1989

Evaluation of system identification algorithms for aspect-independent radar target classification.

Larison, Peter David.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

EVALUATION OF SYSTEM IDENTIFICATION
ALGORITHMS FOR ASPECT-INDEPENDENT
RADAR TARGET CLASSIFICATION

by

Peter David Larison

December 1989

Thesis Advisor: Michael Morgan

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 62	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification) EVALUATION OF SYSTEM IDENTIFICATION ALGORITHMS FOR ASPECT-INDEPENDENT RADAR TARGET CLASSIFICATION

12 PERSONAL AUTHOR(S)
LARISON, Peter David

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1989 December	159

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflec the official policy or position of the Department of Defense or the US Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Prony's Method; Kumaresan-Tufts Algorithm; Cadzow-Solomon Algorithm

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

A radar target, acting as a scatterer of an incident electromagnetic wave, can be considered as a linear time-invariant system. Previous work has shown that the target's pole locations are independent of the incident electromagnetic excitation, including incident wave shape, aspect and polarization. This thesis develops the Kumaresan-Tufts and Cadzow-Solomon signal processing algorithms into computer routines and evaluates their pole extraction performance. Data used to evaluate the extraction algorithms includes synthetic and integral equation generated signals with additive noise, in addition to measurements of scattering by scale models made in an anechoic chamber.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL MORGAN, Michael	22b TELEPHONE (Include Area Code) 408-646-2677 22c OFFICE SYMBOL 62Mw

DD Form 1473, JUN 86 *Previous editions are obsolete*

S/N 0102-LF-014-6603

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Approved for public release; distribution is unlimited

EVALUATION OF SYSTEM IDENTIFICATION ALGORITHMS FOR ASPECT-INDEPENDENT RADAR TARGET CLASSIFICATION

by

Peter David Larison
Captain, United States Marine Corps
B.S., Xavier University, 1981

Submitted in partial fulfillment of requirements
for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1989

ABSTRACT

A radar target, acting as a scatterer of an incident electromagnetic wave, can be considered as a linear time-invariant system. Previous work has shown that the target's pole locations are independent of the incident electromagnetic excitation, including incident wave shape, aspect and polarization. This thesis develops the Kumaresan-Tufts and Cadzow-Solomon signal processing algorithms into computer routines and evaluates their pole extraction performance. Data used to evaluate the extraction algorithms includes synthetic and integral equation generated signals with additive noise, in addition to measurements of scattering by scale models made in an anechoic chamber.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. THE PROBLEM 2

B. BACKGROUND 3

C. HISTORY 8

II. POLE EXTRACTION ALGORITHMS 9

A. PREVIOUS WORK 9

1. Direct Minimization 9
2. Prony's Method 11

B. KUMARESAN-TUFTS ALGORITHM 12

1. Equations 13
2. Singular Value Decomposition 14
3. Bias Compensation 16
4. Kumaresan and Tufts Compensation . . 16
5. Compensation Based on Eigenvalue Shifting Theorem 17
6. Performance 19
   a. Synthetically Generated Data . . 19
      1. Noise Performance 19
   b. Thin Wire Integral Equation Generated 29
   c. Scale Model Measurements 38
      1. Wire Targets 50
      2. Aircraft Models 50

C. CADZOW-SOLOMON ALGORITHM 63

1. Applicability 68
2. Equations 68
3. Excess Poles and Noise Removal . . . 69
4. Singular Value Decomposition 70
5. Bias Compensation in the Cadzow-Solomon Formulation 70
6. Performance 72
   a. Synthetically Generated Data . . 72
      1. Noise Performance 72
   b. Thin Wire Integral Equation Generated Data 80
   c. Scale Models 80
      1. Wire Targets 92
      2. Model Aircraft 92

III. SUMMARIES AND CONCLUSIONS 108

A. KUMARESAN-TUFTS 108

B. CADZOW-SOLOMON 111

C. CONCLUSIONS 112

APPENDIX A. THE KUMARESAN-TUFTS POLE EXTRACTION ALGORITHM 113

APPENDIX B. THE CADZOW-SOLOMON POLE EXTRACTION ALGORITHM 125

APPENDIX C. MATRIX MULTIPLICATION 140

APPENDIX D. GRAPHICS ROUTINE 141

LIST OF REFERENCES 145

INITIAL DISTRIBUTION LIST 147

LIST OF FIGURES

Figure 1. Signal Containing two S-Plane Poles, 90.0 dB SNR 21

Figure 2. Kumaresan-Tufts Poles, Synthetic Data, 90.0 dB SNR 22

Figure 3. Kumaresan-Tufts Poles, Synthetic Data, 30.0 dB SNR 23

Figure 4. Kumaresan-Tufts Poles, Synthetic Data, 20.0 dB SNR 24

Figure 5. Kumaresan-Tufts Poles, Synthetic Data, 10.0 dB SNR 25

Figure 6. Kumaresan-Tufts Poles, Synthetic Data, 7.0 dB SNR 26

Figure 7. Signal Containing Two S-Plane Poles, 7.0 dB SNR 27

Figure 8. Kumaresan-Tufts Pole Extraction, 7.0 dB SNR 28

Figure 9. Double Gaussian Pulse 30

Figure 10. Integral Equation Thin Wire Scattering, 30 Degree Aspect 32

Figure 11. Integral Equation Thin Wire Scattering, 45 Degree Aspect 33

Figure 12. Integral Equation Thin Wire Scattering, 60 Degree Aspect 34

Figure 13. Integral Equation Thin Wire Scattering, 90 Degree Aspect 35

Figure 14. Kumaresan-Tufts Poles, Noiseless Thin Wire Data 36

Figure 15. Integral Equation Thin Wire Scattering, 20.0 dB SNR, 45 Degree Aspect 37

Figure 16. Kumaresan-Tufts Poles, 20.0 dB SNR . . 39

Figure 17. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 30 Degree Aspect 40

Figure 18. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 45 Degree Aspect 41

Figure 19. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 60 Degree Aspect 42

Figure 20. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 90 Degree Aspect 43

Figure 21. Integral Equation Thin Wire Scattering, 7.0 dB SNR, 45 Degree Aspect 44

Figure 22. Kumaresan-Tufts Poles, 7.0 dB SNR . . . 45

Figure 23. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 30 Degree Aspect 46

Figure 24. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 45 Degree Aspect 47

Figure 25. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 60 Degree Aspect 48

Figure 26. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 90 Degree Aspect 49

Figure 27. Measured Thin Wire Scattering, 30 Degree Aspect 51

Figure 28. Measured Thin Wire Scattering, 45 Degree Aspect 52

Figure 29. Measured Thin Wire Scattering, 60 Degree Aspect 53

Figure 30. Measured Thin Wire Scattering, 90 Degree Aspect 54

Figure 31. Kumaresan-Tufts Poles, Measured Thin Wire 55

Figure 32. Thin Wire Comparison, Measured vs. Integral Equation 56

Figure 33. Target 1 Scattering, 30 Degrees from Nose on 57

Figure 34. Target 1 Scattering, Nose on 58

Figure 35. Target 2 Scattering, 30 Degrees from Nose on 59

Figure 36. Target 2 Scattering, Nose on 60

Figure 37. Kumaresan-Tufts Poles, Target 1 61

Figure 38. Kumaresan-Tufts Poles, Target 1 63

Figure 39. Kumaresan-Tufts Poles, Target 1 64

Figure 40. Kumaresan-Tufts Poles, Target 2 65

Figure 41. Kumaresan-Tufts Poles, Target 2 66

Figure 42. Kumaresan-Tufts Poles, Target 2 67

Figure 43. Signal Containing Two S-Plane Poles, 90.0 dB SNR 73

Figure 44. Cadzow-Solomon Poles, Synthetic Data, 90.0 dB SNR 74

Figure 45. Cadzow-Solomon Poles, Synthetic Data, 30.0 dB SNR 75

Figure 46. Cadzow-Solomon Poles, Synthetic Data, 20.0 dB SNR 76

Figure 47. Cadzow-Solomon Poles, Synthetic Data, 10.0 dB SNR 77

Figure 48. Cadzow-Solomon Poles, Synthetic Data, 7.0 dB SNR 78

Figure 49. Cadzow-Solomon Poles, Noiseless Thin Wire Data 81

Figure 50. Cadzow-Solomon Poles, 20.0 dB SNR . . . 82

Figure 51. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 30 Degree Aspect 83

Figure 52. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 45 Degree Aspect 84

Figure 53. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 60 Degree Aspect 85

Figure 54. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 90 Degree Aspect 86

Figure 55. Cadzow-Solomon Poles, 7.0 dB SNR . . . 87

Figure 56. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 30 Degree Aspect 88

Figure 57. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 45 Degree Aspect 89

Figure 58. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 60 Degree Aspect 90

Figure 59. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 90 Degree Aspect 91

Figure 60. Cadzow-Solomon Poles, Measured Thin Wire 93

Figure 61. Thin Wire Comparison, Measured vs. Integral Equation 94

Figure 62. Cadzow-Solomon Poles Target 1, Three Aspects 95

Figure 63. Cadzow-Solomon Poles Target 1, Three Aspects 96

Figure 64. Cadzow-Solomon Poles Target 1, All Six Aspects 97

Figure 65. Cadzow-Solomon Poles Target 2, Three Aspects 98

Figure 66. Cadzow-Solomon Poles Target 2, Three Aspects 99

Figure 67. Cadzow-Solomon Poles Target 2, All Six Aspects 100

Figure 68. Pole Comparisons, Target 1, All Six Aspects 101

Figure 69. Pole Comparisons, Target 2, All Six Aspects 102

Figure 70. Target 3 Scattering, Nose-on 104

Figure 71. Target 4 Scattering, Nose-on 105

Figure 72. Cadzow-Solomon Pole Comparisons, 4 Targets, Nose on 106

I. INTRODUCTION

A radar target, acting as a scatterer of a specified incident electromagnetic wave, can be considered as a single input, single output, linear time-invariant (LTI) system for a fixed field observation point. The target can thus be considered as a transfer function with poles and zeros. Baum demonstrated at the Air Force Weapons Laboratory that a target's induced current response to an incident electromagnetic wave has identifiable poles determined by the composition and structural geometry of the target [1]. In 1974, Moffatt and Mains proposed that the target's scattered field pole locations are independent of the incident electromagnetic excitation, including aspect and polarization [2]. Morgan has proven theoretically that, for the case of a conducting target, the scattering response contains complex natural resonances which are independent of the incident electromagnetic excitation [3]. By determining the poles of a target's response, aspect independent target identification can be accomplished through the use of electromagnetic natural resonances.

Although the concept of radar target identification through the use of natural resonances was first proposed in 1974 by Mains and Moffatt [2], only recently have signal

processing techniques been applied to locate the poles in a radar target's response in the presence of noise. Kumaresan-Tufts [4] and Cadzow-Solomon [5] have each developed algorithms which have proven successful in the presence of noise. This thesis develops computer routines based upon these two algorithms and examines their respective performance and appropriateness using a variety of scattering data.

A. THE PROBLEM

Since the performance of signal processing methods varies under different conditions, a system employed to identify targets would possibly reach a decision based on the combined output of several signal processing methods. For example, the Kumaresan-Tufts and Cadzow-Solomon methods could be used to extract poles from the response of scale model targets. The information so gathered could be used to build a data base for comparison with data similarly obtained in actual field use. The results of this system would serve as one input to a larger system. Other methods would provide input to the system, such as the K-pulse method of Kennaugh [6] and the annihilation filter used by Dunavin [7], Morgan and Dunavin [8] and Chen [9]. As the name suggests, an annihilation filter annihilates the target's poles. A system using the annihilation filter concept would contain many such filters, each previously designed to cancel the poles of a specific

known target. In actual field use, a radar target's response would be input into each of the filters, and the target selected would be that matching the filter whose output exhibits the lowest signal energy.

A system used to identify radar targets would require the following concept of employment. First, information required by each of the sub-systems would be obtained for every target class of concern. In actual field use, this information would be compared against actual radar target responses. The system would then determine the identity of the target based on the input from each of its sub-systems.

B. BACKGROUND

Consider a perfectly conducting target illuminated by an electromagnetic field. The current induced on the surface of this target at a given point must satisfy the magnetic field integral equation (MFIE),[10]

$$\bar{J}(r,t)=2\hat{n}\times\bar{H}^{i}(\bar{r},t)+\int_{S}\int_{pv}\tilde{K}(\bar{r},\bar{r}',t)\bar{J}(r,\frac{t-|\bar{r}-\bar{r}'|}{c})dS \qquad (1)$$

where $\hat{n}$ is an outward unit vector normal to the surface of the object, $\bar{J}$ is the surface current density, $\bar{H}^{i}$ is the incident magnetic field, and $\tilde{K}$ is a Green's function dyadic. The entire equation is most easily understood as the sum of driven currents and "feedback" currents corresponding to the

cross-product term and surface integral term respectively. The term driven by the magnetic field, $2\hat{n}\times\bar{H}^{i}$, forms the physical optics portion of the total current. Physical optics describes the cross-product term as the induced current without interaction with the rest of the body. The Green's function kernel describes the current at a point on the object due to the feedback of currents from every other point on the object, as previously illuminated by the incident field. The current at each point is then summed over the surface of the object. Note that the surface integral term is of principal-value type; the integral excludes the point $\bar{r}=\bar{r}'$.

Once the incident magnetic field is no longer present, the solutions of (1) are considered the natural modes of the object. These natural modes are of the form, $J_n\exp(s_n)$. The natural resonance frequencies s_n are of the form,

$$s_n=\sigma_n+j\omega_n \tag{2}$$

where σ_n is the damping rate in Nepers/sec and ω_n is the frequency in radians/sec. The natural resonances of (2) are functions of the structural geometry of the object and are independent of the incident magnetic field. To understand how these natural resonances are unique to the geometry and composition of the object, consider a set of points on the object previously illuminated by the incident field, so that $\bar{H}^{i}=0$. The current at a given point in the set is due to the

infinite number of feedback currents from every other point in the set. Recall that these feedbacks are described by the Green's function kernel in the integral term of (1). Since the set of points previously illuminated is physically located on the same object, the infinite number of paths that connect a point with all other points in the set is the same for all points in the set. The infinite number of paths are unique to the structural geometry of the object and correspond exactly to the infinite number of paths taken by currents which feedback to a given point via the Green's function kernel. Finally, the composition of the target determines the surface current density on the object. Although an infinite number of resonances exists in any object, only a limited number of these will be measurably excited by an incident field of finite bandwidth. These resonances described in (2) appear as complex conjugate pairs in the left-half portion of the s-plane.

In the far-field, the back-scattered response of a target to an incident plane wave is of the form

$$\bar{H}^{s}(-r\hat{p},t)=\frac{1}{4\pi cr}\frac{\partial}{\partial t}\int\int_{s}\hat{p}\times\bar{J}(\bar{r}',t-\left|\bar{r}-\bar{r}'\right|/c)\,dS' \qquad (3)$$

where c is the speed of light and $\hat{p}$ is the unit vector whose direction matches that of the plane wave's propagation.

Equation (3) is the result of integrating the current at each point on the target surface for a fixed point in the far-field. Recall that the current at each point on the target is defined by (1). Thus, the back-scattered far-field can be obtained by substituting (1) into (3):

$$\bar{H}^{s}(-r\hat{p},t)=u(t-r/c)\left\{ H_{po}(-r\hat{p},t)+\sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} H_{n}(-r\hat{p},t)\exp(s_{n}t) \right\} \quad (4)$$

The currents in (1) produce the field in (4). In fact, each term in (4) corresponds to the term in (1) which produced it. Specifically, the first term in (4) describes the physical optics scattered field generated by the $2\hat{n}\times\bar{H}^{i}$ current which, of course, is the first term in (1). Similarly, the second term in (4) is produced by the source-free currents defined by the second term in (1). Like the current described in (1), the field in (4) is the sum of two terms, a driven term and a term containing feedbacks.

The results of (4) can also be seen as two forms of the Singularity Expansion Method (SEM) developed by Baum [1]. As shown by Morgan [10], during the early-time portion of the target's response, the scattered field is composed of the physical optics scattered field and a "Class 2" form of the SEM expansion. The class 2 SEM expansion corresponds to the second term of (4), wherein the coefficients H_n are time-varying as the wave passes over the target, since the currents

producing this portion of the field are integrated over a time-varying surface area. At the instant the wave passes the last point of the target, the physical optics field vanishes and the remaining term in (4) is produced by constant coefficients H_{n_c}. The coefficients H_n are constant at this instant since the surface area in the integral in (3) is now constant. This instant also marks the transition of (4) from a "class 2" SEM expansion of time-varying coefficients to a "class 1" SEM expansion of constant coefficients. The scattered field due to a plane wave is therefore composed of a physical optics term and a class 2 SEM expansion in the early-time, and a simple class 1 expansion in the late-time.

Actual measurement of the scattered far-zone field would be greatly aided by knowledge of the transition time of the field from early time to late time. From [10], this transition for a monostatic radar would occur at $\Delta t=T+2(D+d)/c$ seconds after radar turn-on. Here, T is the pulse duration, D is the target's dimension along the direction of wave propagation, d is the distance between the target and the measurement point and c is the speed of light.

The discussion presented in this section was extracted from work done by Morgan in [10]. The reader is referred to this work for a more detailed treatment of the material in this section.

C. HISTORY

The results of the previous section form the basis for the hypothesis that the natural resonances found in the scattering response of a target to an incident electromagnetic wave are unique to that target. Additionally, only a finite set of these natural resonances are measurably excited by a wave of finite bandwidth. In 1974, Moffatt and Mains proposed that the extraction of resonances from a target's response to electromagnetic excitation could be used for target identification. This work related to earlier work in 1965, when Kennaugh and Moffatt first developed the concept of a radar target as a linear time invariant system. Poles in the z-plane are directly related to the natural resonances of a target

$$z_n = e^{s_n \Delta t} \tag{5}$$

where s_n is given by (2) and Δt is the sampling interval in seconds. Hence, pole extraction involves resonance identification. The use of pole extraction algorithms is discussed in the next chapter.

II. POLE EXTRACTION ALGORITHMS

The use of pole extraction algorithms to identify radar targets is discussed in this chapter. A brief discussion of two methods precedes the in-depth evaluation of the Kumaresan-Tufts and Cadzow-Solomon algorithms. The evaluation of the latter two algorithms occurs in two stages. First, each algorithm will be evaluated in its ability to extract poles from data with known poles. Some of the data processed was generated at various signal to noise ratios by a computer program written by Morgan [11]. Additional data was produced by Morgan's time-domain thin wire integral equation computer program [12]. In the second stage, a side by side comparison is made of poles extracted by each method using transient scattering measurements for a thin wire and for various model aircraft. Comparisons between the two methods are made as the aspect of the aircraft is varied.

A. PREVIOUS WORK

1. Direct Minimization

The most direct way to determine the natural resonances in a target's response is to minimize the mean-square error between the modeled signal and the received signal. In [10], Morgan determined that the late-time target

response to a radar could be represented as a sum of damped sinusoids given by

$$\hat{y}(t)=\sum_{i=1}^{\infty} A_i e^{\sigma_i t}\cos(\omega_i t+\theta_i) \tag{6}$$

The frequency, ω_i, and damping rate, σ_i, are the same parameters found in the natural resonance defined in (2). Phase, θ_i, and amplitude, A_i, are the remaining parameters. The representation in (6) is the sum of an infinite number of resonances. The sampled response to an incident wave of finite bandwidth can be modeled as

$$\hat{y}(n\Delta t) = \hat{y}_n=\sum_{i=1}^{N} A_i e^{\sigma_i n\Delta t}\cos(\omega_i n\Delta t+\theta_i) \tag{7}$$

where Δt is the sampling interval in seconds. The four parameters of (7) must be adjusted to minimize the sampled mean-square error signal

$$e_n^2=(y_n-\hat{y}_n)^2 \tag{8}$$

between the actual discrete sampled received signal y_n and the modeled signal $\hat{Y}_n$. The processing required in this minimization problem is both inefficient and highly non-linear. Nevertheless, Chong used this method to process mathematically-generated data down to 15.0 dB signal-to-noise (SNR) ratio [13].

2. Prony's Method

As in direct minimization, Prony's approach to resonance classification focuses on the late-time portion of a radar target's response. However, linear processing and root solving are used. The late-time response is modeled as the output of an LTI system of order K_D. Each signal received at some discrete sample, n, is considered to be the weighted sum of K_D previous signals. Thus, the finite term approximation of the received late-time signal, $\hat{y}_n$, is defined by

$$y_n = \sum_{i=1}^{K_D} b_i y_{n-i} \tag{9}$$

The z-transform of (9) is

$$z^{K_D} - b_1 z^{K_D-1} - b_2 z^{K_D-1} \ldots - b_{K_D} = 0 \tag{10}$$

The roots of this polynomial in z are the poles of the system model. Therefore, the key to extracting the poles in the system's response lies in solving for the coefficients b_i of (9).

A set of K_D+M received signals in M equations (9) can be arranged in matrix form as

$$\begin{bmatrix} y_0 & \cdots & y_{K_D-1} \\ \vdots & & \vdots \\ y_{M-1} & \cdots & y_{K_D+M-2} \end{bmatrix} \begin{bmatrix} b_{K_D} \\ \vdots \\ b_1 \end{bmatrix} = \begin{bmatrix} y_{K_D} \\ \vdots \\ y_{K_D+M-1} \end{bmatrix} \tag{11}$$

In Prony's original method, the data matrix D_y is exactly determined, and the coefficient vector, b, is solved using linear computations. In the presence of noise, Prony overdetermines the data matrix by setting $M>K_D$ and solves for the coefficient vector by obtaining the least-squares solution to the system of equations.

The Prony method has two major problems. First, the poles obtained by the least squares solution to the overdetermined matrix may be strongly perturbed by noise [14], since noise does not satisfy the causal model of the system. Second, the order of the system is generally not known *a priori*. When the estimated order is greater than the actual order, poles due to noise are generated. Prony's method offers no technique for distinguishing between the signal poles and the extra poles caused by overestimation of the system's order. If the estimated system order is less than the actual order, actual poles are lost and the remaining poles are perturbed from their true positions.

B. KUMARESAN-TUFTS ALGORITHM

The Kumaresan and Tufts pole extraction algorithm was developed by adapting Prony's method to reduce the problems addressed in the preceding section. The Kumaresan-Tufts algorithm modifies the least-squares Prony method in three ways:

1. Processed signals are arranged in a data matrix based on a non-casual model of the system.

2. The model of the system is deliberately overestimated.

3. The system of equations determined by the above two criteria is solved by using singular value decomposition (SVD).

Kumaresan demonstrates in [15] that the use of singular value decomposition tends to force the extra poles of the excess-order system inside the unit circle, while the non-causal arrangement of the signals tends to force the signal poles outside the unit circle. The excess order of the system model reduces the effects of noise on the actual poles. Since the noise is stationary and stable, it looks the same in forward and backward time.

1. Equations

Recall that in (9), Prony's technique defines the received late-time signal as the weighted sum of K_D previous signals, where K_D is presumed to be the order of the system. Kumaresan models the same late-time signal as the weighted sum of K_D future signals, where K_D is greater than the estimated order of the system. This non-casual model is given by

$$y_M = \sum_{l=1}^{K_D} b_l' y_{M+K_D+1-l} \tag{12}$$

A system of M such prediction equations can be written in matrix form as

$$\begin{bmatrix} y_1 \cdots y_{K_D} \\ \vdots \quad \vdots \\ y_M \cdots y_{K_D+M-1} \end{bmatrix} \begin{bmatrix} b'_{K_D} \\ \vdots \\ b'_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_{M-1} \end{bmatrix} \tag{13}$$

Or, in matrix notation,

$$D_y \cdot b = y \tag{14}$$

As in Prony's method, the coefficients b'_l are coefficients of a polynomial in z that models the system's late-time response. Two simple manipulations of either data matrix leads to the relationship between the coefficients of the Prony model and the prediction coefficients of the Kumaresan-Tufts model. With $b_0=-1$, a prediction coefficient is related to an autoregressive coefficient by

$$b'_l = - \frac{b_{l-1}}{b_{K_D}} \tag{15}$$

From the above relationship, it can be shown that the complex pole pairs of the causal model are merely conjugate reflections across the unit circle of the pole pairs in the non-causal model.

2. Singular Value Decomposition

The non-causal arrangement of late-time signals in a set of system equations, and subsequent processing through singular value decomposition, combine to separate the signal

and noise into orthogonal spaces. As discussed in the preceding paragraph, poles of the non-causal model are reflected outside the unit circle. Kumaresan demonstrates in [15] that the extra poles of the excess-order system can be forced inside the unit circle through the use of SVD.

Singular value decomposition factors the MXK_D data matrix D_y into the product of the matrices:

$$D_y=U\Sigma V^T \tag{16}$$

The columns of U (MXM) are eigenvectors of $D_yD_y^T$ and the columns of V (K_DXK_D) are eigenvectors of $D_y^TD_y$. If r is the rank of the data matrix, D_y, the diagonal matrix Σ (MXK_D) contains r singular values which are the square roots of the nonzero eigenvalues of both $D_y^TD_y$ and $D_yD_y^T$. By rearranging the three matrices in the product, the pseudoinverse of D_y can be obtained as

$$D_y^+=V\Sigma^+U^T \tag{17}$$

where Σ^+ is a (K_DXM) matrix whose singular values on the diagonal are the reciprocals of those in the Σ matrix. Finally, the coefficient vector b^+, of minimum Euclidian norm, is given by

$$b^+=D_y^+y \tag{18}$$

The coefficient vector b^+ so obtained is the minimum length least-squares solution to (14). In other words, b^+ is the best possible solution to (14). In the case of noiseless data, the extraneous poles generated by the excess-order model will always be inside the unit circle when b^+ is used. This result is generally true for noisy data.

3. Bias Compensation

Kumaresan and Tufts [4] observed that the addition of noise perturbed the singular values of the Σ matrix of (16). If the perturbation of these singular values is not compensated, both the signal poles and extraneous poles are biased towards the unit circle. Kumaresan and Tufts used a compensation method which reduced the bias in their work, but did not derive an analytical justification. In [16], Norton derived a more valid bias compensation method based on the eigenvalue shifting theorem.

4. Kumaresan and Tufts Compensation

If the actual order of the system is K_D', then the first K_D' singular values of the Σ matrix in (16) are non-zero. The remaining K_D-K_D' singular values are considered noise singular values and are zero in the case of noiseless data. The addition of noise perturbs the first K_D signal singular values and increases the noise to some non-zero value. Kumaresan and Tufts compensated for this increase in the singular values due to the noise by subtracting the

average of the noise singular values from the signal singular values. The noise singular values were then set to zero.

5. Compensation Based on Eigenvalue Shifting Theorem

As described in the previous section, the singular values of the matrix D_y are the square roots of the eigenvalues of $D_yD_y^T$ and $D_y^TD_y$. Assume the noisy data matrix can be represented by $D_y=S+N$, where N is composed of the wide-sense stationary white noise process v_i, given by

$$N_y=\begin{bmatrix} v_1 & \cdots & v_{K_D} \\ \vdots & & \vdots \\ v_M & \cdots & v_{M+K_D} \end{bmatrix} \tag{19}$$

The expected value of $D_yD_y^T$ can be obtained by

$$D_yD_y^T=E\left[(S+N)(S+N)^T\right]=E[SS^T]+E[SN^T]+E[NS^T]+E[NN^T] \tag{20}$$

Since S is deterministic, $E[SS^T]=SS^T$. Assuming the noise is zero mean, the two cross products are zero. Because we assume the noise is wide-sense stationary and white, $E[NN^T]=\sigma_v^2I$, where σ_v^2 is the noise variance and I is the identity matrix. The expected value of $D_yD_y^T$ thus becomes

$$E\left[D_yD_y^T\right]=SS^T+\sigma_v^2I \tag{21}$$

Similarly, the expected value of $D_y^T D_y$, the other source of singular values, is

$$E[D_y^T D_y] = S^T S + \sigma_v^2 I \tag{22}$$

The assumption in the results of (21) and (22) is that the diagonals of $E[N^T N] = E[NN^T]$ equals the noise variance σ_v^2. Equations (21) and (22) show that in the mean, the squares of the singular values of D_y are increased by the noise variance.

The results lead to the method of eigenvalue compensation recommended by Norton in [16]. Recall from (16) that the eigenvalues of D_y are on the diagonal of the Σ matrix returned by the singular value decomposition of D_y. If K_D' is the actual order of the system, and K_D is the estimated order of the system then the remaining $K_D - K_D'$ singular values of the Σ matrix can be squared and averaged to obtain an estimate of the noise variance, σ_v^2. These noise singular values can then be set to zero. The first K_D singular values of the Σ matrix are then squared and reduced by subtracting the estimate of the noise variance. The square root of the difference becomes the new first K_D singular values of the compensated Σ matrix. Calculations according to (17) and (18) can then be carried out in a normal manner to obtain poles in the presence of the noise. Eigenvalue compensation requires an estimate of the actual order of the system. Methods to obtain this estimate are discussed in Chapter III.

6. Performance

The Kumaresan-Tufts algorithm was programmed in Fortran and tested on various types of data. The program appears in Appendix A.

a. Synthetically Generated Data

The starting point for evaluating the performance of the Kumaresan-Tufts algorithm was with synthetically generated data of the form given by (8) and shown here again for convenience

$$\hat{y}_n = \sum_{i=1}^{N} A_i e^{\sigma_i n\Delta t} \cos(\omega_i n\Delta t + \theta_i) \qquad (8)$$

Again, $A_i, \sigma_i, \omega_i, \theta_i$, are the amplitude, damping rate, frequency and phase of a set of N damped sinusoids. Noisy data was created by adding stationary white noise.

1. Noise Performance

The algorithm was evaluated at various SNR's, ranging from 90.0 dB to 7.0 dB. These SNR's are ratios of signal energy to noise energy rather than the ratio of signal-to-noise power. Synthetic data so generated more closely resembles the exponential decay of signal power typical in actual radar measurements.

Figure 1 shows the signal produced by two s-plane poles at 90.0 dB. Figures 2 through 6 depict the poles extracted from this signal at SNR's ranging from 90.0 dB to 7.0 dB. Obtained poles are shown at their positions within the upper right hand quadrant of the unit circle in the z-plane. Not shown are conjugates of each pole which are located below the real axis outside the figure boundaries.

Figures 2 through 6 demonstrate outstanding performance on noisy data, even at SNR's of 7.0 dB. The scaling needed to show a discernible difference between results obtained at 30.0 dB and 7.0 dB would necessarily exclude one of the poles from the enlarged figure. The average distance of the trial poles obtained in the 7.0 dB SNR signal from the true poles is on the order of 10^{-3}. This magnitude corresponds to that of the average estimate of the noise variance obtained in successive trials with this signal. The correlation between the distance of trial poles from true poles and the noise variance estimate was consistently observed with each of the different signal-to-noise ratios used. Figure 7 depicts the signal of Figure 1 severely corrupted by noise having 7.0 dB SNR.

As discussed previously, the signal-to-noise ratio used in the synthetically generated data is the ratio of energy. Figure 8 depicts the results of pole extraction from the signal shown in Figure 7, but with a late-time

