Terahertz multi-film material parameter extraction using gradient descent algorithms

THEO SMITH[1], ARTA LATIFI [2] NICHOLAS T. KLOKKOU,[1] BEN BEDDOES,[1,*], JON GORECKI 2, GEORGIOS BARMPARIS, VASILIS APOSTOLOPOULOS [5,6,1]

[1]*School of Physics and Astronomy, University of Southampton, United Kingdom*
[2]*Royal School of Mines, Imperial College London, United Kingdom*
Germany
[5]*University of Crete, Department of Physics, Heraklion, GR-70013, Greece*
[6]*Foundation for Research and Technology-Hellas, Institute of electronic structure and Laser, Heraklion, Crete*
[4]*Optoelectronics Research Centre, University of Southampton, United Kingdom*
[*]*scr1n19@soton.ac.uk*

Abstract: Terahertz time domain spectroscopy is a technique to examine the properties of materials using THz radiation. Extracting these properties from experimental data can be unstable and laborious, especially when the data contains multiple reflections, such as in thin or multi-layer samples. In these cases, traditional extraction algorithms typically fail as there are too many free parameters to solve for. Here we develop an extraction method that allows for stable and accurate extraction of material parameters from multi-layer samples. Our method utilises a combination of Bayesian optimization and gradient descent algorithms operating in the time domain. We test these optimization algorithms on both simulated and experimental data, demonstrating their ability to extract sample parameters from multi-layer systems without prior knowledge of the samples. Our method will be good for some practical reasons and examples of things that it would be good for and why people want to us it.

1. Introduction

Terahertz Time Domain Spectroscopy (THz-TDS) is a powerful analytical technique for determining the complex optical properties of materials. Importantly, THz-TDS has an advantage over nearly all spectroscopic methods in that it is able to directly measure the time varying electric field of the electromagnetic pulse, as opposed to simply sampling the time averaged magnitude, thereby retaining information on the phase of the electric field components. Due to this fact, THz-TDS is able to determine the complex frequency-dependent optical parameters of a material (either refractive index, conductivity, or permittivity) without having to rely on approximations or assumptions. The analysis of material spectra generally involves fitting the experimental data to a theoretical transfer function, and modifying the material properties to minimise the difference between theory and experiment. For the simplest case of a flat, homogeneous, single-layer, optically thick (thickness $> \lambda_0$) sample the existing THz-TDS fitting routines are extremely stable and generally converge on a single solution within a few iterations. However, many samples do not satisfy this list of prerequisites, and once the sample geometry becomes more complicated (such as for thin or multi-layer samples) the transfer functions produce multiple mathematically-valid solutions; numerical fitting routines encounter non-convex loss functions, multiple local minima appear, and therefore determining the true material parameters becomes challenging.

what is the angle for this introduction? are we saying that we are borrowing the optimisation algorithms from ML, and then using them in a more traditional setting?

Material parameter extraction can be formulated as an inverse problem. Simplifying the problem of a multi-layered sample by assuming a discrete, frequency independent, complex refractive index for each layer. This reduces the number of free parameters being solved for. In this framework, a forward model maps a set of material parameters to a predicted system response in our case, producing the time-domain sample pulse given the material parameters and a measured reference pulse. The inverse problem is then solved by adjusting the material parameters until the model output matches the experimental data within a defined tolerance. This approach implicitly assumes that the forward model provides an accurate physical description of the system, making the choice of model particularly one suited to the sample geometry critical to obtaining meaningful results. The existing methods typically use the Fresnel model of light propagating through a sample to construct a theoretical transfer function from n and k.

$$H_{\text{theoretical}} = \frac{4\tilde{n}}{(\tilde{n}+1)^2} e^{-\frac{i(\tilde{n}-1)\omega d}{c}}, \tag{1}$$

An objective function is then constructed using this and the experimental data.

$$f(\tilde{n},\omega) = \ln\left(H_{theoretical}\right) - \ln\left(H_{experimental}\right) \tag{2}$$

The root of this objective function ($f(\tilde{n},\omega)$) with respect to $\tilde{n}$ is the solution to this inverse problem. The root is found using Newton's method, an iterative root finding algorithm that uses the following parameter update rule.

