[19] Codes for Variable Projection method dnsg.f and nlsr.f, available in NetLib.

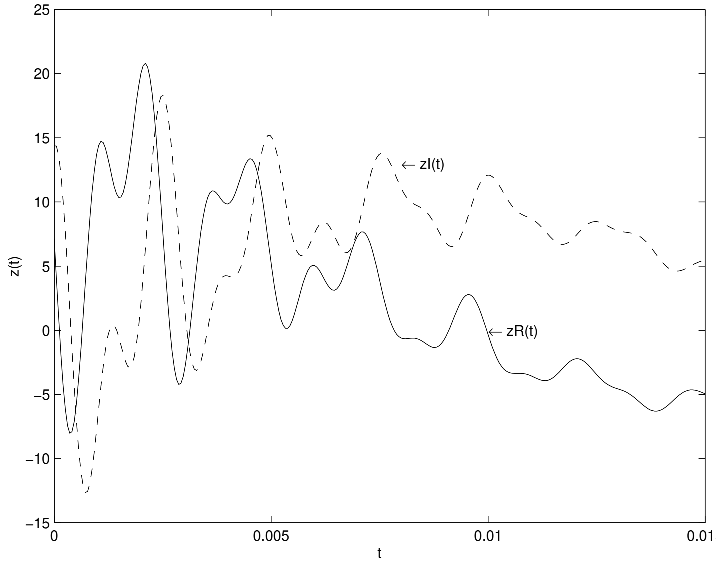


FIG. 3. *Correct signal* $z_c(t)$ *for sum of 7 complex exponentials test problem. Real and imaginary parts.* $0 \leq t \leq 0.015$.

Wiley, 1979.

[2] Å. Björck. *Numerical Methods for Least Squares Problems*, Chapter 9. SIAM, Philadelphia, PA, 1996.

[3] H. Chen, S. Van Huffel, D. Ormondt, and R. deBeer. Parameter estimation with prior knowledge of known signal poles for the quantification of NMR spectroscopy data in the time domain. *J. Magnetic Resonance*, Series A 119, 225–234, 1996.

[4] R. Fletcher. *Practical Methods of Optimization*, 2nd Ed. John Wiley, New York, 1987.

[5] G.H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM J. Numer. Anal.*,10:413–432, 1973.

[6] L. Kaufman. A variable projection method for solving separable nonlinear least squares problems. *BIT*, 15:49–57, 1975.

[7] S.M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice-Hall, 1993.

[8] L. Ljung. *System Identification Theory for the User.* Prentice-Hall, 1987.

[9] K. Jittorntrum and M. R. Osborne. Strong uniqueness and second order convergence of nonlinear discrete approximation. *Numer. Math.*, 34:439–455, 1980.

[10] J.M. Mendel. *Lessons in Estimation Theory for Signal Processing, Communications, and Control.* Prentice-Hall, 1995.

[11] J.B. Rosen, H. Park, and J. Glick. Total least norm formulation and solution for structured problems. *SIAM Journal on Matrix Anal. Appl.*, 17-1:110-128, 1996.

[12] J.B. Rosen, H. Park, and J. Glick. Total least norm for linear and nonlinear structured problems, Recent advances in total least squares techniques and errors-in-variables modeling. Ed. S. Van Huffel, SIAM, pp. 203-214, 1997.

[13] J.B. Rosen, H. Park, and J. Glick. Structured total least norm for nonlinear problems. *SIAM Journal on Matrix Anal. Appl.*, 20-1:14–30, 1998.

[14] J. B. Rosen, H. Park, J. Glick, and L. Zhang. Accurate solution to overdetermined linear equations with errors, using L_1 norm minimization. UMSI 98/161, University of Minnesota, 1998. To be published in in *Computational Optimization and Applications*, 2000.

[15] L.L. Scharf. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis.* Addison-Wesley, 1991.

[16] H.W. Sorenson. *Parameter Estimation: Principles and Problems.* Marcel Dekker, 1980.

[17] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem, Computational Aspects and Analysis*, SIAM, Philadelphia, 1991.

[18] S. Van Huffel, H. Park, and J. B. Rosen. Formulation and solution of structured total least norm problems for parameter estimation. *IEEE Trans. on Signal Processing*, 44-10:2464–2474, 1996.

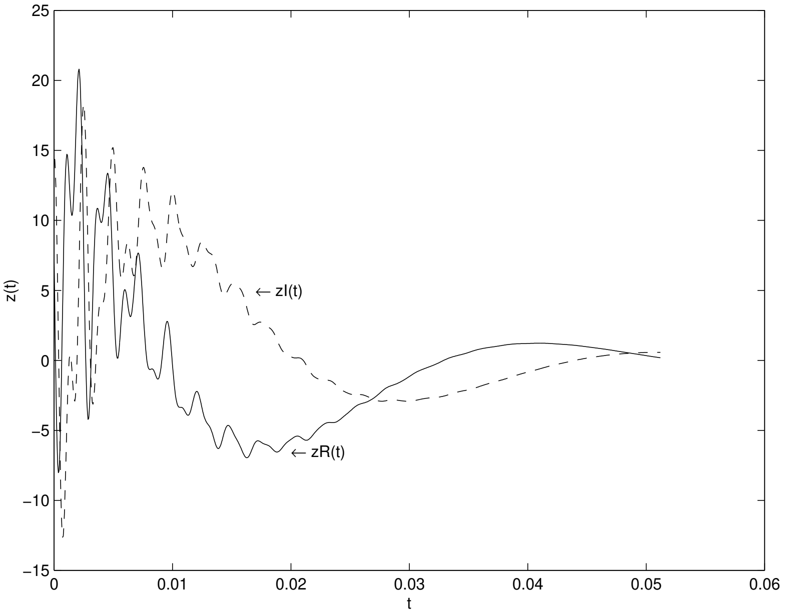


FIG. 2. *Correct signal $z_c(t)$ for sum of 7 complex exponentials test problem. Real and imaginary parts.* $0 \leq t \leq 0.06$.

for the other 4 sets, with $\delta = 0.01$ were essentially the same as given in Table 2. An additional set of 10 problems, with $\delta = 0.001$ was run to determine the effect of changing the magnitude of the 25 larger errors. The final relative errors for SNTLN1 were essentially unchanged, but those for VarPro decreased by approximately a factor of 10. Additional tests showed that the final relative errors for SNTLN1 are essentially independent of δ, while those for VarPro are approximately proportional to δ. The maximum number of iterations required by SNTLN1 for any problem was 9. The largest final relative error in z for any problem was RE_z = 6.1e-7.