Figure 1. Signal Containing two S-Plane Poles, 90.0 dB SNR

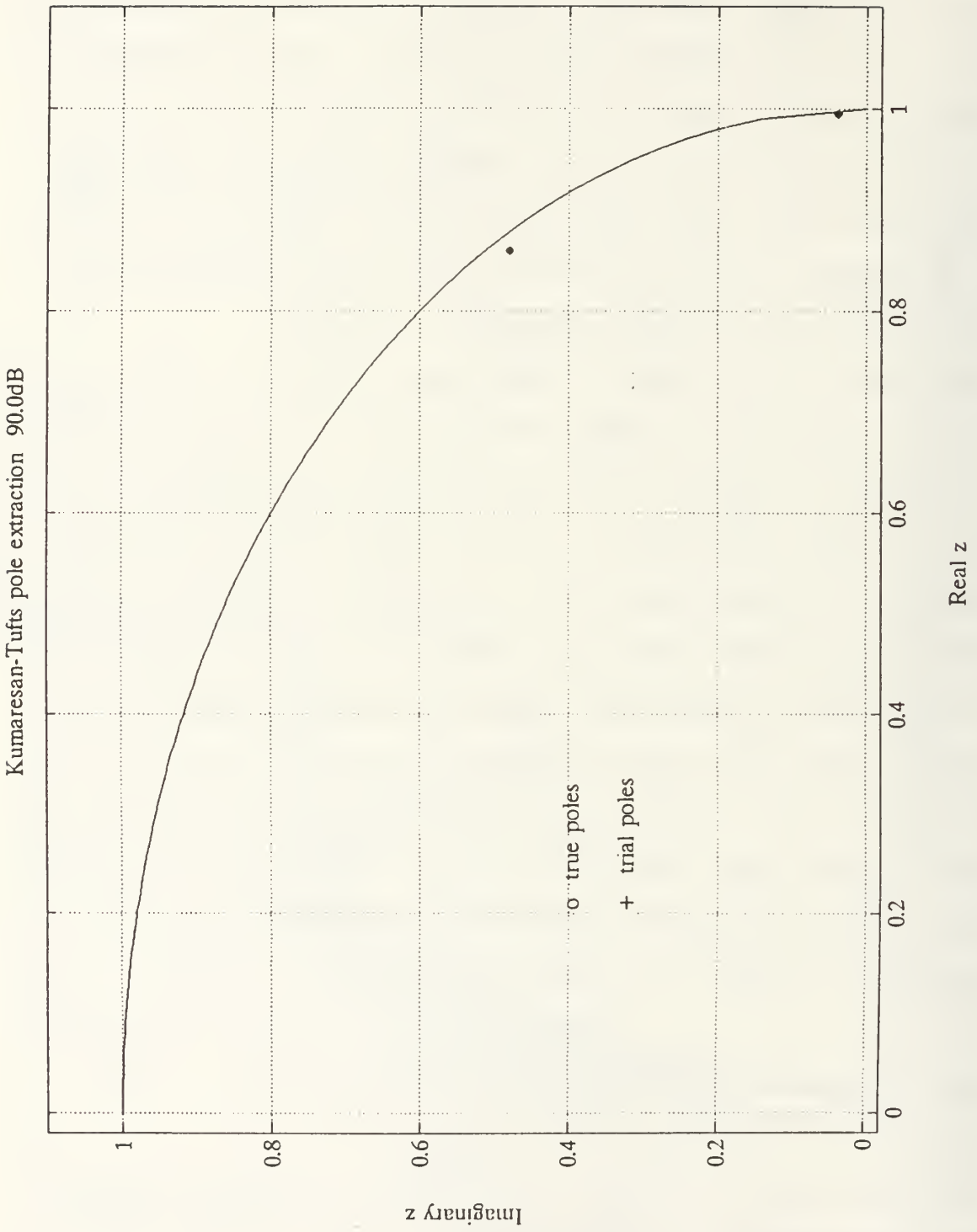

Figure 2. Kumaresan-Tufts Poles, Synthetic Data, 90.0 dB SNR

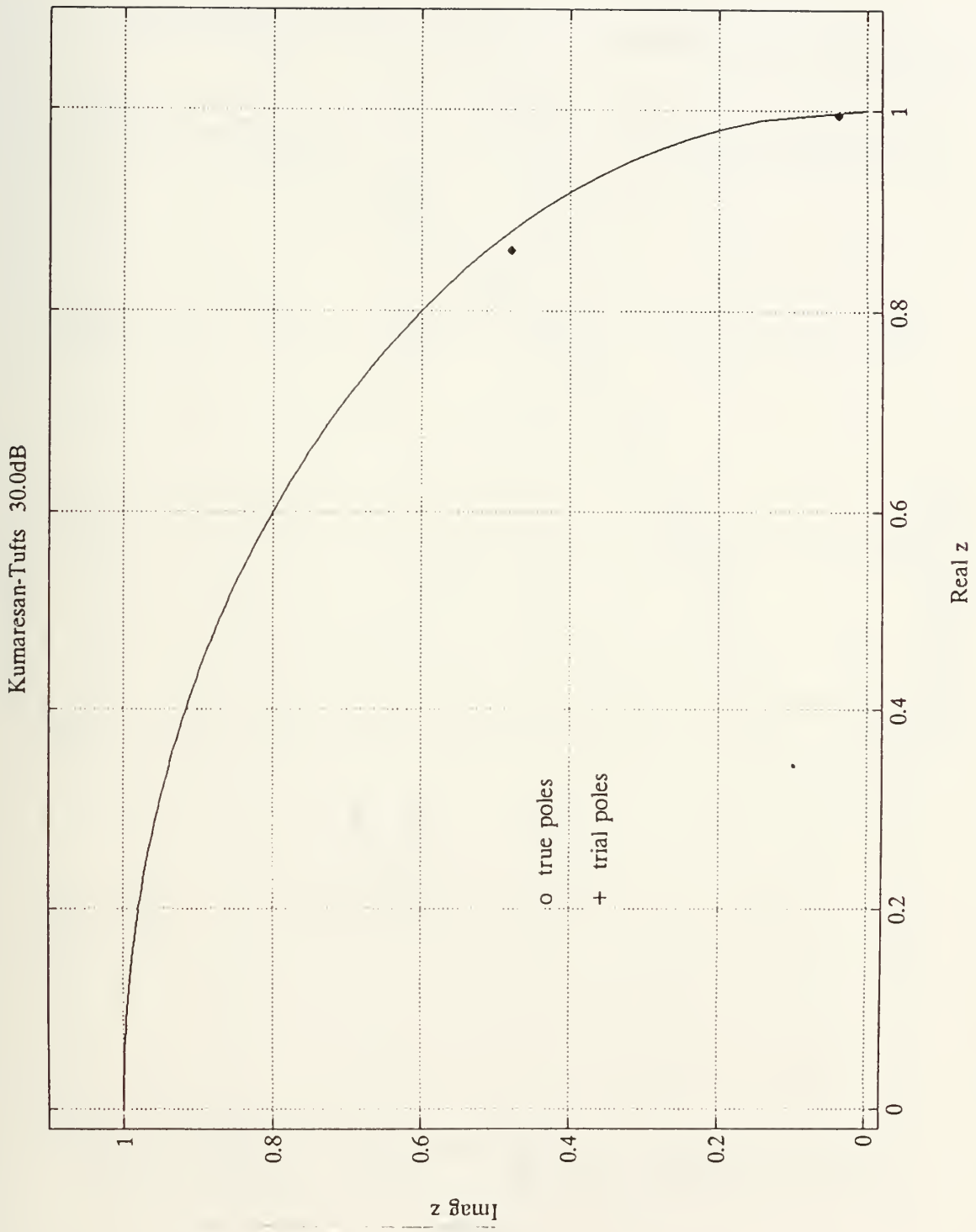

Figure 3. Kumaresan-Tufts Poles, Synthetic Data, 30.0 dB SNR

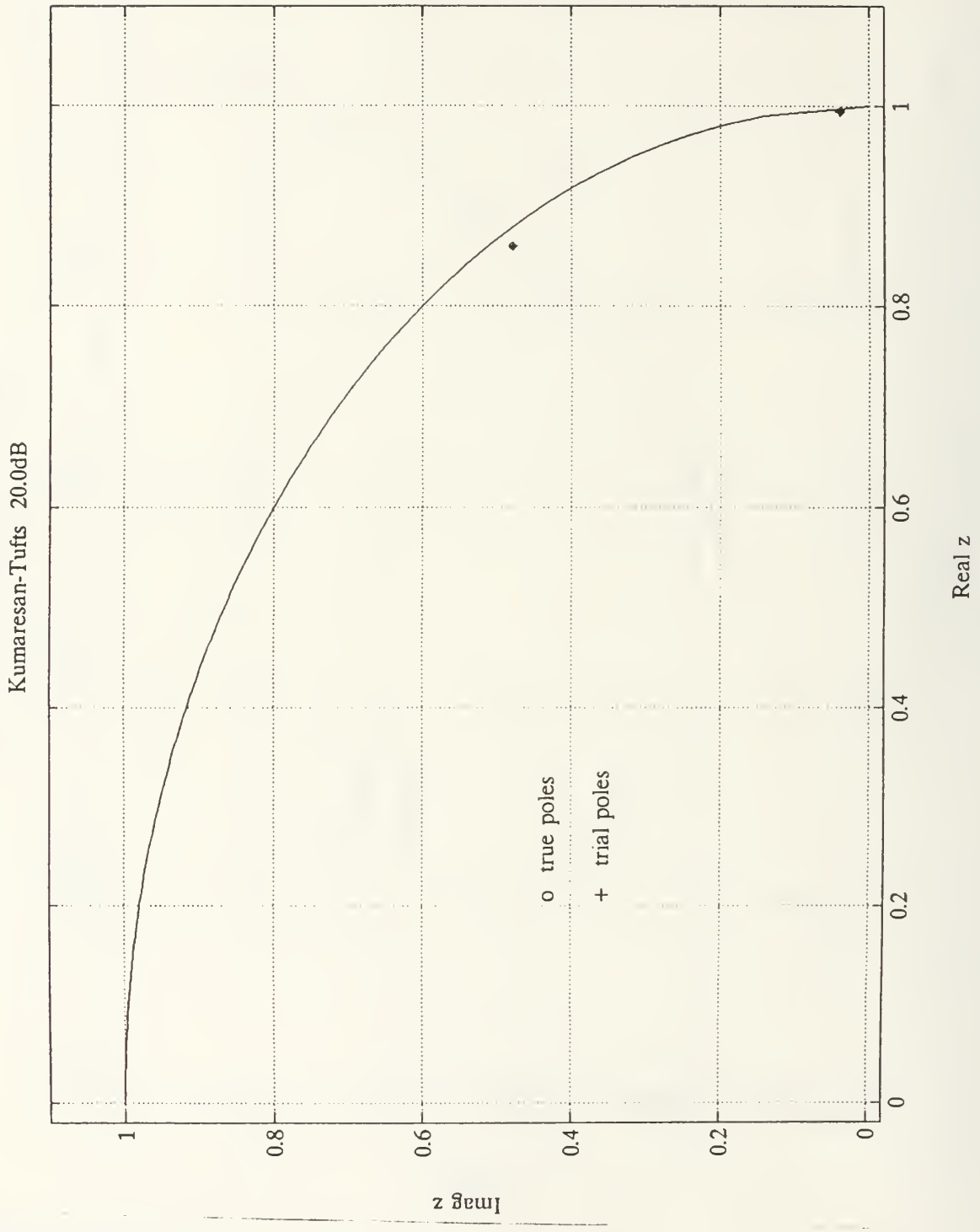

Figure 4. Kumaresan-Tufts Poles, Synthetic Data, 20.0 dB SNR

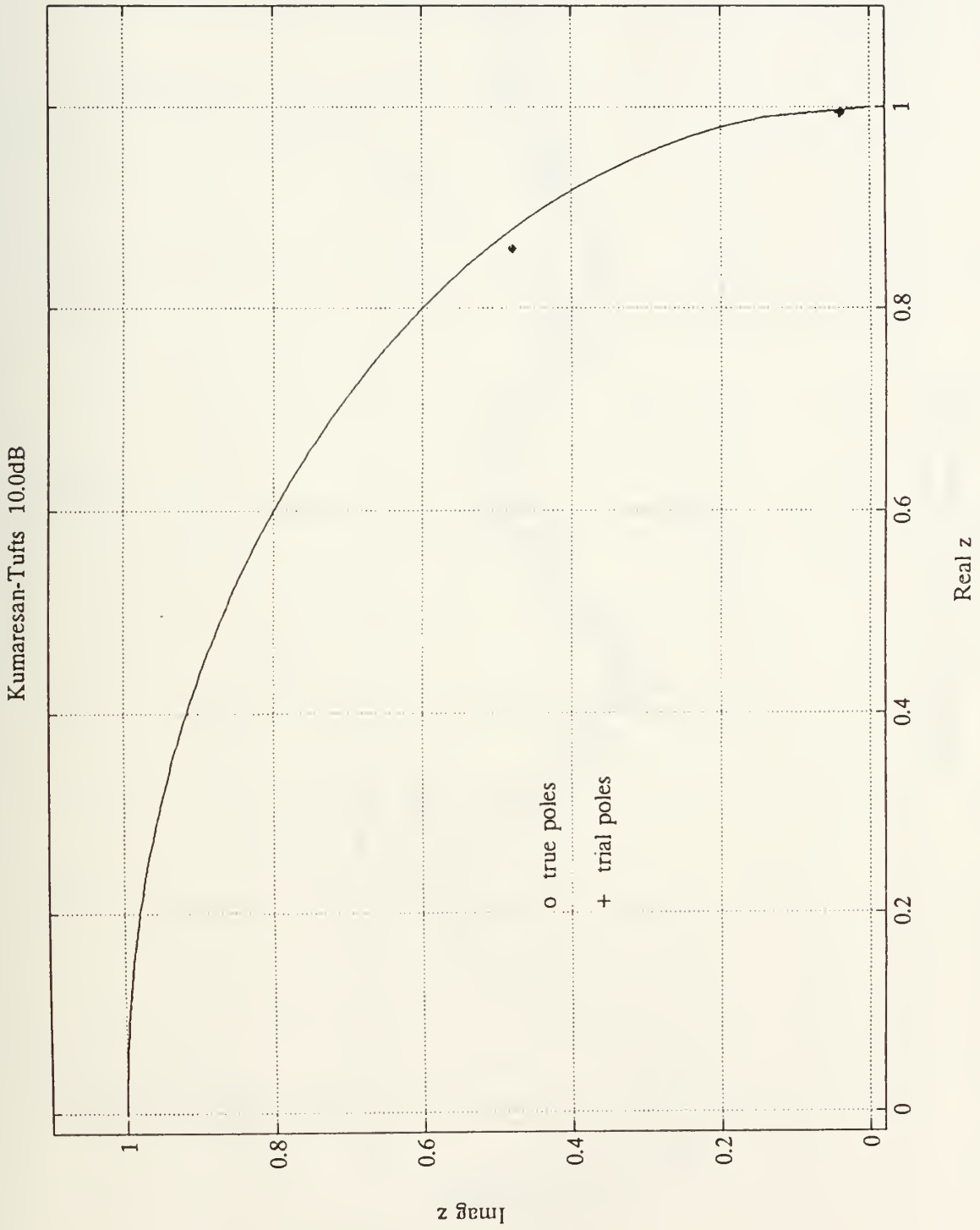

Figure 5. Kumaresan-Tufts Poles, Synthetic Data, 10.0 dB SNR

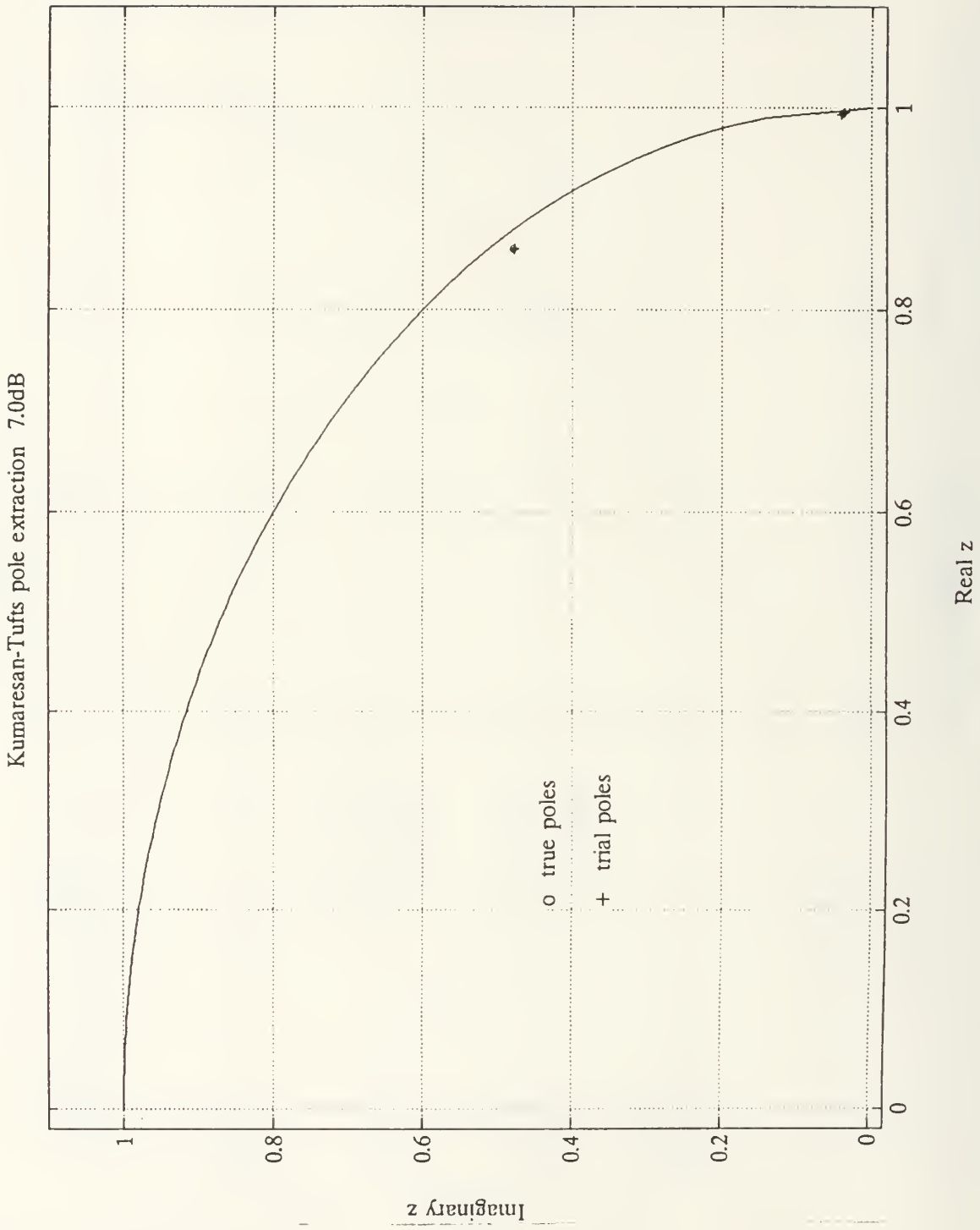

Figure 6. Kumaresan-Tufts Poles, Synthetic Data, 7.0 dB SNR

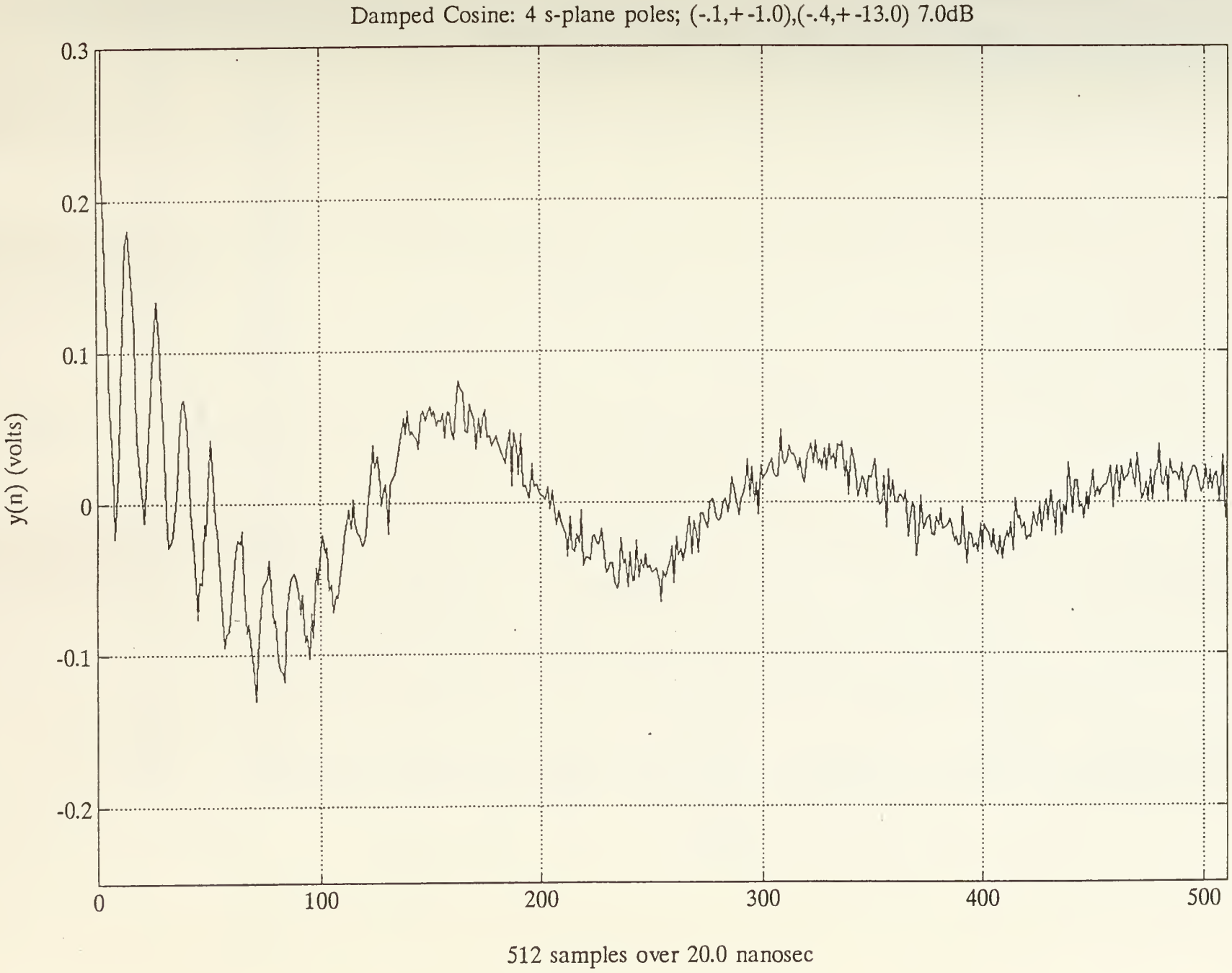

FIGURE 7. SIGNAL CONTAINING TWO S-PLANE POLES, 7.0dB

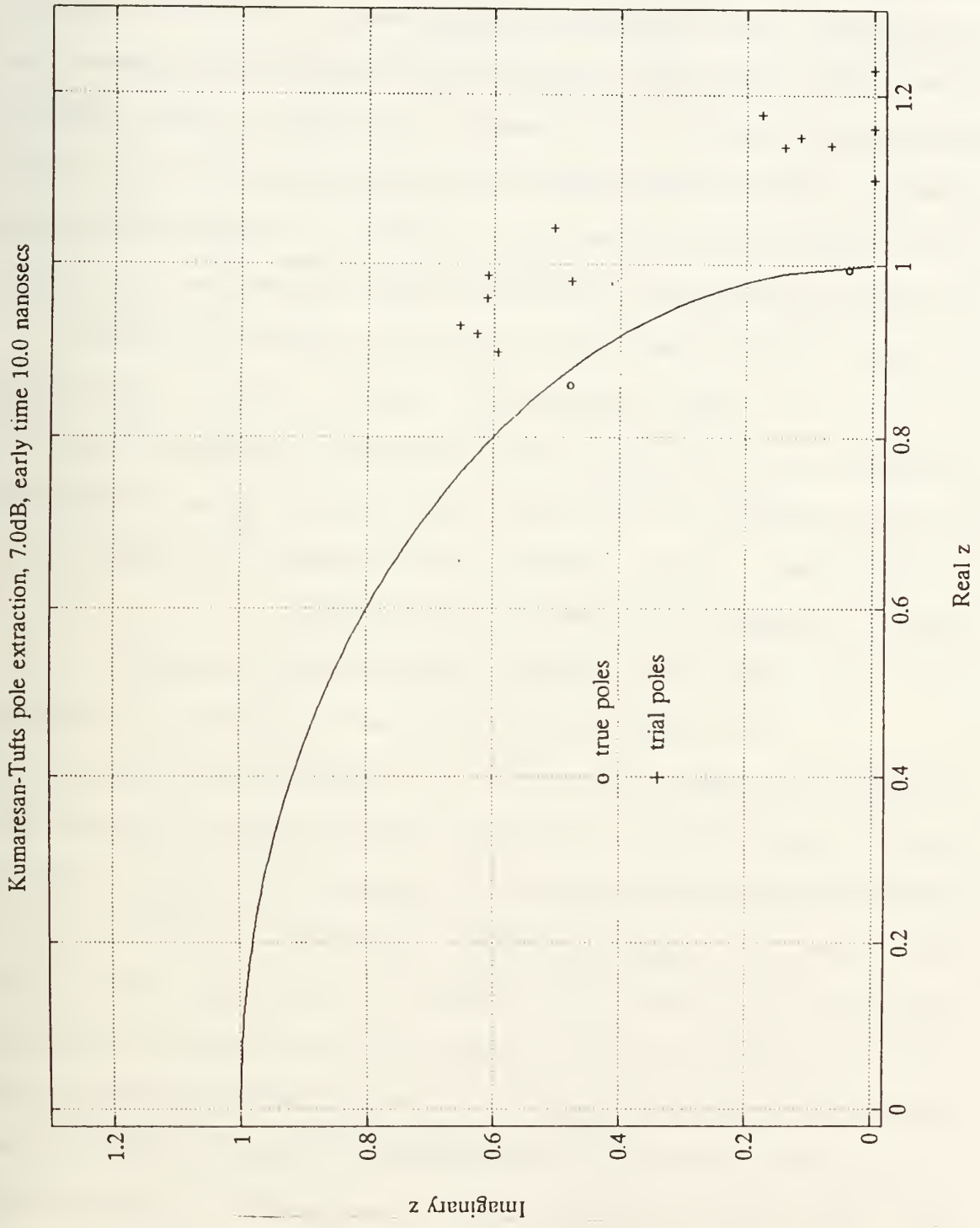

Figure 8. Kumaresan-Tufts Pole Extraction, 7.0 dB SNR

beginning ten nanoseconds later. Since the SNR is calculated over twenty nanoseconds for both signals, the signal power at some later time will clearly be less than the power ten nanoseconds earlier. The results in Figure 8 show complete breakdown of the algorithm's ability to extract poles. The trial poles shown are the poles closest to the true poles, and yet they are located at positions whose reflections are inside the unit circle where noise poles are typically located.

The preceding results show outstanding accuracy for full-length noisy data but a complete breakdown of the algorithm for the same signal with a later transition to late-time. These initial observations are supported by similar findings presented in this thesis.

b. Thin Wire Integral Equation Generated Data

For simple objects such as a thin wire, the radar response of that object can be computed by establishing boundary conditions on the object and numerically solving the integral equations that describe the surface current. Recall the magnetic field integral equation given by (1). Simulations produced by Morgan's time-domain thin wire integral equation computer program [12] were used to evaluate the pole extraction algorithm. The excitation waveform used is the double Gaussian pulse depicted in Figure 9. This pulse is a wide Gaussian pulse with a ten percent width of 0.3 nanoseconds subtracted from a narrow Gaussian pulse with a ten

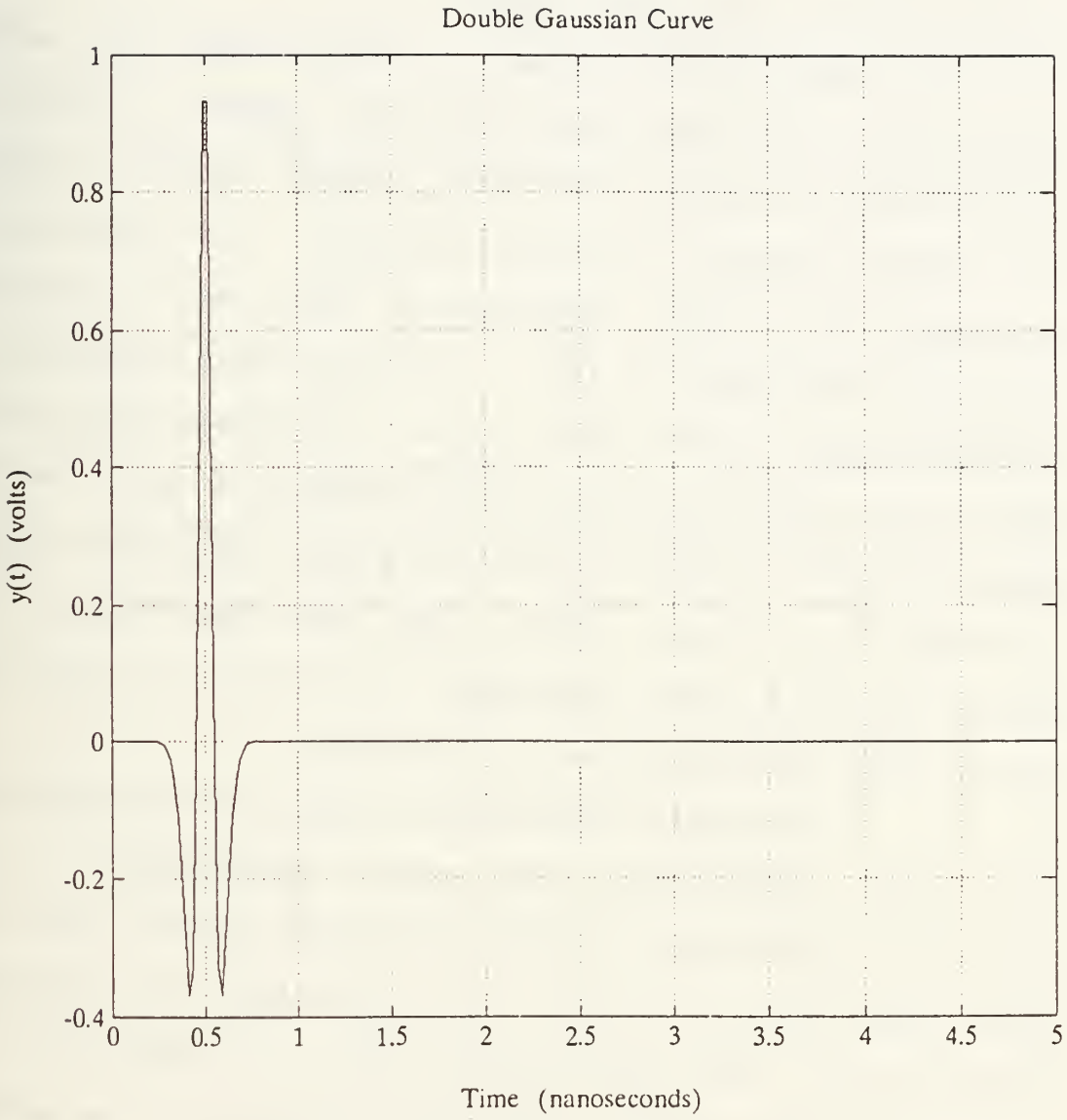

Figure 9. Double Gaussian Pulse

nanoseconds subtracted from a narrow Gaussian pulse with a ten percent width of 0.15 nanoseconds.

Figures 10 through 13 depict back scattering response of a 0.1 meter length thin-wire, having a radius of 0.00118 meter, computed at various incident aspects, ranging from thirty degrees to ninety degrees. The laboratory arrangement for actual measurements simulated by Morgan's program is described in [17]. Ninety degrees represents a broadside aspect, while thirty degrees represents the incident plane wave having nearly grazing incidence on the wire. The poles extracted at each of the four aspect angles are plotted in Figure 14. In this figure, and those that follow which depict extracted poles, the signal poles lie in or on the unit circle, and the noise poles lie outside.

The results obtained with this rigorous numerical computation demonstrate the aspect independence of the extracted poles using the Kumaresan-Tufts method. Note that only half of the poles were obtained for broadside illumination; two even-numbered poles can easily be seen outside the unit circle. This results because of the physical symmetry of both the wire and the incident field, thus precluding excitation of odd-symmetric modal currents and their associated natural resonances.

Figure 15 exemplifies the computed back-scattering response of the 0.1 meter thin wire corrupted artificially

