

# Wiring Diagnostics via $\ell_1$ -Regularized Least Squares

Stefan Schuet, *Member, IEEE*

**Abstract**—A new method for detecting and locating wiring damage using time domain reflectometry is presented. This method employs existing  $\ell_1$  regularization techniques from convex optimization and compressed sensing to exploit sparsity in the distribution of faults along the length of a wire, while further generalizing commonly used fault detection techniques based on correlation and peak detection. The method’s effectiveness is demonstrated using a simulated example, and it is shown how Monte Carlo techniques are used to tune it to achieve specific detection goals, like a certain false positive error rate. In addition, this method is easily implemented by adapting readily available optimization algorithms to quickly solve large, high resolution, versions of this estimation problem. Finally, the technique is applied to a real data set, which reveals its impressive ability to identify a subtle type of chaffing damage on real wire.

**Index Terms**—diagnostics, fault detection, inverse scattering, lossless media, sparsity, time domain reflectometry, TDR, wiring

## I. INTRODUCTION

**T**HIS paper considers the specific problem of detecting faults in wiring systems using time domain reflectometry. Generally, this is performed by launching a known signal into a wire, and examining the signal reflected back for potential issues (Figure 1 below). An important aspect of this technique is that one can detect and locate wiring problems before hard short or open conditions occur. One specific application is to aircraft wiring systems that are hard to inspect visually, and where it is critical to identify problems before components start to fail. Wiring diagnostics aside, the general idea is related to *inverse scattering*, which appears in many other areas, including the identification of layered earth systems in geology, and vocal tract area reconstruction in acoustics [1], [2]. This problem is certainly not new, and has been studied in various forms for more than half a century.

The setup is presented in Figure 1. A Time Domain Reflectometer (TDR) is connected to the transmission line we want to check, and is used to send a signal down the wire. The reflected signal is then measured, and checked for anomalies that might indicate possible wiring problems along the line. For example, consider a simple case where the original transmission line is perfect (and has matched source and load impedance). In this case, we will see the incident signal pass right through the line without receiving any reflected signal back. Now imagine that

during the course of its lifetime, the outer shielding along a section of the wire is damaged, a common problem with aging aircraft caused by decades of wires rubbing together, among other things. This sort of damage will cause the incident voltage wave to reflect and travel back along the line where it will be measured by the TDR.

Existing methods for detecting wiring problems often fall into one of two categories. The first category contains techniques that solve the transmission line partial differential equations, using discrete or continuous methods, and then simply invert the solution process without incorporating the effects of measurement noise [2], [3]. The second category consists of methods that use simple linear models, which account for noise, and apply various least-squares based techniques, such as Kalman filtering [1]. Furthermore, some current developments in this field are focused on using specific spread spectrum TDR (SSTDR) input signals that can interrogate a wire for faults in live systems [4]. For these systems, fault sensing is typically performed by using a standard correlation technique, which is optimal in some sense when no other information is available to improve detection capability.

In this paper, we develop an improved method that also uses a linear model, but in addition to noise, directly accounts for the prior information that wiring faults are generally sparsely populated along the line. This method is extremely effective, and seems to be new to the field of time domain reflectometry. Furthermore, the application of this technique is described in general terms that are readily applicable to any particular input interrogation signal, which should hopefully allow the technique to find immediate practice in existing systems.

This paper is organized as follows. First we present a linear model for the TDR setup and measurement process just described. Next, the problem of detecting the location and severity of wiring damage is posed as an estimation problem, and a heuristic is introduced to find effective solutions to the original problem, by solving a convex optimization problem. Finally, we will show how the Fast Fourier Transform (FFT) makes it possible to efficiently solve large-scale problems. Numerical examples are presented along the way, including one that uses real TDR data.

## II. A LINEAR TDR MODEL

We assume the transmission line or just wire is lossless (and hence also distortionless), that any voltage wave traveling through it moves at constant velocity, and that the line is initially quiescent. To simplify the conceptual discussion, we

Manuscript created on March 30, 2009; revised on . This work was supported by NASA’s Aviation Safety Program, as part of Challenge Problem 8 within the Aircraft Aging and Durability Project.

S. Schuet is with NASA Ames Research Center, Intelligent Systems Division, Moffett Field, CA 94035. (e-mail: stefan.r.schuet@nasa.gov)

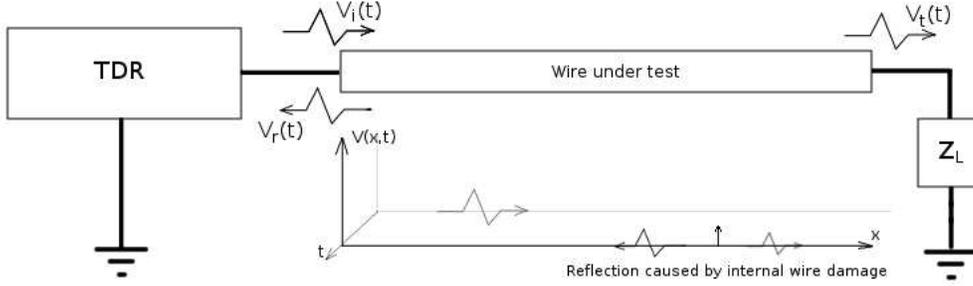


Fig. 1. Basic TDR Setup. The TDR interrogates the wire with input signal  $V_i(t)$ , which propagates along the wire and reflects off of impedance discontinuities caused by damage. The reflected signal,  $V_r(t)$ , is measured at the input of the wire and used to determine the location and severity of the damage.

also assume that the load and source impedance are matched to prevent significant reflection from the ends of the line.