$$\tilde{n}_{t+1} = \tilde{n}_t - \frac{f(\tilde{n_t})}{f'(\tilde{n_t})} \tag{3}$$

This works well for basic geometries but modified versions for thin samples with reflections or multiple layers struggle to stably converge to a single solution. In this work, we are not using the Fresnel model, opting instead for a forward model based on the matrix transfer method [1]. This allows the model to produce data for samples with multiple layers. Given the complexity of solving the objective function with this method, we dispense with Newton's method in favour of

gradient descent. This means we do not require an analytical derivative of the objective with respect to the forward models parameters. This is commonly used in deep learning to train neural networks to fit their training data. The basic gradient descent algorithm uses the parameter update rule below for a sample with N layers.

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t) \tag{4}$$

$$\text{where } \theta = \{\theta_1, \theta_2, \ldots, \theta_N\} \text{ with each } \theta_j = \{n_j,\ k_j,\ d_j\} \text{ for } j = 1, \ldots, N.$$

θ is the parameter set being updated and $\mathcal{L}(\theta_t)$ describes the objective function, t indicates the optimization step and η is a learning rate (an arbitrary constant).

For gradient descent to be used, the forward model must also be differentiable with respect to it's input parameters. This means functions such as phase unwrapping cannot be used because this is a discontinuous process. The objective here is to minimize the root mean squared error (rMSE) between the predicted time domain pulse and the experimental one. By doing this, we are fitting an average refractive index (w.r.t frequency) for each layer by calculating loss in the time domain.

1.1. Transfer Matrix methods for multilayer samples in THz-TDS

Any model composed of differentiable operations (e.g., as implemented in PyTorch) can, in principle, be optimized using gradient-based methods such as Adam. However, differentiability alone does not guarantee successful or accurate convergence due to potential issues such as non-convex loss landscapes, ill-conditioned gradients, or sensitivity to initial parameter values. In this work, the transfer matrix method (TMM) is adopted as the forward model, owing to its ability to compute the total transmission and reflection from stratified media with high physical fidelity. TMM provides a well-established, physically grounded framework that integrates well into a differentiable pipeline, enabling parameter inference for multi-layered structures via backpropagation.

This approach is based on the transfer matrix method (TMM) [1], a well-established technique for modeling wave propagation in stratified media. At each interface between layers, a portion of the incident THz pulse is reflected and transmitted, while a phase shift is accumulated during propagation through each layer [2]. By enforcing electromagnetic boundary conditions namely, the continuity of the components of the electric and magnetic fields the reflection and transmission coefficients can be derived analytically for each interface. Consider a pulse propagating from an incident medium with refractive index n_0 into a layer with complex refractive index $n_j = n'_j + ik_j$ and physical thickness d_j. The corresponding complex optical path length is given by $D_j = n_j \cdot d_j$. The single-layer reflection and transmission coefficients can then be expressed as [1]:

$$r = \frac{i}{2} \cdot \frac{\frac{n_0}{n_j} - \frac{n_j}{n_0}}{\cos(n_j D_j) + \frac{i}{2}\left(\frac{n_0}{n_j} + \frac{n_j}{n_0}\right)\sin(n_j D_j)} \tag{5}$$

$$t = \frac{1}{\cos(n_j D_j) + \frac{i}{2}\left(\frac{n_0}{n_j} + \frac{n_j}{n_0}\right)\sin(n_j D_j)} \tag{6}$$

These coefficients are computed for each layer in the sample. The overall reflection (R) and transmission (T) coefficients for the entire stack are then obtained by recursively combining the individual layer matrices using the standard TMM formulation. The complex transmission coefficient T captures both amplitude attenuation and phase delay, thereby accounting for absorption and dispersion effects across the structure. This enables the simulation of time domain THz responses by applying the computed T to a reference pulse in the frequency domain, followed by an inverse Fourier transform.

Once T is calculated, the sample's response to a reference pulse can be found. This is calculated in the frequency domain with an FFT (fast Fourier transform) to find $X(\omega)$. The frequency domain response is calculated using the transmission coefficient as the transfer function.

$$Y(\omega) = T(\omega) \cdot X(\omega) \tag{7}$$

An IFFT (inverse fast Fourier transform) is used to convert this back to the time domain $y(t)$.

$$y(t) = \mathcal{F}^{-1}[T(\omega) \cdot \mathcal{F}[x(t)]] \tag{8}$$

Figure 1 shows a completely simulated system with 3 layers n = 2,3,2 and d = 1, 0.1, 1 mm.