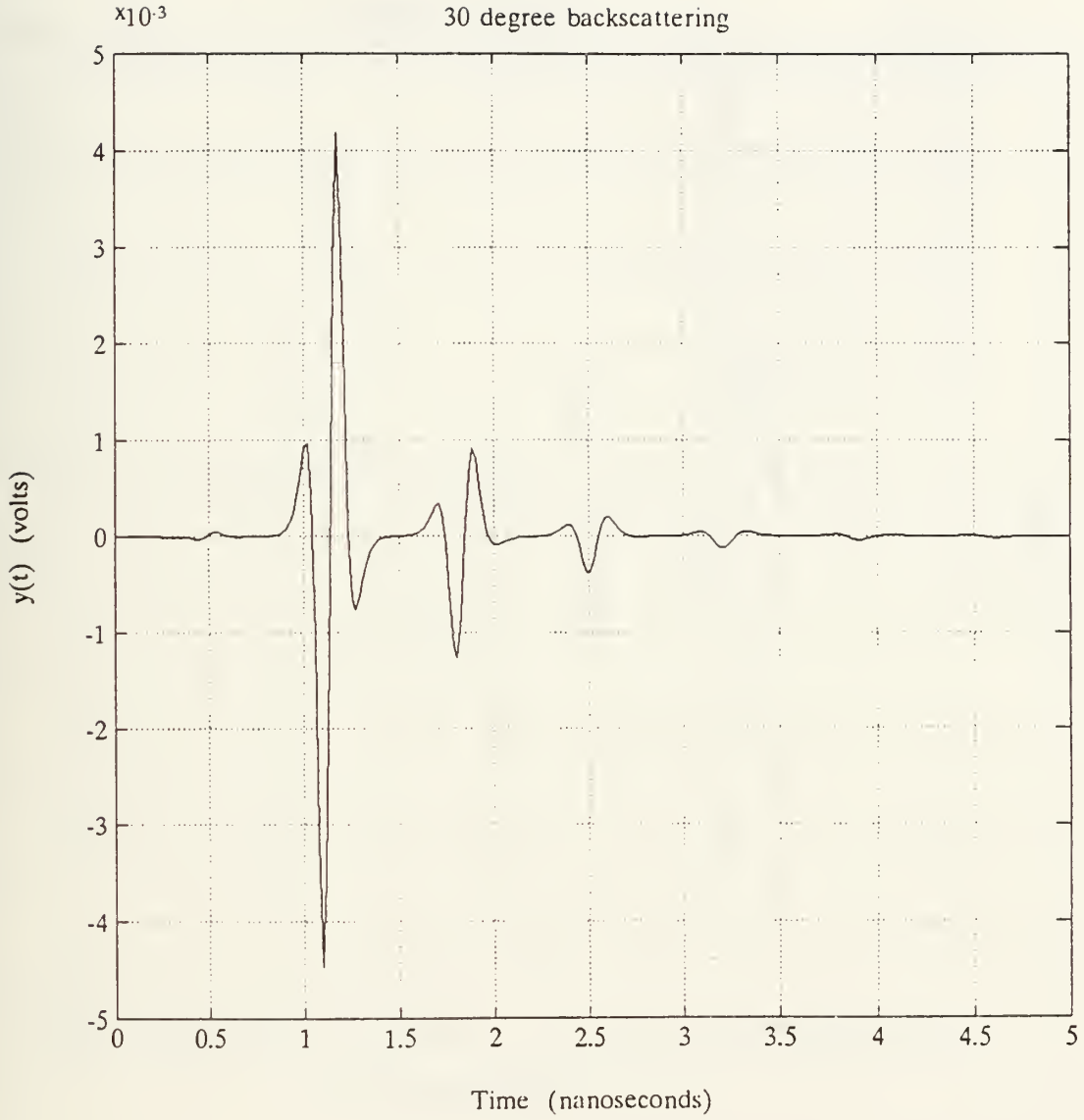

Figure 10. Integral Equation Thin Wire Scattering, 30 Degree Aspect

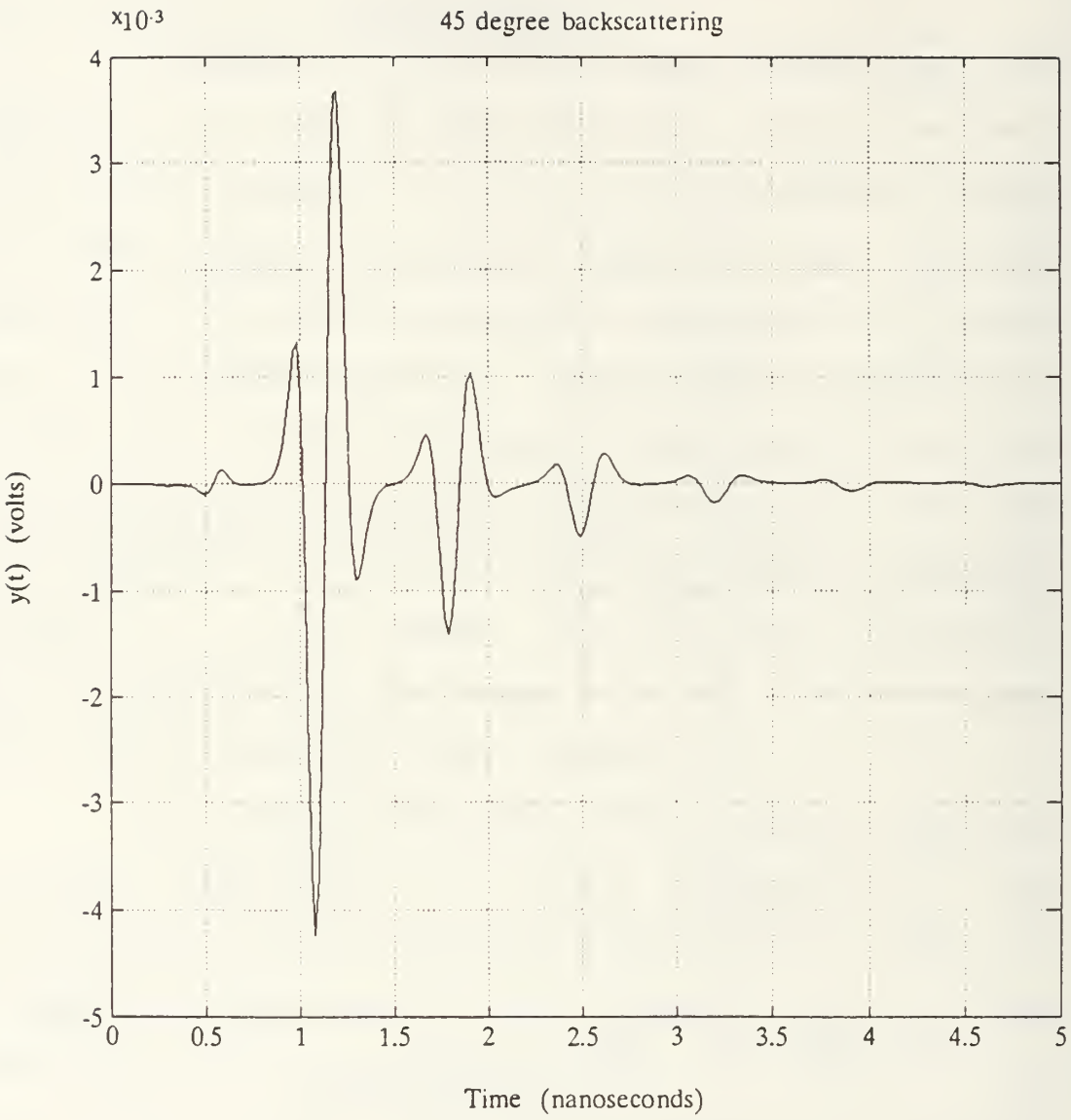

Figure 11. Integral Equation Thin Wire Scattering, 45 Degree Aspect

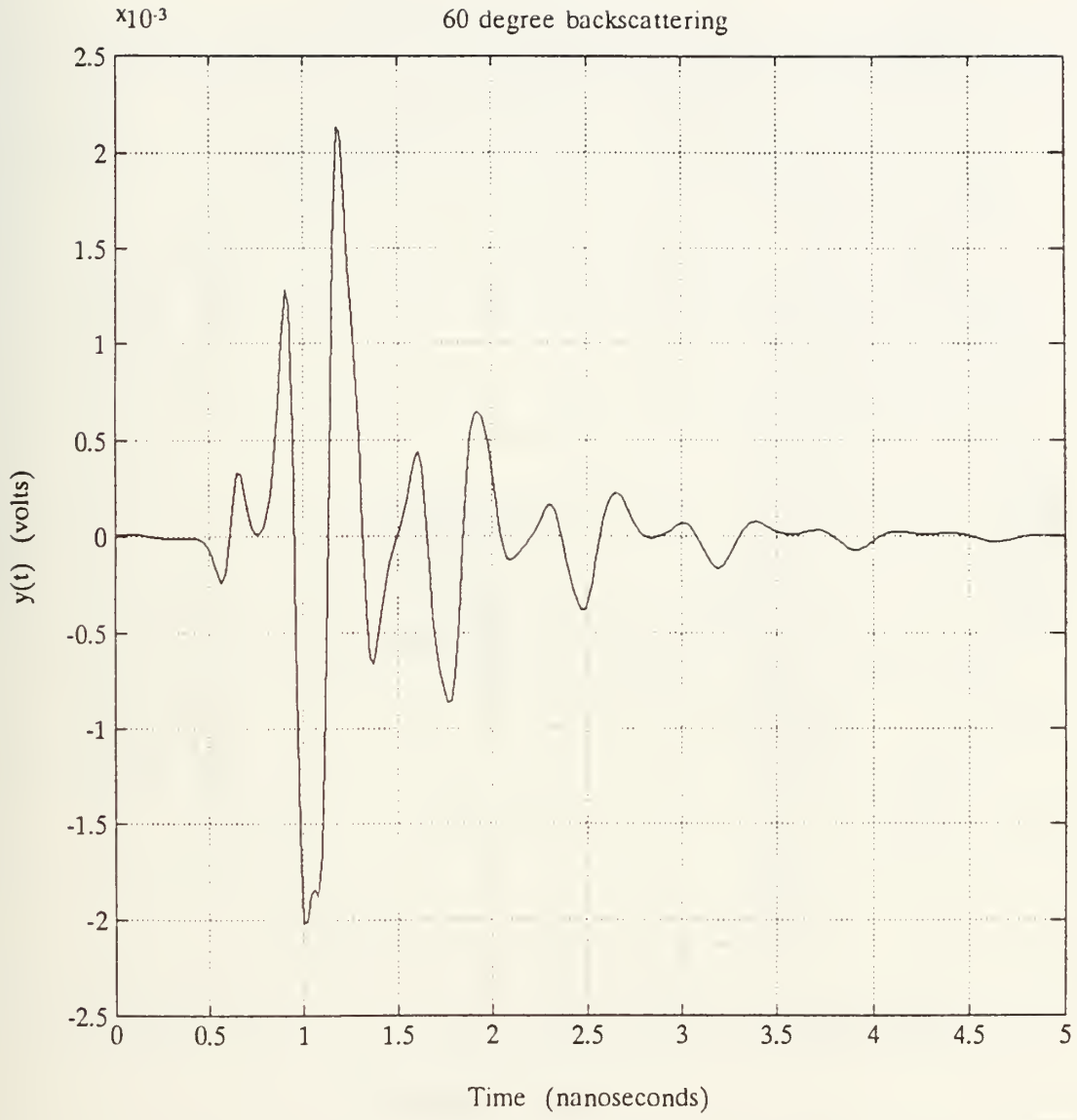

Figure 12. Integral Equation Thin Wire Scattering, 60 Degree Aspect

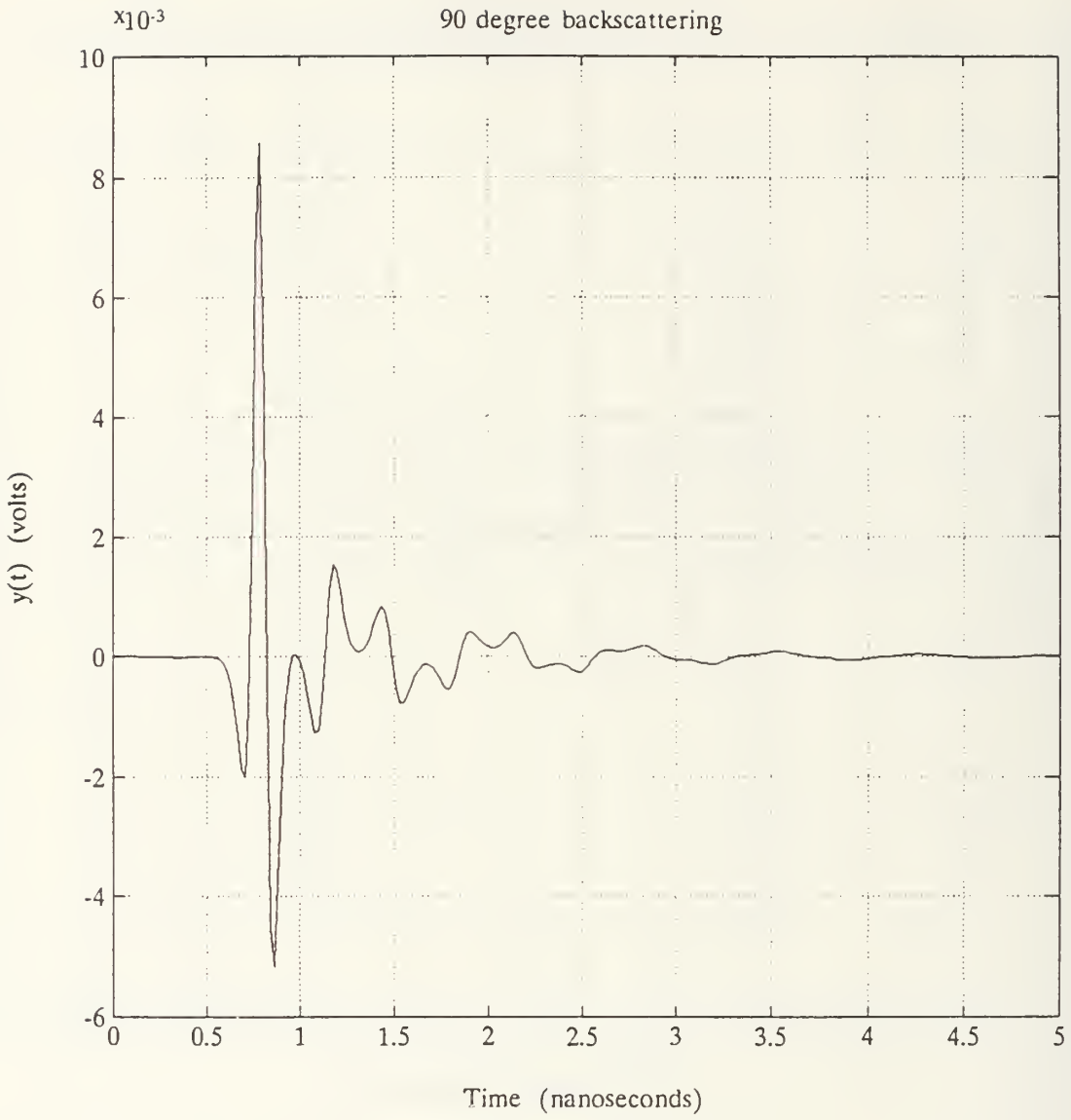

Figure 13. Integral Equation Thin Wire Scattering, 90 Degree Aspect

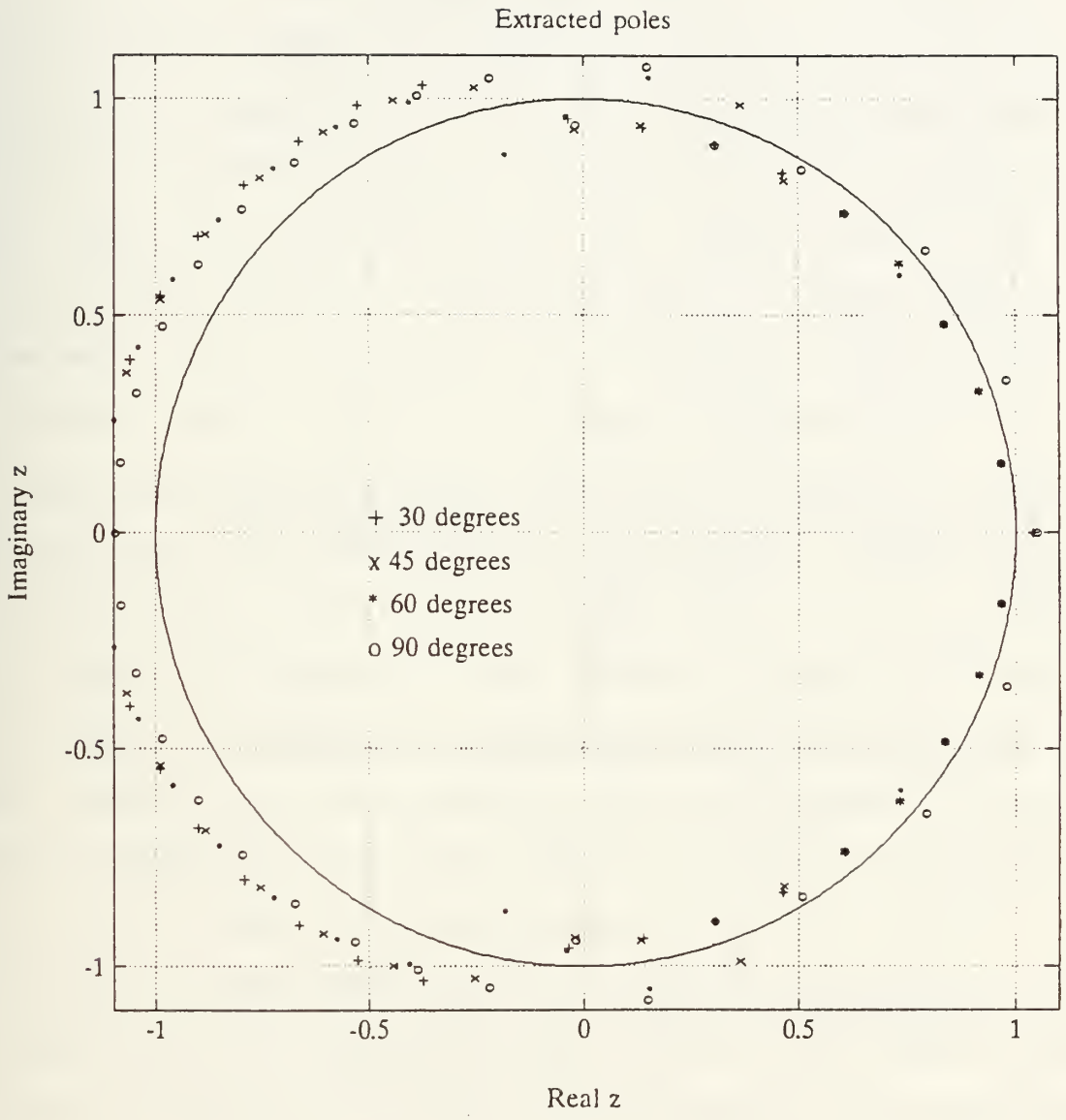

Figure 14. Kumaresan-Tufts Poles, Noiseless Thin Wire Data

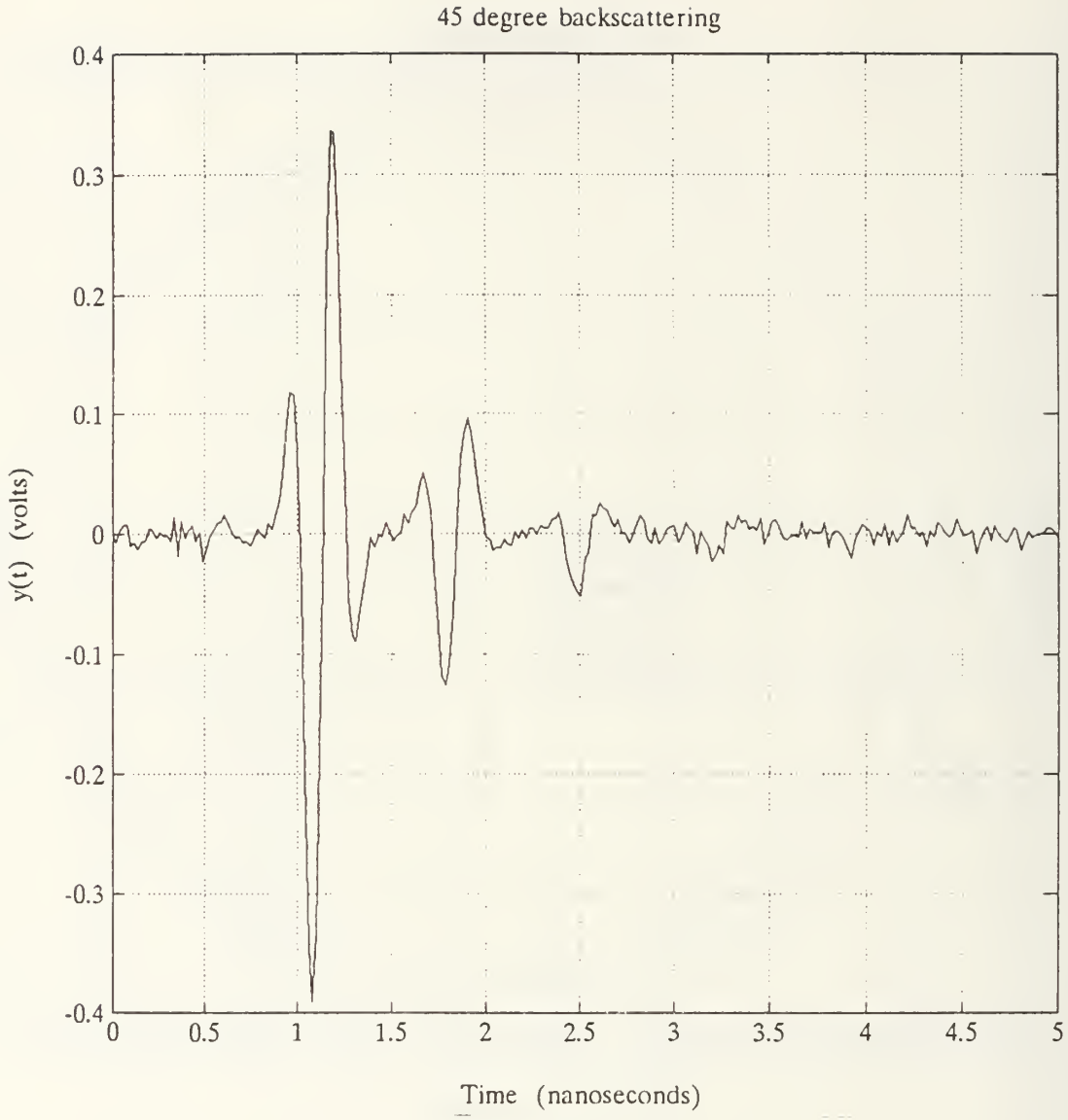

Figure 15. Integral Equation Thin Wire Scattering, 20.0 dB SNR, 45 Degree Aspect

with noise at a 20.0 dB SNR. Figure 16 shows the poles extracted at each of the four angles of incidence used previously in Figure 14. Poles of Figure 14 at 90° are now missing in Figure 16, and only the first three low frequency poles are tightly grouped. The loss of high frequency poles is expected because these have the highest damping and thus lose their energy at the fastest rate. Further comparison between results computed at 20.0 dB SNR and infinite SNR are offered, angle by angle, in Figures 17 through 20.

One additional test of the computed thin wire scattering was conducted at a 7.0 dB SNR. The corrupted waveforms are exemplified by Figure 21; the extracted poles are shown in Figure 22. The number of poles obtained has decreased with respect to the number obtained at 20.0 dB SNR. The grouping of the clusters has also expanded. Angle by angle comparisons are again offered in Figures 23 through 26.

c. Scale Model Measurements

The transient scattering measurements of scale models used for evaluation in this section were made by Walsh using the anechoic chamber of the Transient Electromagnetic Scattering Laboratory at the Naval Postgraduate School. The entire measurement process and laboratory setup are described in detail in [17].

