube.* IEEE Trans. Electron Devices, 1976. **ED-23**(7): p. 730-738.
3. Kosmahl, H. and J. Peterson. *A TWT amplifier with a linear power transfer characteristic and improved efficiency*. in *10th Communication Satellite Systems*. 1984. Orlando, Florida: American Institute of Aeronautics and Astronautics.
4. D. Abe, B.L., T. Antonsen Jr., D. Whaley, B. Danly, *Design of a Linear C-Band Helix TWT for Digital Communications Experiments Using the CHRISTINE Suite of Large-Signal Codes.* IEEE Trans. Plasma Science, 2002. **30**(No. 3): p. 1053-1062.
5. Wilson, J., *Design of high-efficiency wide-bandwidth coupled-cavity traveling-wave tube phase velocity tapers with simulated annealing algorithms.* IEEE Trans Elect Dev, 2001. **48**(1): p. 95-100.
6. Lewis, B., et al., *Design of an electron gun using computer optimization.* IEEE Trans. on Plasma Science, 2004. **32**(3): p. 1242-1250.
7. Kelley, C., *Iterative Methods for Optimization*. SIAM Frontiers in Applied Mathematics. 1999, Philadelphia PA: SIAM.
8. Nelder, J. and R. Mead, *A Simplex Method for Function Minimization.* Computer Journal, 1965. **7**(4): p. 308-313.
9. Dantzig, G., *Linear Programming and Extensions*. 1963, Princeton: Princeton University Press.
10. Press, W., et al., *Numerical Recipes in Fortran 77*. 1992, Cambridge: Cambridge University Press.
11. Perttunen, C., D. Jones, and B. Stuckman, *Lipschitzian optimization without the Lipschitz constant.* J. of Opt. Theory and Appl. **79**(1): p. 157-181.
12. Finkel, D., *Direct Optimization Algorithm User Guide*, in *CRSC-TR03-11*. 2003, Center for Research in Scientific Computation at North Carolina State University: Raleigh, NC.
13. Finkel, D. and C. Kelley, *Convergence Analysis of the DIRECT Algorithm*, in *CRSC-TR03-11*. 2004, Center for Research in Scientific Computation, North Carolina State University: Raleigh.
14. Whaley, D.R., et al., *Sixty-percent-efficient miniature C-band vacuum power booster for the microwave power module.* IEEE Trans. Plasma Science, 1998. **26**(3): p. 912-921.
15. Kory, C., et al. *Microfabricated W-band traveling wave tubes*. in *The Joint 30th International Conference on Infrared and Millimeter Waves and 13th International Conference on Terahertz Electronics*. 2005. Williamsburg, Virginia.
16. Bhattacharjee, S., et al., *Folded waveguide traveling wave tube souces for THz radiation.* IEEE Trans Plasma Dev, 2004. **32**(3): p. 1002-1014.
17. *CST Microwave Studio Suite*, Computer Simulation Technology (CST).