The most significant conclusion is that SNTLN1 is able to identify the correct signal, with final errors about 1000 times smaller than VarPro, using the identical data. In fact, the final errors in SNTLN1 are determined by the small ($< 10^{-6}$) data errors, and are essentially independent of the 25 larger errors. This is shown clearly by the fact that the SNTLN1 solution is changed only slightly by a change in δ. In contrast, the VarPro solution final error is determined primarily by the largest data errors, and will increase in proportion to δ, as δ is increased. This will also be true for any approximation method based on L_2 norm minimization.

It is also significant that SNTLN1 is able to reduce the relative error from an initial value of about 10% to about 3e-7, in an average of 6.1 iterations. This demonstrates the rapid convergence rate of the SNTLN1 algorithm for this class of problems. The robustness of the algorithm is also shown by the relatively small range for final errors (from min to max) in Table 2, for the 10 different problems solved. This robust performance was observed in all test problems solved.

REFERENCES

[1] M.S. Bazaraa and C.M. Shetty. *Nonlinear Programming Theory and Algorithms*, Chapter 7. John

	Initial			Final: VarPro			Final: SNTLN1		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
RE_x	3.74e-1	6.50e-1	1.35	7.84e-3	4.75e-2	1.83e-1	2.73e-6	2.39e-5	1.27e-4
RE_d	4.14e-2	5.39e-2	7.0e-2	9.67e-4	4.14e-3	9.21e-3	5.94e-7	2.47e-6	7.95e-6
RE_f	1.44e-2	1.82e-2	2.52e-2	1.82e-4	4.60e-4	1.18e-3	6.05e-8	1.99e-7	4.39e-7
RE_z	7.44e-2	1.32e-1	3.16e-1	4.24e-4	6.88e-4	8.27e-4	2.06e-7	3.09e-7	4.29e-7
# Iters.	-	-	-	6	8.1	16	5	6.1	7

TABLE 2

Relative Errors in x, f, d, and z. Comparison of SNTLN1 and VarPro. Results for 10 runs. $\delta = 0.01$.

For the purpose of the test problems we used the following values: $m = 128$, $n = 7$, $\Delta t = 0.0004$, and $t_{max} = 0.0512$. For SNTLN1, bounds were imposed on d and f:

$$\begin{array}{l} d_{lj} \le d_j \le d_{uj} \\ f_{lj} \le f_j \le f_{uj} \end{array} \qquad j = 1, \ldots, 7,$$

where

$$\begin{aligned} d_l &= (40, 40, 130, 100, 160, 190, 240), \\ d_u &= (65, 65, 160, 130, 190, 220, 280), \\ f_l &= (8, 18, 32, 120, 370, 530, 790), \\ f_u &= (13, 28, 42, 140, 400, 560, 825). \end{aligned}$$

The complex errors ϵ_i at each data point consisted of two parts. One was a random (uniformly distributed) error with magnitude $\le 10^{-6}$, at every data point. The second part was an error $\delta z(t_i)$, added to 25 randomly selected data points. Two values of δ, (0.001 and 0.01) were used, each in different runs.

Each additional problem solved was different. The correct values x_c, d_c, and f_c were generated for each problem by selecting a random value within the bounds. For the $x_{cj} = x_{Rj} + \sqrt{-1} x_{Ij}$, each x_{Rj} and x_{Ij} were randomly selected in the range $[-10, 10]$. These test problems were chosen so as to be similar to the NMR signal data problem described in [3].

Since each problem was different, we used the same initial values:

$$\begin{aligned} d_{init} &= \tfrac{1}{2}[d_l + d_u] \\ f_{init} &= \tfrac{1}{2}[f_l + f_u] \end{aligned}$$

for each problem.

An illustration of the correct signal $z_c(t)$ is shown in Figures 2 and 3.

A total of 5 different sets of problems were solved using SNTLN1, with 10 different correct signals for each set. For comparison, VarPro was also used to solve 2 of these sets, using the same 10 correct signals as for SNTLN1. This was done with $\delta = 0.01$ and $\delta = 0.001$.

Typical results for 10 different problems with $\delta = 0.01$ are summarized in Table 2. The initial errors are listed in the first 3 columns, showing the minimum, average, and maximum for each relative error, over the 10 different correct signals. The next 3 columns show the same information for the final relative errors obtained by VarPro. The last 3 columns show the corresponding final relative errors obtained by SNTLN1. The number of iterations needed by VarPro and SNTLN1 is also summarized. Results

# of Large Errors	Percentage $FRE \le 10^{-10}$		Percentage $FRE \ge 0.01$		Average # itns	
	SNTLN1	VP	SNTLN1	VP	SNTLN1	VP
0	100	28	0	72	7.1	27
1	100	0	0	100	6.8	-
2	100	0	0	100	7.4	-
5	95	-	5	-	6.9	-
10	95	-	5	-	6.9	-
20	60	-	40	-	6.9	-
25	30	-	70	-	7.2	-

TABLE 1

Effect of Large Errors in Data Vector b on Parameter Estimate Accuracy. Sum of Gaussians Test Problem.

4^{th} and 5^{th} columns give the percentage of cases for which $FRE \ge 0.01$, and the last two columns give the average number of iterations each algorithm required.

The remarkable ability of SNTLN1 to obtain the correct parameter and amplitude vectors in spite of large errors in the data and initial estimates, is clearly shown. This is due both to the use of the L_1 norm and the imposed bounds. It is seen in Table 1 that even with large errors in the data at 10 of the 60 data points, the correct vectors were obtained in 95% of the cases tested.

In contrast, the VarPro algorithm was only able to recover correct vectors in 28% of the cases, even with no errors in the data. For these cases, the only errors were in the initial parameter estimates. It was observed that the cause of difficulty with VarPro was that two of the α_i would tend toward the same value (since there are no bounds to prevent this). This leads to a very ill-conditioned matrix $A(\alpha)$, and a corresponding very large vector x . Furthermore, as soon as even one large data error was present, VarPro produced errors equal, or greater than, the data error. These results clearly show the great advantage of SNTLN1 when only some of the data points are in error. We will now describe results for a more realistic situation, where there may be at least some small error in every data point.