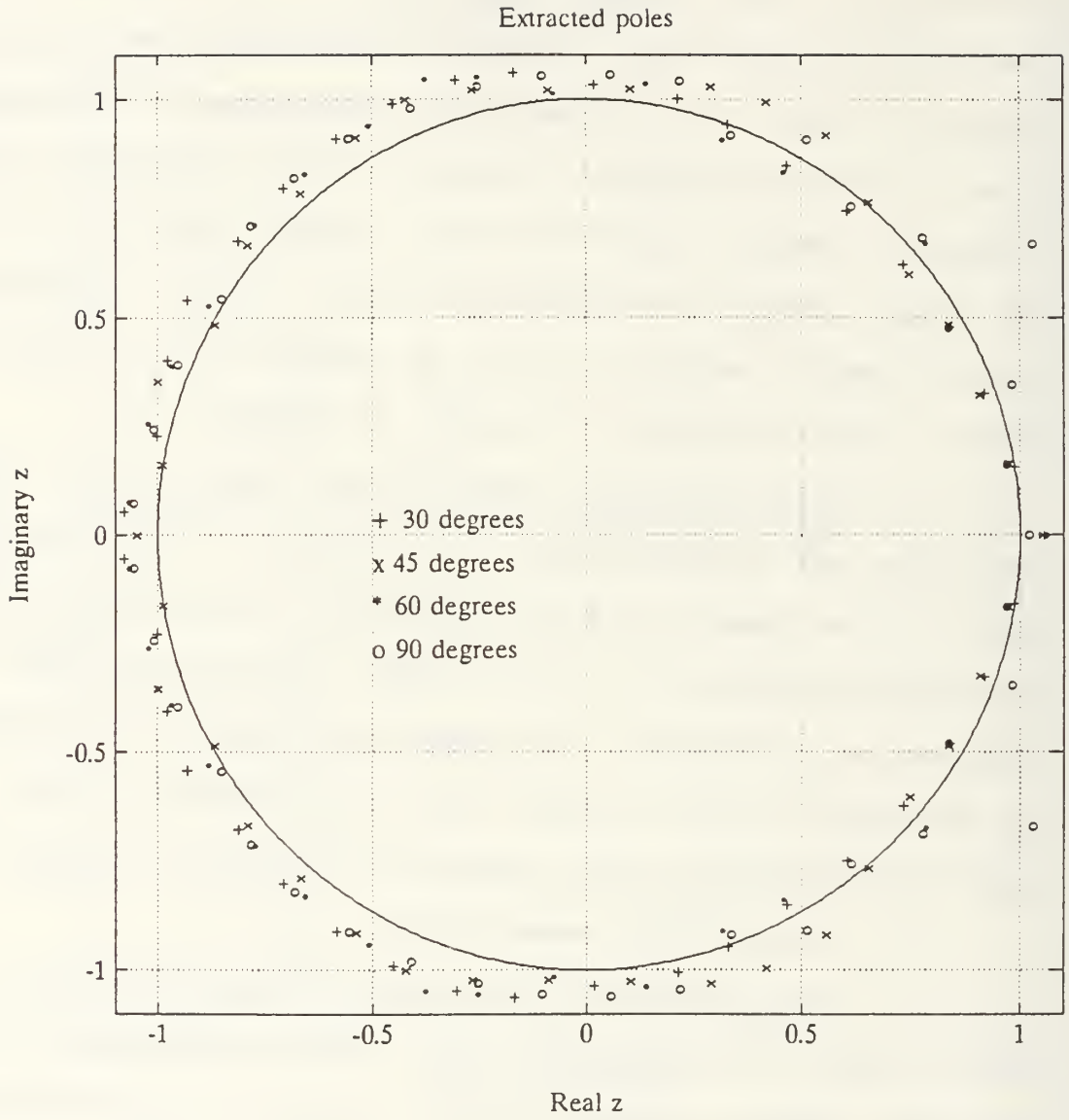

Figure 16. Kumaresan-Tufts Poles, 20.0 dB SNR

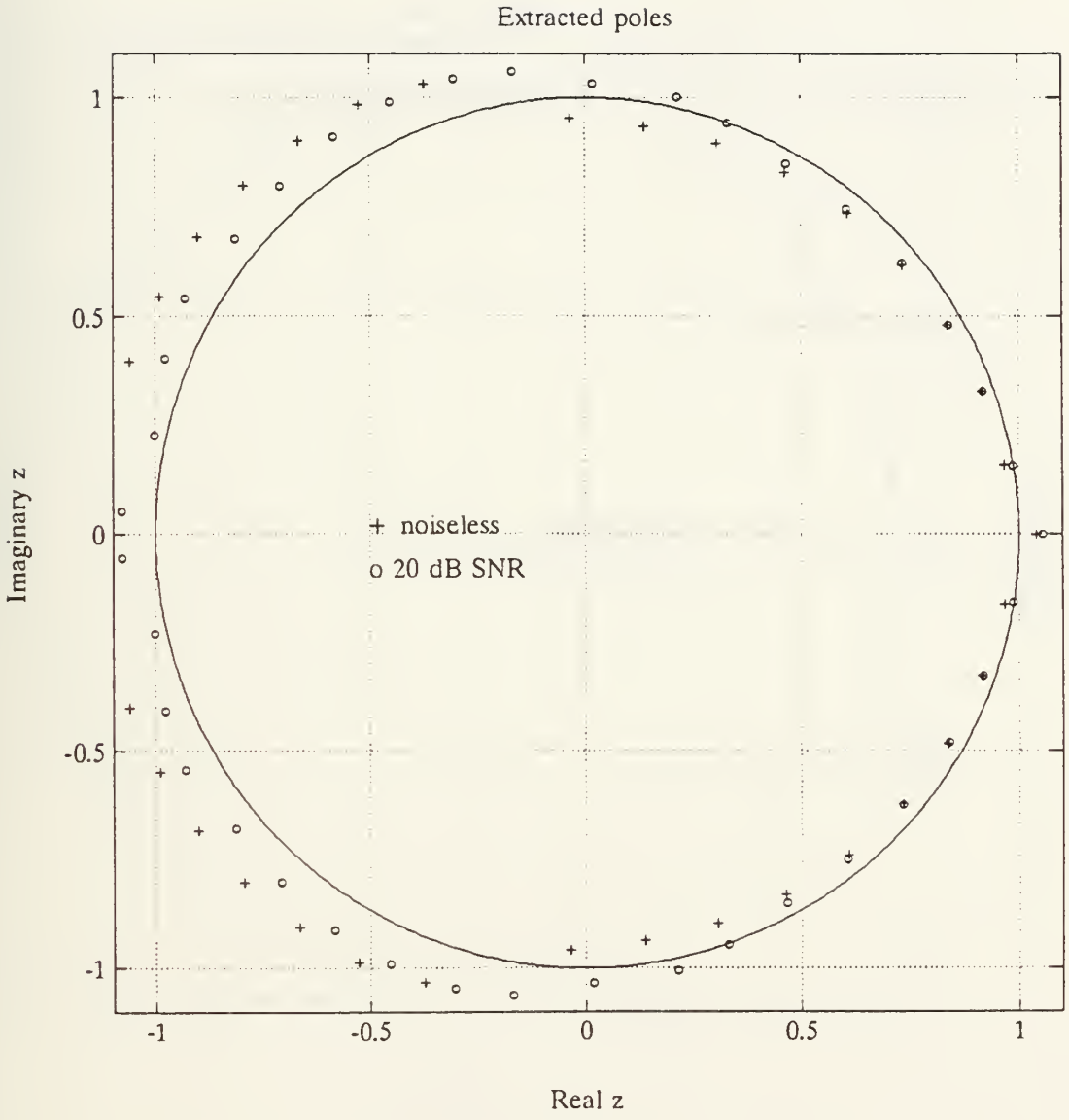

Figure 17. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 30 Degree Aspect

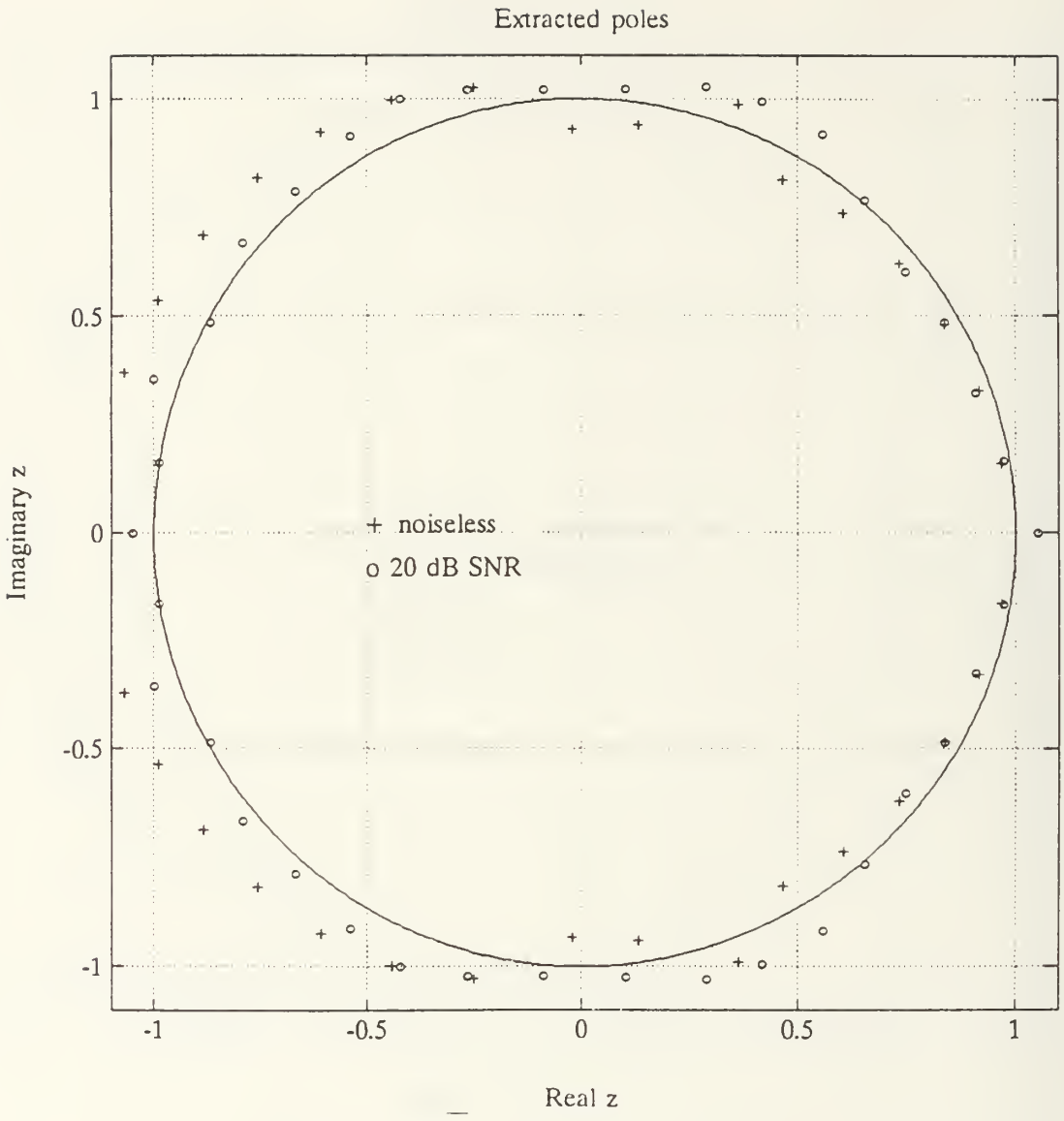

Figure 18. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 45 Degree Aspect

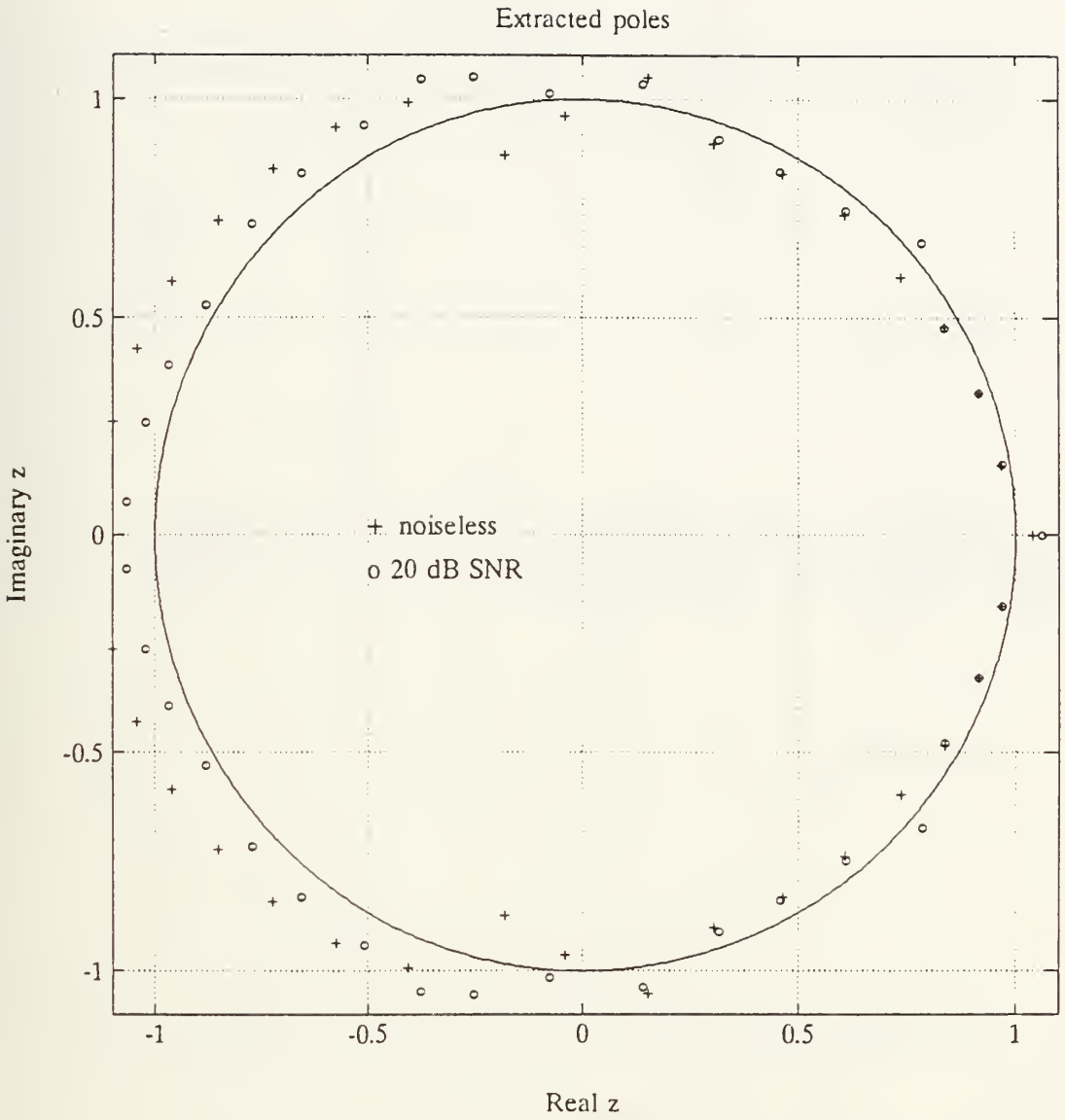

Figure 19. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 60 Degree Aspect

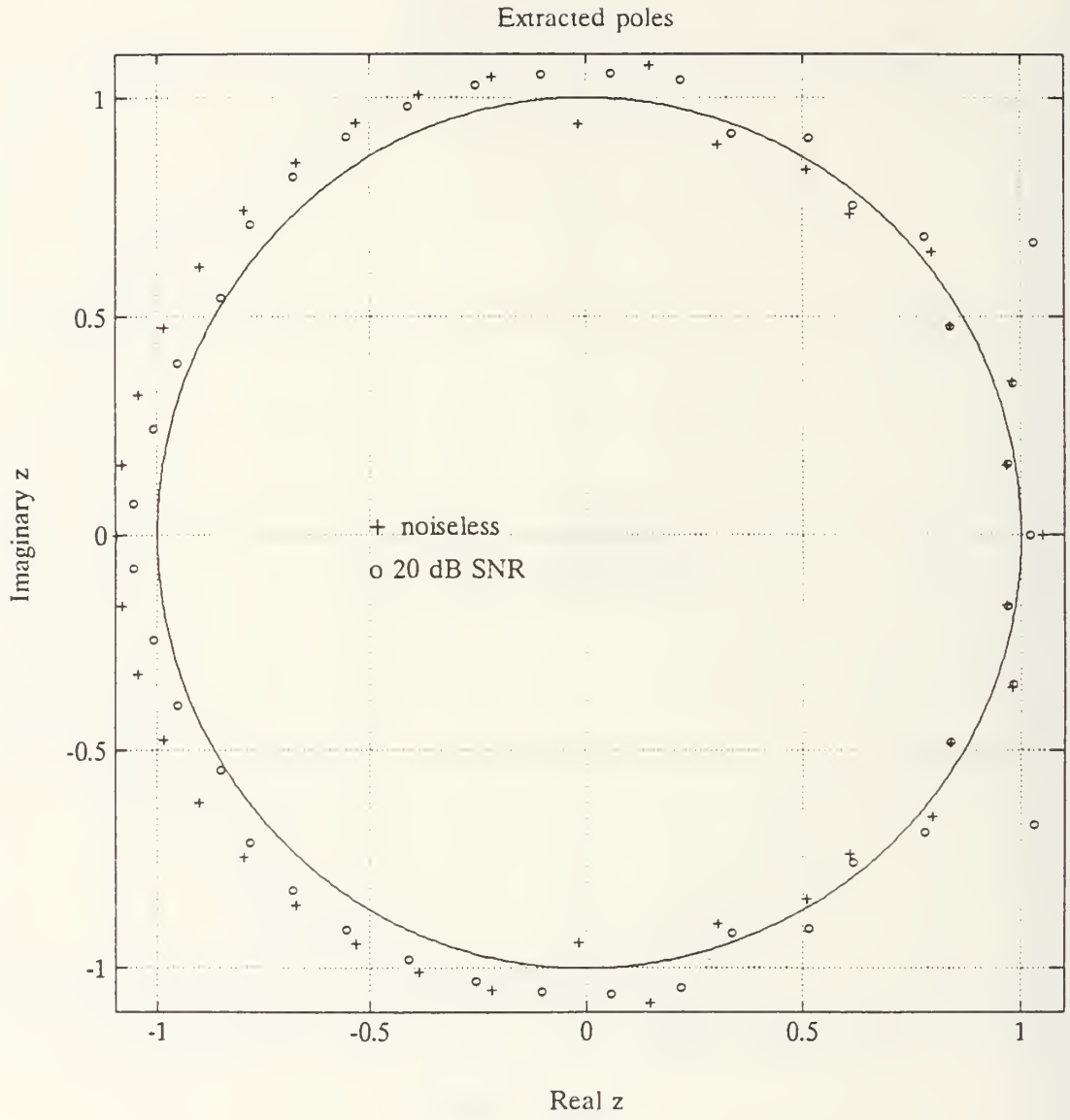

Figure 20. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 90 Degree Aspect

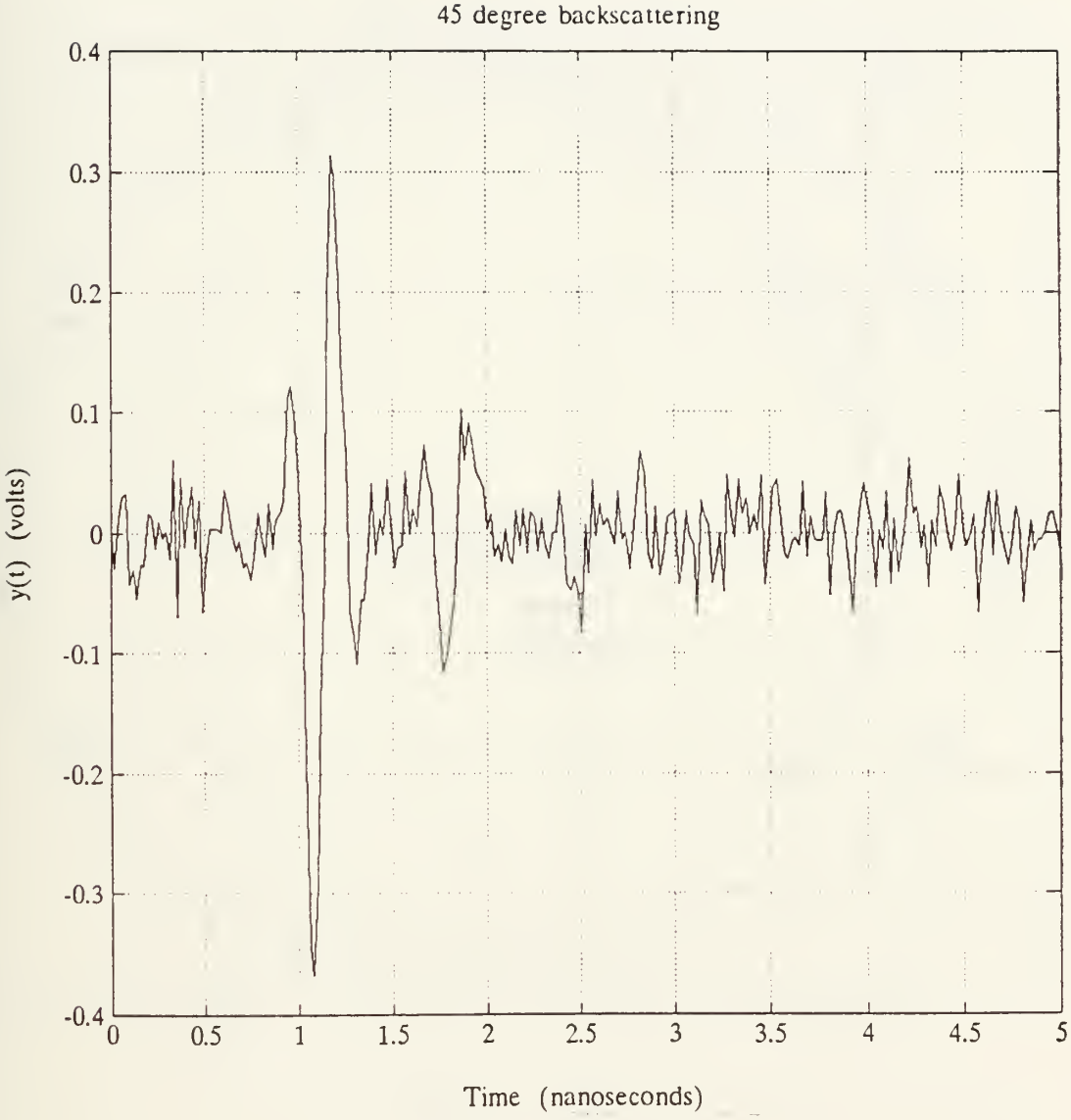

Figure 21. Integral Equation Thin Wire Scattering, 7.0 dB SNR, 45 Degree Aspect

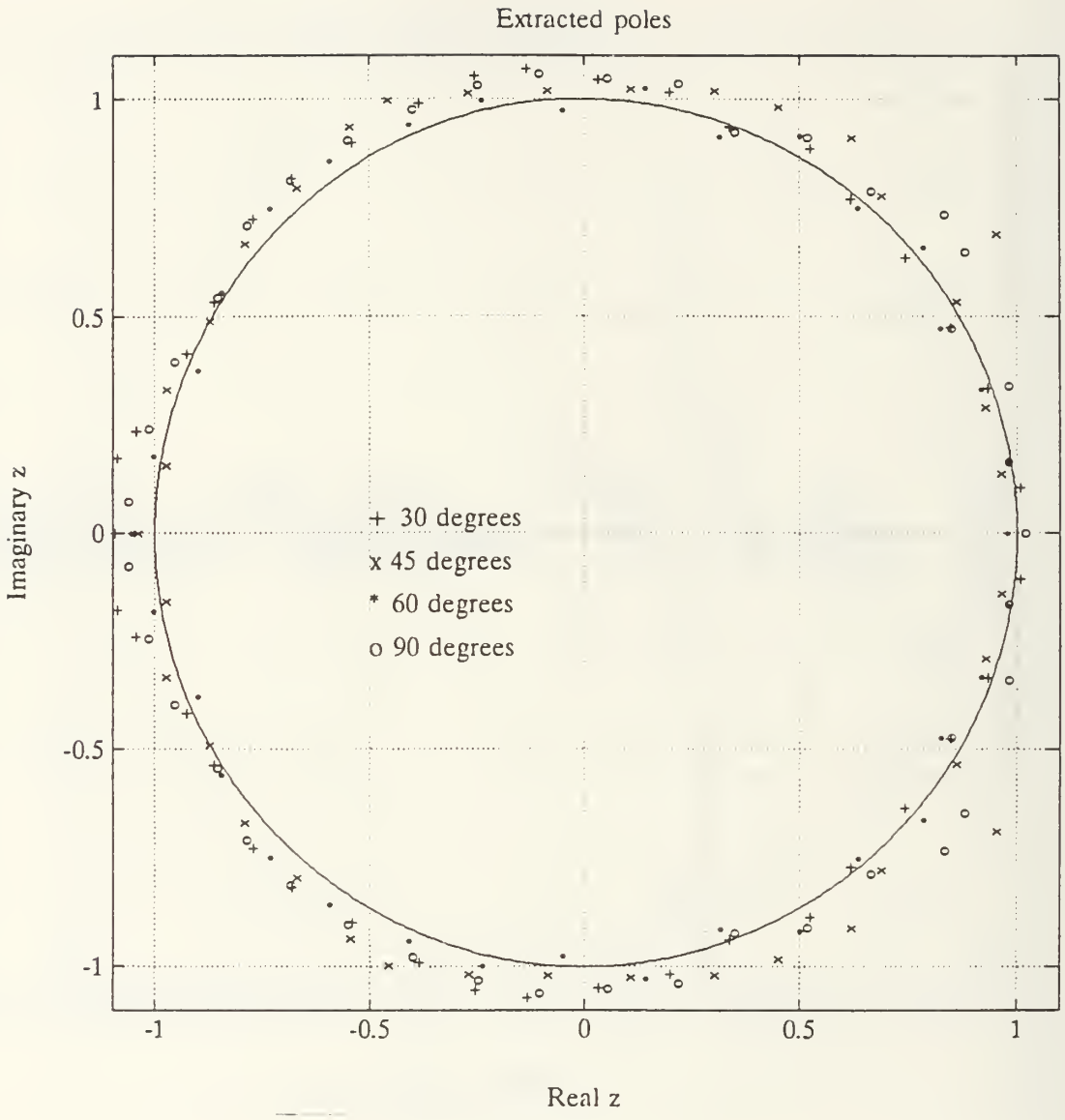

Figure 22. Kumaresan-Tufts Poles, 7.0 dB SNR

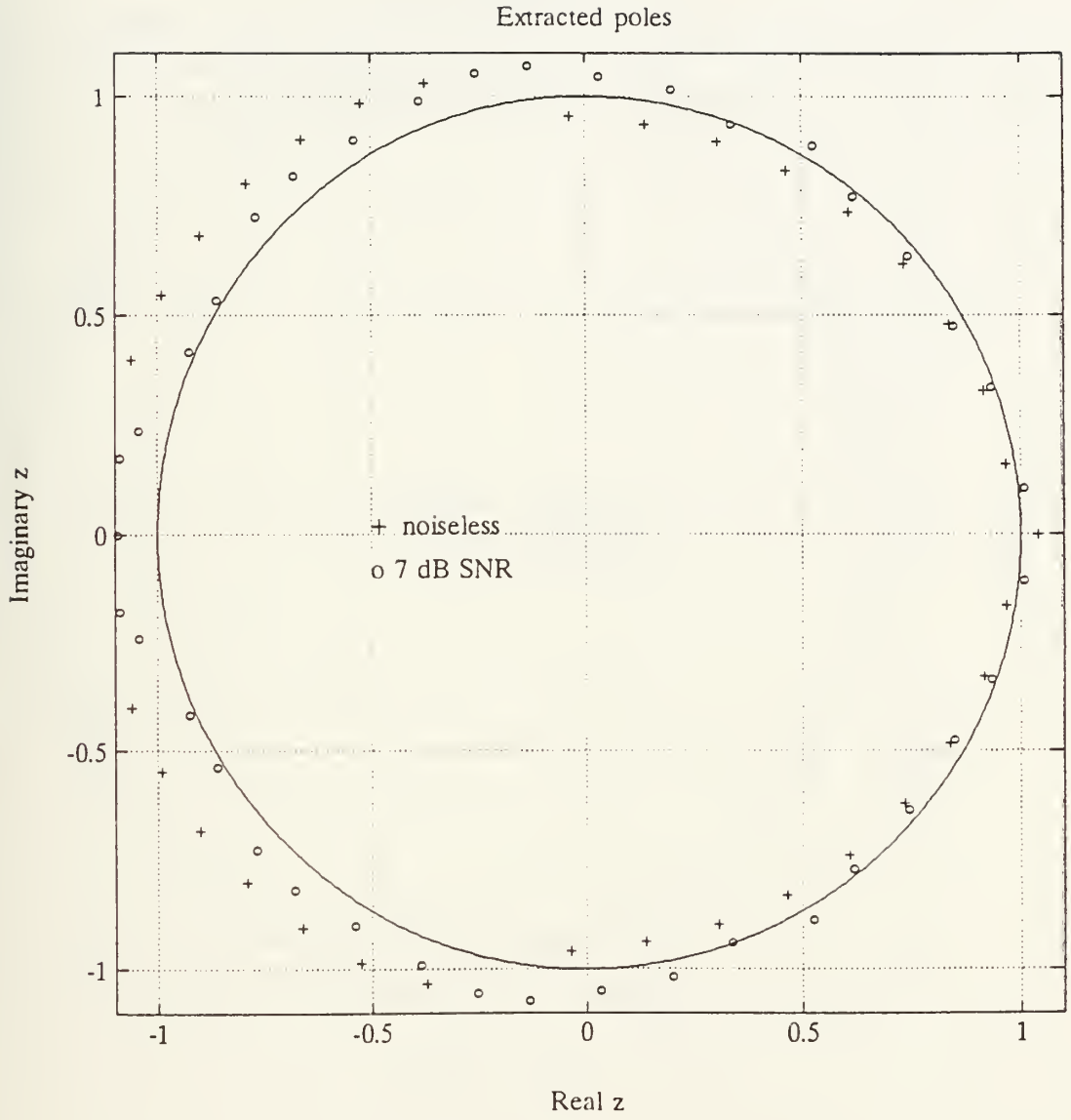

Figure 23. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 30 Degree Aspect

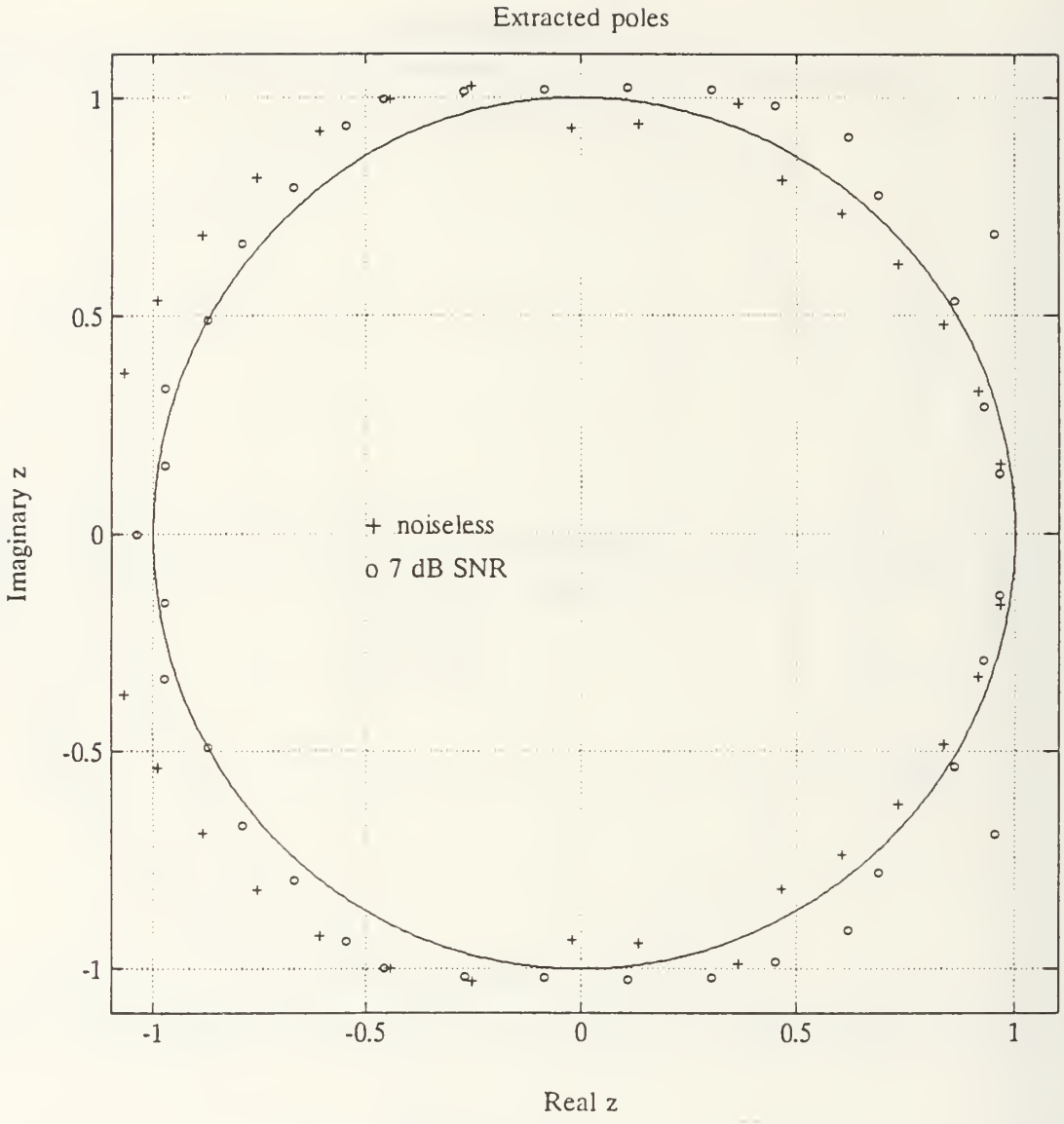

Figure 24. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 45 Degree Aspect

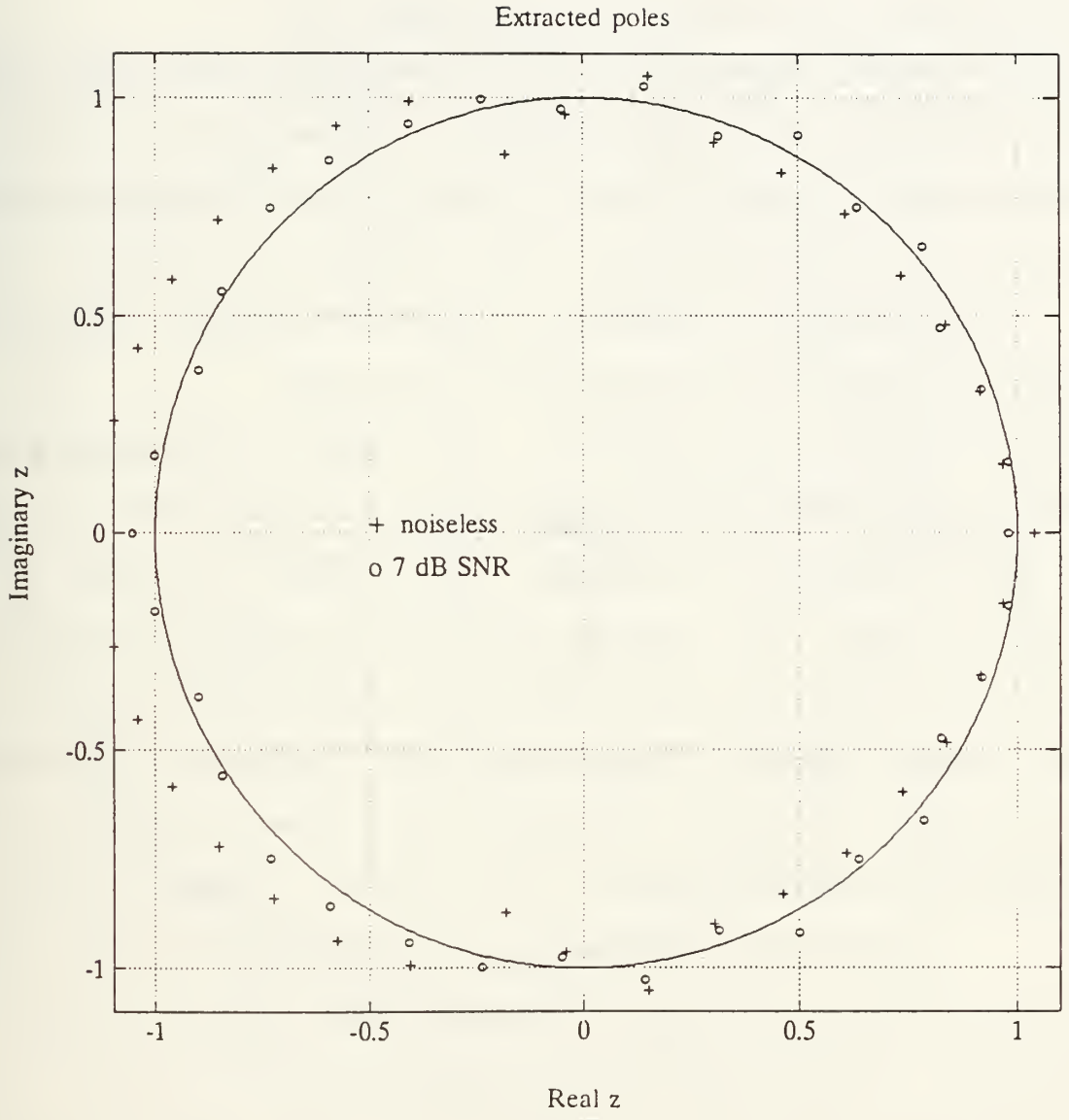

Figure 25. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 60 Degree Aspect

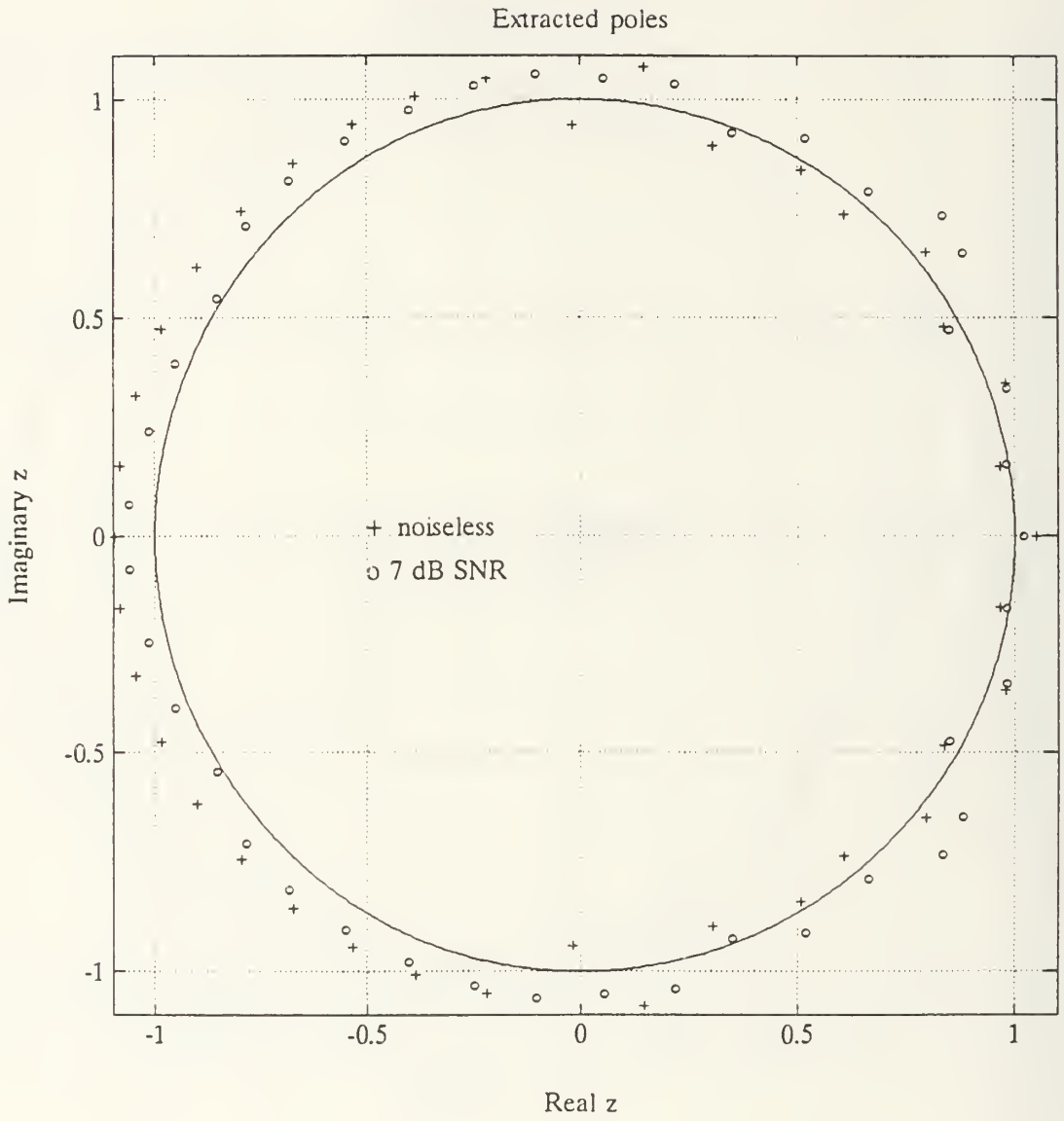

Figure 26. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 90 Degree Aspect

1. Wire Targets

The thin wire measurements were obtained from the scattering response of a 0.1 meter length thin wire having radius 0.00118 meter. Recall that these are the same dimensions as the wire whose computed response was processed in the previous section. The measurements at each of four incident aspects are shown in Figures 27 through 30.

The poles extracted from the four measurements are depicted in Figure 31. As before in the computed noisy data, tight clusters occur only at the lowest frequencies. The poles in these tight clusters are those which are measurably present at various aspects. The poles extracted at higher frequencies are those which possessed sufficient measurable energy at the given aspect. Figure 32 depicts the comparison between poles extracted from the measured and computed signals. Again, the closest agreement between the two sets of poles occurs at the lowest frequences.

2. Aircraft Models

Plastic 1/72 scale aircraft models, coated with silver, were used for transient scattering measurements. Representative scattering signatures of two aircraft targets, measured at six different aspects, are shown in Figures 33 through 36.

The results of pole extraction in target 1 are shown for a total of six different aspects in Figures 37 and