We consider the following discrete convolution model for the TDR measurement process:

$$V_r(k) = \sum_{j=0}^{n-1} \mu(j)V_i(k-j) + \eta(k), \quad (1)$$

where for  $k = 0, 1, 2, \dots, n-1$ ,  $V_r(k)$  is the measured response,  $\mu(k)$  is a series of impulse response *reflection coefficients* that characterize the damaged wire,  $V_i(k)$  is the known incident wave launched into the transmission line, and  $\eta(k)$  is random measurement noise. This model has a simple interpretation: the measured signal is the sum of time shifted and scaled replicas of the input signal  $V_i(k)$ , plus noise.

The model can represent either causal or circular (periodic) convolution. For circular convolution we put  $V_i(-k) = V_i(n-k)$ . For causal convolution, we simply define  $V_i(k) = 0$  for all  $k < 0$ . Obviously, for either case  $V_r(k)$  must get the same treatment.

It is both instructive and notationally convenient to rewrite (1) in an equivalent matrix vector form:

$$v_r = V\mu + \eta \quad (2)$$

where,

$$\begin{aligned} v_r &= [V_r(0), \dots, V_r(n-1)]^T \\ \mu &= [\mu(0), \dots, \mu(n-1)]^T \\ \eta &= [\eta(0), \dots, \eta(n-1)]^T \end{aligned}$$

and,

$$V = \begin{bmatrix} V_i(0) & V_i(-1) & \dots & V_i(1-n) \\ V_i(1) & V_i(0) & \ddots & \vdots \\ \vdots & \vdots & \ddots & V_i(-1) \\ V_i(n-1) & V_i(n-2) & \dots & V_i(0) \end{bmatrix}.$$

So,  $v_r$ ,  $\mu$ , and  $\eta \in \mathbf{R}^n$ .  $V$  is a Toeplitz matrix in  $\mathbf{R}^{n \times n}$  entirely determined by the input signal  $V_i(k)$ .

### III. REFLECTION COEFFICIENT ESTIMATION

Nonzero values of  $\mu(k)$  indicate the location and severity of faults along the wire. Given the prior information that

wires are typically undamaged for most of their length, except perhaps at a few locations, the reflection coefficient vector  $\mu$  should contain only a few nonzero values. In other words, we expect  $\mu$  to be sparse. Thus, we are interested in solving the optimization problem:

$$\begin{aligned} &\text{minimize} && f_0(\mu) \\ &\text{subject to} && \mu \text{ sparse,} \end{aligned} \quad (3)$$

where,

$$f_0(\mu) = \frac{1}{2\sigma^2} \|V\mu - v_r\|^2 \quad (4)$$

is the objective representing the negative log-likelihood of observing the signal  $V_r$  given  $\mu$ , under the assumption that the noise  $\eta(k)$  is IID  $N(0, \sigma^2)$ .

One heuristic to handle the rather vague sparsity constraint in (3), is to add an  $\ell_1$ -norm penalty to the objective. This regularization technique is well known to produce sparse solutions (see [5]–[8] and [9] §6.3.2). To this end, we consider solving the convex  $\ell_1$ -regularized least squares problem (LSP):

$$\text{minimize } f_0(\mu) + \lambda \|\mu\|_1, \quad (5)$$

with  $\ell_1$ -norm defined as  $\|\mu\|_1 = \sum_{j=0}^{n-1} |\mu(j)|$ . Intuitively, the solution is sparse because in the process of finding an optimal solution, the solver will routinely reduce a small coefficient identically to zero at the cost of increasing the associated squared error  $f_0(\mu)$  by a smaller amount. The key observation is perhaps that square error measured by  $f_0$  stays relatively flat near a minimum, while absolute error measured by  $|\mu(k)|$  decreases to zero at a constant rate and does not level off (it is also not differentiable at  $\mu(k) = 0$  for each  $k$ ). Please see the references just cited for more examples and discussion.

The parameter  $\lambda \geq 0$  adjusts the trade-off between sub-optimality in the likelihood of the measured response, and the sparsity of  $\mu$ . Since effective values of  $\lambda$  for a given problem depend on the measurement noise variance  $\sigma^2$ , we will frequently specify the product  $\lambda\sigma^2$  (rather than just  $\lambda$ ) to highlight the interdependence between these two constants.