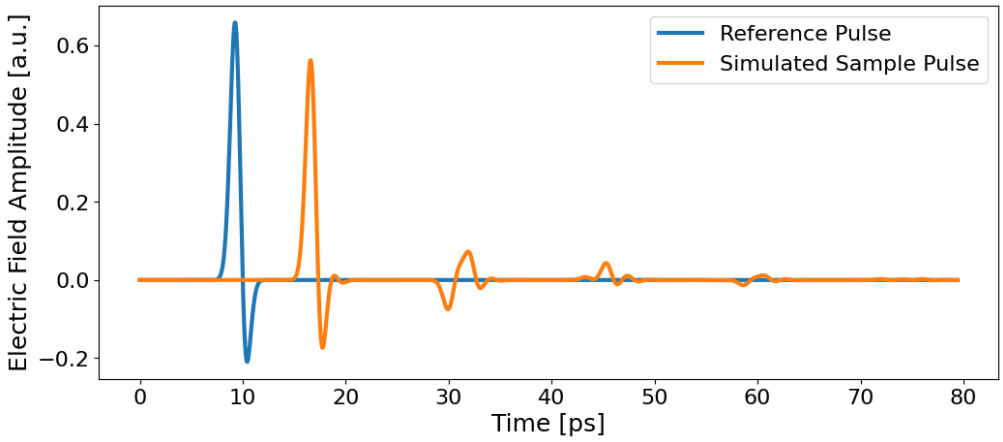

Fig. 1. Time domain of simulated reference and sample pulse for a 3 layered material with material parameters: $n = 2, 3, 2$ and $d = 1, 0.1, 1$ mm respectively. These were simulated using TMM.

1.2. Gradient descent for material parameter extraction

In order to fit for thickness at the same time as refractive index, we assume a constant refractive index across the frequency range. This will be addressed in future work. We have a discrete parameter set θ_j for each layer (j).

$$\theta_j = \{n_j,\ k_j,\ d_j\} \tag{9}$$

These parameters are passed to the TMM model to compute T giving $T(\theta)$. A prediction of the sample pulse from these parameters is obtained from equation 8.

$$\hat{y}(\theta) = \mathcal{F}^{-1}[T(\theta) \cdot \mathcal{F}[x(t)]] \tag{10}$$

We then construct a loss function to be minimized by a gradient based algorithm. Root mean squared error provides a lightweight and robust error metric for accurate comparison of the pulses.

$$L(\theta) \ = \ \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{\text{sim}}(t_i;\theta) - y_{\text{exp}}(t_i)\right)^2} \tag{11}$$

where N is the number of data points in the time domain, $y_{\text{exp}}(t_i)$ is the experimentally measured pulse and $y_{\text{sim}}(t_i;\theta)$ is the pulse simulated with TMM using parameters θ. Gradient descent updates parameters based on a learning rate and the gradient of the loss function with respect to the free parameters.

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t) \tag{12}$$

η is the learning rate and helps to determine the rate and resolution of the fitting and t denotes the index of the iteration. The number of iterations and the learning rate depend on the loss landscape of the problem. This can be affected by the level of noise, number of parameters and the physical properties of the sample. To improve convergence stability and reduce sensitivity to manual hyperparameter tuning, we employ the Adam optimizer [3], a gradient-based method that adaptively adjusts the learning rate for each parameter. Adam tracks two exponential moving averages, the first and second moments (mean of gradients). These are computed and modified in the following way:

The first moment at step t:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla_\theta \mathcal{L}(\theta_t) \tag{13}$$

and the second moment at step t:

$$\nu_t = \beta_2 \nu_{t-1} + (1-\beta_2)(\nabla_\theta \mathcal{L}(\theta_t))^2 \tag{14}$$

These are bias corrected using:

$$\hat{m_t} = \frac{m_t}{1-\beta_1}, \quad \hat{\nu_t} = \frac{\nu_t}{1-\beta_2} \tag{15}$$

This leads to the overall parameter update rule:

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{\nu}_t} + \epsilon} \tag{16}$$

Adam combines the advantages of momentum and RMSProp [3], making it well-suited for problems with noisy gradients or poorly scaled loss landscapes, such as those arising in inverse reconstruction of layered media.