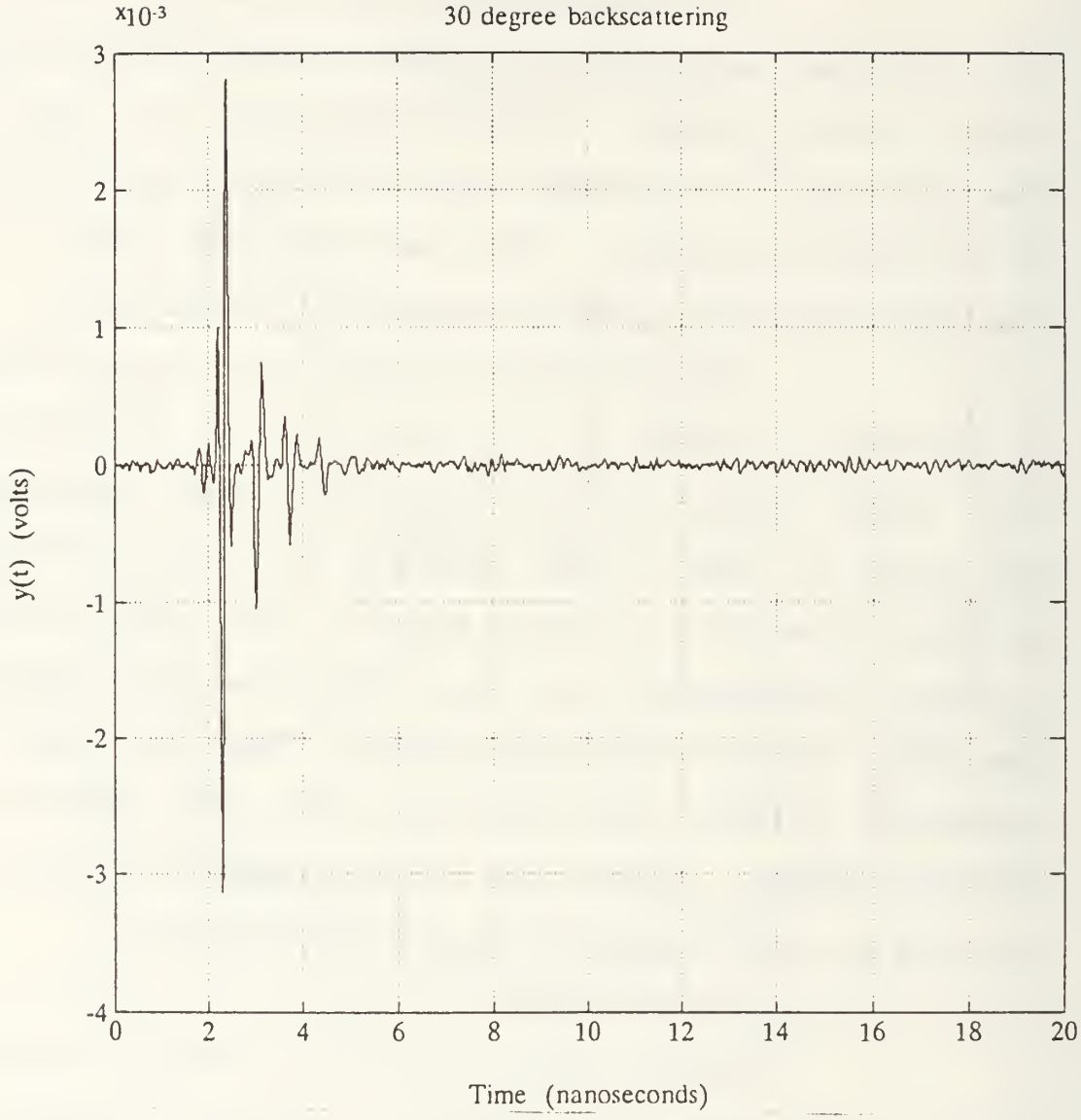

Figure 27. Measured Thin Wire Scattering, 30 Degree Aspect

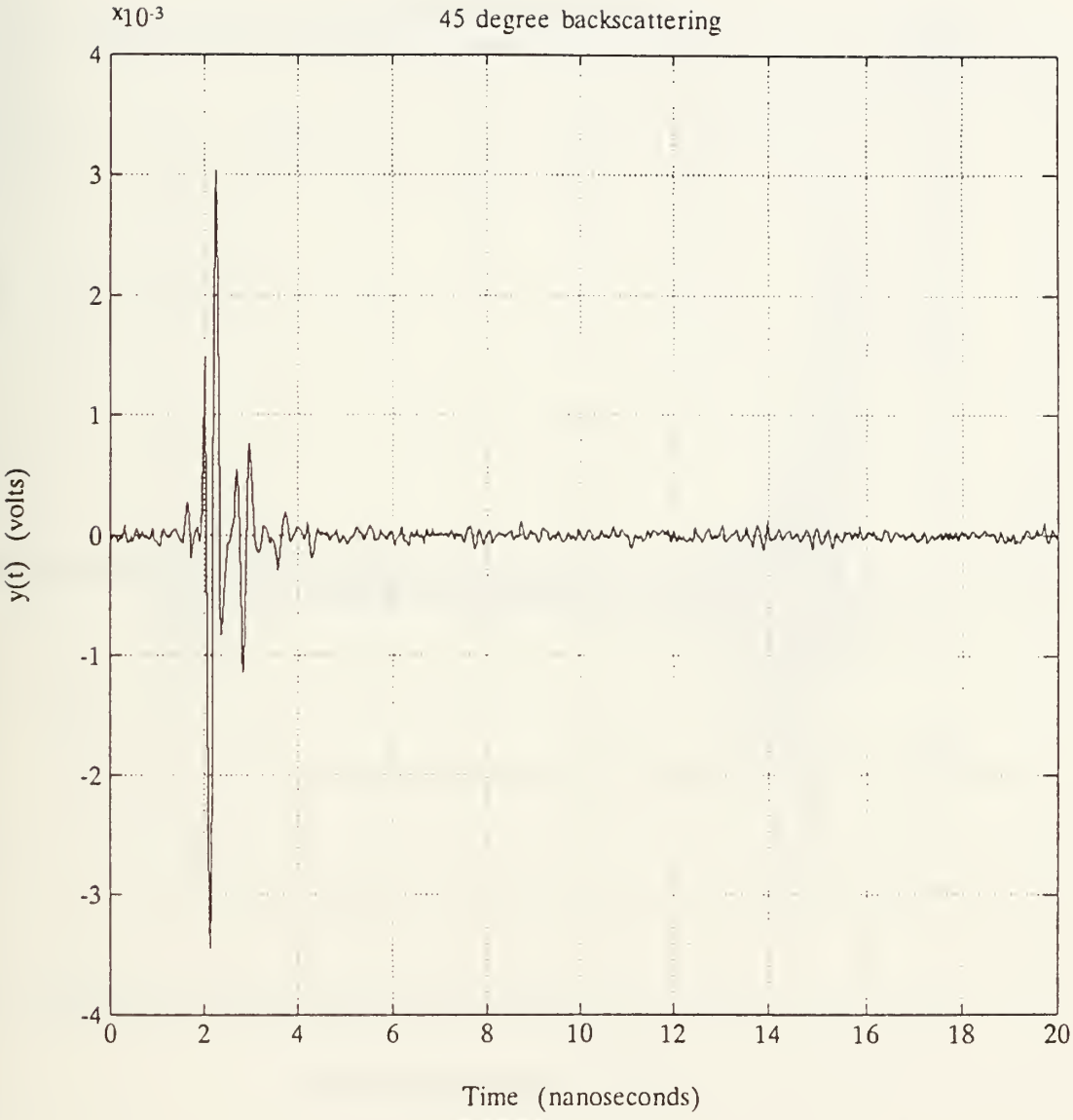

Figure 28. Measured Thin Wire Scattering, 45 Degree Aspect

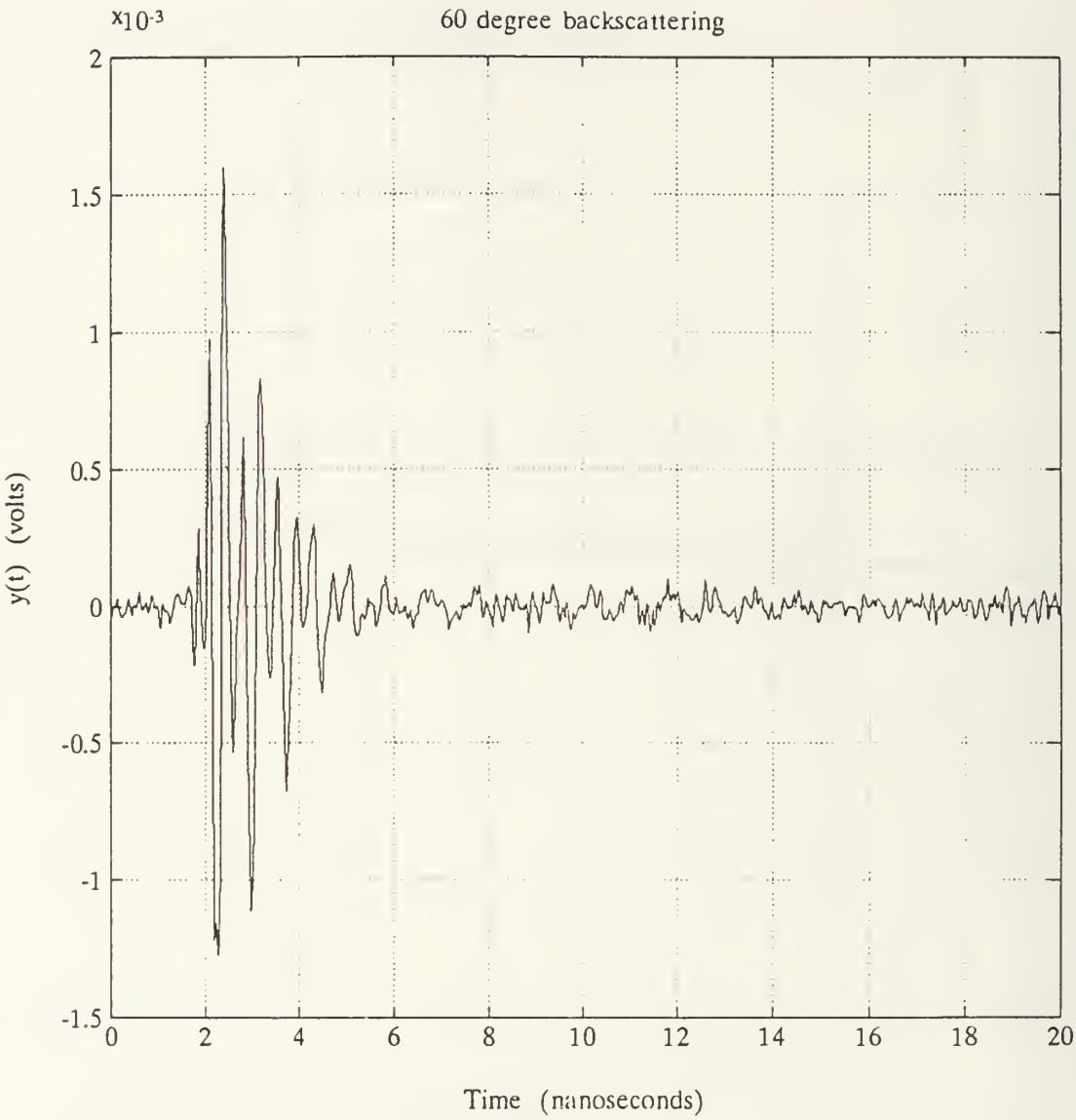

Figure 29. Measured Thin Wire Scattering, 60 Degree Aspect

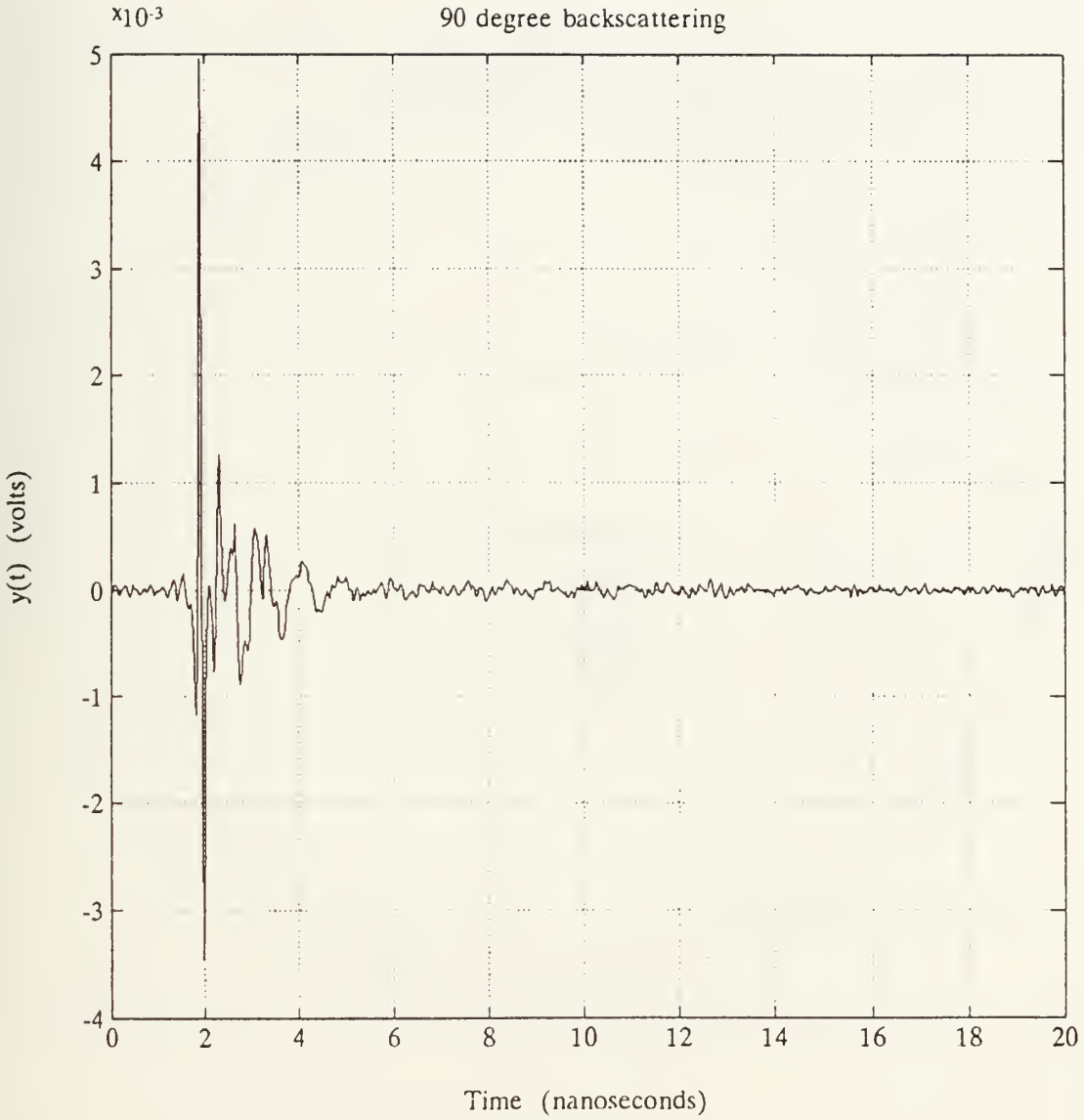

Figure 30. Measured Thin Wire Scattering, 90 Degree Aspect

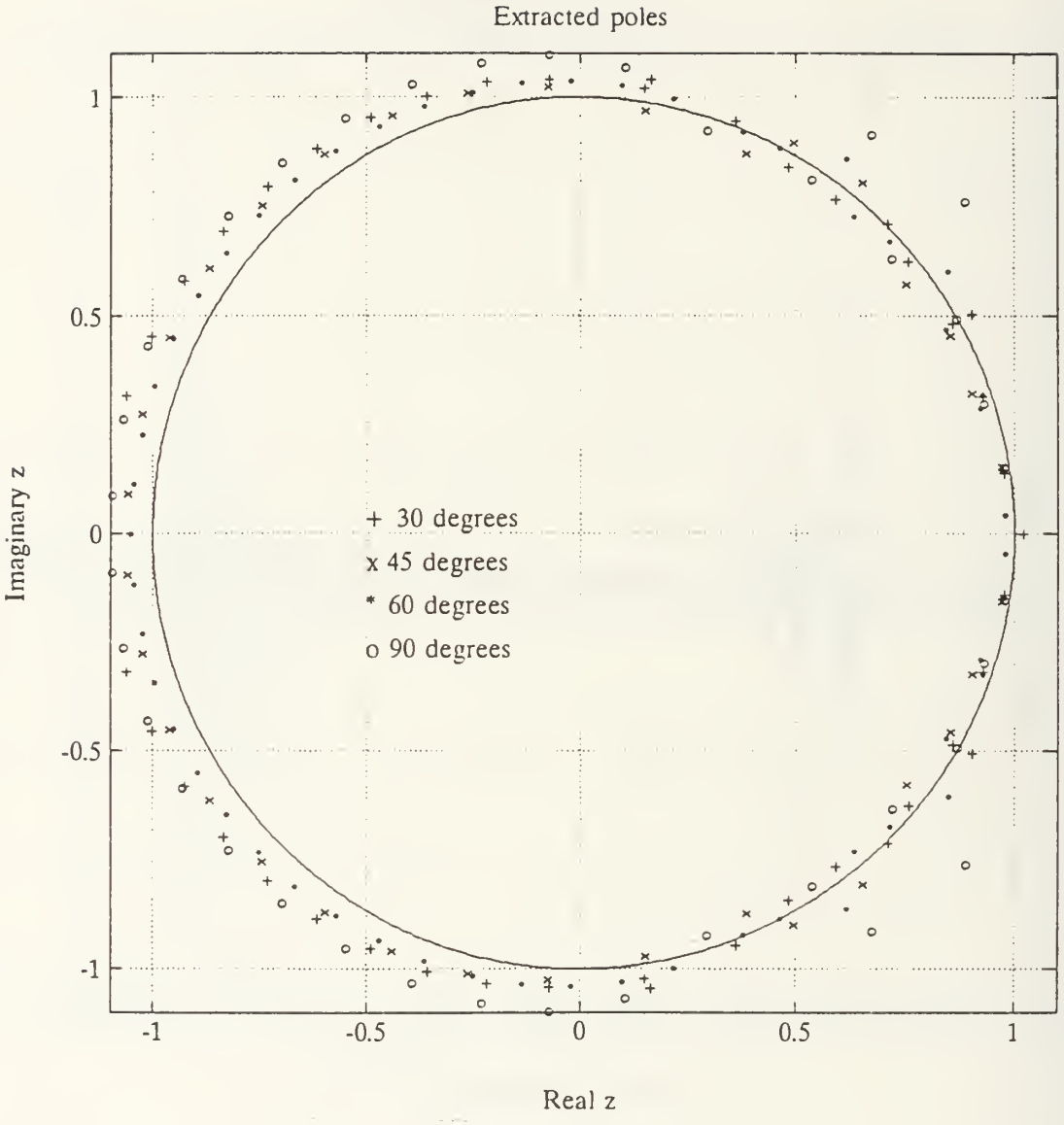

Figure 31. Kumaresan-Tufts Poles, Measured Thin Wire

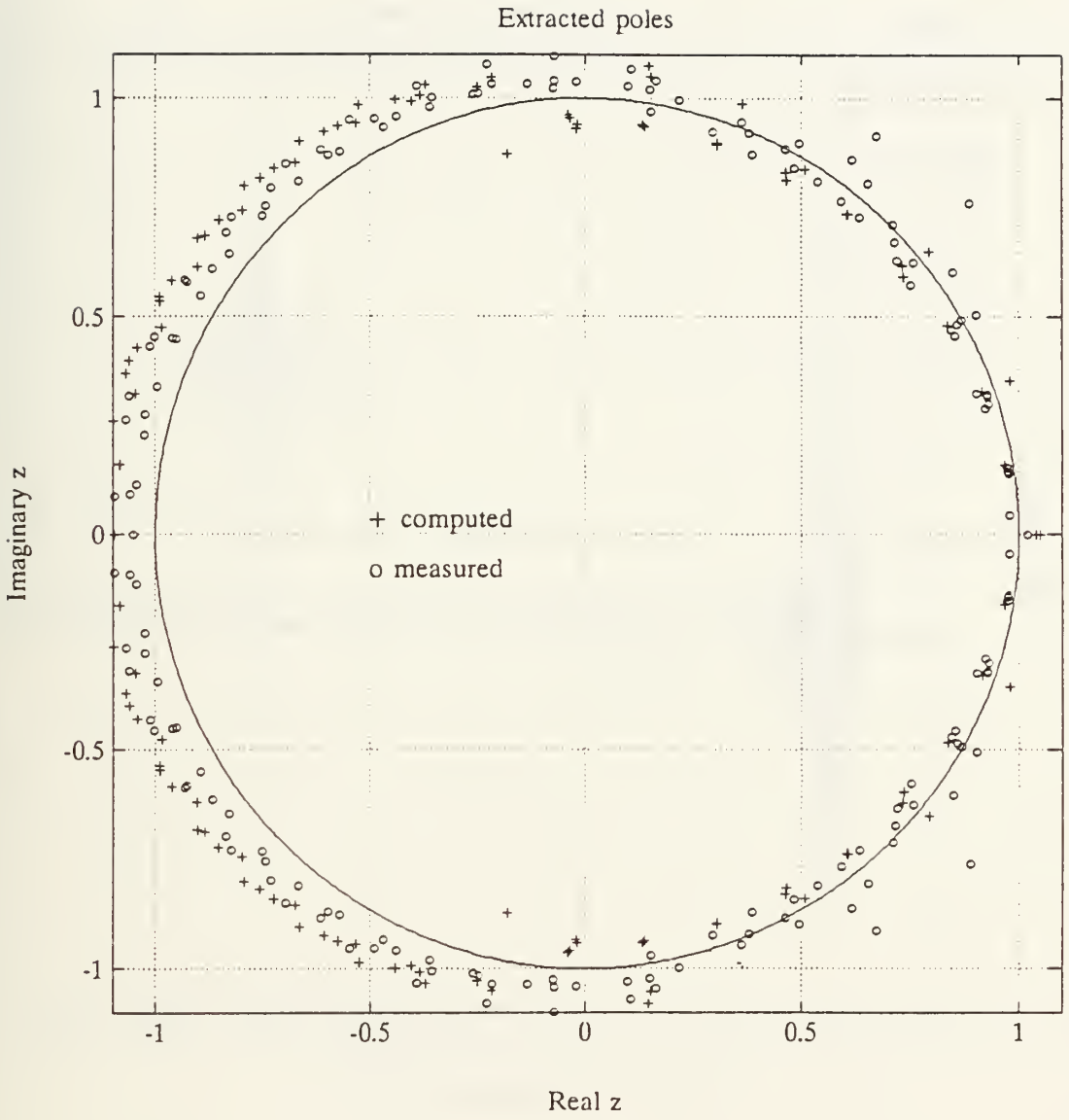

Figure 32. Thin Wire Comparison, Measured vs. Integral Equation

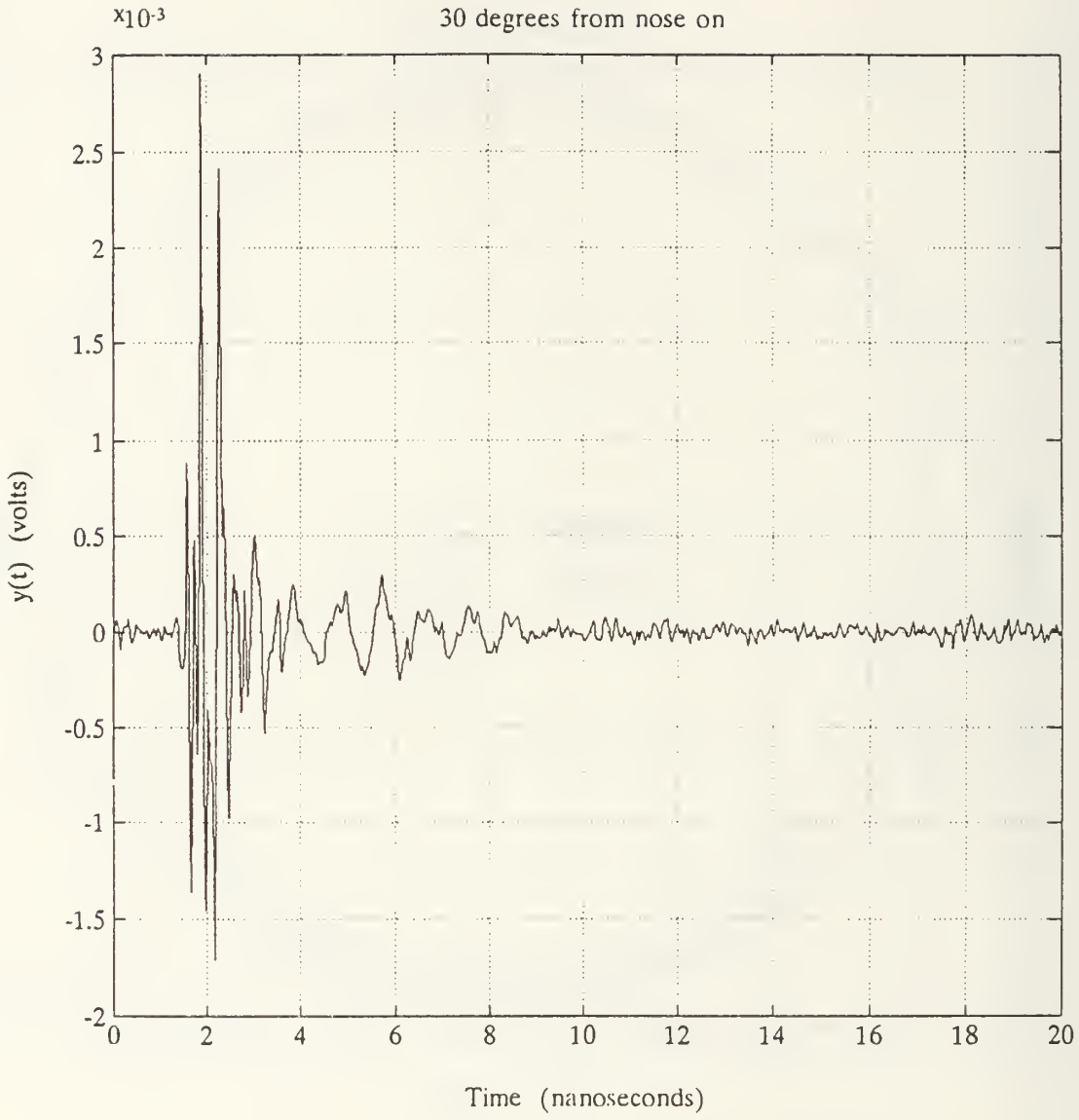

Figure 33. Target 1 Scattering, 30 Degrees from Nose on

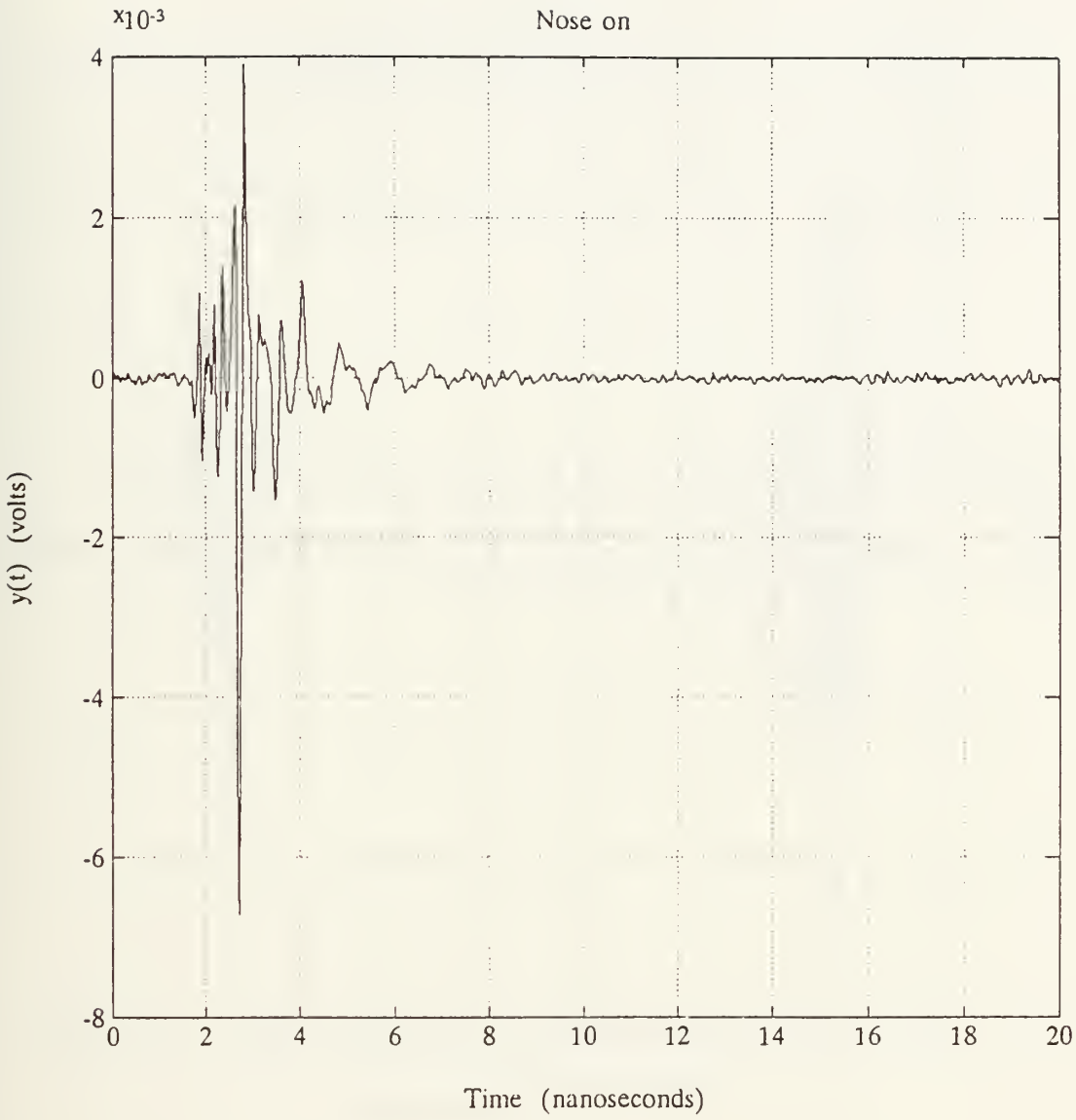

Figure 34. Target 1 Scattering, Nose on

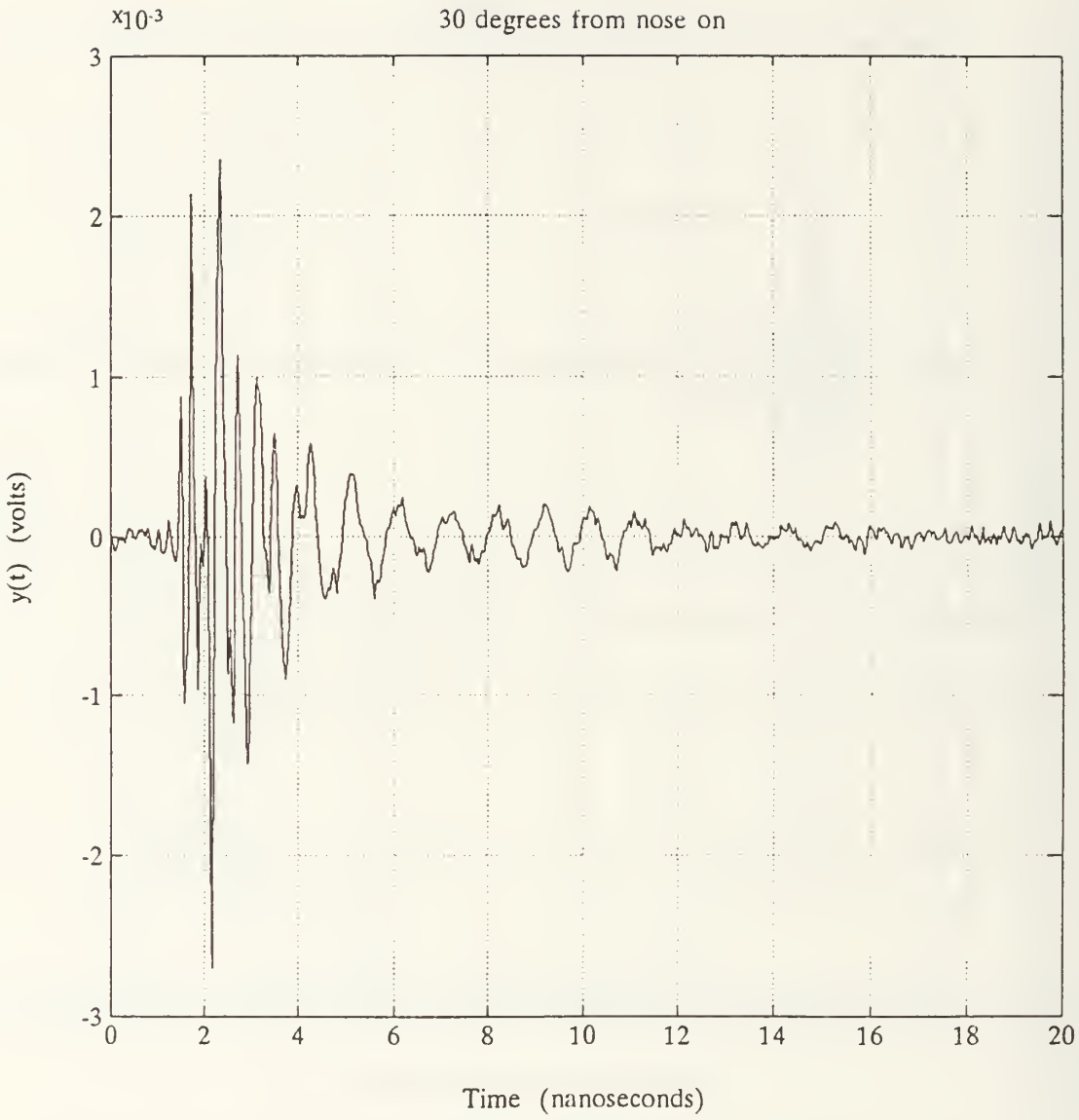

Figure 35. Target 2 Scattering, 30 Degrees from Nose on

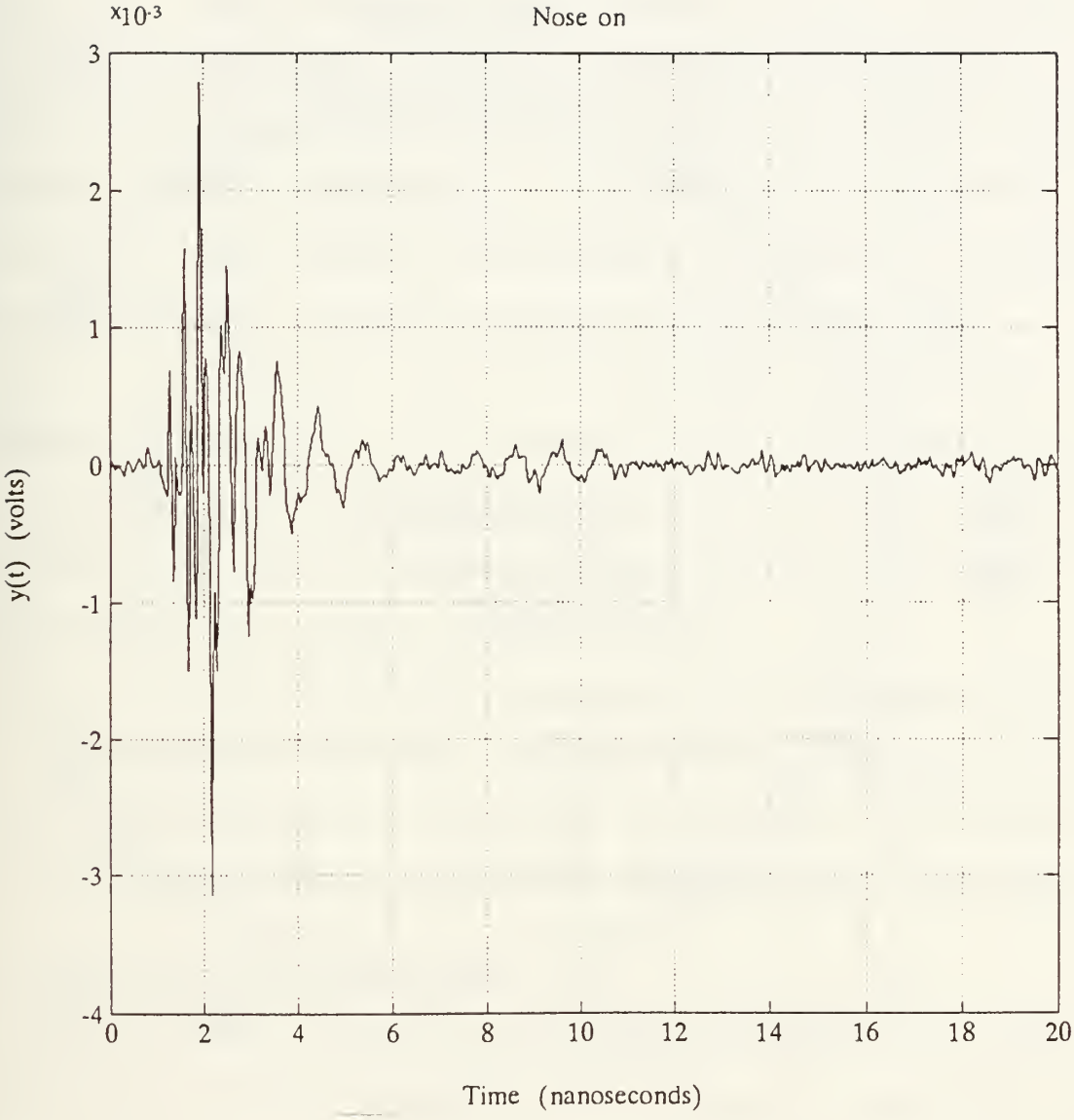

Figure 36. Target 2 Scattering, Nose on

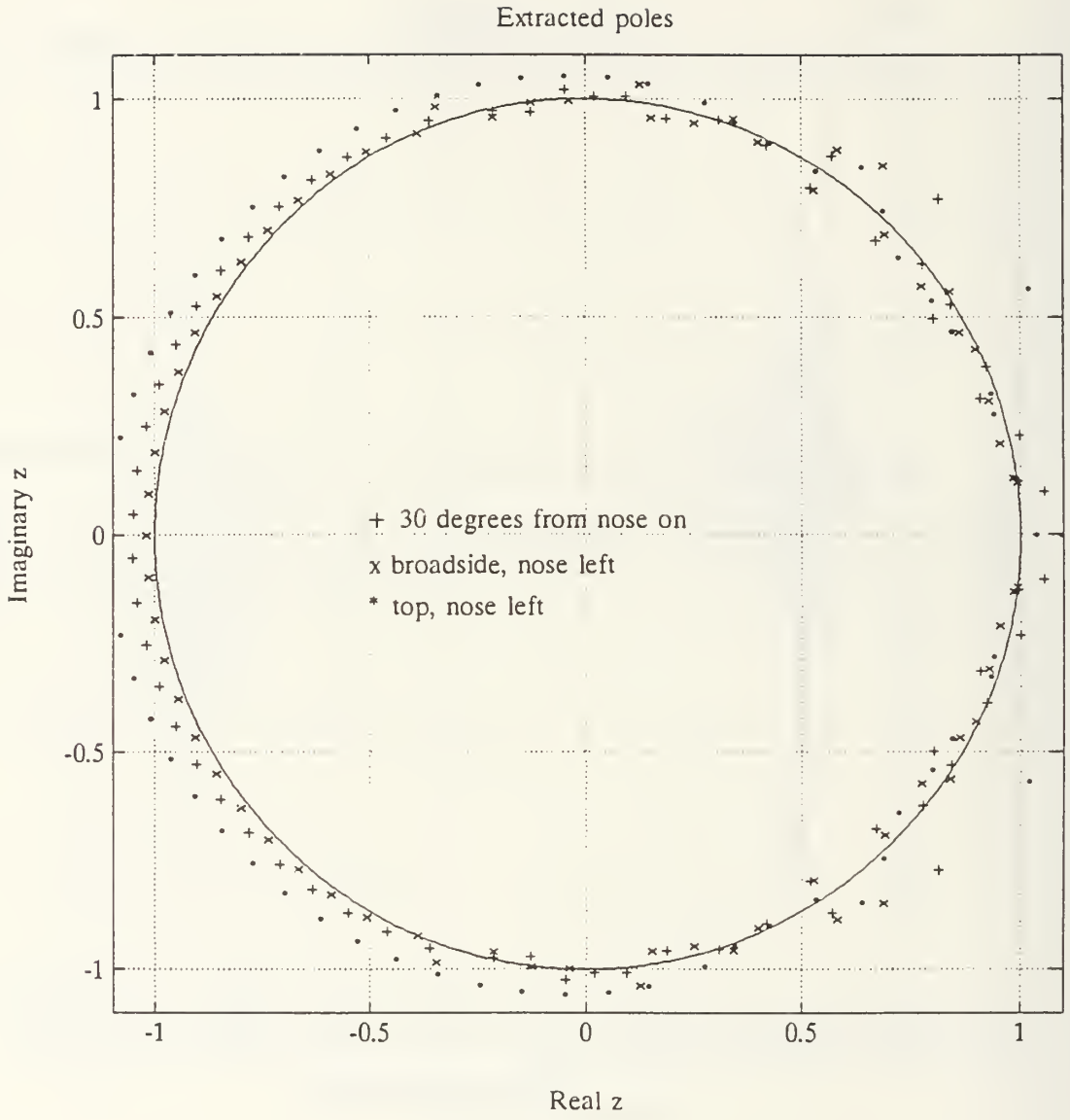

Figure 37. Kumaresan-Tufts Poles, Target 1

38. The poles extracted at all six aspects are shown in Figure 39. Only one clearly discernible cluster is present in each of the three figures. At higher frequencies, no useful information is imparted by the data. Results of similar, though slightly improved quality, were obtained from target 2. These results are presented in Figures 40 through 42 in the format of Figures 37 through 39 respectively.

Although the Kumaresan-Tufts algorithm is capable of extracting low frequency poles acceptably, the inconsistent results at higher frequences reveals the inherent weakness in an algorithm capable of processing only the late-time portion of a target's radar response.

A side-by-side comparison of poles obtained from both aircraft by both the Kumaresan-Tufts method and the Cadzow-Solomon method is presented at the end of the chapter to illustrate the gains afforded by processing the early-time.

C. CADZOW-SOLOMON ALGORITHM

Recall from the results depicted in Figure 8 that a late transition to late-time, and the consequent reduction of signal power, caused complete breakdown of the Kumerasan-Tufts algorithm. The Cadzow-Solomon algorithm addresses this shortcoming by processing the signal at the instantaneous onset of early-time. Thus, the Cadzow-Solomon algorithm is capable of processing the earliest response of a target to

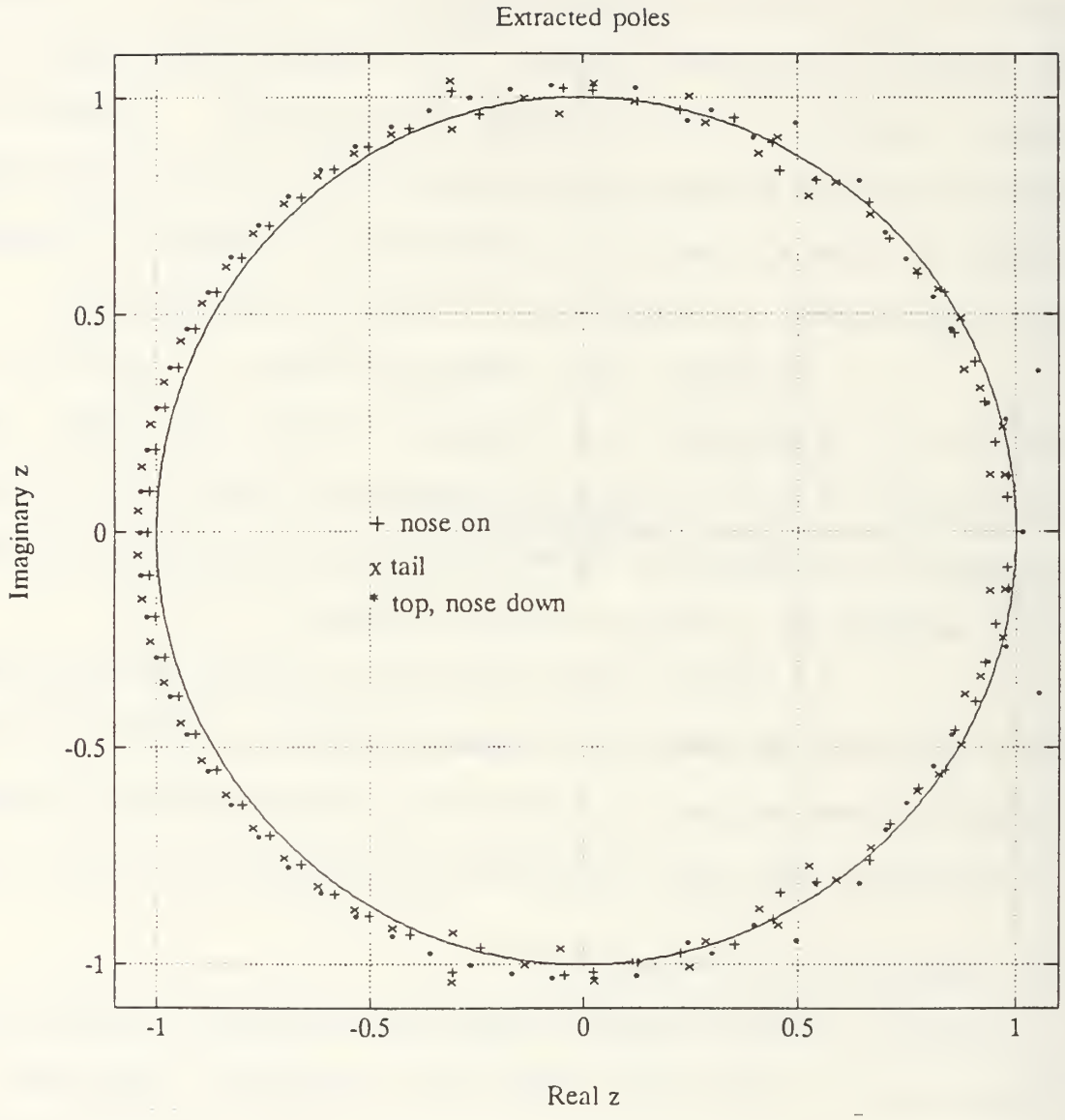

Figure 38. Kumaresan-Tufts Poles, Target 1

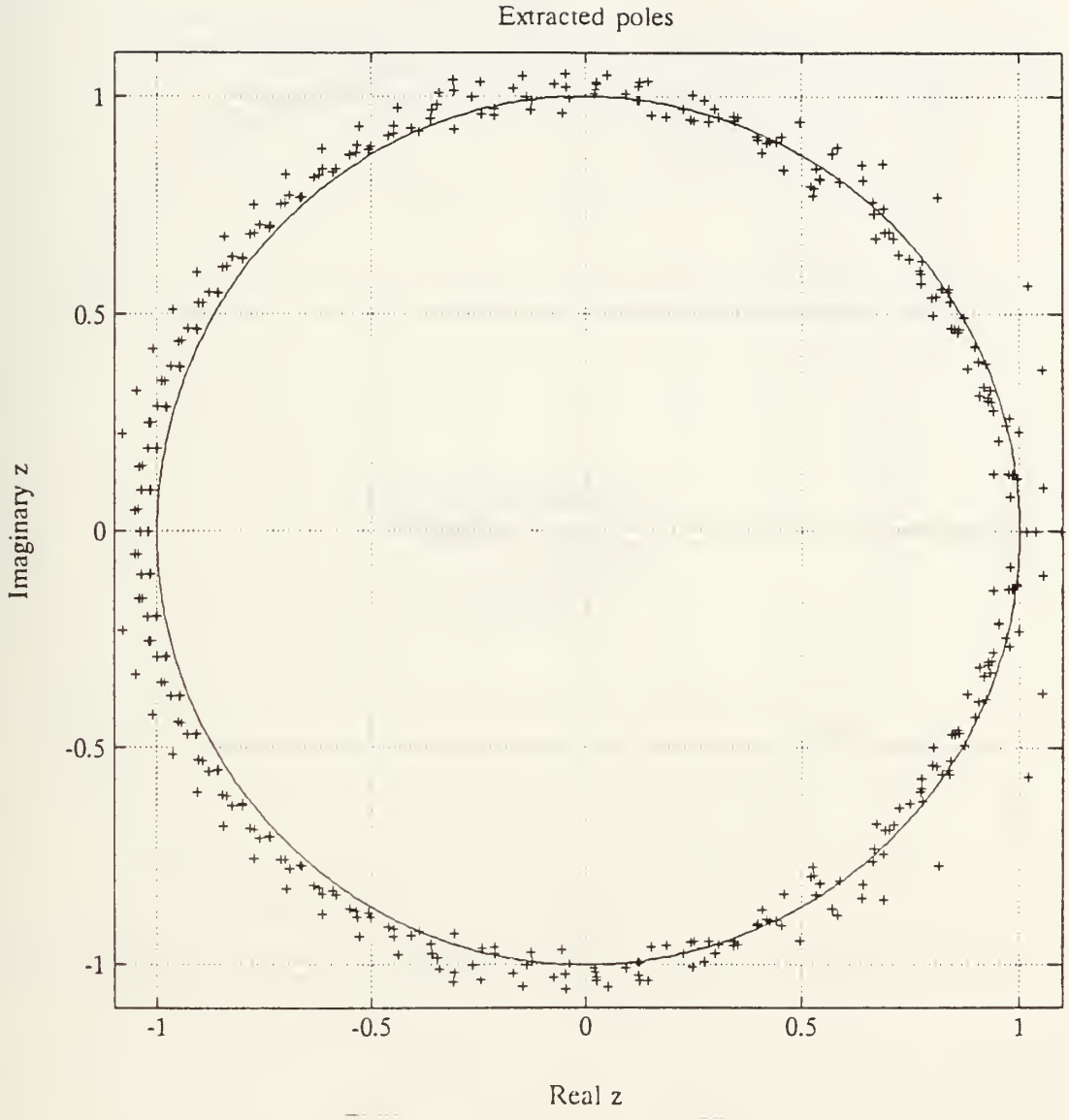

Figure 39. Kumaresan-Tufts Poles, Target 1

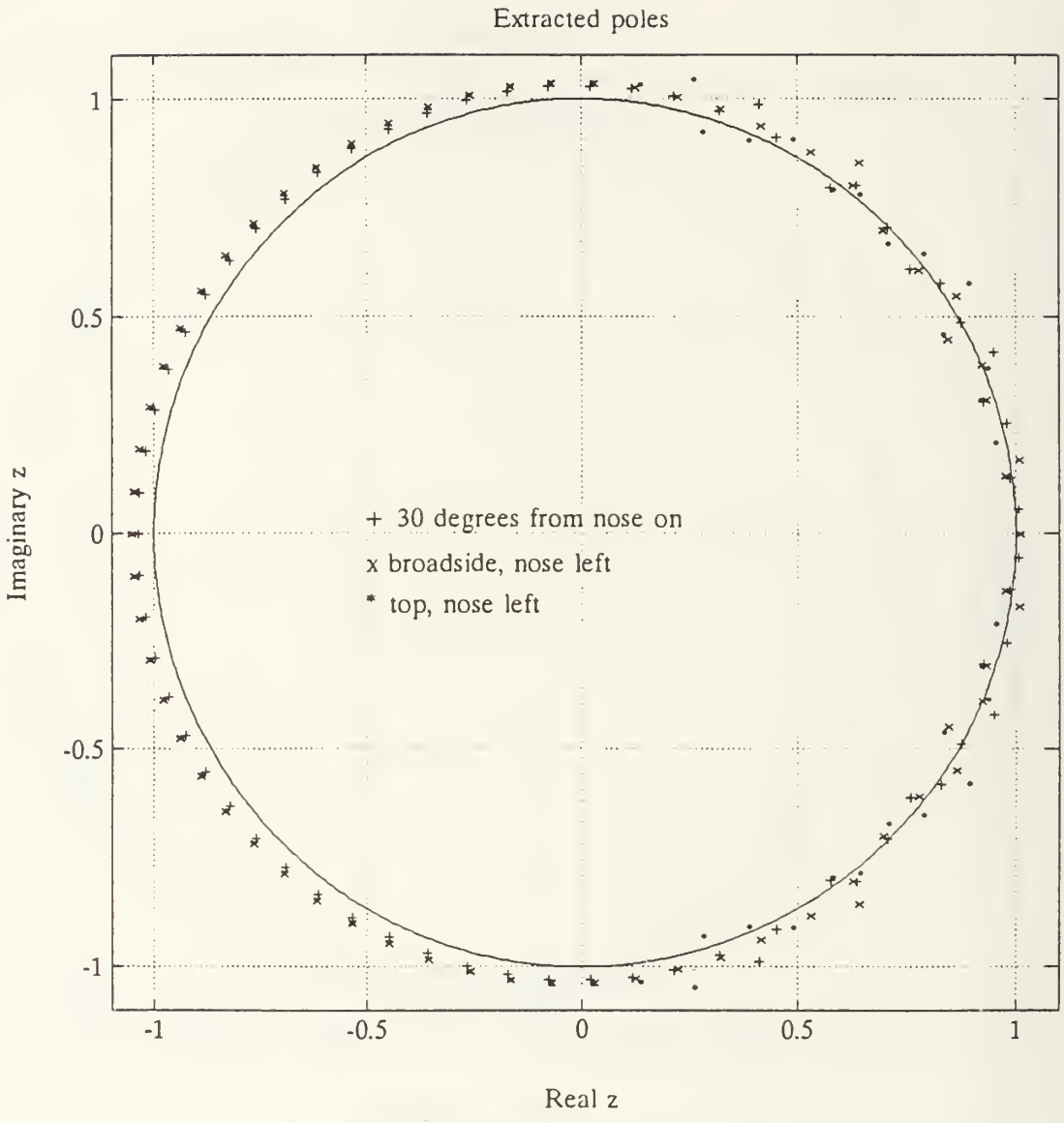

Figure 40. Kumaresan-Tufts Poles, Target 2

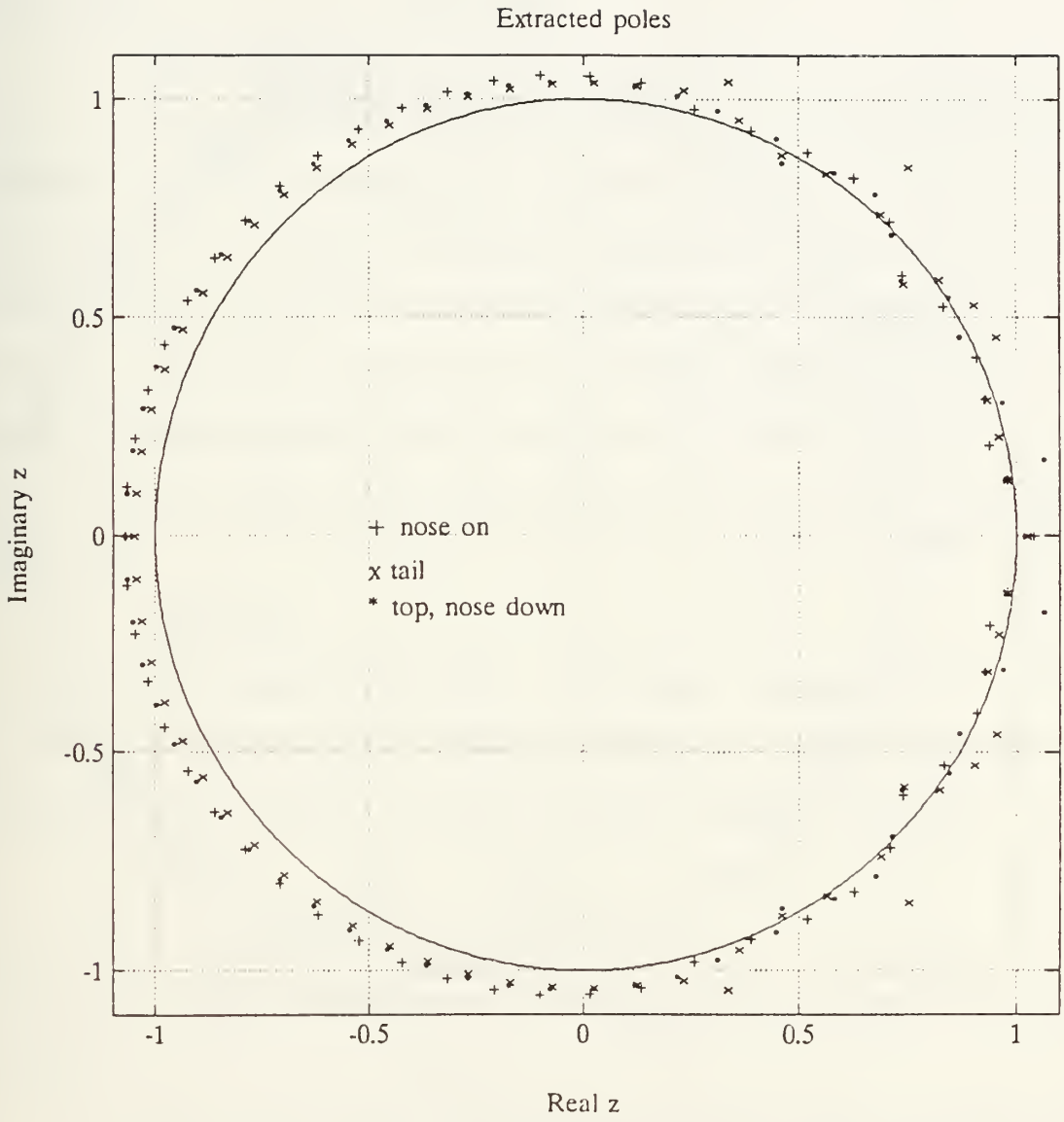

Figure 41. Kumaresan-Tufts Poles, Target 2

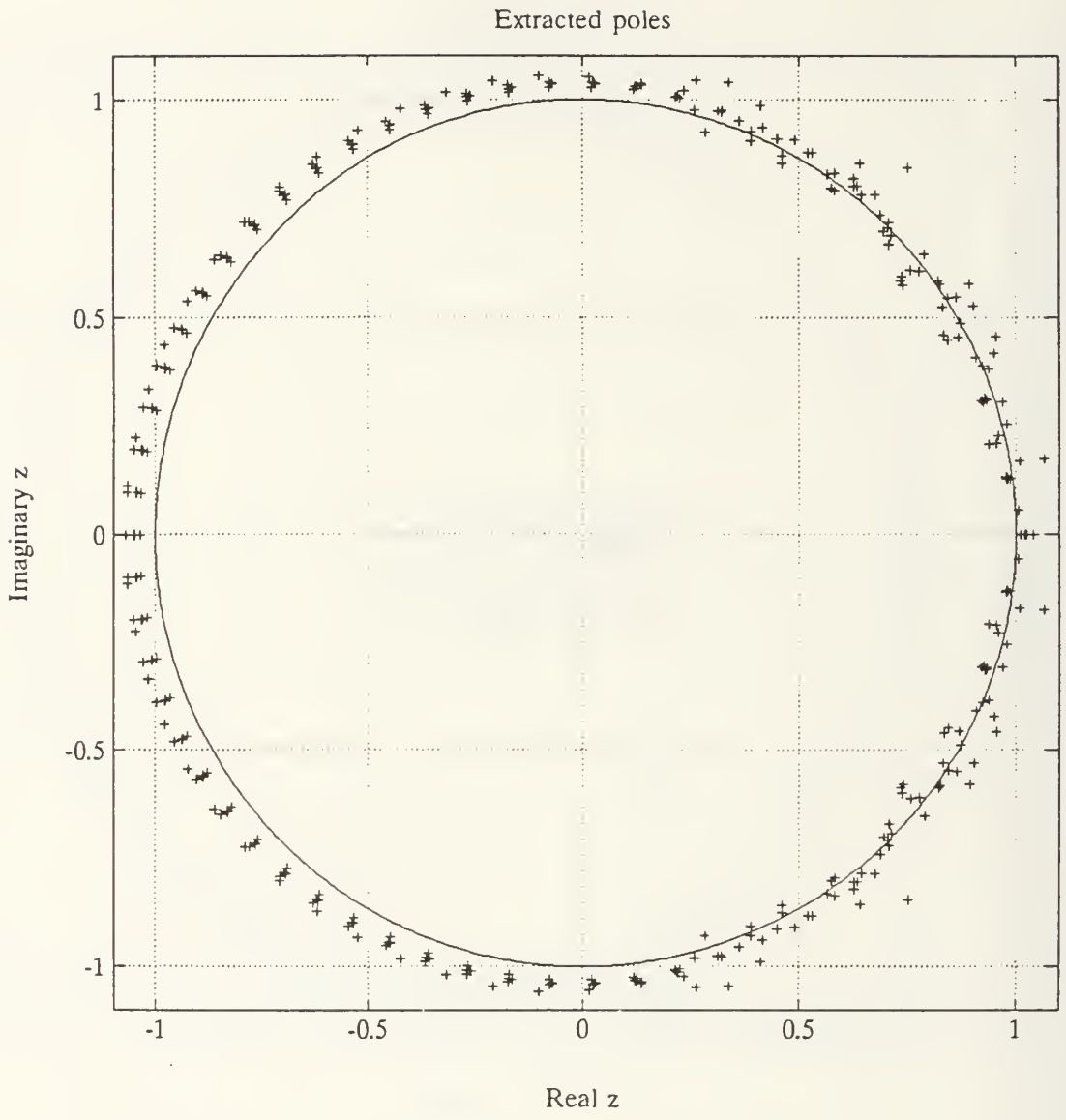

Figure 42. Kumaresan-Tufts Poles, Target 2

electromagnetic excitation where the response has the greatest magnitude.