The fact that (5) is a convex optimization problem is an important feature for practical applications. Primarily, it means the optimal solution can be computed *globally*, in a robust and efficient manner [9].

### A. Relation to Least-Squares and Correlation Detectors

To see that (5) is a generalization of the least-squares problem we need only set  $\lambda = 0$ . In this case, the optimal solution is well known:  $\mu^* = (V^T V)^{-1} V^T v_r$ , assuming  $(V^T V)^{-1}$  exists. From here we observe two things.

The first is that for our problem  $V \in \mathbf{R}^{n \times n}$ , so in the best case scenario  $V$  is full rank, and the optimal least-squares solution simplifies to  $\mu^* = V^{-1} v_r$  (and  $f_0(\mu^*) = 0$ ). Unfortunately with this approach, we are making  $n$  observations corrupted by  $n$  independent noise terms (*i.e.*,  $\eta(k)$ ), which leaves us with no “redundancy” for obtaining a robust estimate and places it at the mercy of the measurement noise. What’s worse is that sometimes in practice, when the actual  $V_i(k)$  generated by the TDR hardware is measured,  $V$  becomes ill-conditioned or not even full rank, which makes reliably estimating  $\mu$  extremely problematic if not impossible. In fact, this approach is no better than inversion techniques that do *not* at all account for noise in the measurement and modeling process (*e.g.*, the  $\mu$  that minimizes  $f_0$ , which accounts for the presence of noise, is the same  $\mu$  one gets by removing the noise term  $\eta$  from (2) and solving for  $\mu$  directly).

The second observation is that if the columns of  $V$  are orthogonal (so that  $V^T V = I$ ), then the least-squares estimate reduces to  $\mu^* = V^T v_r$ , which when written out becomes the familiar discrete equation for the correlation between the input and output signals:

$$\mu^*(k) = \sum_{j=0}^{n-1} V_i(j-k) V_r(j), \quad (6)$$

This is a common detection technique employed in practice. For example, by selecting a pseudo-noise sequence for  $V_i(k)$ , one can satisfy the orthogonality condition (but only approximately in the finite causal discrete case), and we have the beginnings of the SSTDR systems that can detect faults on live wires, see [4].

With these observations in mind, it becomes apparent that for this application least-squares estimates do not enhance immunity to the effects of system noise, and that things might be improved if we can incorporate additional prior information. One straightforward approach would be to regularize the least-squares objective by an  $\ell_2$  penalization as in:

$$\text{minimize } f_0(\mu) + \lambda \|\mu\|^2 \quad (7)$$

This well known technique, sometimes called *Tikhonov regularization*, is designed to tradeoff “closeness of fit” to the observed data and the size of the reflection coefficients. This is a reasonable approach, since one is often interested in detecting small reflections. However, by solving (5) we can simultaneously obtain estimates that are both small and sparse, which is a more desirable goal for fault detection in many wiring systems.

### B. Example

To simulate the TDR measurement process, we begin by generating a sparse vector of reflection coefficients  $\mu \in \mathbf{R}^n$  as follows:

- 1) Randomly pick an integer  $N$  between 0 and 10 (with equal probability).  $N$  is the number of faults on the wire.
- 2) Draw  $N$  random reflection coefficients from a uniform distribution on  $[-0.5, 0.5]$ .
- 3) Assign the coefficients to  $N$  randomly chosen (equally probable) locations in  $\mu$ , and set all other elements to zero.

Next,  $n$  measurements of the reflected signal  $V_r(k)$ , for  $k = 0, 1, 2, \dots, n-1$ , are obtained by using the TDR measurement model (1), with some specified input signal  $V_i(k)$ . This method is used to generate simulated TDR data for the rest of the paper.

Consider an example with  $n = 200$ . The above procedure was used to generate a sparse reflection coefficient vector  $\mu \in \mathbf{R}^{200}$ , and a measurement of the reflected signal  $V_r(k)$ , from a unit step input signal  $V_i(k)$ , and measurement noise  $\sigma = 0.02$ . The  $\ell_1$ -regularized LSP (5) was then solved for several different values of  $\lambda$  using CVX, a package for specifying and solving convex programs [10], [11]. The results are plotted in Figure 2.

### C. Polishing

The previous example shows that for larger values of  $\lambda$ , the estimated reflection coefficients appear in the correct location, but typically have reduced amplitude (see Figure 2). This can be viewed as an artifact of the  $\ell_1$ -norm penalty function, since it favors smaller elements in  $\mu$ .

A simple technique called *polishing* alleviates this problem, simply by solving the original problem (3) with the sparsity pattern obtained from the solution to the  $\ell_1$ -regularized heuristic (5). Of course, when problem (3) has a fixed sparsity pattern, it becomes a simple least-squares problem.