1.3. Loss landscapes and Bayesian fitting for efficient parameter search

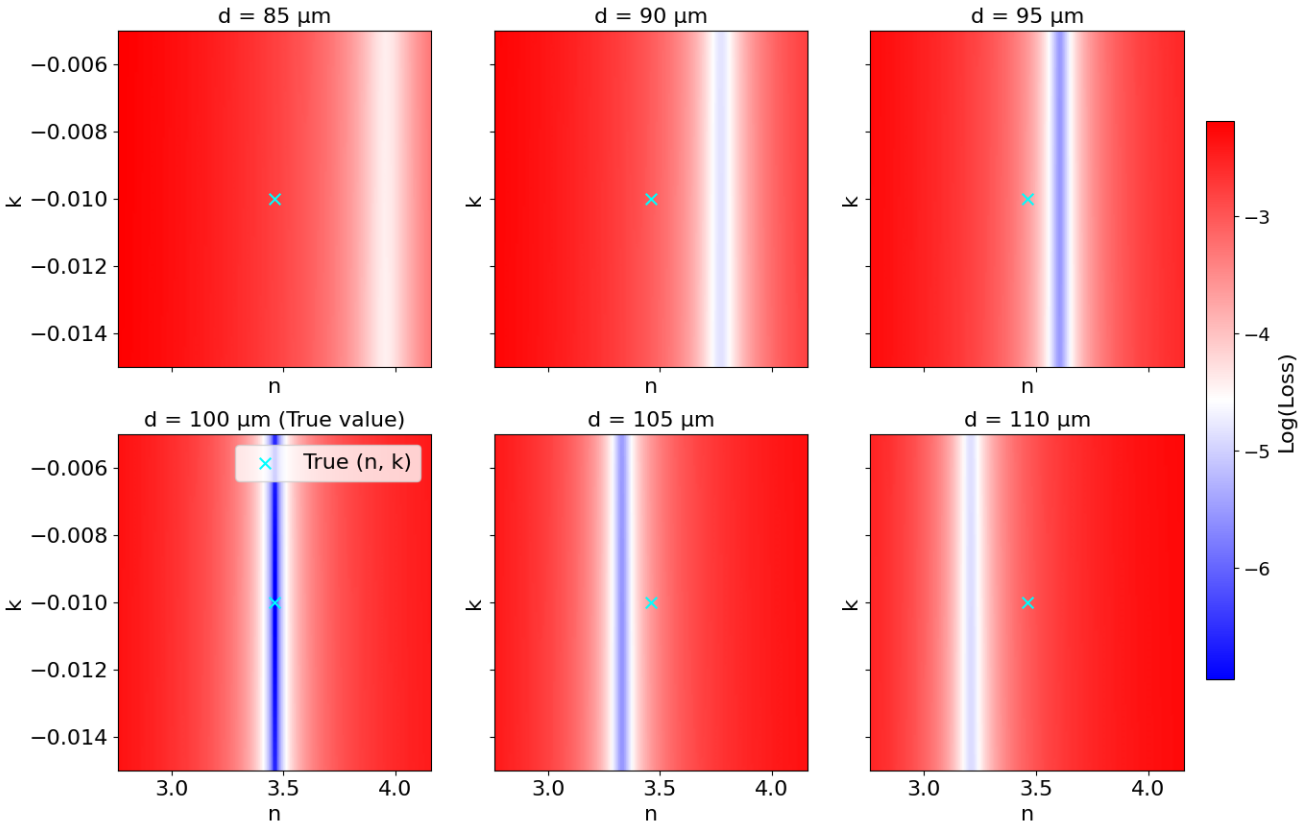

Fig. 2. Logarithmic loss landscapes as a function of refractive index parameters n and k for different assumed thickness values d. Each subplot corresponds to a fixed thickness between 85–110 μm. The black dot marks the true (n, k). At the correct thickness (d = 100 μm, bottom left), the true values coincide with the minimum-loss region (blue). For mismatched thicknesses, the minimum-loss region shifts n, illustrating the coupling between thickness and refractive index in the fitting process. These regions also have a higher value of minimum loss, representing a poorer overall fitting than that with the correct thickness.