1. Applicability

The early-time portion of a target's scattered field occurs as long as there is a driven portion of the total field. Once the field no longer contains a scattered response due, in part, to the incident excitation at points on the object, early-time ceases and late—time begins. Hence, the Cadzow-Solomon models both the system's input and output, and equivalently, the poles and zeros of the system transfer function.

2. Equations

The Cadzow-Solomon algorithm extends the auto-regressive equation (9), used in Prony's method, to the more general autoregressive moving average (ARMA) equation

$$y_n = \sum_{i=1}^{K_D} b_i y_{n-i} + \sum_{i=0}^{K_N} a_i x_{n-i} \tag{23}$$

where the second summation term models the excitation to the system.

A set of M such equations in matrix form is given by

$$\begin{bmatrix} y_0 & \cdots & y_{K_D-1} & x_0 & \cdots & x_{K_N} \\ \vdots & & \vdots & \vdots & & \vdots \\ y_{M-1} & \cdots & y_{K_D+M-2} & x_{M-1} & \cdots & x_{K_N+M-1} \end{bmatrix} \begin{bmatrix} b'_{K_D} \\ \vdots \\ b'_1 \\ \hline a_{K_N} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} y_{K_D} \\ \vdots \\ y_{K_D+M-1} \end{bmatrix} \tag{24}$$

As in the Kumaresan-Tufts method, M is selected to be greater than the column dimension of the data matrix which is K_D+K_N+1.

3. Excess Poles and Noise Removal

The Cadzow-Solomon method used in this thesis is a modification which incorporates the non-causal arrangement of the system equations used by Kumaresan-Tufts. This modification was first discussed by Norton in [16]. The Kumaresan approach of overestimating the system order can be used as before in a non-causal model to constrain the noise poles inside the unit circle, while SVD forces the signal poles outside the unit circle.

Since the input waveform is known, its order can be almost exactly determined. In all the work of this thesis, the input waveform used is the double Gaussian depicted in Figure 14. Approximately 25 samples defining this pulse of 0.5 nanoseconds duration makes K_N equal 25 in equation (23). Since the input is causal, the signal zeros fall inside the unit circle where they cannot be easily segregated from similarly located noise poles. However, the signal zeros impart no information about the target and need not be extracted. The inclusion of the input in the data matrix is nevertheless vital to the model of the system and the accurate determination of the signal poles.

The ARMA equation of (23) can be modified to obtain

$$y_n = \sum_{i=1}^{K_D} b'_i y_{K_D+n-i+1} + \sum_{i=0}^{K_N} a_i x_{n-i} \qquad (25)$$

The recursive portion of (25) is now in a non-causal form similar to expression (12). A set of M such equations in matrix form is given by

$$\begin{bmatrix} y_{K_N+1} & \cdots & y_{K_N+K_D} & x_0 & \cdots & x_{K_N} \\ \vdots & & \vdots & \vdots & & \vdots \\ y_{K_N+M} & \cdots & y_{K_N+K_D+M-1} & x_{M-1} & \cdots & x_{K_N+M-1} \end{bmatrix} \begin{bmatrix} b'_{K_D} \\ \vdots \\ b'_1 \\ \hline a_{K_N} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} y_{K_D} \\ \vdots \\ y_{K_D+M-1} \end{bmatrix} \qquad (26)$$

Or, in matrix notation

$$[D_{yx}]\left[\frac{b'}{a}\right] = y \quad \text{where} \quad [D_{yx}] = [D_y : D_x] \qquad (27)$$

4. Singular Value Decomposition

Like the system equations of the Kumaresan-Tufts model, the system equations in (26) are processed using singular value decomposition. The coefficient vector is again the minimum-norm solution, which constrains the extraneous poles and extraneous zeros to be inside the unit circle.

5. Bias Compensation in the Cadzow-Solomon Formulation

By compensating the eigenvalues of the Σ matrix in (16), the performance of the Kumaresan-Tufts algorithm is significantly improved in the presence of noise. Cadzow-Solomon have shown [5] that if the actual orders K'_D and K'_N

are overestimated to be K_D and K_N, min (K_D-K_D', K_N-K_N') singular values are zero in noiseless data. Since the input data is known, the eigenvalues of the data matrix may be compensated in the same manner as in the Kumaresan-Tufts algorithm for noiseless data.

To understand the compensation required in noisy data, an analysis of additive noise is required. As given by Norton [16], if the input data noise is w_1 and the output data noise is v_1, the data matrix may be modeled as

$$[D_{yx}]=[D_y:D_x]=S_{yx}+N_{yx} \tag{28}$$

where

$$[N_{yx}]=[N_y:N_x] \tag{29}$$

and

$$N_x=\begin{bmatrix} 1 & K_D \\ \vdots & \vdots \\ w_M \cdots & w_{M+K_D} \end{bmatrix} \qquad N_y=\begin{bmatrix} v_1 \cdots & v_{K_D} \\ \vdots & \vdots \\ v_M \cdots & v_{M+K_D} \end{bmatrix} \tag{30}$$

The expected value of $D_{yx}D_{yx}^T$ is then

$$E[D_{yx}D_{yx}^T]=S_{yx}S_{yx}^T+E[N_{yx}N_{yx}^T] \tag{31}$$

If the input and output noise variances are not equal, the eigenvalue shifting theorem used in Kumaresan-Tufts cannot be used to analytically predict the requisite eigenvalue compensation of $D_{yx}D_{yx}^T$. Nevertheless, when the input and output variances were assumed equal, and eigenvalue compensation similar to that used in Kumaresan-Tufts was

performed, the results were consistently superior to those obtained without compensation. Therefore, the results of Cadzow-Solomon signal processing presented in this thesis were obtained using eigenvalue compensation and the assumption of equal noise variance.

6. Performance

The Cadzow-Solomon algorithm was programmed in Fortran and tested on the same data used for evaluating the Kumaresan-Tufts algorithm. Note that the Cadzow-Solomon algorithm can use the early-time portion of the data that the Kumaresan-Tufts algorithm can not use. The program appears in Appendix B.

a. Synthetically Generated Data

The starting point for evaluating the performance of the Cadzow-Solomon algorithm was with synthetically generated data of the form given by (8) plus the addition of input data required to model early time data.

1. Noise Performance

The algorithm was evaluated at various signal-to-noise ratios, ranging from 90.0 dB to 7.0 dB. Figure 43 shows the signal produced by two s-plane poles at 90.0 dB, with a late-time beginning at 10.0 nanoseconds. Figures 44 through 48 depict the poles extracted from this signal at the different signal-to-noise ratios.

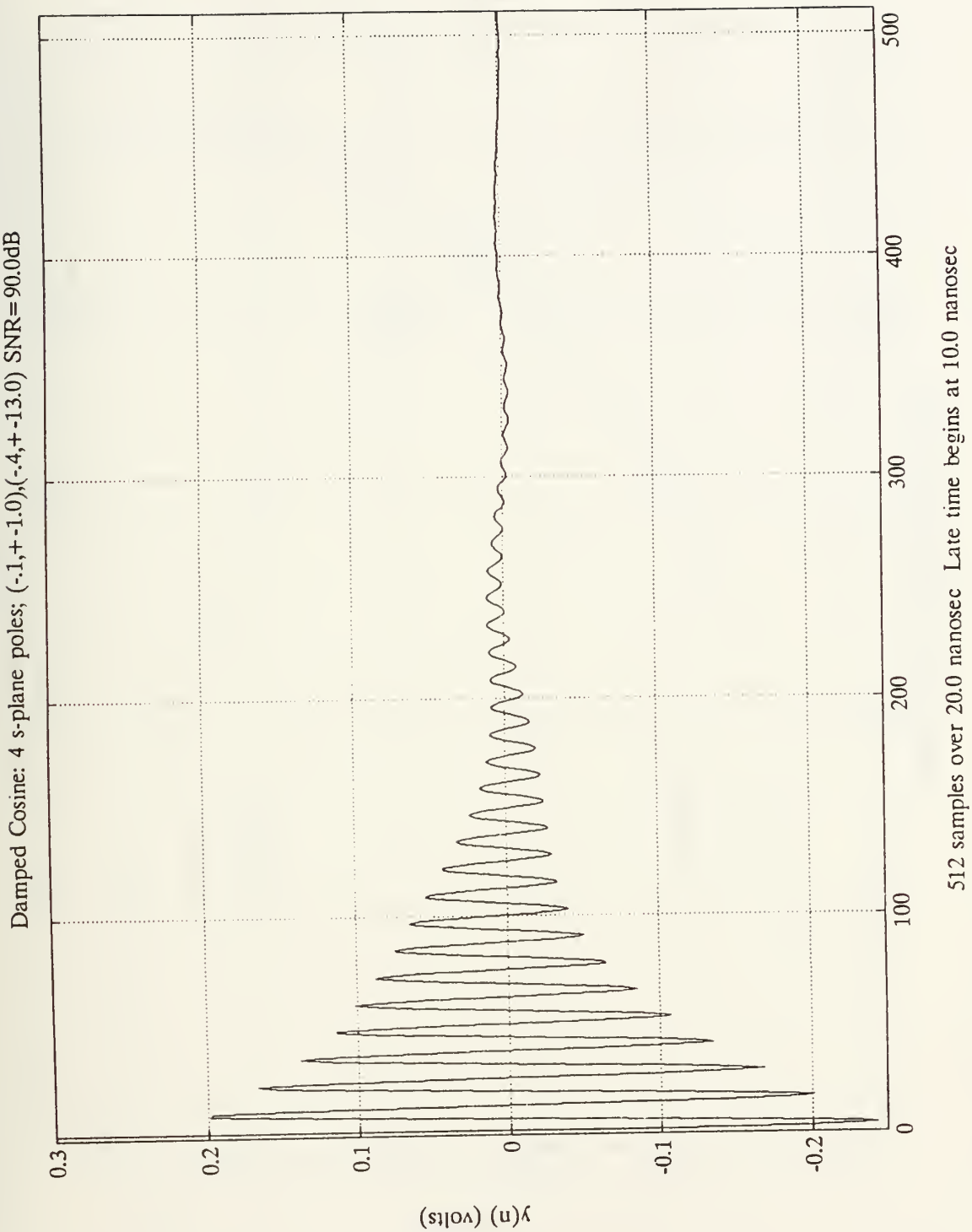

Figure 43. Signal Containing Two S-Plane Poles, 90.0 dB SNR

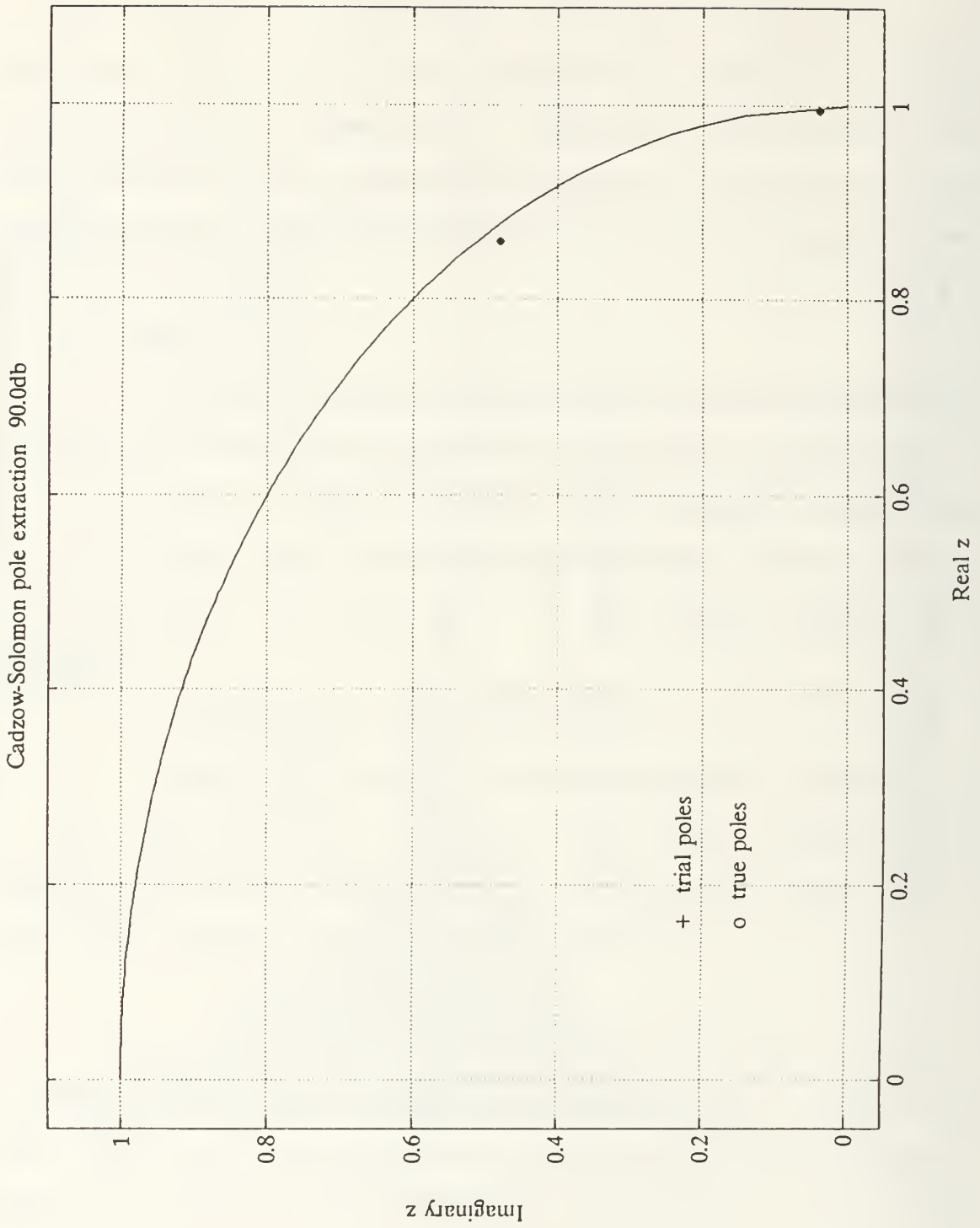

Figure 44. Cadzow-Solomon Poles, Synthetic Data, 90.0 dB SNR

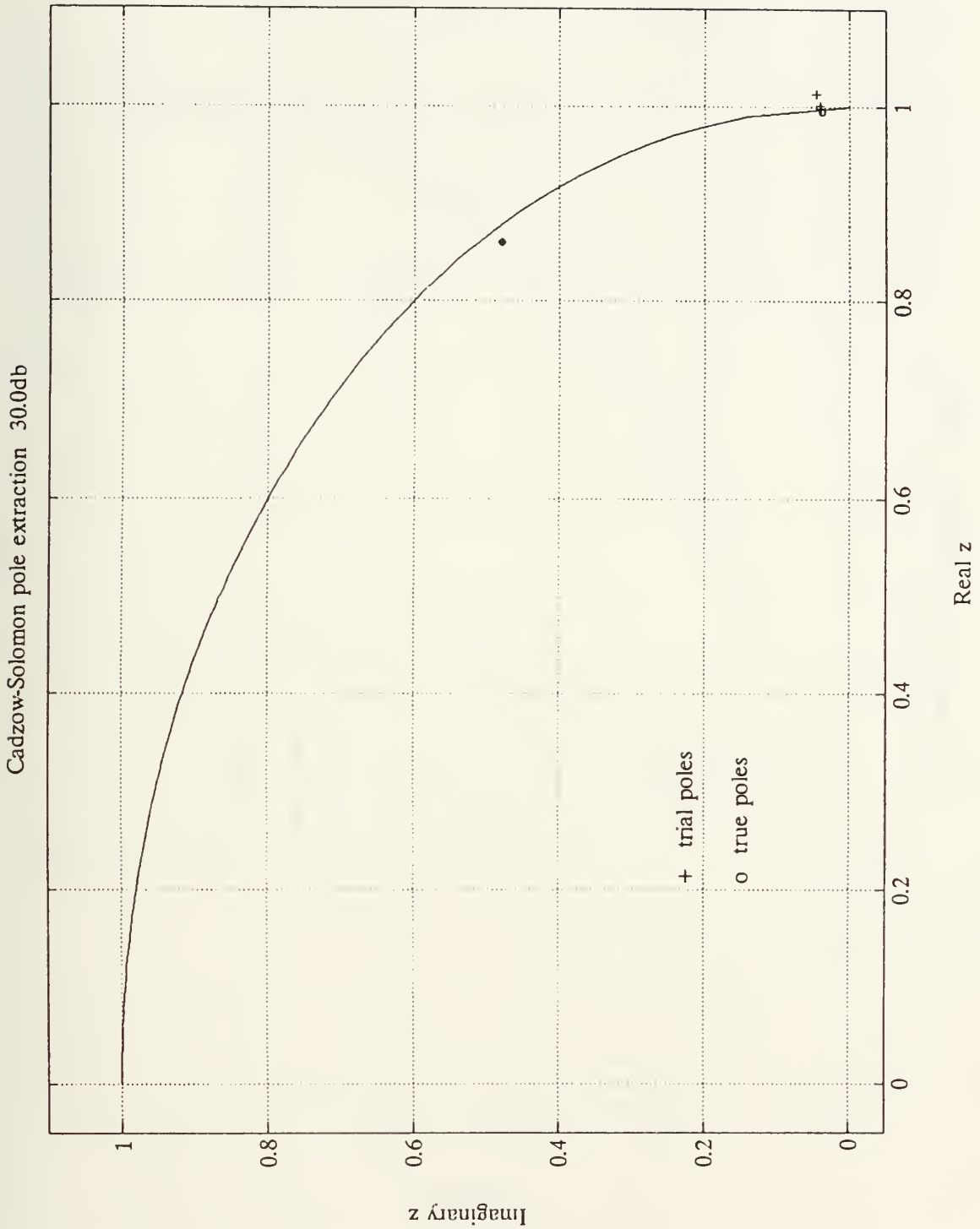

Figure 45. Cadzow-Solomon Poles, Synthetic Data, 30.0 dB SNR

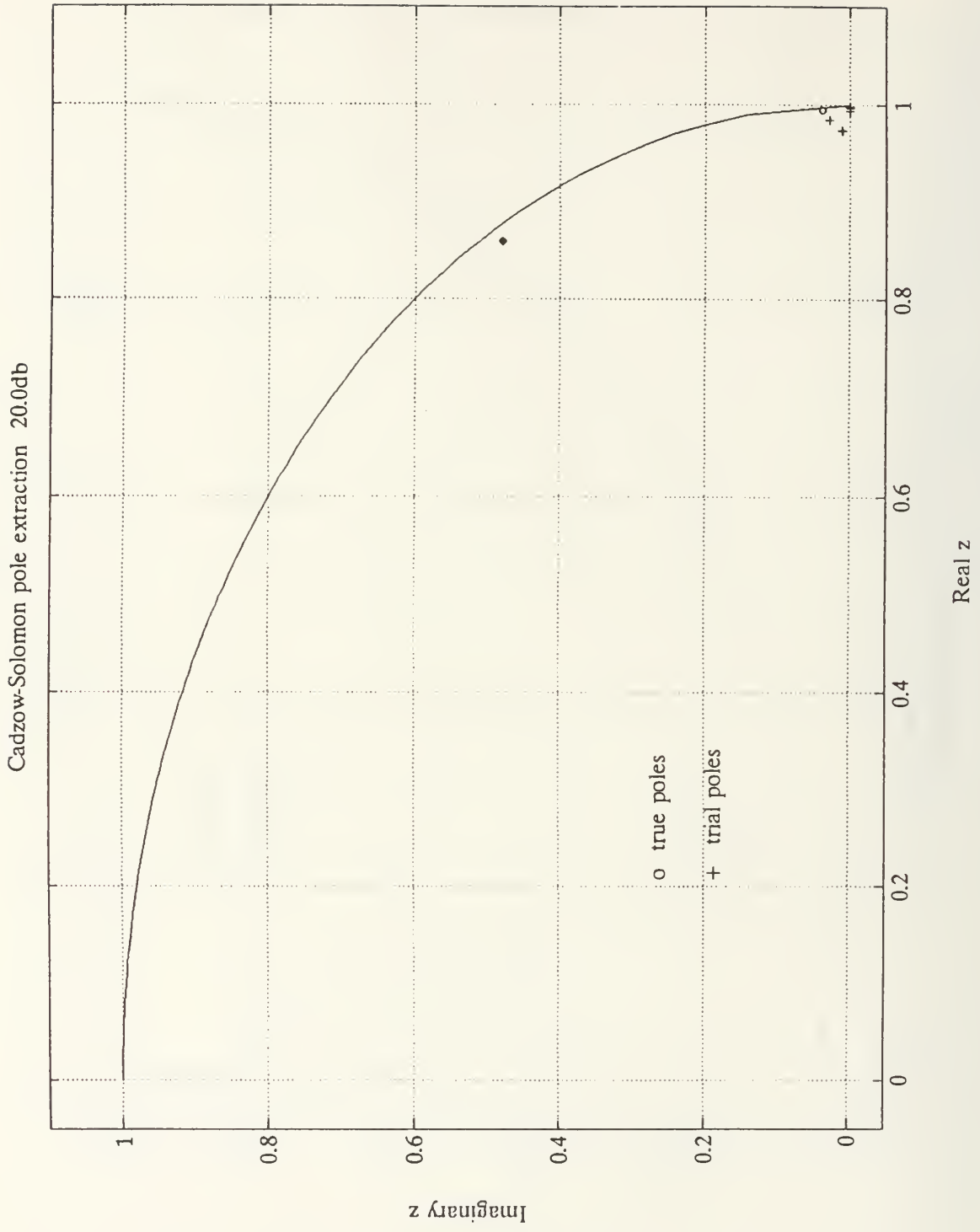

Figure 46. Cadzow-Solomon Poles, Synthetic Data, 20.0 dB SNR

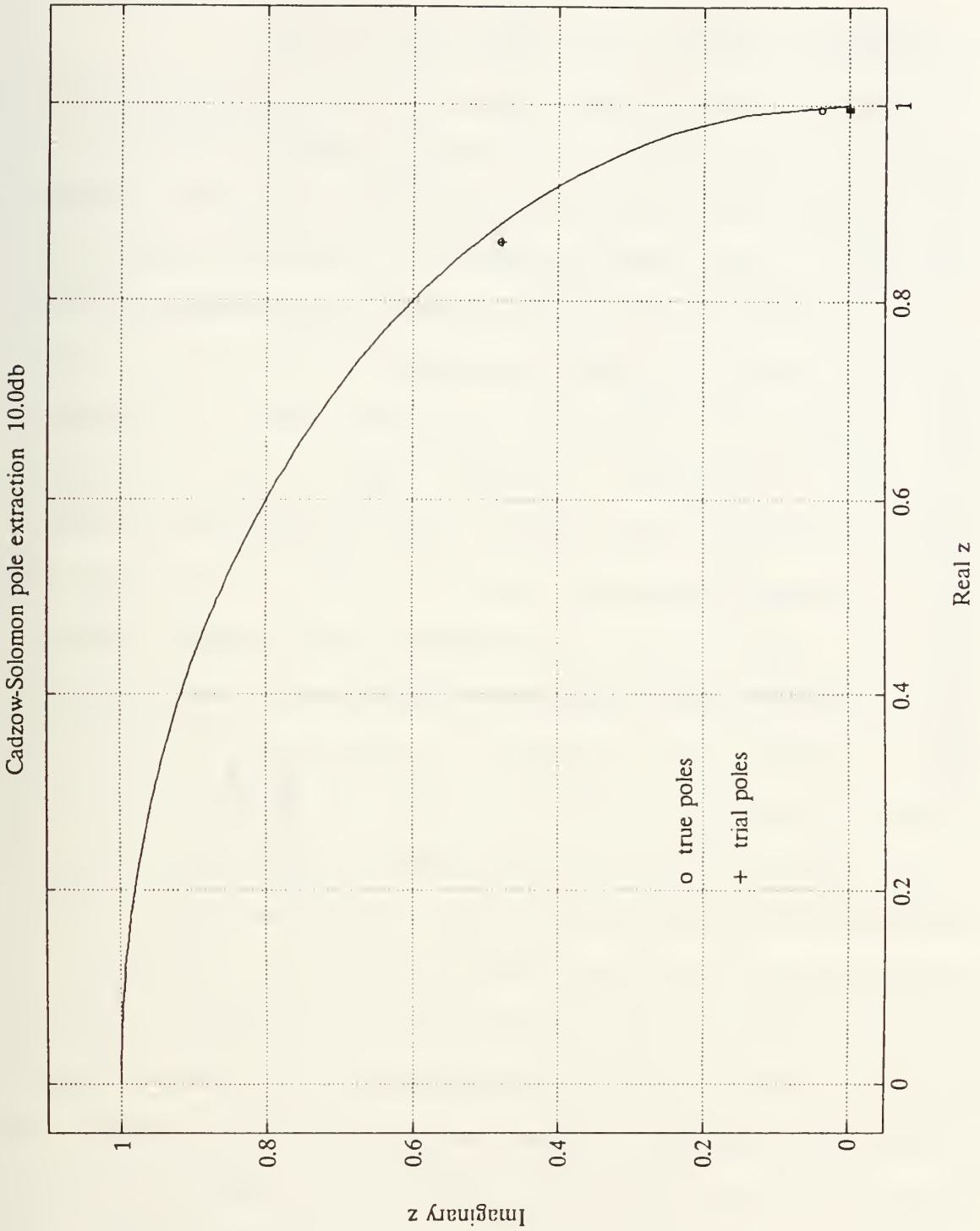

Figure 47. Cadzow-Solomon Poles, Synthetic Data, 10.0 dB SNR

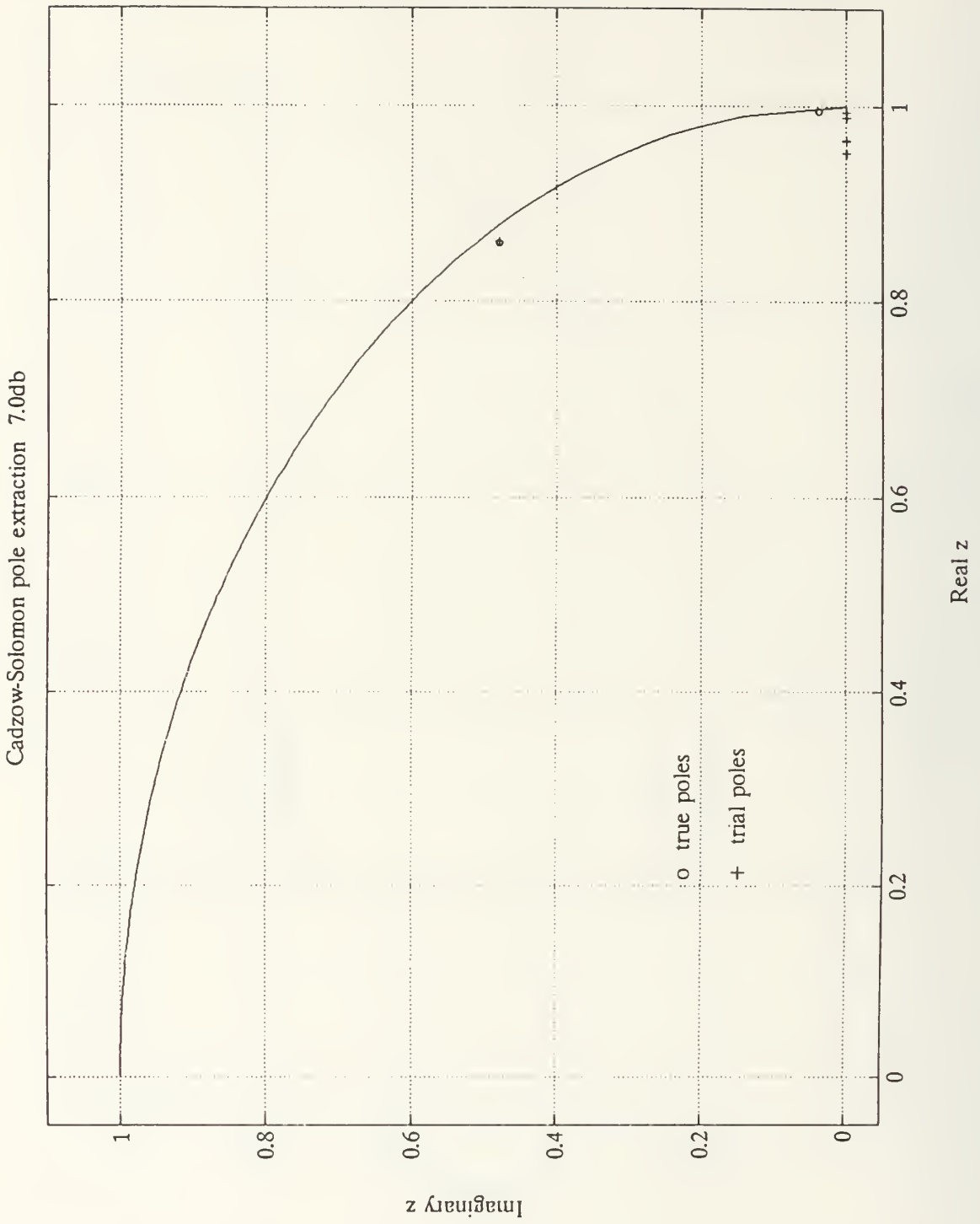

Figure 48. Cadzow-Solomon Poles, Synthetic Data, 7.0 dB SNR

The figures chart the steady degradation of the algorithm's performance with the increase of noise. At 30.0 dB, the location of the low frequency pole is already slightly displaced. More significant is the location of one of the extracted poles in the noise signal space. At 20.0 dB, the low frequency pole is located in some trials on the real axis. At 10.0 dB, all the extractions are located on the real axis and at 7.0 dB their locations there are dispersed. The extraction of the higher frequency pole is uncharacteristically more accurate than that of the low frequency pole. Even at 7.0 dB, the high frequency pole is located with excellent accuracy. The location of the low frequency pole near the real axis was chosen deliberately to illustrate the difficulty in resolving the slight frequency difference between the true pole and a noise pole located on the real axis. Also, fewer points were processed using the Cadzow-Solomon method than were processed using the Kumaresan-Tufts method, since the largest data matrix allowed by the programs in Appendices A and B contain fewer data points in the Cadzow-Solomon data matrix than in the Kumaresan-Tufts data matrix. The results demonstrate the need to process a substantial number of points in order to accurately extract low frequency poles.

b. Thin Wire Integral Equation Generated Data

The performance of the Cadzow-Solomon algorithm was evaluated using the same set of data tested by the Kumaresan-Tufts algorithm. The results are presented in Figure 49. Tight clusters appear at frequencies higher than those obtained with the Kumaresan-Tufts algorithm. Figure 50 depicts the poles extracted from the same signal at a 20.0 dB SNR. The clustering at this SNR is comparable to the results obtained by the Kumaresan-Tufts method with the noiseless data. Further angle-by-angle comparisons of the poles extracted from the noiseless data and the 20.0 dB data are depicted in Figures 51 through 54. Note the small number of poles in Figure 54 due to the unexcited odd-symmetric poles at 90° aspect.

One further test was conducted on computed data at a 7.0 dB SNR. The results are depicted in Figure 55. Even at 7.0 dB, discernible clusters are present. Angle-by-angle comparisons of the poles obtained in 7.0 dB data and those obtained in noiseless data are presented in Figures 56 through 59.

c. Scale Models

The same scale models used to evaluate the Kumaresan-Tufts algorithm were used to evaluate the Cadzow-Solomon algorithm.