Figure 3 shows the effect of polishing on the previous example (for the largest value of  $\lambda$  considered). Note, at least in this case, the technique almost always does the right thing:  $\mu_{est}(k)$  is brought closer to the actual value (even when the actual value is 0).

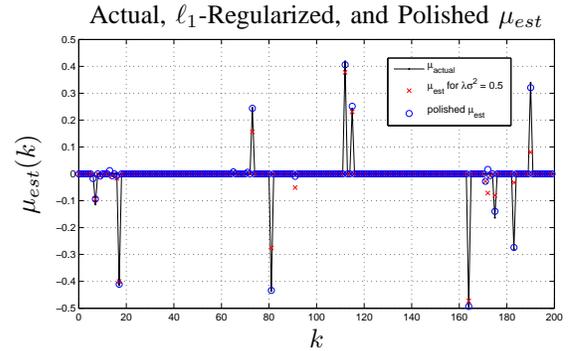


Fig. 3. Polishing example. The plot shows how the polished reflection coefficients  $\mu_{est}$ , are significantly closer to the actual values than the original set of estimated coefficients.

### D. The No-Fault Condition

This section presents how the selection of  $\lambda$  determines the no-fault condition (*i.e.*, all estimated reflection coefficients are zero).

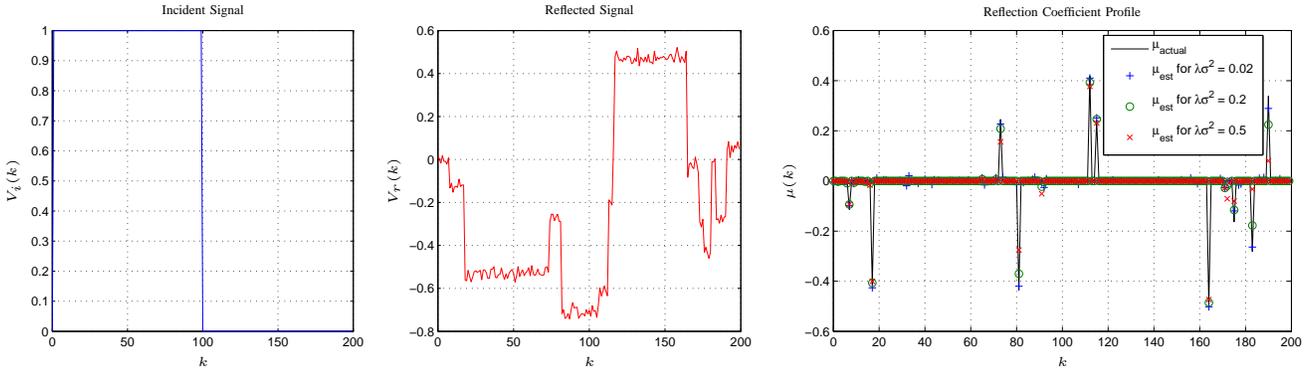


Fig. 2. A reflection coefficient estimation example. The estimation results for different values of  $\lambda\sigma^2$  are shown on the plot to the right. Note, that in all cases most of the reflection coefficients  $\mu(k)$  are zero as desired.

We begin by defining the correlation signal  $y(k)$  as,

$$y(k) = \sum_{j=0}^{n-1} V_i(j-k)V_r(j), \quad (8)$$

For each value of  $k$ , this signal measures the correlation between the measured response, and the input signal shifted  $k$  units in time.

Using subgradient calculus it is readily shown the optimal solution to (5) is  $\mu = 0$ , if and only if

$$\|\nabla f_0(0)\|_\infty = \max_{k=0, \dots, n-1} \left\{ \left| \frac{\partial f_0(0)}{\partial \mu_k} \right| \right\} \leq \lambda. \quad (9)$$

For our problem this implies:

$$\left| \frac{y(k)}{\sigma^2} \right| \leq \lambda \quad \text{for all } k = 0, 1, \dots, n-1. \quad (10)$$

This sensitivity condition simply states that if the best case correlation  $y(k)$  to noise ratio is less than  $\lambda$ , then the optimal solution to (5) will indicate no faults on the line (of course in reality faults may still be present). The condition could be important for designing sensors that are less prone to accidental tripping, but only if one can afford decreased sensitivity (and more false negative readings for smaller faults). This is further explored in the next section.

### E. Estimation Performance Example

In this section Monte Carlo simulation is used to investigate how the selection of  $\lambda$  affects our ability to correctly identify the reflection coefficient profile.