For the adam algorithm, the rate of convergence depends on the gradient of loss at that point in the parameter space. Initial guesses that are far from the correct values tend to have a flat gradient, as can be seen in Figure 2. Areas with low or no gradient cause the algorithm to need more steps to converge and in some cases, can prevent convergence all together. We are using Bayesian optimisation in order to speed up the gradient descent method and improve initial guess passed to the adam algorithm. Bayesian optimization is a global optimization strategy for expensive black-box functions. It models the objective using a probabilistic surrogate, typically a Gaussian Process [4]. Based on this model, it selects the next evaluation point by maximizing an acquisition function like Expected Improvement (EI) or Upper Confidence Bound [5, 6]. This method is particularly efficient in scenarios where function evaluations are costly. It has it has a range of uses including hyperparameter tuning for machine learning models [7]. The balance between exploration and exploitation is controlled explicitly through the acquisition function. Bayesian optimization provides an efficient method for searching wide parameter spaces with

a few features as in Figure 2. These parameters are then fine tuned using the gradient based method. Figure 3 shows the loss landscape at the correct thickness and zoomed into a smaller range of n values. This shows a clear gradient to a minimum value of loss that coincides with the true values of n and k. This landscape is optimal to be solved by gradient descent.

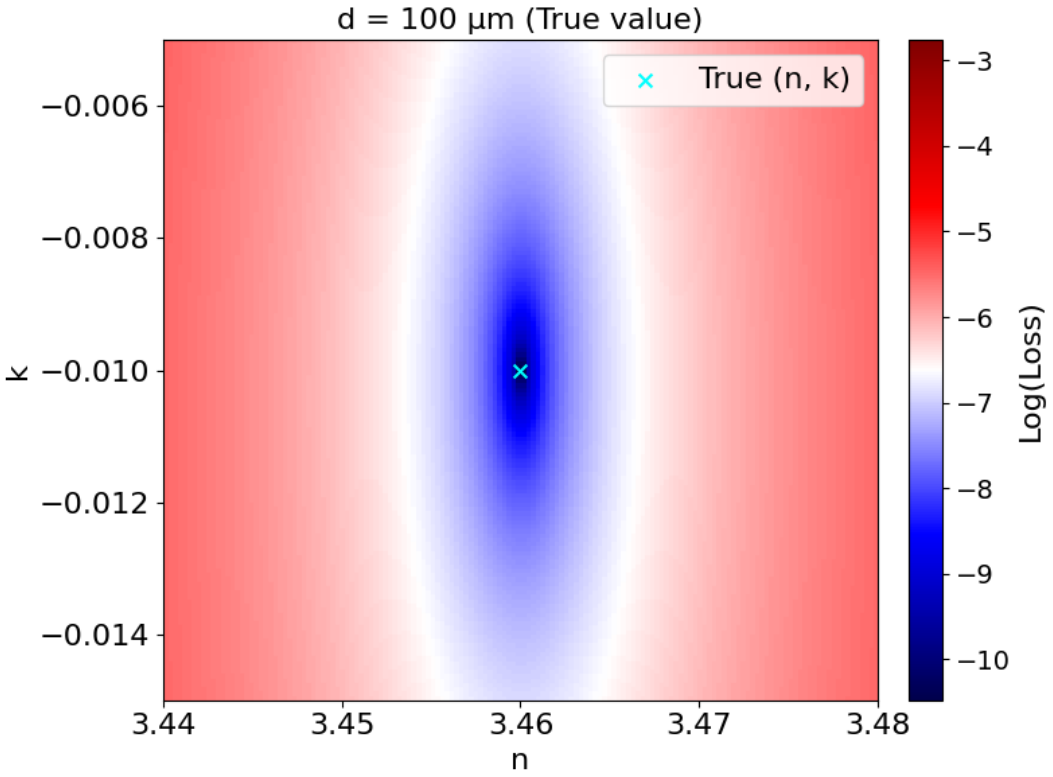

Fig. 3. Log-loss landscape for d = 100 μm (true value). The refractive index axis is shown in a narrowed range, highlighting the steep gradient of the loss surface with respect to n and a visible gradient with respect to k.