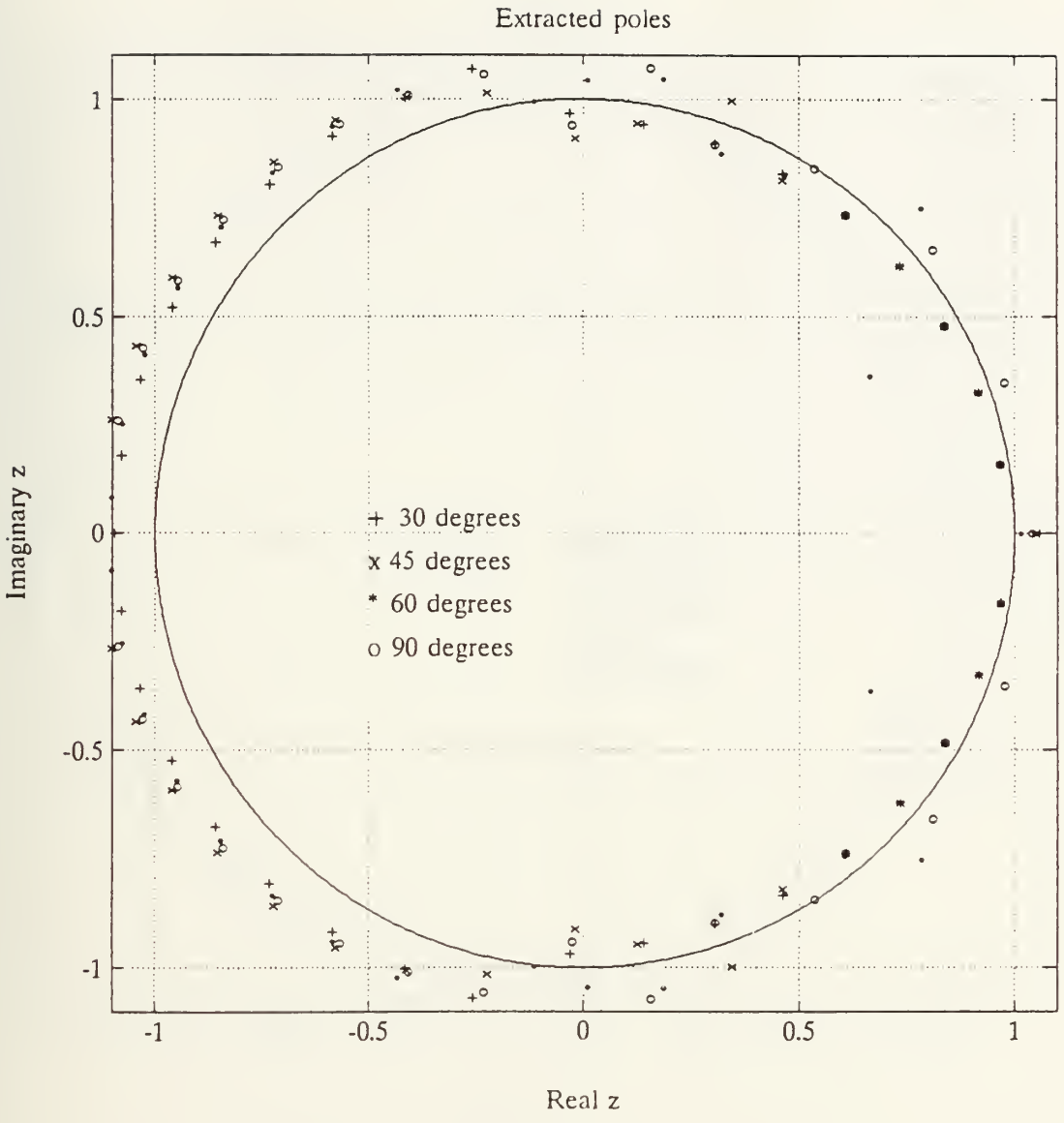

Figure 49. Cadzow-Solomon Poles, Noiseless Thin Wire Data

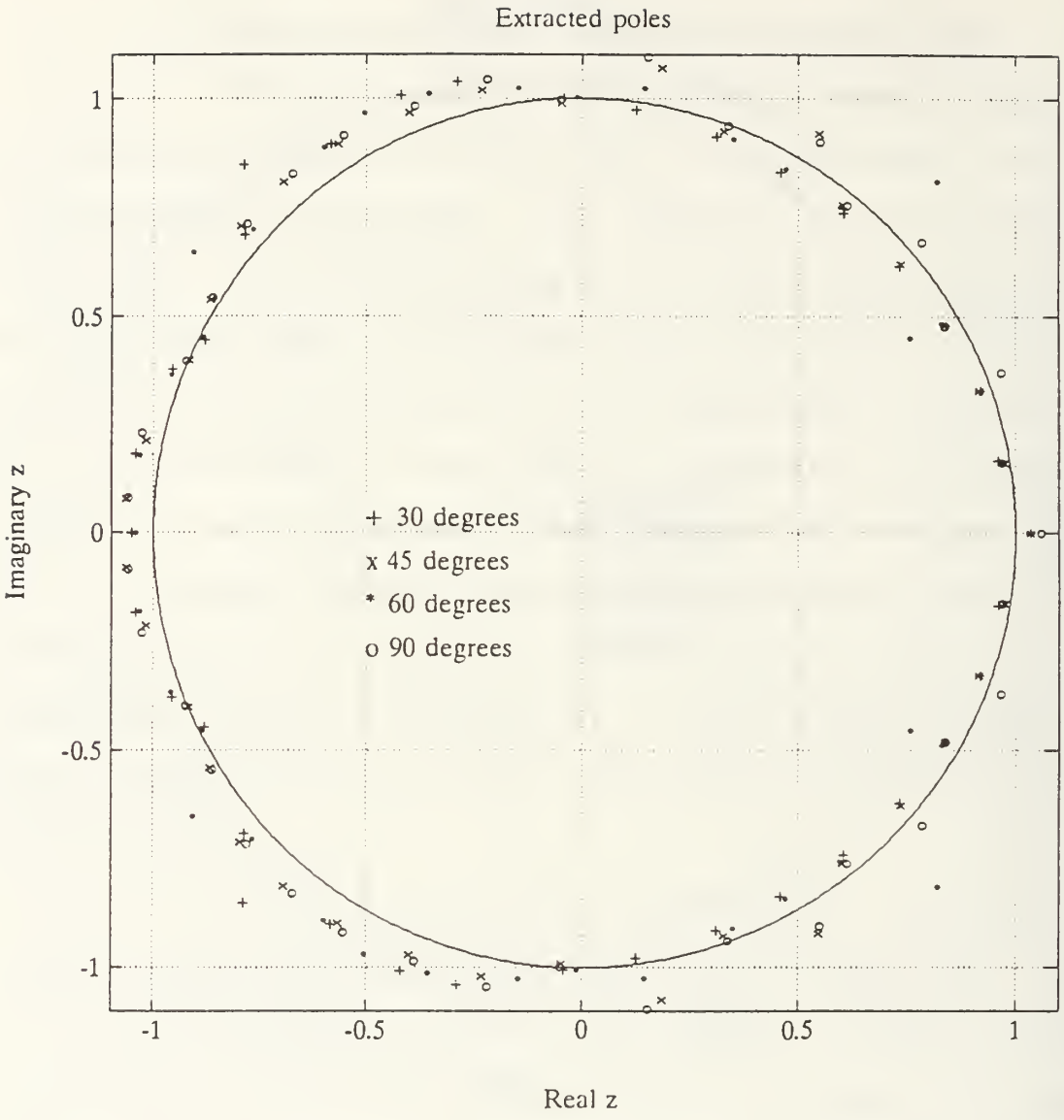

Figure 50. Cadzow-Solomon Poles, 20.0 dB SNR

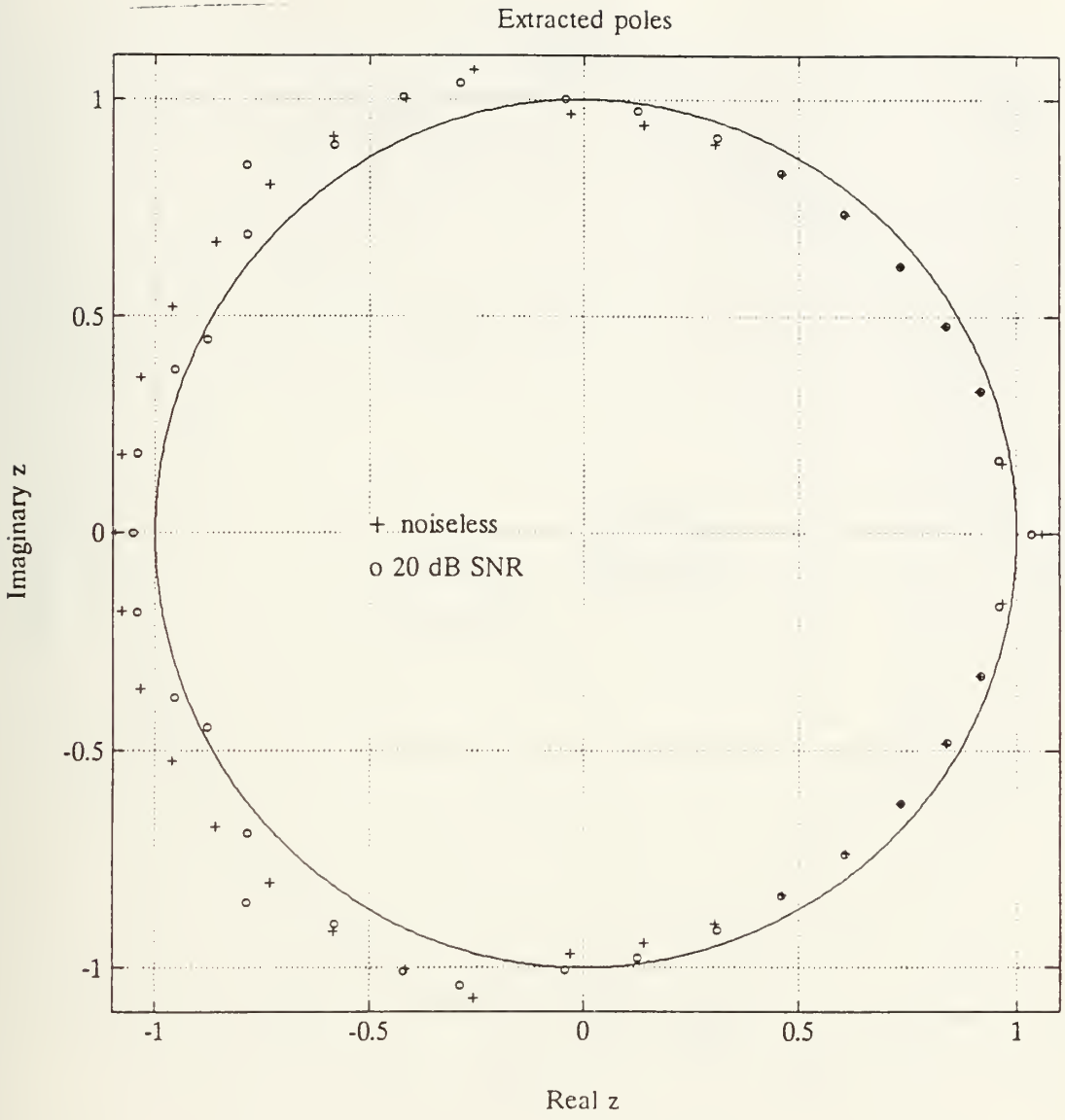

Figure 51. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 30 Degree Aspect

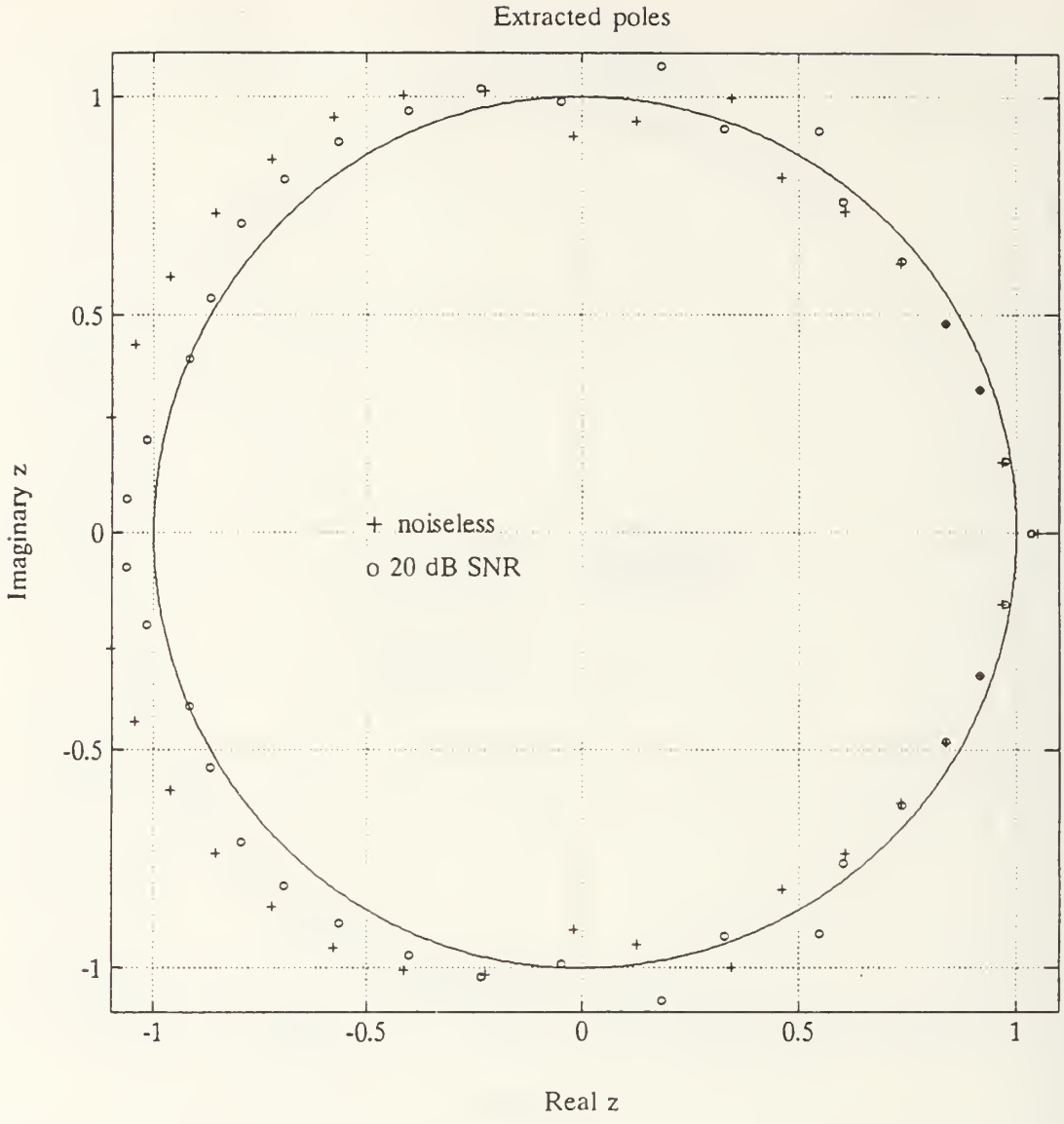

Figure 52. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 45 Degree Aspect

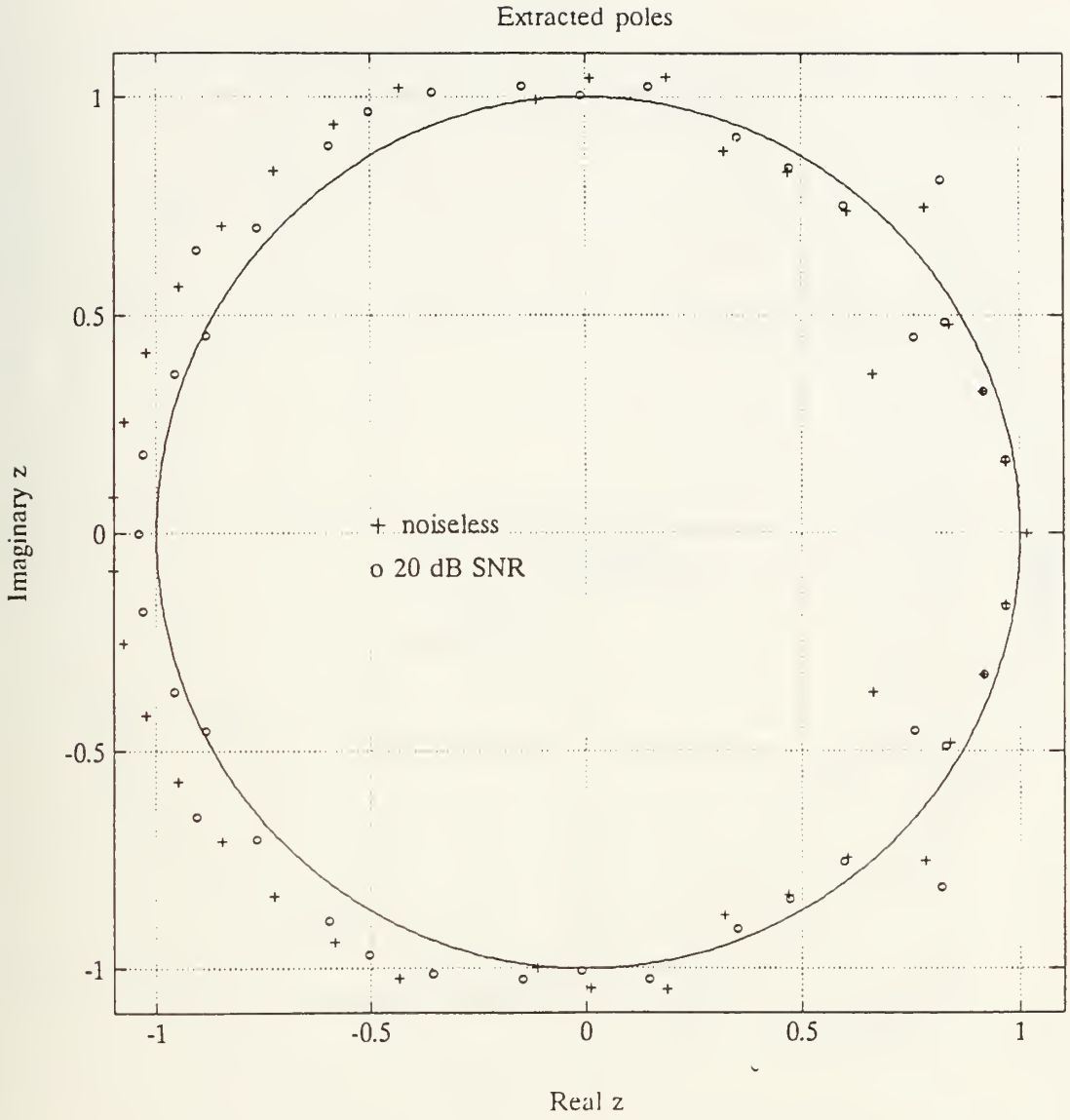

Figure 53. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 60 Degree Aspect

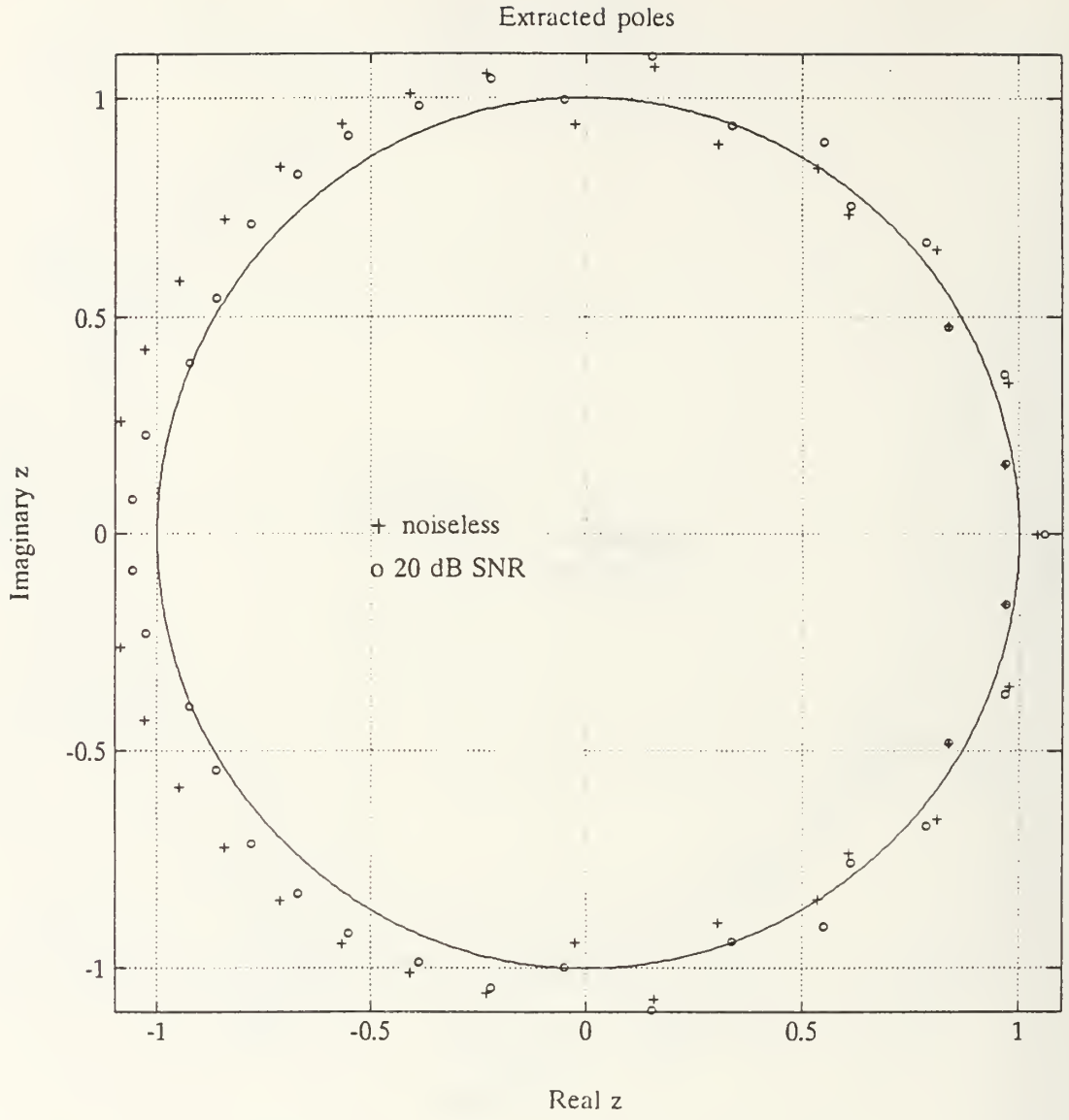

Figure 54. Integral Equation Thin Wire Comparison, Noiseless vs. 20.0 dB SNR, 90 Degree Aspect

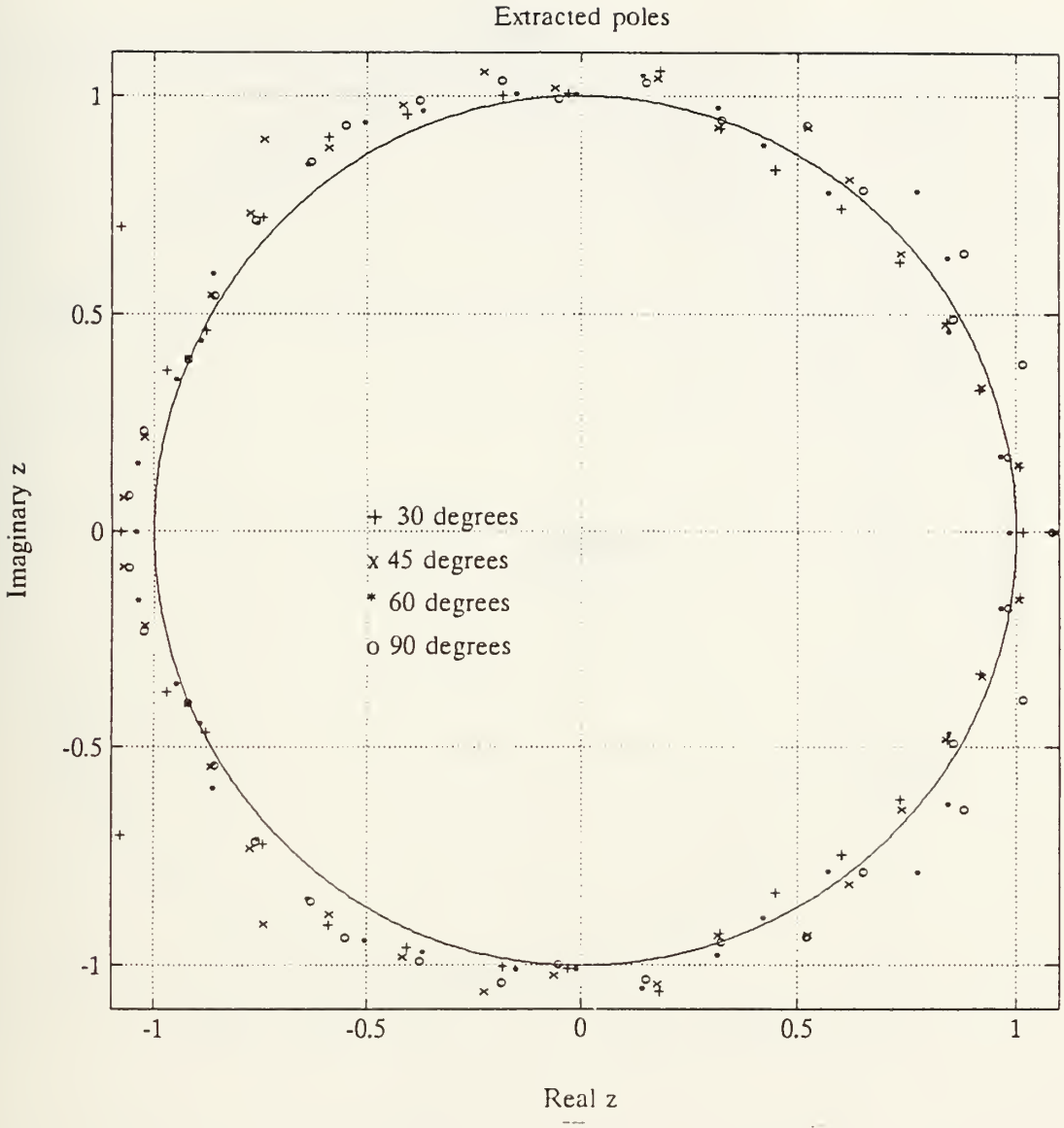

Figure 55. Cadzow-Solomon Poles, 7.0 dB SNR

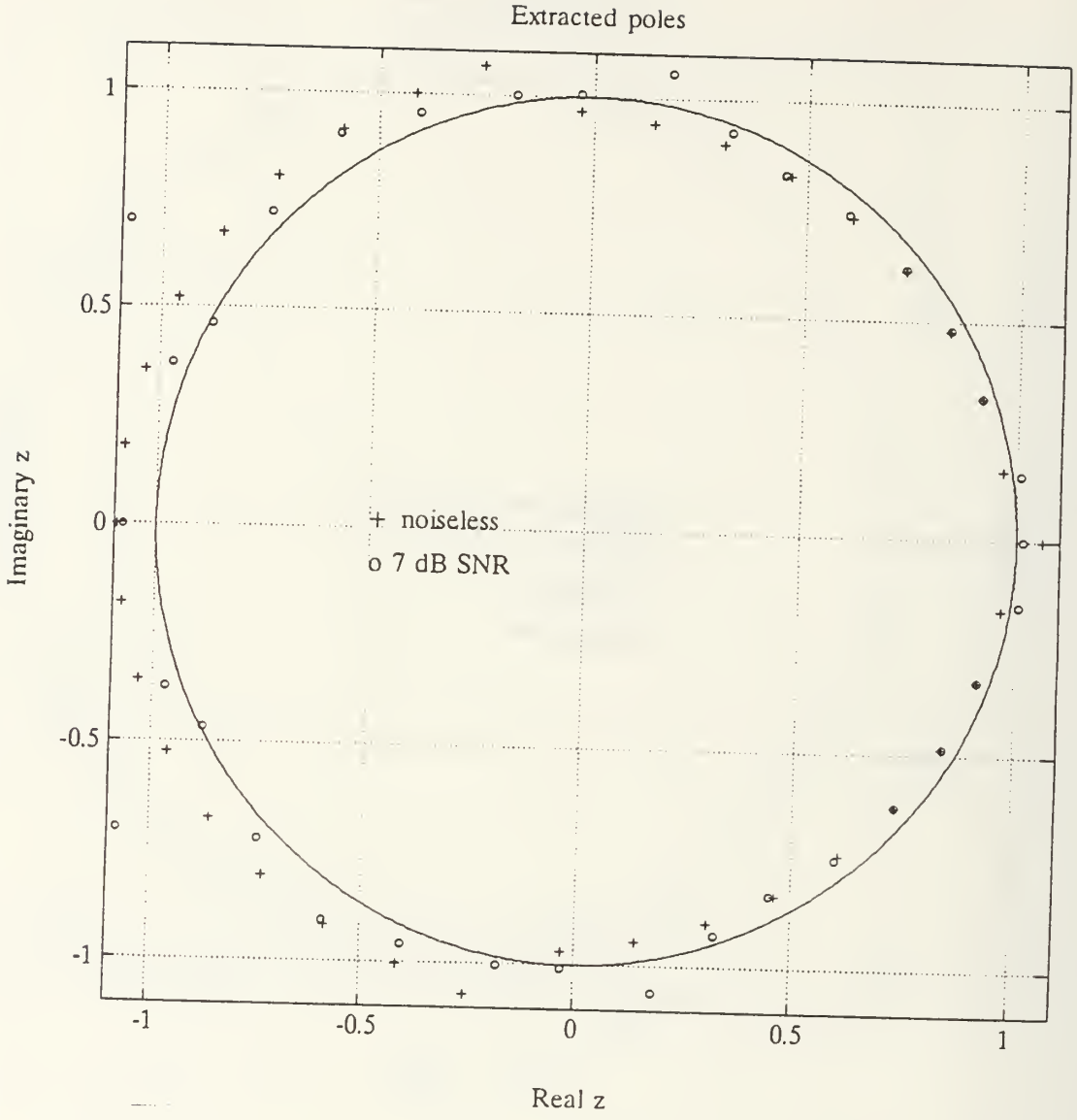

Figure 56. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 30 Degree Aspect

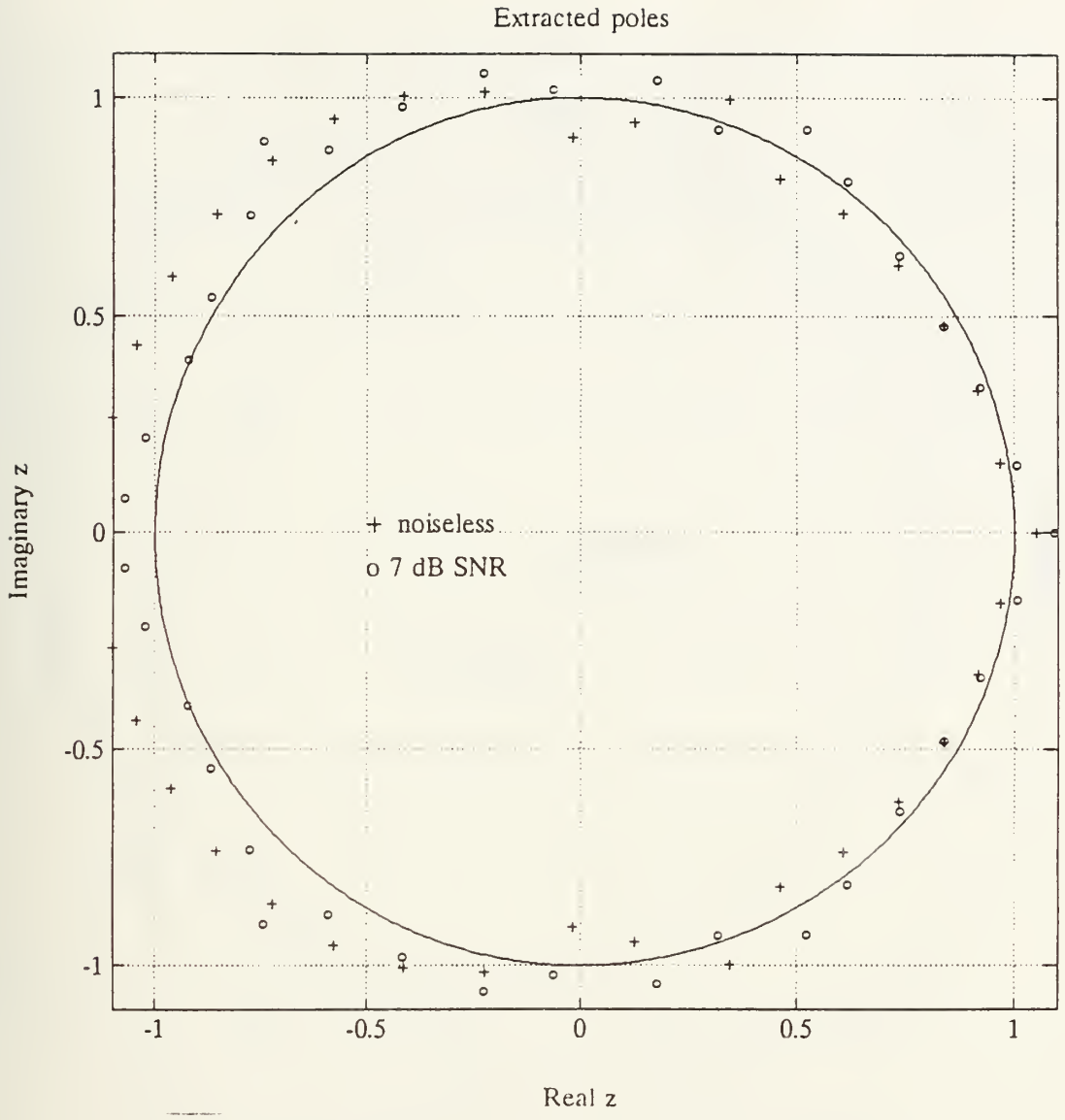

Figure 57. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 45 Degree Aspect

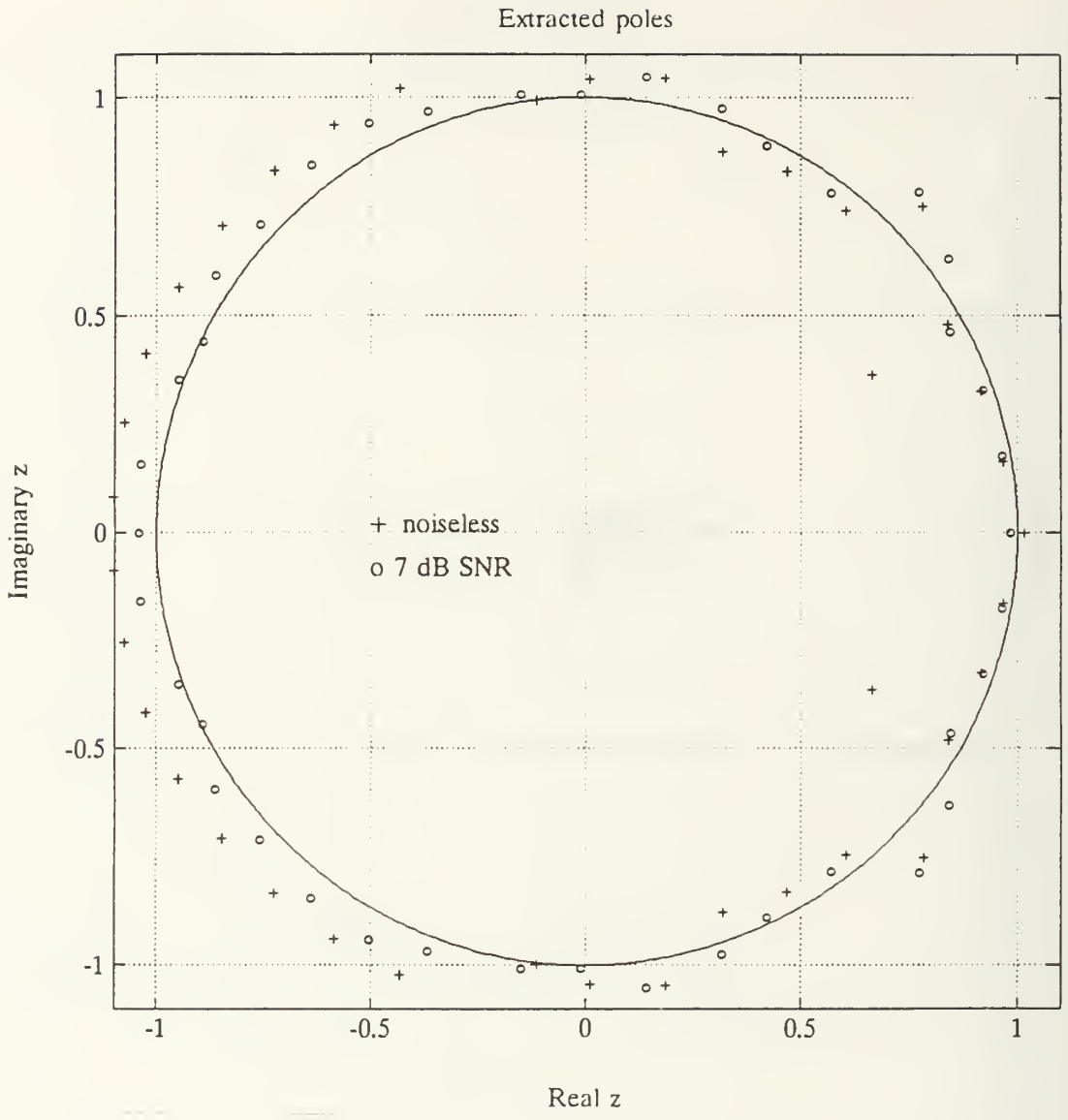

Figure 58. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 60 Degree Aspect

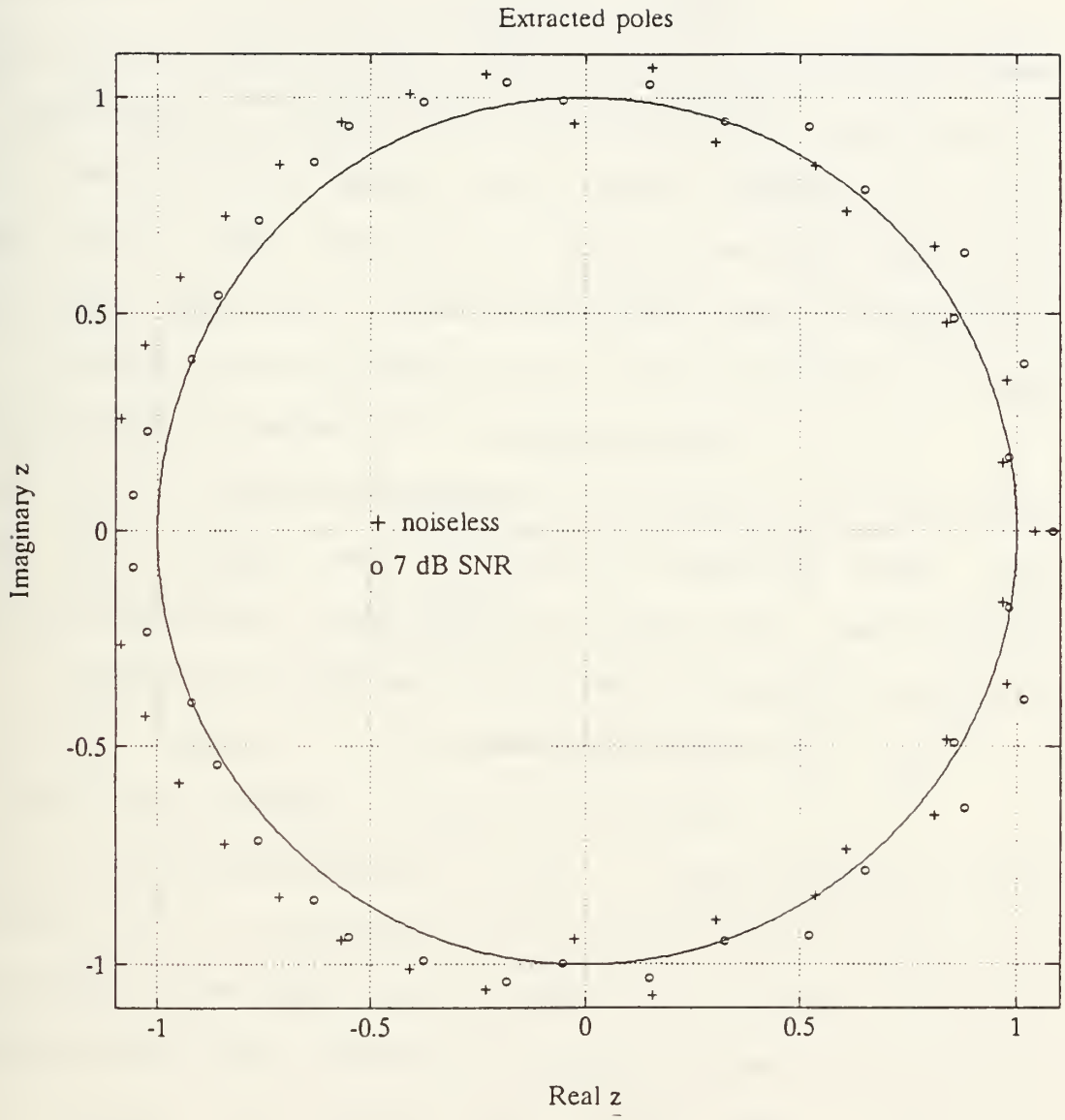

Figure 59. Integral Equation Thin Wire Comparison, Noiseless vs. 7.0 dB SNR, 90 Degree Aspect

1. Wire Targets

Figure 60 depicts the poles extracted from measurements of a 0.1 meter wire. Three tight clusters appear at the lowest frequencies and at the highest frequencies. The poles in between can not be easily discriminated. The dispersion of these poles is apparently due to the aspect dependence of their measurable power. In other words, these poles are excited more at some aspects then at others.

Figure 61 depicts the comparison between poles extracted from computed data and measured data. As in Figure 60, close agreement exists at the highest and lowest frequencies. The results are much more favorable than those similarly obtained by the Kumaresan-Tufts algorithm.

2. Model Aircraft

Figures 62 through 64 depict poles extracted from aircraft target 1. As in the Kumaresan-Tufts testing, the Cadzow-Solomon testing was conducted at six different aspects. Results for target 2 are depicted in Figures 65 through 67. The results of both targets show clearly defined clusters. The first two clusters of target 2 are exceptionally tight. However, the mid-frequency clusters of target 2 are not as clearly formed as those of target 1.