To do this we will continue to build on the previous example with  $n = 200$ . First, a set of 10 random reflection coefficient profiles, and corresponding TDR response data, were generated via the same process described earlier in §III-B (again with fixed noise standard deviation  $\sigma = 0.02$ , and an input step voltage wave). For each measured response, the estimation problem (5) was solved for a series of values  $\lambda$ . The number of false positives (detected reflections that are not real) and false negatives (real reflections not detected) were counted. Figure 4 shows the Receiver Operating Characteristic (ROC) curve for these results. As one might expect, larger values of

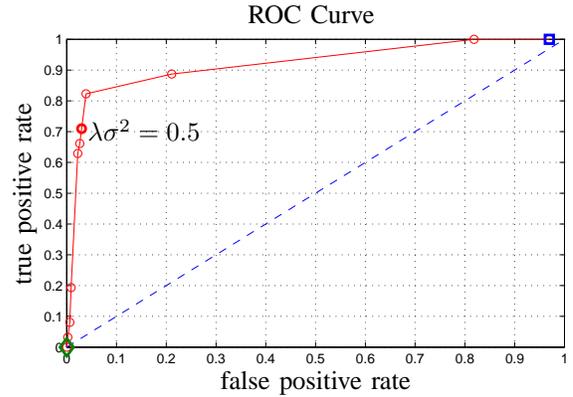


Fig. 4. ROC performance curve. The  $\square$  corresponds to  $\lambda\sigma^2 = 0$ , and the  $\diamond$  corresponds to  $\lambda\sigma^2 \simeq 75$ , which is the value that causes the method to report no faults along the wire (see §III-D).

$\lambda$  lead to fewer false positives (because we are encouraging sparsity) and as a consequence, more false negatives.

Figure 4 also provides us with a way to make decisions about which value of  $\lambda$  we want to use. For example, if we require a false positive rate of less than 5% (and consequently a true negative rate greater than 95%), we might select  $\lambda\sigma^2 = 0.5$ . With this setting now fixed, we evaluated the estimation performance on a new *test set* of 50 more random coefficient profiles and TDR response data. For this set of 10000 test points the false positive rate was 2.34%, with a corresponding true negative error rate of 97.6%. Figure 3 already presented an example comparing the actual, estimated, and polished reflection coefficients achieved with this value of  $\lambda\sigma^2$ .

## IV. SOLVING LARGE SCALE PROBLEMS

The  $\ell_1$ -regularized LSP (5) is readily solved for small to medium sized problems through any one of a variety of existing solvers (most of which are available online under the GNU Public License): CVX [10], [11], MOSEK [12], 11-magic [7], and LASSO [13], [14] to name a few. For example CVX can handle problems with up to a few thousand reflection coefficients.

Here we consider using yet another solver, `11_ls` [6]. This Matlab based solver uses a truncated Newton interior-point

method that computes search directions with a preconditioned conjugate gradient algorithm [5]. Through these techniques, `l1_ls` allows us to solve our particular estimation problem for a large number of reflection coefficients ( $n = 100000$  or more) by taking advantage of algorithms that efficiently compute convolution.

### A. Implementation

The `l1_ls` algorithm solves the general  $\ell_1$ -regularized LSP problem:

$$\text{minimize } \|Ax - y\|_2^2 + \hat{\lambda}\|x\|_1, \quad (11)$$

with variable  $x \in \mathbf{R}^n$ , given the observations  $y \in \mathbf{R}^m$ , and data matrix  $A \in \mathbf{R}^{m \times n}$ . Clearly, this handles the estimation problem (5) we are interested in with  $x = \mu$ ,  $y = V_r$ ,  $\hat{\lambda} = 2\lambda\sigma^2$ , and  $A = V$ . Note that  $A$  is an  $n \times n$  convolution matrix entirely determined by the input interrogation signal  $V_i(k)$ .

Conveniently, the `l1_ls` routine allows one to overload matrix multiplication by  $A$  and  $A^T$  (by creating a new `Matlab` object), when there is a more efficient way of performing the calculation. This is important because the cost of solving (11), via `l1_ls`, is dominated by the cost of performing matrix vector multiplies by  $A$  and  $A^T$ , which is up to order  $n^2$  floating point operations (flops). However, it is often possible to achieve a substantial improvement by exploiting the structure in  $A$ . For our estimation problem, multiplication by  $A$  computes convolution, and multiplication by  $A^T$  computes correlation. As we will review in the next section, both of these operations are performed efficiently with the FFT in order  $n \log(n)$  flops.

### B. Fast Convolution

This section reviews how the FFT algorithm is used to efficiently compute the convolution needed for our problem.

We start by defining the *circulant Toeplitz* matrix  $C(r)$  as:

$$C(r) = \begin{bmatrix} r_0 & r_{-1} & r_{-2} & \dots & r_{1-n} \\ r_1 & r_0 & r_{-1} & \dots & r_{2-n} \\ r_2 & r_1 & r_0 & \dots & r_{3-n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{n-1} & r_{n-2} & r_{n-3} & \dots & r_0 \end{bmatrix}, \quad (12)$$

where  $r_{-k} = r_{n-k}$ . With this definition it is not hard to see that  $Cx$  computes the circular convolution between  $r \in \mathbf{R}^n$  (the first column of  $C$ ) and a vector  $x \in \mathbf{R}^n$  in order  $n^2$  flops.