4.2. Complex Exponential Test Problem. In this section, we present test results for a difficult complex signal identification problem which represents realistic data obtained in typical applications. The measured data at each time point consists of the true signal plus some error. That is, ε_i consists of a small random error for each i, and much larger errors at some specified number of randomly selected values of i. The results presented again compare SNTLN1 and VarPro. The true signal used is given by:

$$
\begin{aligned}
z(t) &= z_R(t) + \sqrt{-1} z_I(t) \\
&= \textstyle\sum_{j=1}^{7} (x_{Rj} + \sqrt{-1} x_{Ij}) e^{(-d_j + 2\pi\sqrt{-1} f_j)t}, \qquad t \in [0, 0.0512].
\end{aligned} \tag{31}
$$

The correct signal is therefore specified by 14 parameters, and 14 amplitudes, $d_j, f_j, x_{Rj}, x_{Ij}, j = 1, \ldots, 7$. The measured data is given by:

$$
b_i = z(t_i) + \varepsilon_{Ri} + \sqrt{-1}\varepsilon_{Ii}, \qquad i = 1, \ldots, 128. \tag{32}
$$

The values of t_i chosen were with a fixed interval size, so that $t_i = i\Delta t$, with $\Delta t = 0.0004$, although a nonuniform set of t_i could equally well have been used. The complex signal identification problem is to determine the correct values of d_j, f_j, x_{Rj} and x_{Ij}, given the b_i and some initial estimates of the d_j and f_j.

true signal $z(t)$ is given by

$$z(t) = \sum_{j=1}^{6} x_j e^{-(t-\alpha_j)^2/\sigma^2}, \qquad t \in [0,\, 1.2]. \tag{30}$$

The signal is measured at 60 uniformly spaced time points, $t_i = i\Delta t$, $\Delta t = .02$, $i = 1, \ldots, 60$. The value of σ was assumed known, with $\sigma^2 = 0.05$, and the bounds

$$(0.09, 0.27, 0.45, 0.78, 0.91, 0.95) \leq \alpha^T \leq (0.11, 0.33, 0.55, 0.90, 0.94, 1.05)$$

were imposed. A plot of $z(t)$, as given by (30) is shown in Figure 1.

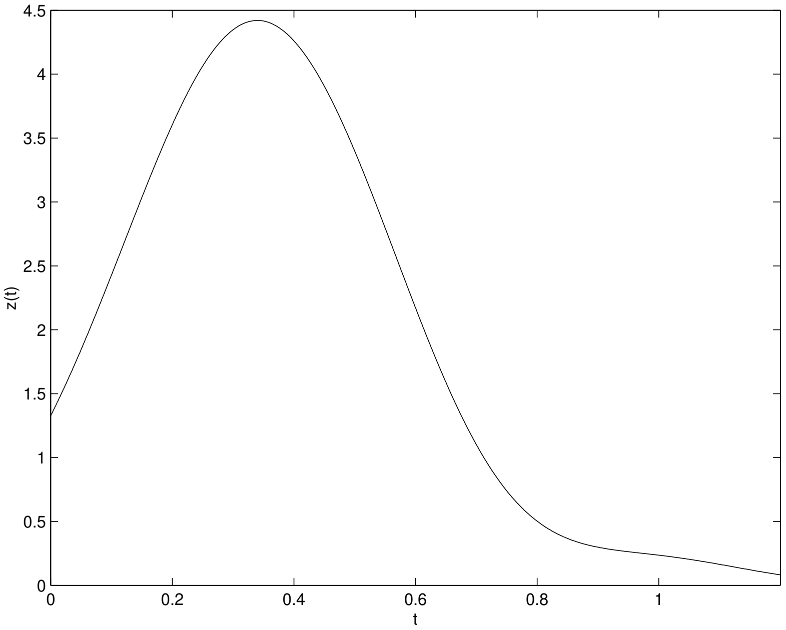


FIG. 1. *Sum of Gaussians test problem. True signal* $z(t)$

For this plot, the following values of α and x were used:

$$\alpha_c = (0.1, 0.3, 0.5, 0.87, 0.92, 0.96),$$
$$x_c = (0.1, 3.0, 2.0, 0.25, -0.5, 0.5).$$

For each test problem, a specified number of large errors were introduced at randomly selected points t_i, with $\varepsilon_i \leq 0.1|z(t_i)|$. The remaining ε_i were all kept at zero. The number of large errors ranged from 0 to 25. For each specified number of large errors, 20 different cases were computed. A total of 140 test problems were run using SNTLN1. The results are summarized in Table 1. This table gives the Final Relative Error (FRE), defined by $FRE = 0.5[FRE_\alpha + FRE_x]$, where $FRE_\beta = \|\beta_A - \beta_c\|/\|\beta_c\|$, and $\beta = \alpha$ or x.

The first column gives the number of large errors introduced. The next two columns give the percentage of cases for which the $FRE \leq 10^{-10}$, for SNTLN1 and VarPro. The size of the errors FRE_α and FRE_x were approximately the same. The

In order to preserve the linearity of Step 2a in the LN algorithm when using the L_1 norm with a complex vector, we define the corresponding L_1 norm:

$$\|r\|_1 = \| \begin{pmatrix} r_R \\ r_I \end{pmatrix} \|_1. \tag{27}$$

This, of course, doubles the number of rows in the corresponding linear program. Thus the problem

$$\min_{\alpha, x_R, x_I} \|r(\alpha, x_R, x_I)\|_1 \tag{28}$$

requires, at each major iteration, the solution of the following linear program:

$$\begin{aligned} \min_{\Delta\alpha, \Delta x_R, \Delta x_I, \gamma_R, \gamma_I} \quad & \sum_{i=1}^{m} (\gamma_{Ri} + \gamma_{Ii}) \\ \text{subject to} \quad & \\ -\gamma_R \le {} & A_R(\alpha)\Delta x_R - A_I(\alpha)\Delta x_I \\ & +[J_R(\alpha, x_R) - J_I(\alpha, x_I)]\Delta\alpha + r_R(\alpha, x_R, x_I) \le \gamma_R \\ -\gamma_I \le {} & A_I(\alpha)\Delta x_R + A_R(\alpha)\Delta x_I \\ & +[J_R(\alpha, x_I) - J_I(\alpha, x_R)]\Delta\alpha + r_I(\alpha, x_R, x_I) \le \gamma_I \\ & \alpha_l \le \alpha + \Delta\alpha \le \alpha_u \end{aligned} \tag{29}$$

where $J_R(\alpha, x)$ and $J_I(\alpha, x)$ are the Jacobians with respect to α, of $A_R(\alpha)x$ and $A_I(\alpha)x$, respectively. This linear program has $2m + 4n$ variables ($\Delta\alpha$, Δx_R, Δx_I, γ_R, γ_I) and $4(m + n)$ inequality constraints. For efficiency of solution, this problem is considered the dual problem (with more inequality constraints than variables), and the equivalent primal problem is solved.