In Figure 4 the time domain of simulated reference and sample pulse for a 2 layered material with material parameters: n = 3.46-0.01j, 2.5-0.015i and d = 1,0.2 mm respectively is shown in 4(a). These were simulated using a matrix transfer methods. In 4(b) the Bayesian optimisation results are shown where for the two layers we obtained n=3.466-0.005j, 2.493-0.023j. In Figure 4(c) using the Bayesian as a first approximation the gradient descent converges to 3.4600-0.010j, 2.50007-0.01501j which is close to the initial values. The Bayesian fitting stage allows for a fast and rough search of a wide parameter space for samples with largely unknown material parameters. In this case the search space ranged from n_1: 2.67-3.67, k_1: -0.025–0.005, d_1: 600-1200 μm for layer 1 and n_2: 1.7-2.7, k_2: -0.027–0.007, d_2: 100-700 μm for layer 2. The values from this stage are then fine-tuned using the adam based algorithm to achieve a higher level of accuracy.

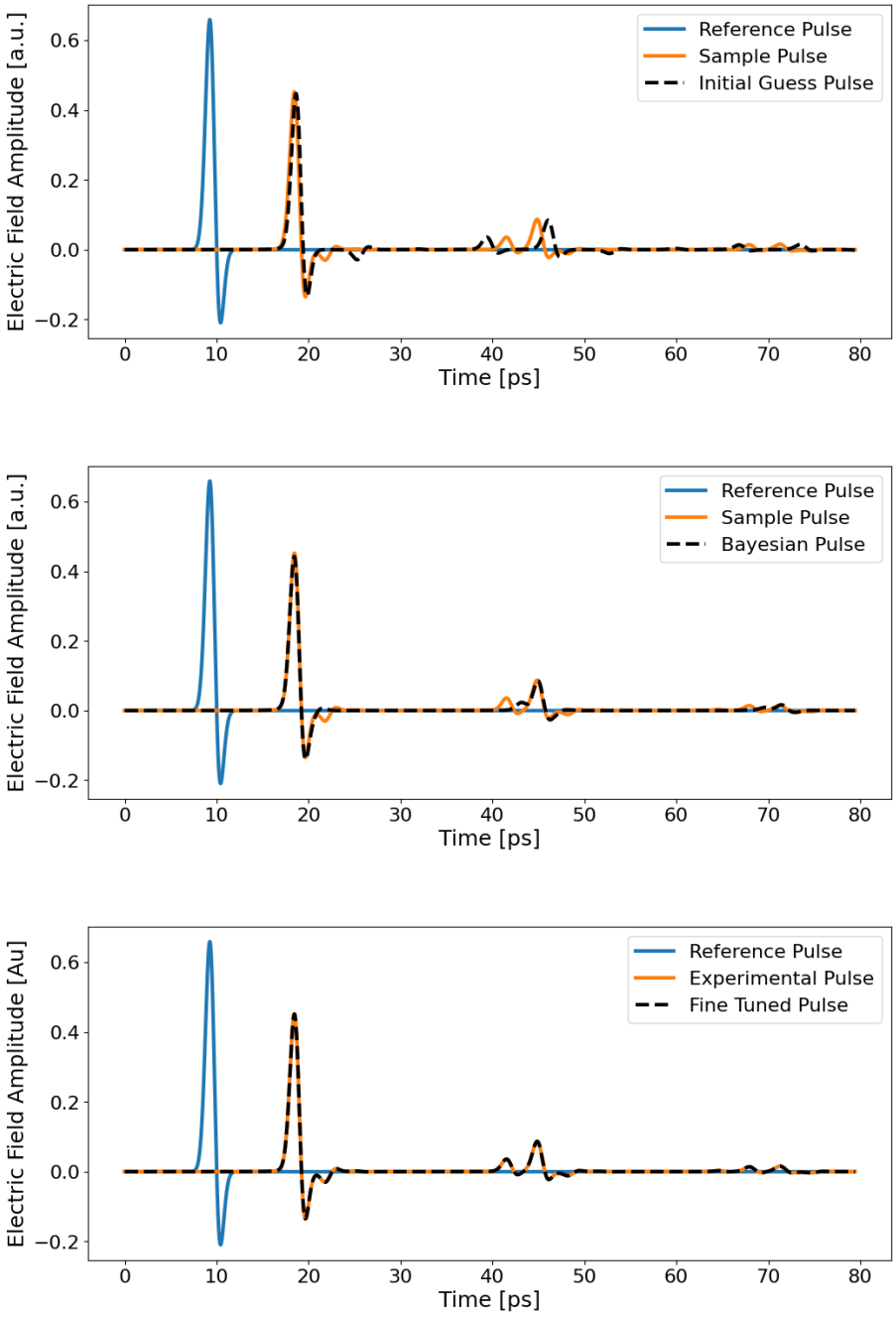

Fig. 4. (a) Time domain of simulated reference and sample pulse for a 2 layered material with material parameters: n = 3.46-0.01i, 2.5-0.015i and d = 1,0.2 mm respectively. These were simulated using a transfer matrix methods. The simulated pulse using the initial guess parameters is shown as well (black dashed). (b) For a 2 layered material with material parameters: n = 3.46-0.01i, 2.5-0.015i and d = 1,0.2 mm, Bayesian optimisation is shown, the Bayesian optimisation results are 3.34-0.014i, 1.11 mm for layer 1 and 2.7-0.016i, 0.10 mm for layer 2. (c) Using the Bayesian optimization's result as the seed value for the gradient descent converges to 3.4601-0.0099i, 1.0002 mm and 2.5001-0.01490i, 0.20003 mm. The reconstructed pulse using these parameters is shown by the black dashed line.

2. Results

In order to test the optimisation method, we have performed extractions on a variety of simulated samples. Initial guesses for the material parameters are randomly selected. The algorithms are tested on a 5 layer sample with 15 free parameters (3 per layer). An initial value is randomly selected within a confidence interval of the material parameters. The Bayesian model then searches a space with the following bounds on each material parameters: [n: ± 0.5, k: ± 0.01, d: ± 50 μm]. The results from this are then passed to the gradient based model that converges to fine-tune each material parameter.

should we include loss vs epoch graphs for with / without Bayesian optimisation?