We can, however, use the FFT to compute the same product in order  $3n \log(n)$  flops, which is significantly less than  $n^2$  for any appreciable value of  $n$ . Let  $F \in \mathbf{C}^{n \times n}$  be the matrix that computes the discrete Fourier transform of a vector in  $\mathbf{R}^n$ , with inverse  $F^H$ , the complex conjugate transpose of  $F$ . Using the fact that the Fourier transform converts convolution in the time domain into multiplication in the frequency domain, we have:

$$y = Cx = F^H \text{diag}(Fr)Fx. \quad (13)$$

Thus, circular convolution is efficiently calculated via the following steps:

- 1) Use the FFT to compute  $u = Fx$  and  $v = Fr$  (order  $2n \log(n)$ ).

- 2) Perform an element by element multiply between  $u$  and  $v$  (order  $n$ ).
- 3) Compute  $y$  by taking the inverse FFT of the result from step 2 (order  $n \log(n)$ ).

Note that we never actually form the matrices  $F$  or  $\text{diag}(Fr)$  in this process. Furthermore, we also get an efficient method for computing  $C^T x$ , by simply noting that from equation (13) we have  $C = F^H \text{diag}(Fr)F$ . Thus,

$$C^H = C^T = F^H \text{diag}(\overline{Fr})F. \quad (14)$$

Therefore, to compute  $C^T x$ , the same process enumerated above is used, except in step 2 we multiply by the complex conjugate of  $v$ .

To implement the causal (rather than circular) convolution version of our problem we simply use zero padding. Specifically, we construct  $C(r) \in \mathbf{R}^{2n \times 2n}$  by setting  $r = (V_i, \mathbf{0})$ , where  $\mathbf{0} \in \mathbf{R}^n$  is a vector with all zero elements. Thus, the causal part of the convolution (this is  $Ax$  with respect to the `l1_ls` algorithm) is just the first  $n$  elements of  $C\hat{x}$ , where  $\hat{x} = (x, \mathbf{0})$ . The same idea holds for  $C^T \hat{x}$ .

Finally, we note for some specific input signals  $V_i$ , it is even possible to implement faster convolution than with the FFT. A trivial example is  $V_i(k) = \delta(k)$ , where  $\delta(k)$  is the discrete delta function. In this case, we do not have to perform a convolution at all. Another example is  $V_i(k) = u(k)$ , where  $u(k)$  is a discrete step function. It is not hard to see that convolution with this function can be computed in order  $n$  flops.

### C. Performance Example

In this section we compare the performance between `CVX` and `l1_ls` (using efficient FFT convolution). To do this we solved our estimation problem (5) for increasing values of the problem size  $n$ , and clocked the time taken to find the optimal solution (on a 1.8GHz Intel Core Duo processor under 64-bit Linux).

The measurement data was generated by the same method presented in §III-B. This data was then used to estimate the reflection coefficients  $\mu$ , with  $\lambda\sigma^2 = 1/2$ , for the different solvers. Figure 5 shows the dramatic improvement obtained by the `l1_ls` solver for increasing values of  $n$ . We note that the solution time for the `l1_ls` solver tends to vary, depending on  $V_i$ , and the actual number of nonzero reflection coefficients (this behavior is not expected of `CVX`). For the test cases we tried, this variance was on the order of minutes for the larger values of  $n$ . However, in general, the `l1_ls` method always performed much better than `CVX`.

### D. Real TDR Data Example

This section presents one final example using real TDR data, collected from a one meter long sample of aircraft cable with and without chaffing damage (a one centimeter section of the cable where the shielding is rubbed away). The goal, obviously, is to use the reflected TDR signal to detect the presence of the chaff. Note, in this example the initial assumptions we started with generally do not hold (*i.e.*, the

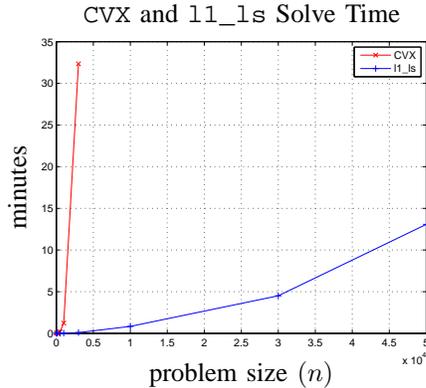


Fig. 5. Comparison between CVX and l1\_ls for solving large-scale reflection coefficient estimation problems.

cable is not distortionless, and the source and load impedance are not even close to matched). Nevertheless, the reflection coefficient estimation method works, allowing us to detect and locate a very subtle type of damage. The detection succeeds without using any additional processing or baseline information (e.g., like subtracting the TDR response of the undamaged wire). That is important because in many applications baseline information is unreliable. Our efficient l1\_ls based solver was used to find the optimal reflection coefficient estimate. The results are plotted in Figure 6, along with an example that uses Tikhonov regularization (as explained in §III-A) for comparison.