4. Computational Results. In this section we summarize the results for two types of test problems. These results were obtained as described in the previous section, using both the SNTLN1 algorithm and a version of VarPro available in NetLib [19].

For each type of test problem, lower and upper bounds (15) were specified. For each individual problem a random vector α_c with $\alpha_l \le \alpha_c \le \alpha_u$, was generated. A random value for each amplitude x_{cj}, with $-10 \le x_{cj} \le 10$ was then generated. Using these in (13), the true signal $z(t)$ was then computed at m points t_i, $i = 1, \cdots, m$. The data vector b, with elements b_i as given by (14) was then computed for this test problem. Thus each individual test problem is different, with different values of α_c and x_c. The initial estimate for the parameter vector was always chosen as $\alpha_{initial} = \frac{1}{2}(\alpha_l + \alpha_u)$. This method of constructing test problems corresponds to a typical application when only bounds on the parameter vector are known, and the (unknown) correct values α_c and x_c are to be determined. Note that the lower and upper bound information is used by both VarPro and SNTLN1. For SNTLN1 the bounds are explicitly imposed, and they are used to determine good initial values for the parameter vector. For VarPro the bounds are used only to determine good initial values, since VarPro does not impose explicit bounds on the parameter values.

4.1. Sum of Gaussians Test Problem. We first summarize the computational results obtained when the true signal is a sum of 6 Gaussian functions. That is the

$$\begin{aligned} RE_x &= \|x_A - x_c\|/\|x_c\|, \\ RE_\alpha &= \|\alpha_A - \alpha_c\|/\|\alpha_c\|, \\ RE_z &= \|z_A - z_c\|/\|z_c\|. \end{aligned} \tag{18}$$

As summarized in Section 1, we attempt to minimize these errors by solving

$$\min_{\alpha,x} \|r(\alpha, x)\| \tag{19}$$

where $r(\alpha, x)$ is given by (2). When the L_1 norm is used the SNTLN1 algorithm consists of solving a linear program at each major iteration, as given by Step 2a of the LN algorithm. The bounds on the α_i are easily enforced in each of these linear program solutions, so that they are satisfied by the final parameter vector α_A.

Two types of test problem in the form (13) were used to study the performance of SNTLN1 and VarPro. These are given by a sum of Gaussians:

$$\varphi_j(\alpha, t) = e^{-(t-\alpha_j)^2/\sigma^2} \tag{20}$$

and a sum of complex exponentials:

$$\varphi_j(\alpha, t) = e^{(-d_j + 2\pi\sqrt{-1}f_j)t} \tag{21}$$

where the d_j are damping factors and the f_j are frequencies. For this case the parameter vector $\alpha \in \mathbf{R}^s$ is given by

$$\alpha^T = (d_1, f_1, d_2, f_2, \cdots, d_n, f_n) \tag{22}$$

with $s = 2n$. Also for this case the amplitudes x_i are complex, so that:

$$x_j = x_{Rj} + \sqrt{-1}x_{Ij}, \quad j = 1, \cdots, n. \tag{23}$$

It follows that the true signal, as given by (13), is a complex function of t :

$$z(t) = z_R(t) + \sqrt{-1}z_I(t). \tag{24}$$

The measured data vector b, as given by (14), is also now complex:

$$\begin{aligned} b &= b_R + \sqrt{-1}b_I \in \mathbf{C}^m \\ b_{Ri} &= z_R(t_i) + \epsilon_{Ri} \\ b_{Ii} &= z_I(t_i) + \epsilon_{Ii} \end{aligned} \tag{25}$$

Finally, the real and complex parts of the residual error are given by:

$$\begin{aligned} r_R(\alpha, x_R, x_I) &= A_R(\alpha)x_R - A_I(\alpha)x_I - b_R \\ r_I(\alpha, x_R, x_I) &= A_I(\alpha)x_R + A_R(\alpha)x_I + b_I \end{aligned} \tag{26}$$

We note that the L_2 norm of the complex residual $r = r_R + \sqrt{-1}r_I$ is given by:

$$\|r\|_2^2 = r^T\bar{r} = r_R^T r_R + r_I^T r_I = \| \begin{pmatrix} r_R \\ r_I \end{pmatrix} \|_2^2.$$

The convergence of the sequence $\{y_k\}$ to a stationary point of $||f(y)||$ is shown by applying a standard convergence theorem [1]. The theorem applies since the sequence $\{y_k\}$ is contained in the compact subset Ω, and the mapping $M : y_k \rightarrow y_{k+1}$ is closed on this subset. Furthermore, as shown above, we have a descent function $||f(y_k)||$, with a strict decrease in its value, if y_k is not a stationary point. With these conditions satisfied, Theorem 7.2.3 [1] applies and $\{y_k\} \rightarrow y^*$, a stationary point of $||f(y)||$.

The LN algorithm using the L_2 norm can also be shown to be equivalent to the Gauss-Newton method with a line search. For $||f(y^*)||_2$ small, and $J(y)$ full rank, the convergence rate will be superlinear [4]. As shown by the computational results in Section 4, the convergence rate using the L_1 norm also appears to be superlinear when $||f(y^*)||_1$ is small, and $J(y)$ is full rank.

3. Construction of Test Problems and L_1 Norm Solution. In this section we describe the construction of realistic computational test problems for which the correct parameter vector α_c and amplitude vector x_c are known. This information is of course not used in the test of the algorithm, but allows us to investigate the ability of the algorithms to recover the correct α and x in spite of errors in the data and in the initial vector α.

A signal $z(t)$ is assumed to be given as a weighted sum of n known functions $\varphi_j(\alpha, t)$, so that:

$$z(t) = \sum_{j=1}^{n} x_j \varphi_j(\alpha, t), \quad t \in [0,\ t_{max}] \tag{13}$$

The correct signal, $z_c(t)$, with no error, is given by (13) with $x = x_c$ and $\alpha = \alpha_c$. We also define a vector $z_c \in C^m$, $z_c^T = (z_c(t_1), \cdots, z_c(t_m))$, which represents the correct signal at the m data points. The measured data vector b is the correct signal at m points t_i, $i = 1, \cdots, m$, $0 < t_i < t_{i+1} \leq t_{max}$, plus a possible error ϵ_i at each t_i :

$$b_i = z_c(t_i) + \epsilon_i, \quad i = 1, \cdots, m > n + s. \tag{14}$$

It is also assumed that lower and upper bounds are known for each parameter:

$$\alpha_{lj} \leq \alpha_j \leq \alpha_{uj}, \quad j = 1, \cdots, s, \tag{15}$$

which are satisfied by the parameter vector α_c.