179 ## 2.1. 5 Layer extraction

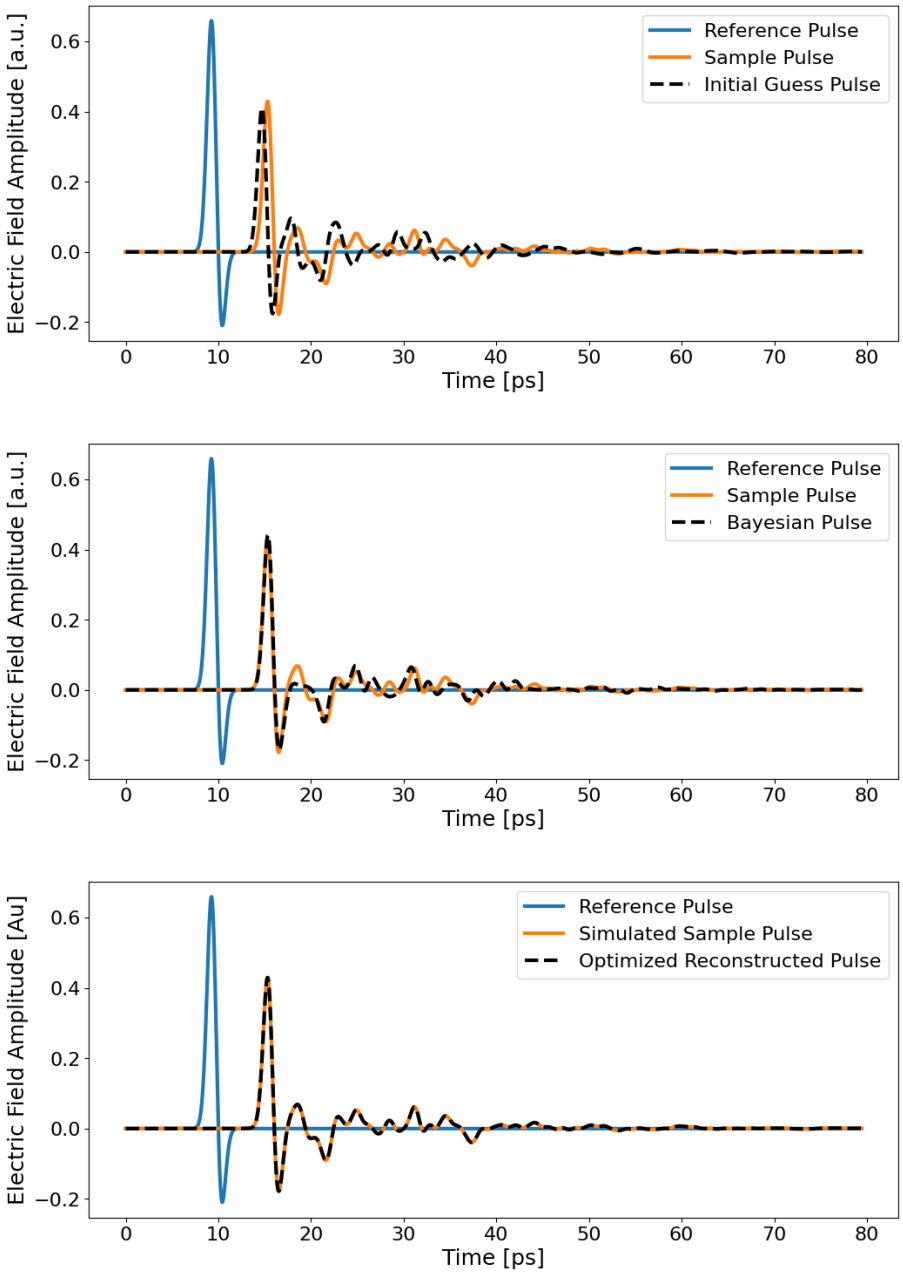

Fig. 5. (a) Time domain of simulated reference and sample pulse for a 5 layered material (b) Bayesian optimisation results with sample pulse and initial guess pulse (c) Using the Bayesian as a first approximation the gradient descent converges to 4th or 5th decimal accuracy

180 Table 1 shows the results from the gradient algorithm when run on clean clean pulses.

Table 1. Optimization results of a clean pulse from a 5 layer simulated sample.

Layer	n				k				d (µm)			
	n_{true}	n_{fit}	**Residual**	**% Error**	k_{true}	k_{fit}	**Residual**	**% Error**	d_{true}	d_{fit}	**Residual**	**% Error**
1	3.364	3.3653	+0.00127	+0.038%	−0.000356	−0.000399	−0.000043	+12.1%	0.150	0.1500	−0.00005	−0.03%
2	2.522	2.5229	+0.00088	+0.035%	−0.003711	−0.003618	+0.000093	−2.5%	0.179	0.1789	−0.00006	−0.03%
3	5.411	5.4104	−0.00058	−0.011%	−0.005331	−0.005254	+0.000077	−1.4%	0.102	0.1020	+0.00000	0.00%
4	4.213	4.2115	−0.00150	−0.036%	−0.008436	−0.008378	+0.000058	−0.69%	0.206	0.2061	+0.00009	+0.04%
5	1.660	1.6597	−0.00032	−0.019%	−0.009043	−0.008860	+0.000183	−2.0%	0.120	0.1200	+0.00003	+0.02%

3. Discussion

4. Conclusions

References

1. P. Yeh, *Optical Waves in Layered Media* (Wiley, 1988).
2. J.-Y. Zhang, J.-J. Ren, L.-J. Li, *et al.*, "Terahertz spectroscopy and effective medium theory for thickness measurement of adhesive bonds," NDT & E Int. **147**, 103216 (2024).
3. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," (2017).
4. C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning* (MIT Press, 2006).
5. J. Mockus, V. Tiesis, and A. Zilinskas, "Application of bayesian approach to numerical methods of global and stochastic optimization," J. Optim. Theory Appl. **34**, 445–459 (1978).
6. N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *ICML,* (2010).
7. J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," (2012).