## V. CONCLUSION

In this paper we have described a method for detecting and locating wiring damage using TDR measurement data. Unlike some other (more general) methods, this one uniquely takes advantage of the fact that faults are often sparsely located along the length of the wire. We demonstrated the effectiveness of our method on a simulated example, and showed how Monte Carlo simulation might be used to tune it (by selecting  $\lambda$ ) to achieve specific detection goals (like a certain false positive error rate). In addition, we saw that preexisting algorithms, like l1\_ls, can be adapted to efficiently solve large-scale (high resolution) versions of our estimation problem. Finally, we applied the method to actual TDR data and revealed its impressive ability to identify a very subtle type of damage. It is hoped the fault detection method presented here will serve as a straightforward improvement to existing techniques that is readily put into practice.

## FUTURE WORK

The largest improvements in the domain of TDR based wire fault detection will perhaps come through more refined physics based models for signal propagation through general wire types (not just the lossless kind considered here). This important research is being actively pursued by at least a couple of communities, and is expected to yield increasingly effective fault detection methods, some of which are quite different from the one presented here.

For this work, physical models accounting for both propagation loss and the fact that faults in real systems tend to

reflect derivatives of the input signal can be derived from first principles (Maxwell's Equations), and readily incorporated in to the approach described here. This is expected to significantly improve the performance of this method. In addition, there are processing techniques that further extend the one presented here, to yield optimal estimates with even better sparsity characteristics, while simultaneously finding an effective  $\lambda\sigma^2$  product (rather than requiring it to be specified or tuned ahead of time) [8], [15].

## ACKNOWLEDGMENT

The author would like to thank Prof. S. Boyd for his guidance in developing many of the ideas presented in this paper as a project component of his EE364b Convex Optimization II course, at Stanford University. Many thanks to Dr. K. Wheeler, Dr. D. Timucin, and Dr. M. Kowalski at NASA Ames Research Center for their leadership and help through many enlightening discussions (and also for getting the author involved with this project).

## REFERENCES

- [1] J. M. Mendel and F. Habibi-Ashrafi, "A survey of approaches to solving inverse problems for lossless layered media systems," *IEEE Trans. Geosci. Remote Sens.*, vol. GE-18, no. 4, pp. 320–330, October 1980.
- [2] A. Bruckstein and T. Kailath, "Inverse scattering for discrete transmission-line models," *SIAM Review*, vol. 29, no. 3, pp. 359–389, September 1987.
- [3] J. Lundstedt and S. Strom, "Simultaneous reconstruction of two parameters from the transient response of a nonuniform LCRG transmission line," *Journal of Electromagnetic Waves and Applications*, vol. 10, no. 1, pp. 19–50, 1996.
- [4] P. Smith, C. Furse, and J. Gunther, "Analysis of spread spectrum time domain reflectometry for wire fault location," *IEEE Sensors J.*, vol. 5, no. 6, pp. 1469–1478, December 2005.
- [5] S. Kim *et al.*, "An interior-point method for large-scale l1-regularized least squares," *Journal of Machine Learning Research*, vol. 8, pp. 1519–1555, July 2007.
- [6] K. Koh, S. Kim, and S. Boyd. (2008) l1\_ls: Simple matlab solver for l1-regularized least squares problems (web page and software). [Online]. Available: [http://stanford.edu/~boyd/l1\\_ls](http://stanford.edu/~boyd/l1_ls)
- [7] E. Candes and J. Romberg. (2006) l1-magic: A collection of MATLAB routines for solving the convex optimization programs central to compressive sampling. [Online]. Available: [www.acm.caltech.edu/l1magic](http://www.acm.caltech.edu/l1magic)
- [8] E. Candes, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, December 2007.
- [9] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.
- [10] M. Grant and S. Boyd. (2008) CVX: Matlab software for disciplined convex programming (web page and software). [Online]. Available: <http://stanford.edu/~boyd/cvx>
- [11] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura, Eds. New York, NY: Springer, 2008, pp. 95–110. [Online]. Available: [http://stanford.edu/~boyd/graph\\_dcp.html](http://stanford.edu/~boyd/graph_dcp.html)
- [12] MOSEK ApS. (2008) The MOSEK optimization tools manual version 5.0. [Online]. Available: [www.mosek.com](http://www.mosek.com)
- [13] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal. Statist. Soc. B.*, vol. 58, no. 1, pp. 267–288, 1996.
- [14] B. Efron, I. Johnstone, T. Hastie, and R. Tibshirani. (2003) LARS software for Splus and R. [Online]. Available: <http://www-stat.stanford.edu/~tibs/lasso.html>
- [15] Y. Lin and D. D. Lee, "Bayesian l1-norm sparse learning," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'06)*, 2006, pp. V605–V608.