Let b_c denote the true signal vector, with $b_{ci} = z_c(t_i)$, and define the $m \times n$ matrix $A(\alpha)$, with

$$A_{ij}(\alpha) = \varphi_j(\alpha, t_i). \tag{16}$$

Then

$$A(\alpha_c) x_c = b_c. \tag{17}$$

Given the data vector b, with error, and the bounds (15), the SI problem is to attempt to recover the correct vectors α_c and x_c, that is, to recover the correct signal with minimum error. Let α_A and x_A be the approximate values obtained by the SI solution, and $z_A(t)$ the corresponding approximation given by (13). We again define a vector $z_A \in C^m$, representing the computed approximation to the measured signal at the data points. We measure the performance of an algorithm on a test problem in terms of the relative errors:

```
                    Algorithm LN
Input – $f(y)$, Jacobian matrix $J(y)$, initial estimate $y_0$, $tol$
Output – $y$, $f(y)$, $||f(y)||$, $\delta$
    1. Set $y := y_0$
    2. repeat
        (a) $\underset{\Delta y}{\text{minimize}} ||f(y) + J(y)\Delta y||$
        (b) Set $\delta := ||f(y)|| - ||f(y) + J(y)\Delta y||$
        (c) $\underset{0<\theta\leq 1}{\text{minimize}} ||f(y + \theta\Delta y)||$
        (d) Set $y := y + \theta\Delta y$
        (e) Compute $f(y)$, $J(y)$, $||f(y)||$
       until $\delta \leq tol$
```

for some $\theta_k > 0$.

First we note that the function $||b + Ax||$ is a convex function of x, since for any x_1, x_2,

$$\left\|A\left(\frac{x_1+x_2}{2}\right)+b\right\| = \left\|\frac{1}{2}(Ax_1+b)+\frac{1}{2}(Ax_2+b)\right\| \leq \frac{1}{2}||(Ax_1+b)|| + \frac{1}{2}||(Ax_2+b)||.$$

Using the assumption that $f(y)$ has bounded second partial derivatives, there is a positive constant μ, such that

$$||f(y_k + \Delta y)|| \leq ||f(y_k) + J(y_k)\Delta y|| + \mu||\Delta y||^2. \tag{9}$$

Define the function:

$$\varphi_k(\theta) = ||f(y_k) + \theta J(y_k)\Delta y_k||, \qquad 0 \leq \theta \leq 1. \tag{10}$$

Since $||f(y_k) + J(y_k)\Delta y||$ is a convex function of Δy, $\varphi_k(\theta)$ is convex in θ. Therefore, for $0 \leq \theta \leq 1$,

$$\begin{aligned} \varphi_k(\theta) &\leq \varphi_k(0) - \theta[\varphi_k(0) - \varphi_k(1)] \\ &= \varphi_k(0) - \theta\delta_k. \end{aligned} \tag{11}$$

From (9), with $\Delta y = \theta\Delta y_k$, we have

$$\begin{aligned} ||f(y_k + \theta\Delta y_k)|| &\leq \varphi_k(\theta) + \mu\theta^2||\Delta y_k||^2 \\ &\leq \varphi_k(0) - \theta\delta_k + \mu\theta^2||\Delta y_k||^2. \end{aligned} \tag{12}$$

Choose $\theta_k = \min\{1, \delta_k/(2\mu||\Delta y_k||^2)\} > 0$. Then $y_{k+1} = y_k + \theta_k\Delta y_k$, and

$$||f(y_{k+1})|| \leq ||f(y_k)|| - \theta_k\delta_k + \frac{1}{2}\theta_k\delta_k,$$

which gives (8). The actual decrease in $||f(y)||$, as given by Step (2c), will always be at least as much as the decrease in (8) if an accurate line search is done. However, the Algorithm LN will still converge with an approximate line search, provided only that the value of θ_k satisfies $0 < \theta_k \leq \min\{1, \delta_k/(2\mu||\Delta y_k||^2)\}$. In the applications (Section 4) it was found that the convergence rate was not very sensitive to the accuracy of the approximate line search used.

We summarize and compare results obtained using the SNTLN algorithm with the L_1 norm(SNTLN1), and using VarPro, for the same set of test problems. These computational results show clearly the benefit of using the L_1 norm, and its robust performance when data includes some larger errors. Specifically, the errors in the approximation obtained by the SNTLN1 algorithm are independent of the set of largest errors in the data vector b, and depend primarily on the set of smallest errors in b. This is in contrast to any method which minimizes the residual in the L_2 norm (such as VarPro), where the errors in the approximation are proportional to the set of largest errors in b. For a complete theoretical and computational analysis of the performance of L_1 norm minimization of the residual error for the overdetermined linear system $Ax \approx b$, see [14]. It is shown there that if no more than k rows of $[A\ b]$ contain errors (which may be large), and the other rows are error-free, then the correct solution is guaranteed if $(m-n)/n \geq 2k/\sigma$, where $\sigma > 0$, is a lower bound of singular values related to A. It is also shown there that if in addition, the other rows contain small errors bounded by δ, then the error in the solution will be $O(\delta)$.

2. Convergence of SNTLN Algorithm. We will now show the convergence of the SNTLN algorithm to a stationary point of the function being minimized. We consider a somewhat more general problem, which includes (4) as a special case. Specifically we consider the following Least Norm (LN) problem:

$$\min \|f(y)\| \tag{5}$$

where $f : \Re^n \to \Re^m$, $m \geq n$, and

$$\|\cdot\| = \|\cdot\|_p, \qquad p = 1, 2 \text{ or } \infty.$$

Also y_0 is a known initial estimate, and the level set $\Omega = \{y \in R^n : \|f(y)\| \leq \|f(y_0)\|\}$ is bounded. We assume that for $y \in \Omega$, the function $f(y)$ has bounded second partial derivatives. We denote by $J(y)$ the $m \times n$ Jacobian matrix of $f(y)$. For certain results we require that $J(y)$ be full rank, for $y \in \Omega$. The LN problem (5) includes (4) by setting

$$y = \begin{pmatrix} \alpha \\ x \end{pmatrix}, \qquad f(y) = \begin{pmatrix} A(\alpha)x - b \\ D(\alpha - \hat{\alpha}) \end{pmatrix}. \tag{6}$$

A stationary point y^* of $\|f(y)\|$ is defined by:

$$\min_{\Delta y} \|f(y^*) + J(y^*)\Delta y\| = \|f(y^*)\|. \tag{7}$$

For the 2-norm, this is equivalent to $J^T(y^*)f(y^*) = 0$. The algorithm for solving LN problem is summarized in Algorithm LN.