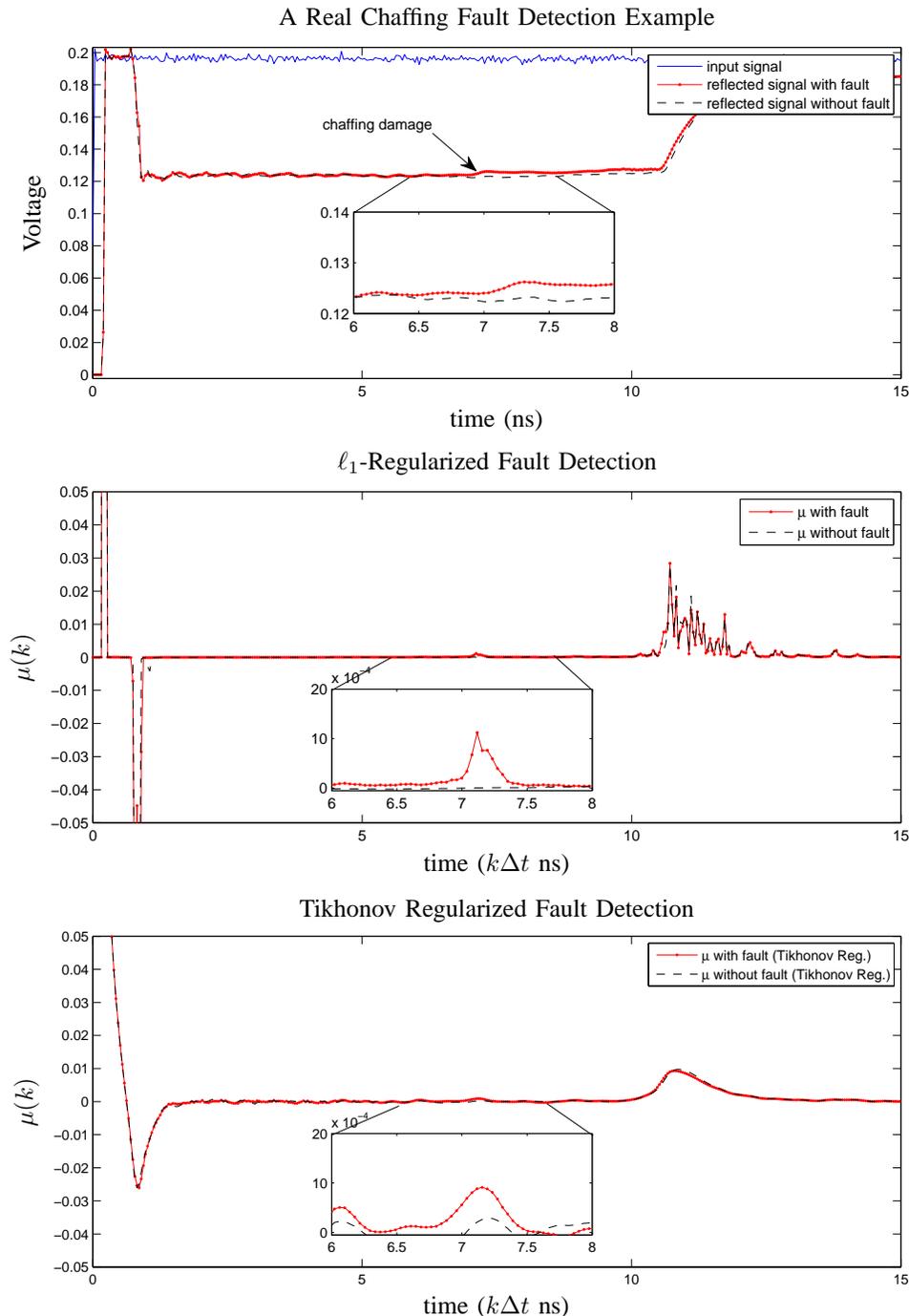


Fig. 6. (Top) Measured input, and reflected voltage waves, recorded with a digital TDR unit (Agilent 54754A). (Middle) Reflection coefficient  $\mu(k)$  estimation results using  $\ell_1$ -regularized least squares, with  $n = 1024$ ,  $\lambda\sigma^2 = 0.005$ , and  $\Delta t = 0.04$  ns (the entire recorded signal is not shown). The large reflection coefficients to the left and right of the chaffing fault are caused by the mismatched load and source impedance of the wire connectors. (Bottom) Reflection coefficient  $\mu(k)$  estimation results using Tikhonov regularized least squares as presented in §III-A. Note, in this case the faulted region is not nearly as apparent.

**Stefan Schuet** is a research and development engineer at NASA Ames Research Center in Moffett Field, California. He received his B.S. and M.S. degree in electrical engineering from Santa Clara University in 2001 and 2004 respectively. Over the last decade he has worked in the field of electronic sensors and instruments to create novel hardware and software signal processing systems. Projects include the 3D imaging of CEV/Orion thermal protection materials, automated Space Shuttle tile defect inspection and Mars rover instrument placement. He is presently working in the Intelligent Systems Division to research and develop advanced applications for optimization, statistical signal processing, and Bayesian inference.