Starting with y_0, Algorithm LN will generate a sequence of points

$$y_{k+1} = y_k + \theta_k \Delta y_k, \qquad k = 0, 1, \ldots,$$

and corresponding δ_k given by Step (2b), with $y_{k+1} \in \Omega$. We now show that for any $tol > 0$ and y_k not a stationary point ($\delta_k > tol$), there is a strict decrease in $\|f(y)\|$. That is:

$$\|f(y_{k+1})\| \leq \|f(y_k)\| - \frac{1}{2}\theta_k \delta_k, \tag{8}$$

Most of these methods [2, 16] use the two norm, so that minimizing the residual norm becomes a nonlinear least squares problem:

$$\min_{\alpha,x} \|r(\alpha,x)\|_2^2 = \min_{\alpha,x} r^T r. \tag{3}$$

The special structure of r is used to advantage in "Separable Nonlinear Least Squares" [5, 2]. This approach has been implemented in the "Variable Projection" (VarPro) method [6, 2, 19], and this software is available for computational testing and comparison. The new Structured Nonlinear Total Least Norm(SNTLN) algorithm for solving this problem in any of the norms L_1, L_2 orL_∞, has recently been proposed [12, 13]. SNTLN is a generalization to nonlinear problems of the earlier structured Total Least Norm (STLN) algorithm [11, 18]. If the total number of variables and parameters is no greater than the number of equations, $n+s \le m$, then minimizing the residual $r(\alpha,x)$ will usually give unique values of α and x. However, in certain types of problem (for example, where $A(\alpha)$ has Toeplitz or Hankel structure), it may be that $n+s \ge m$. In that case we need to solve the augmented problem:

$$\min_{\alpha,x} \left\| \begin{matrix} A(\alpha)x - b \\ D(\alpha - \hat{\alpha}) \end{matrix} \right\|_p, \qquad p = 1, 2, \infty \tag{4}$$

where D is a positive diagonal weighting matrix and $\hat{\alpha}$ is an initial estimate for α. For details, see [11, 13]. When $A(\alpha)$ is affine in $\alpha, \hat{\alpha} = 0$, and D is properly chosen, the SNTLN algorithm with the L_2 norm gives the same solution as Total Least Squares [17]. It has also been shown [13] that the SNTLN algorithm using the L_2 norm is equivalent to the Gauss-Newton method.

In this paper, we give a convergence proof for the SNTLN algorithm which is valid for all three norms. For the purposes of this proof a slight modification of the original SNTLN algorithm is required. Specifically, a simple line search is added at each major iteration in order to insure a strict decrease in the function value. However, our computational experience with the SNTLN algorithm shows that in practice the convergence is usually not adversely affected by using the algorithm without a line search. Convergence of a "generalized Gauss-Newton" method has been investigated in [9]. This method is similar to the Least Norm algorithm described in the next section, but does not include a line search. It is shown in [9] that with additional assumptions on the nature of the stationary point, a second order convergence rate to the stationary point is obtained. In the next section we give a simple constructive convergence proof of the Least Norm algorithm (valid for the L_1, L_2, and L_∞ norms) to a stationary point, without any additional assumptions about its nature. The proof shows convergence from any initial point, to some stationary point.

In the next section we formulate the Least Norm(LN) problem, which includes (4) as a special case. We give the LN algorithm, including the line search, and then prove that with appropriate assumptions, the LN algorithm converges to a stationary point. In Section 3, we formulate the Signal Identification(SI) problem in a form most useful for computational testing and comparison. We also give the formulation and linear programming method for the L_1 solution of SI problems where the signal consists of a sum of complex exponentials. In Section 4, we present the computational results for two different SI problems:

1. Signals represented by a sum of Gaussians,
2. Signals represented by a sum of complex exponentials.

SIGNAL IDENTIFICATION USING A LEAST L_1 NORM ALGORITHM

J. B. ROSEN *, HAESUN PARK †, AND JOHN GLICK ‡

Abstract. In many important applications a signal consists of a sum of exponential terms. The signal is measured at a discrete set of points in time, with possible errors in the measurements. The Signal Identification (SI) problem is to recover the correct exponents and amplitudes from the noisy data. An algorithm (SNTLN) has been developed which can be used to solve the SI problem by minimizing the residual error in the L_1 norm. In this paper the convergence of the SNTLN algorithm is shown, and computational results for two different types of signal are presented, one of which is the sum of complex exponentials with complex amplitudes. For comparison, the test problems were also solved by VarPro, which is based on minimizing the L_2 norm of the residual error. It is shown that the SNTLN algorithm is very robust in recovering correct values, in spite of some large errors in the measured data and the initial estimates of the exponents. For the test problems solved, the errors in the exponents and amplitudes obtained by SNTLN1 were essentially independent of the largest errors in the measured data, while the corresponding errors in the VarPro solutions were proportional to these largest data errors.

EDICS number: SP 3.9.3 – System Identification, Time-Varying Systems.

Corresponding author – J. Ben Rosen, Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093, phone: (619) 534-5932, fax: (619) 534-7029, email: jbrosen@cs.ucsd.edu.

1. Introduction. Many important applications in parameter estimation, signal processing, signal identification [16, 10, 7, 15, 8], and structured approximation can be represented in the form:

$$A(\alpha)x \approx b. \tag{1}$$

The matrix $A(\alpha)$ is $m \times n$, with $m \geq n$, and each of its elements is a differentiable function of the parameter vector $\alpha \in R^s$. Typically, the matrix $A(\alpha)$ has a special structure (such as Toeplitz or Vandermonde), or is sparse. The data vector $b \in R^m$, is often subject to noise or errors. This may consist of small random errors in every element, and possibly larger errors in a few elements.

It is assumed that there are correct values of α_c and x_c, and a corresponding error free data vector b_c, such that $A(\alpha_c)x_c = b_c$. Some a priori information about the parameter vector α is usually available. A common situation is that realistic lower and upper bounds on the individual parameters α_i are known. It is desired to recover good approximations to α_c and x_c given the noisy data vector $b = b_c + \varepsilon$, where ε represents error in b.

Essentially all methods proposed for solving this problem are based on minimizing the norm of the residual error:

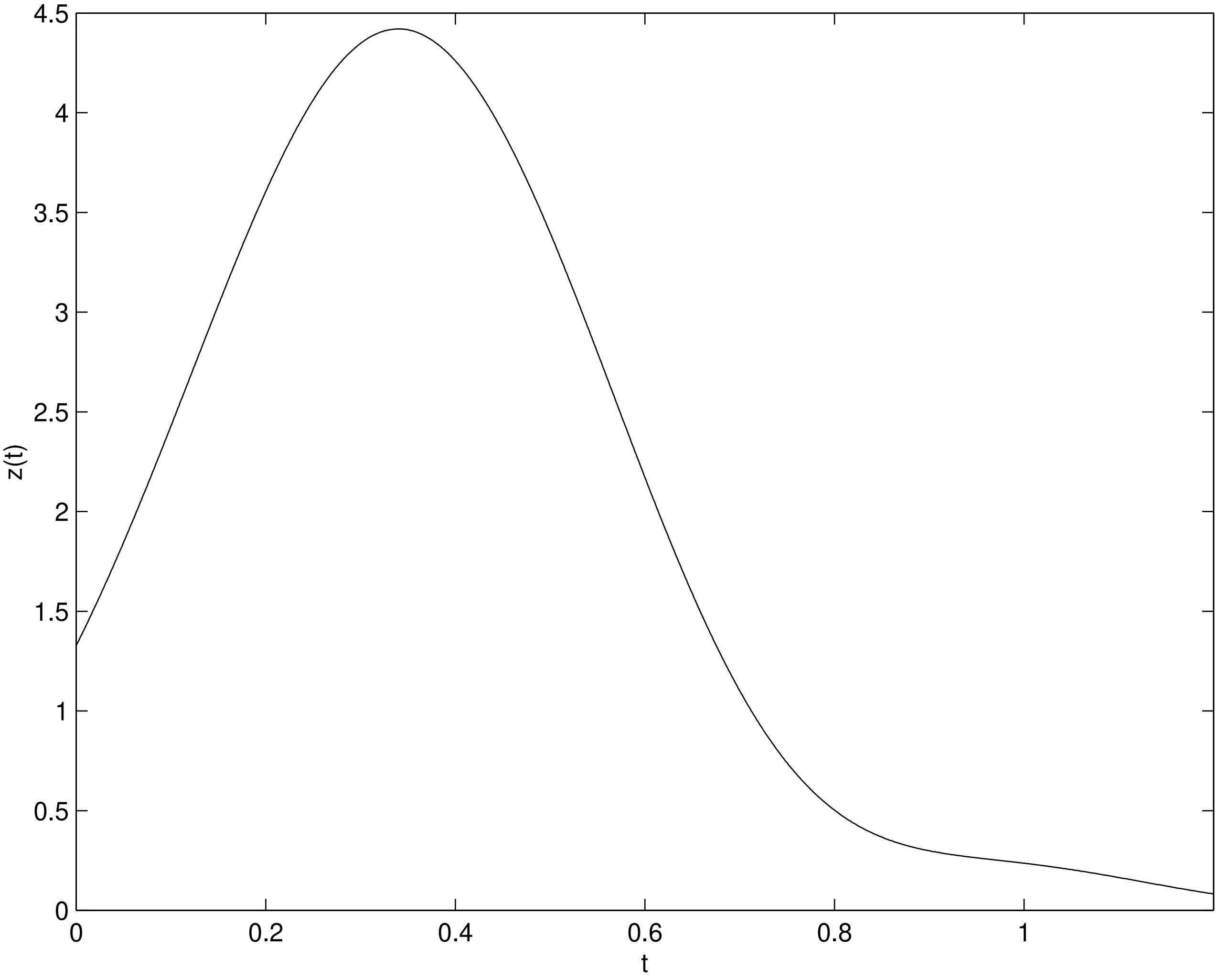
$$r(\alpha, x) = A(\alpha)x - b. \tag{2}$$

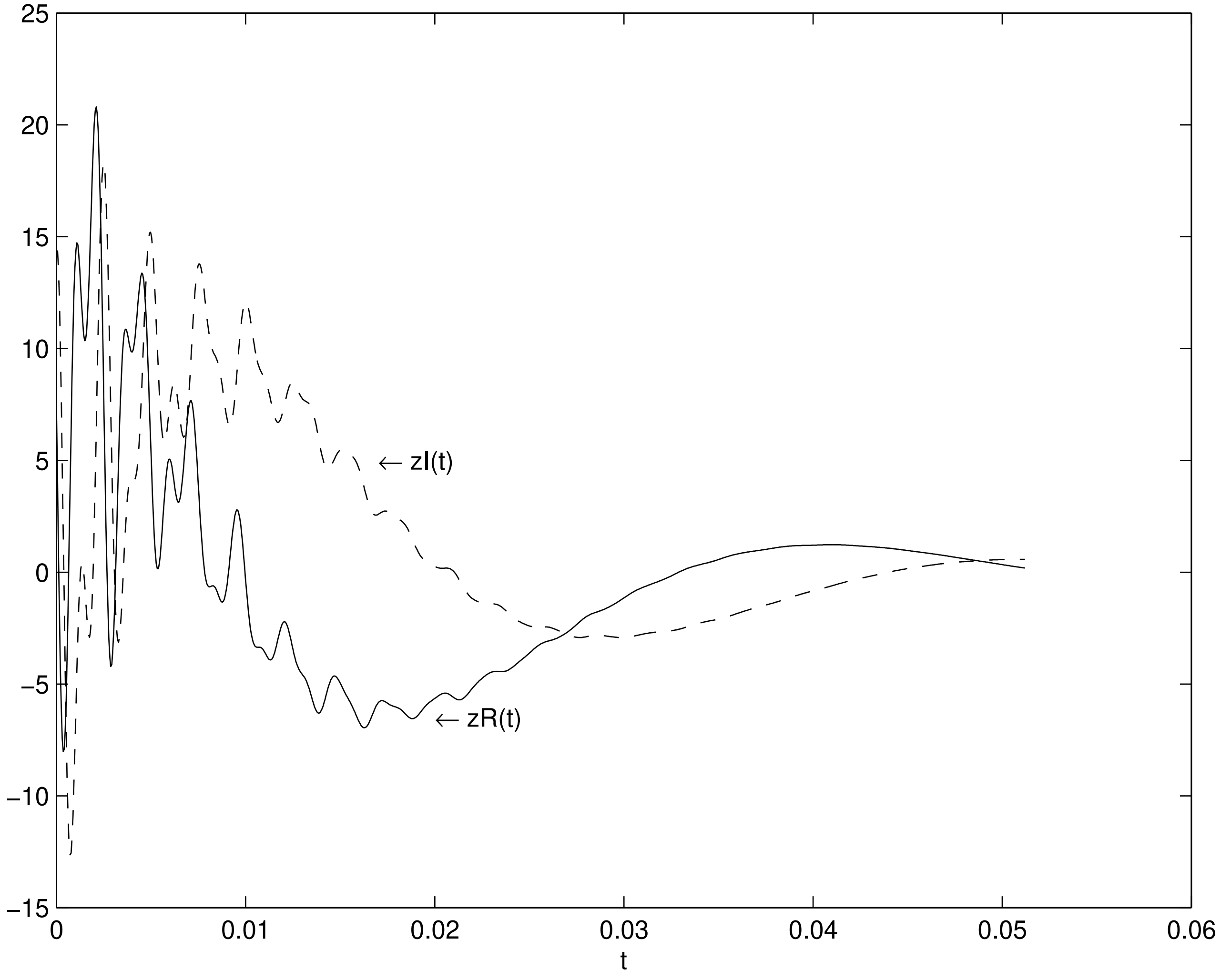
---

[*] Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, and Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093 (`jbrosen@cs.ucsd.edu`). The work of this author was supported in part by the National Science Foundation grant CCR-9509085. This research was also supported by the Minnesota Supercomputer Institute.

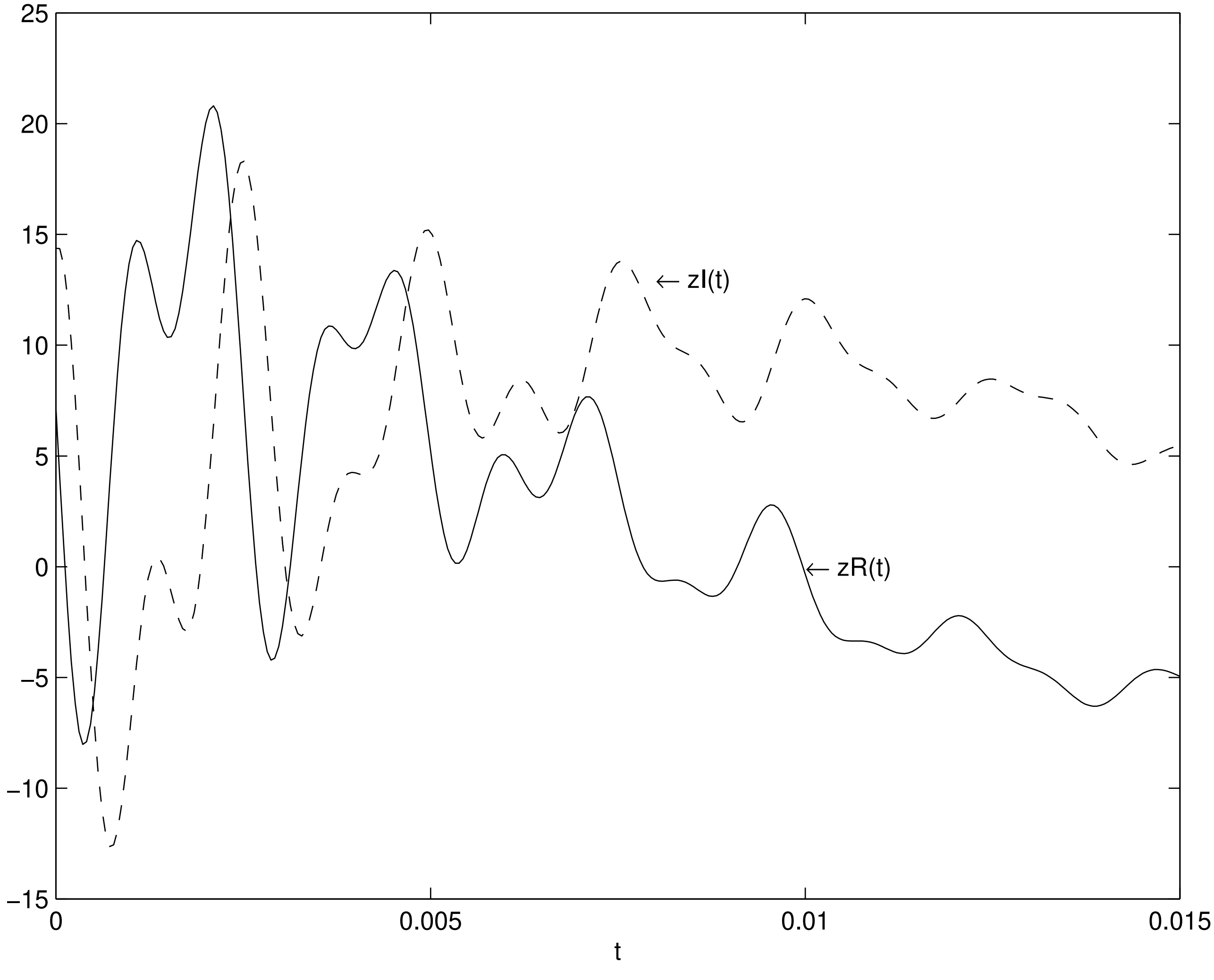
[†] Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 (`hpark@cs.umn.edu`). The work of this author was supported in part by the National Science Foundation grant CCR-9509085.

[‡] Department of Mathematics and Computer Science, University of San Diego, San Diego, CA 92110 (`glick@acusd.edu`).


4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
z(t)
0
0.2
0.4
0.6
0.8
1
t



25
20
15
10
5
0
−5
−10
−15
z(t)
0
0.01
0.02
0.03
0.04
0.05
0.06
t
← zI(t)
← zR(t)



25
20
15
10
5
0
−5
−10
−15
z(t)
0
0.005
0.01
0.015
t
← zI(t)
← zR(t)