Comparisons of poles obtained with each method for target 1 and 2 are depicted in Figure 68 and 69 respectively. These two figures graphically depict the clear

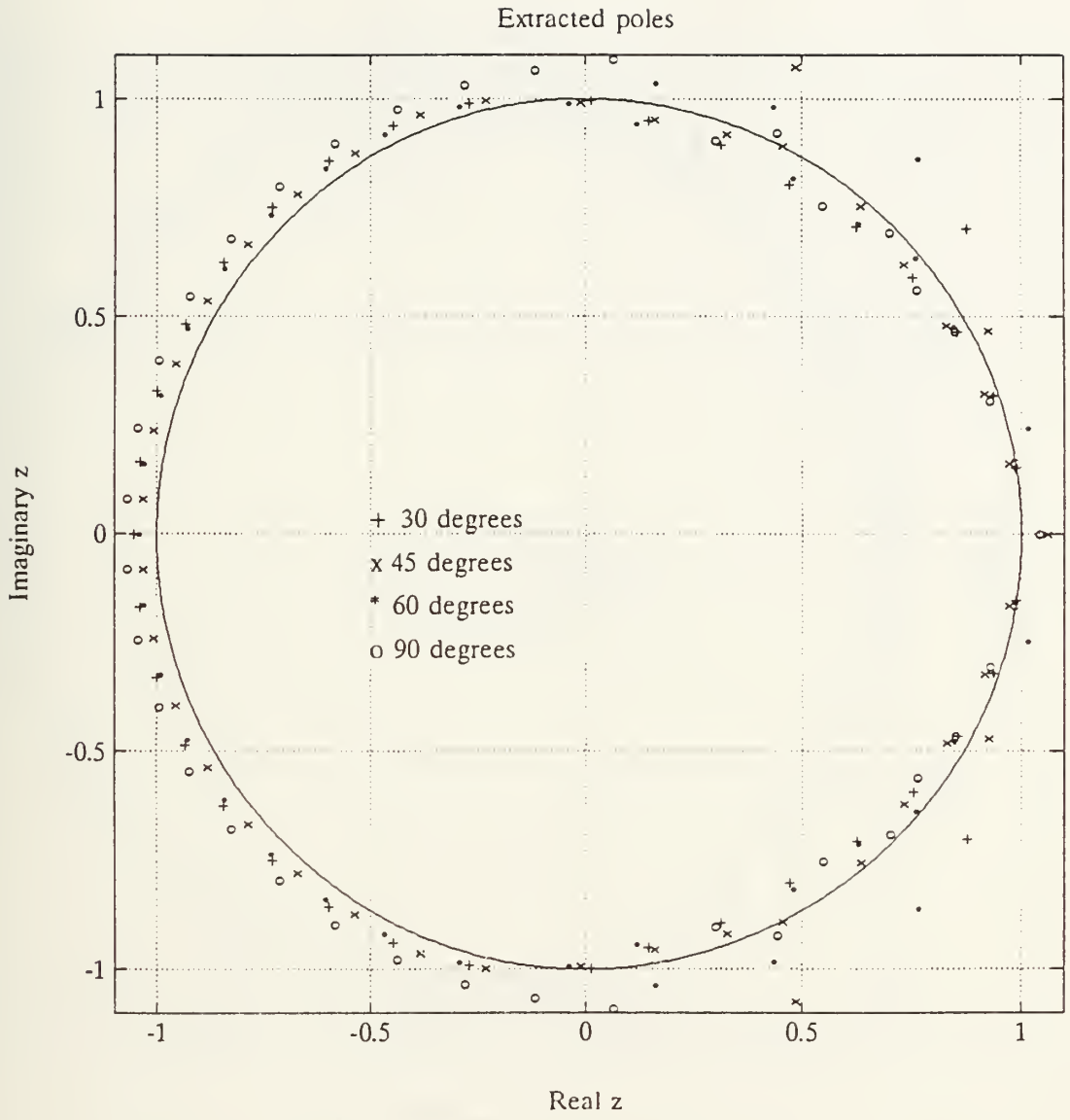

Figure 60. Cadzow-Solomon Poles, Measured Thin Wire

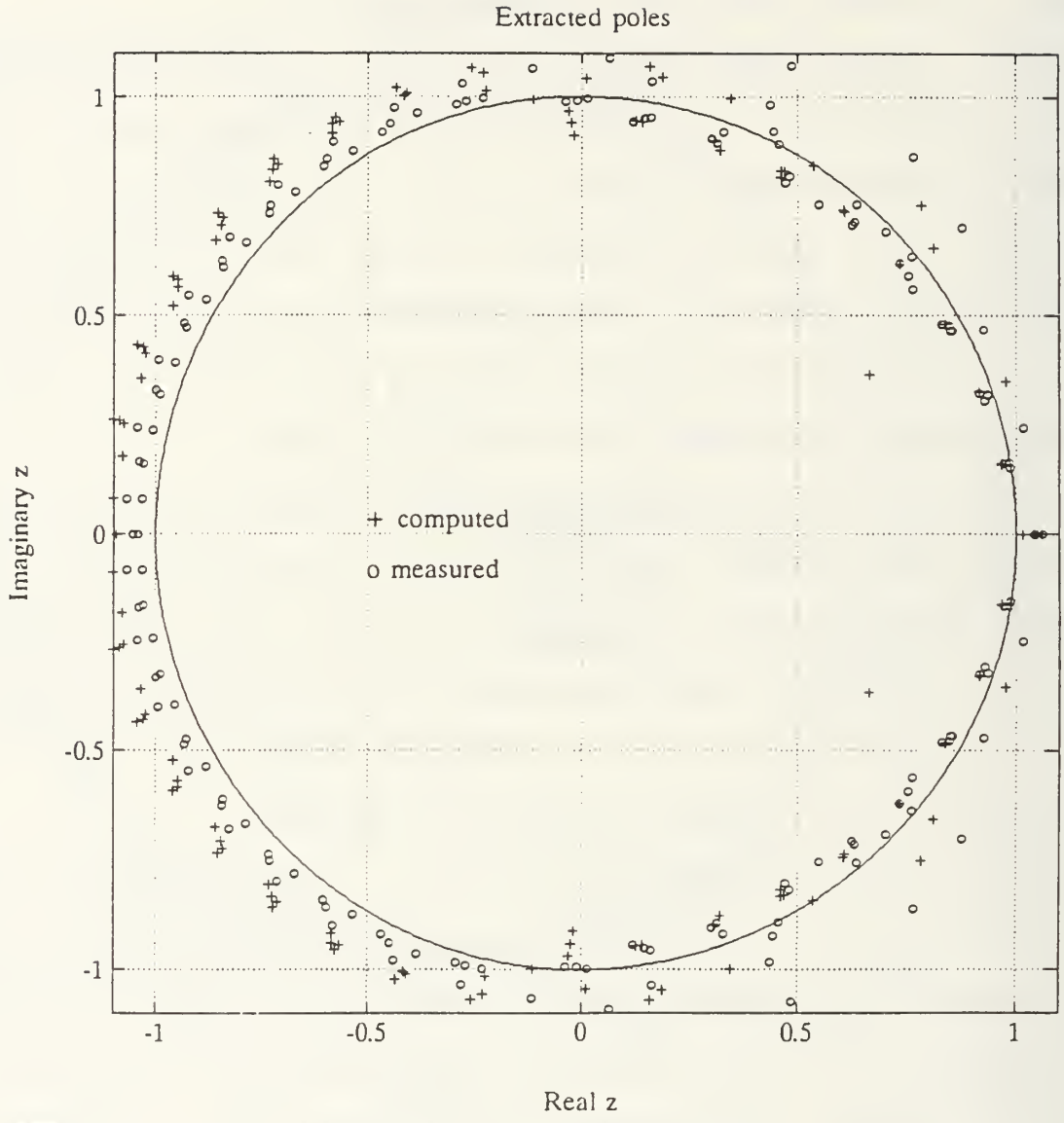

Figure 61. Thin Wire Comparison, Measured vs. Integral Equation

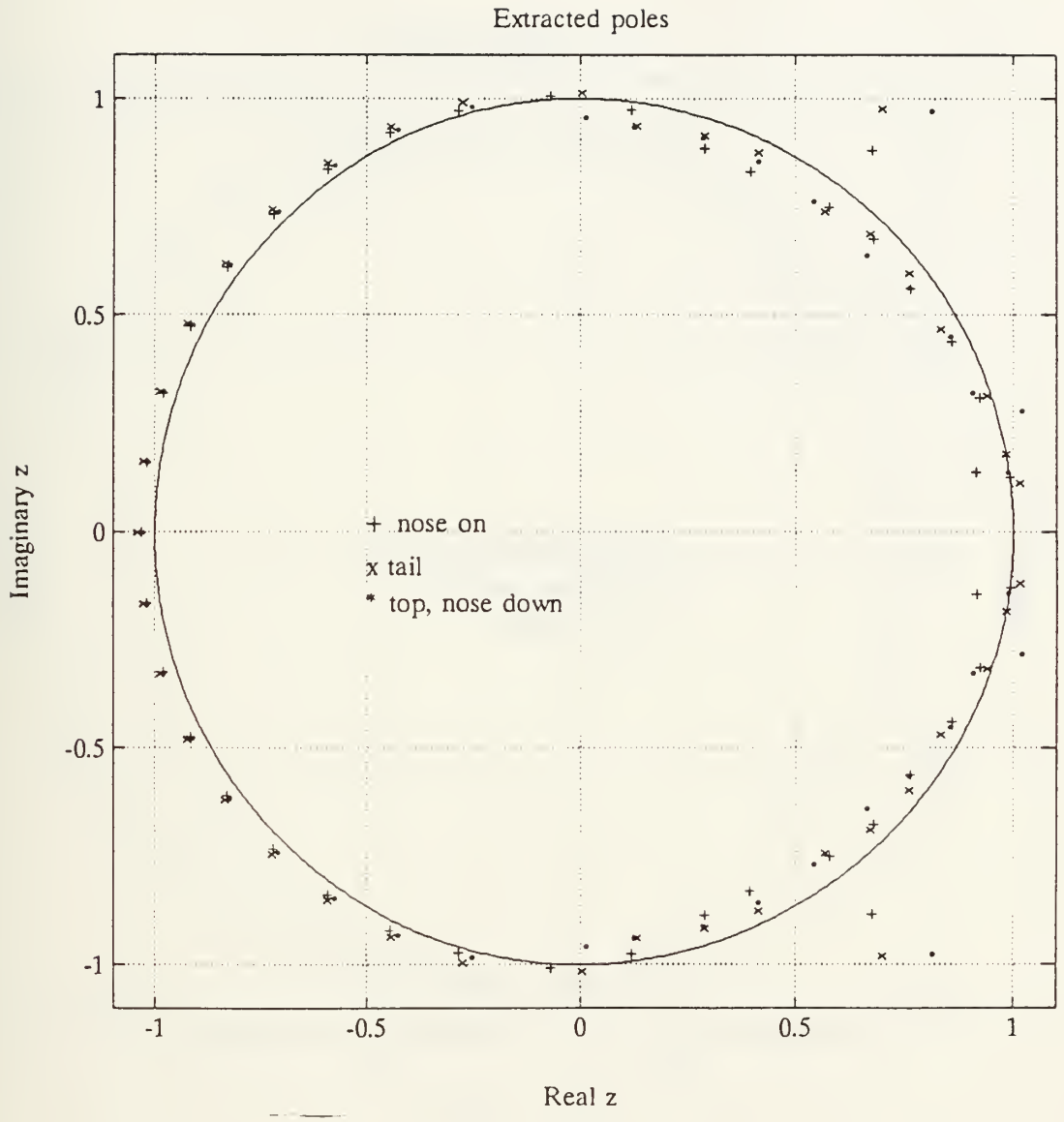

Figure 62. Cadzow-Solomon Poles Target 1, Three Aspects

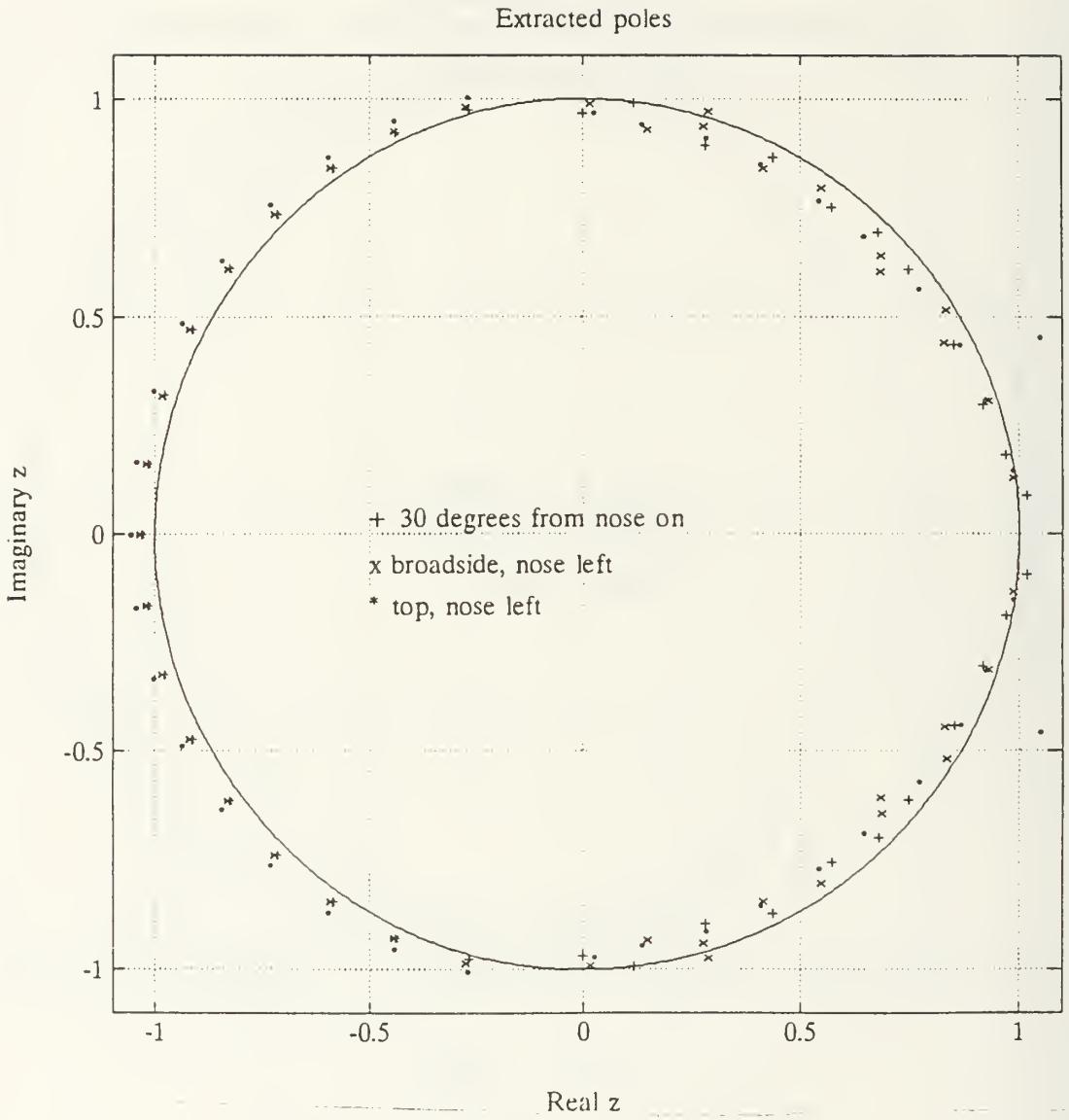

Figure 63. Cadzow-Solomon Poles Target 1, Three Aspects

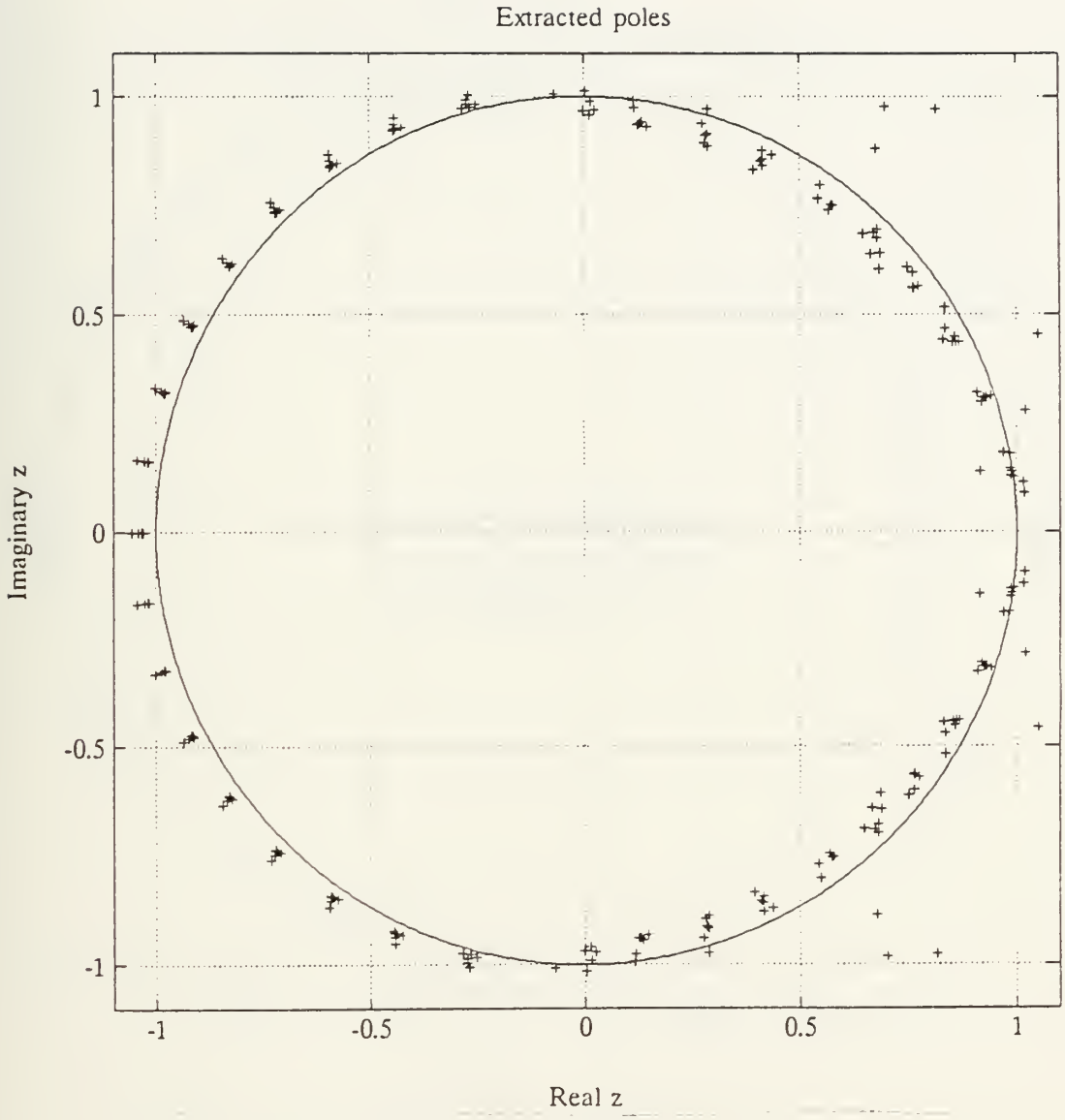

Figure 64. Cadzow-Solomon Poles Target 1, All Six Aspects

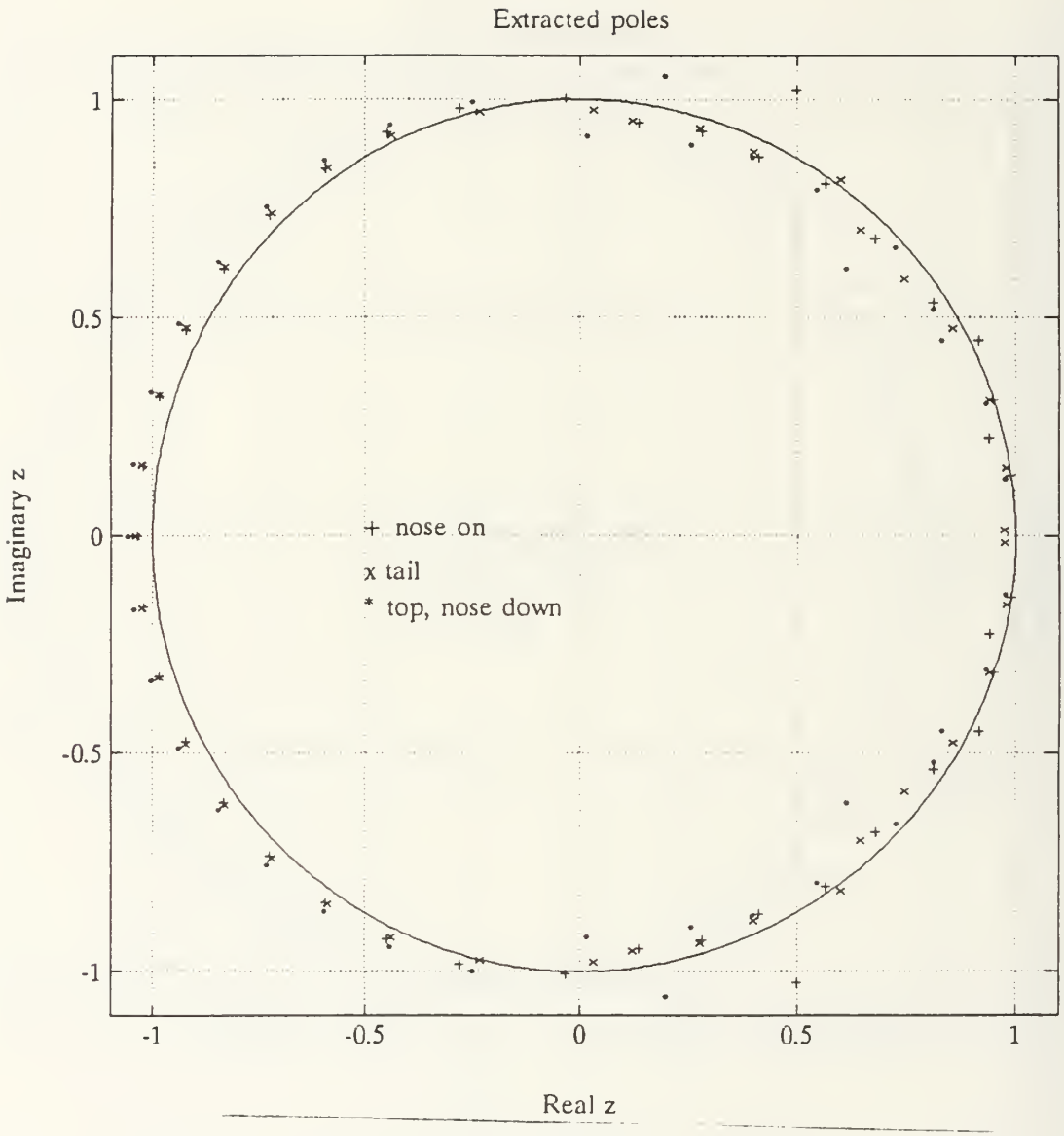

Figure 65. Cadzow-Solomon Poles Target 2, Three Aspects

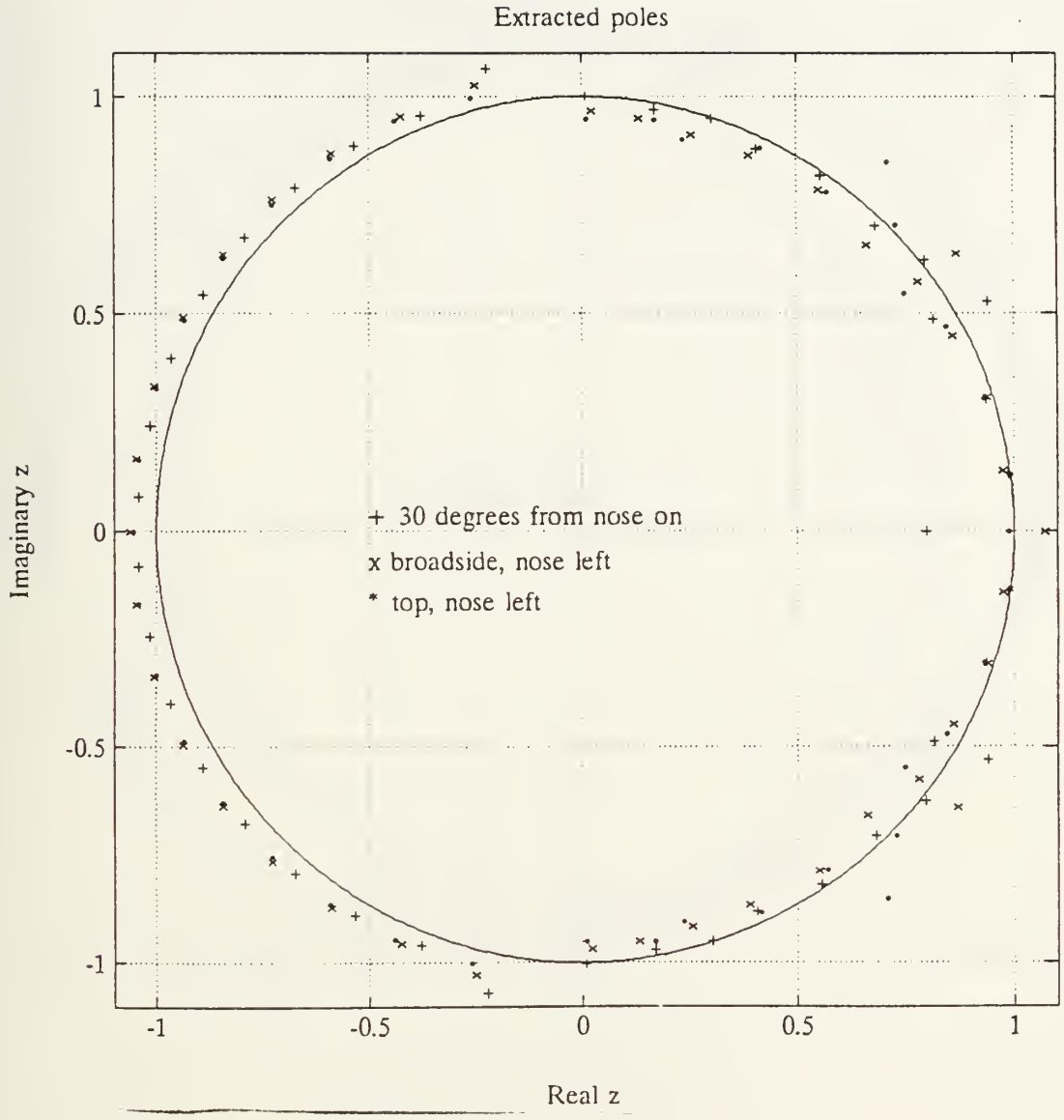

Figure 66. Cadzow-Solomon Poles Target 2, Three Aspects

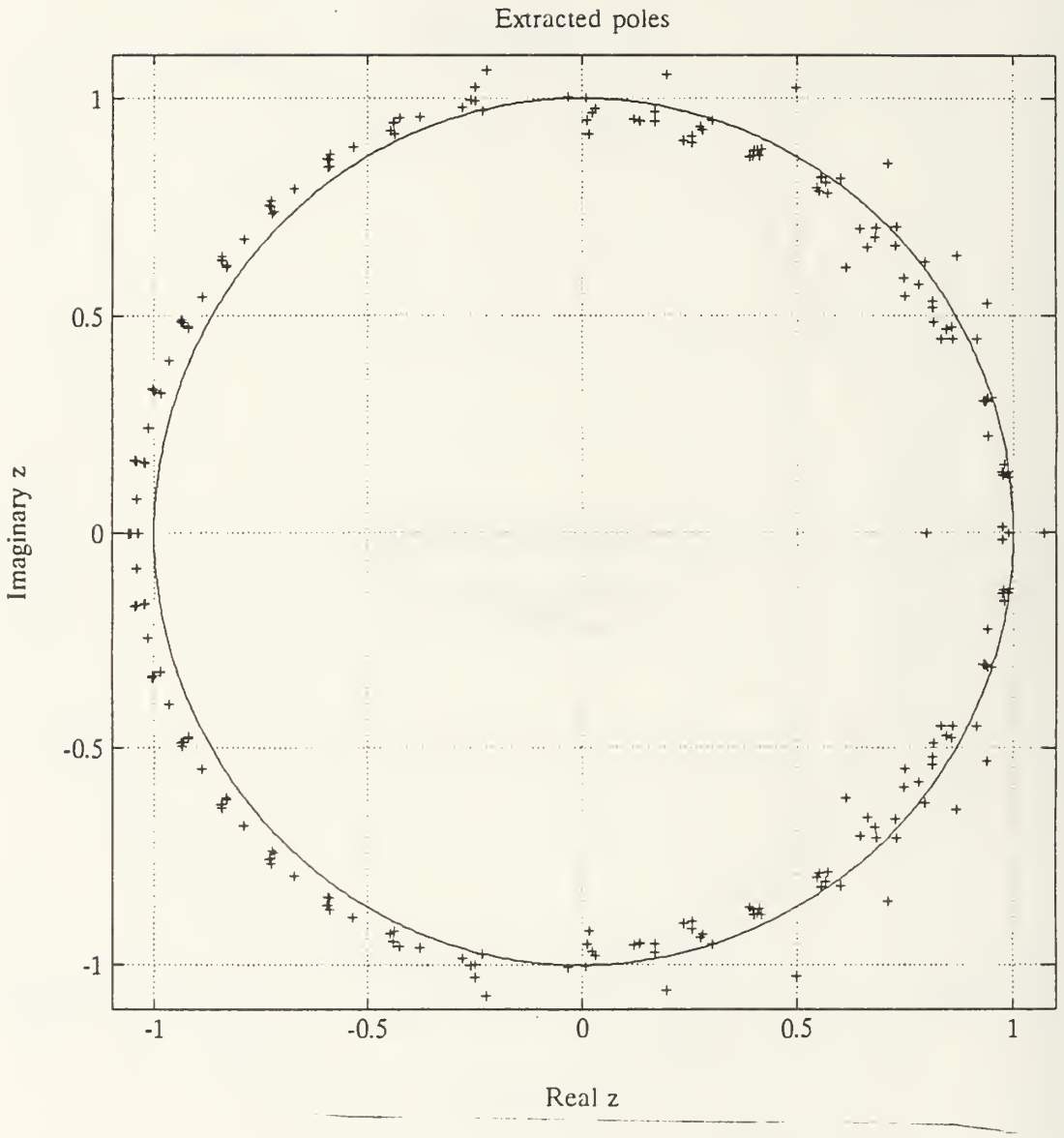

Figure 67. Cadzow-Solomon Poles Target 2, All Six Targets

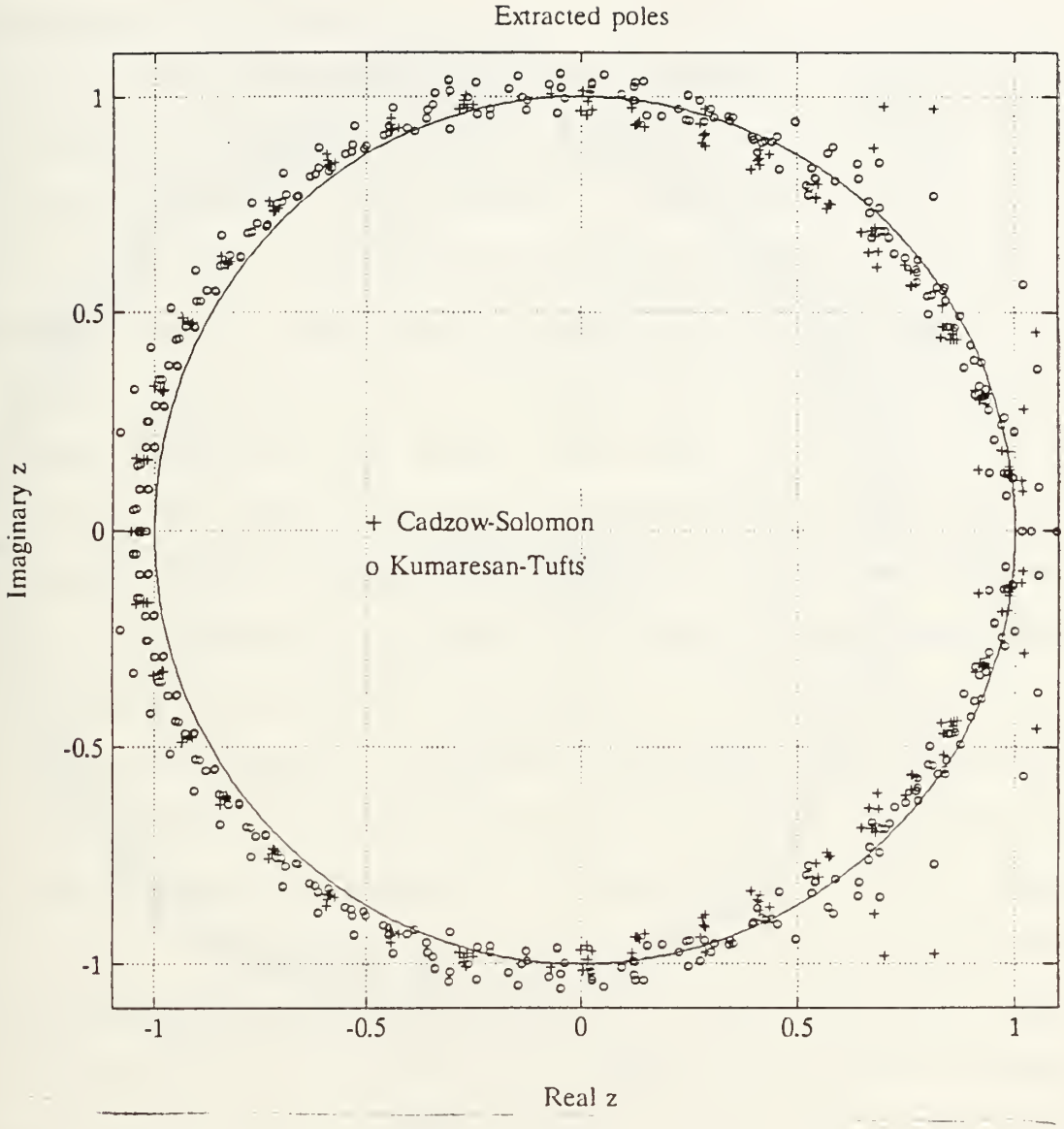

Figure 68. Pole Comparisons, Target 1, All Six Targets

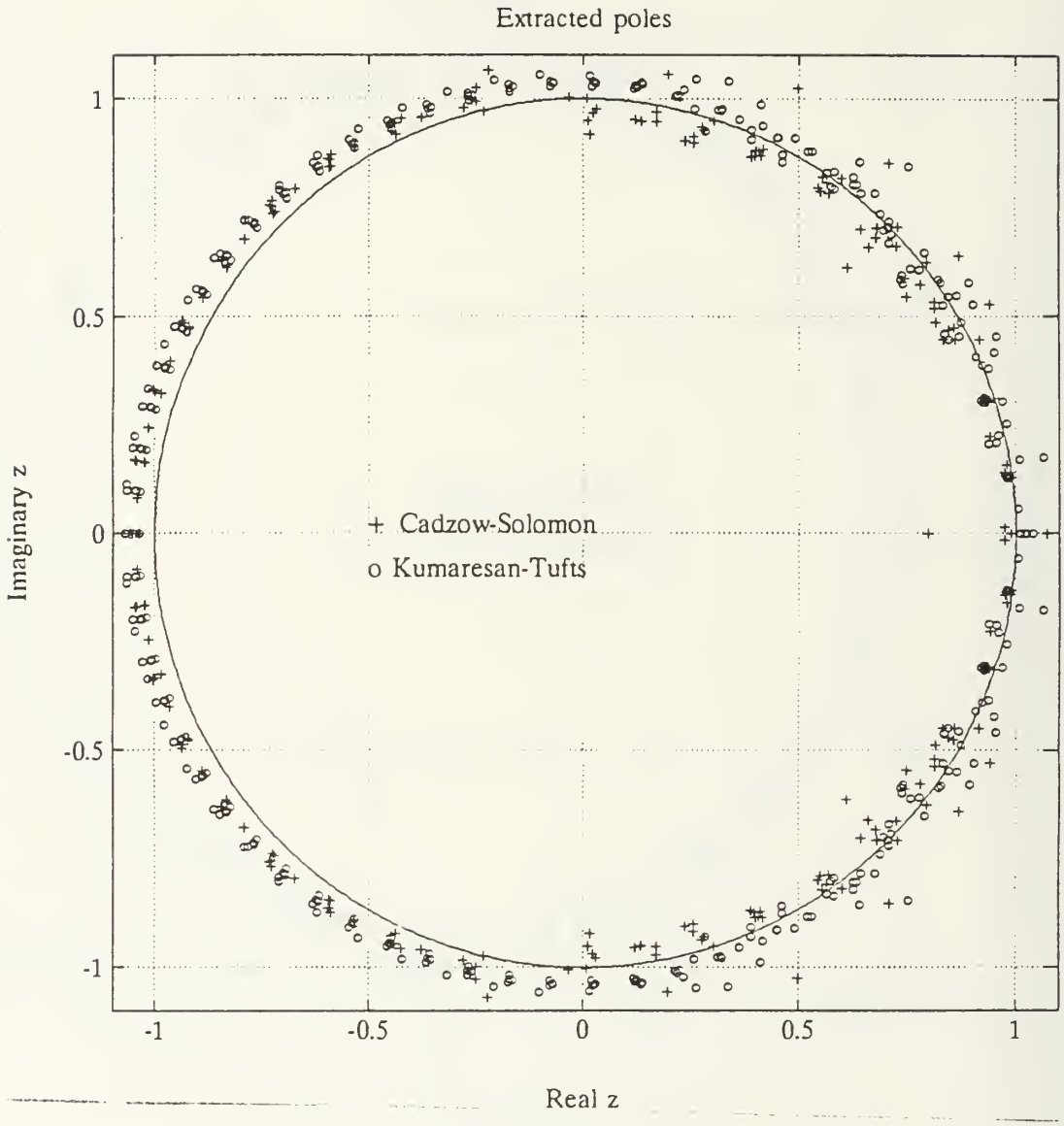

Figure 69. Pole Comparisons, Target 2, All Six Aspects

superiority of the Cadzow-Solomon algorithm over the Kumaresan-Tufts algorithm.

In order to obtain an initial indication of the possibility for target classification through pole extraction, nose-on measurements of two additional aircraft models were made, processed and compared with the results of targets 1 and 2. The nose-on measurements of targets 3 and 4 appear in Figures 70 and 71 respectively. A comparison plot of poles extracted from each of the four targets is depicted in Figure 72. Each of the four aircraft measured are fighters of similar size and shape (see Table 1). The poles for each target are sufficiently different in this single measurement to identify each aircraft individually. However, some of the poles are arranged in clusters which appear with a harmonic pattern similar to that obtained for either of the first two aircraft at various aspects. In order to more fully assess the target classification capability of pole extraction, several measurements should be made of a given aircraft model. A plot of the poles extracted from each of these measurements would form clusters at the locations of the true poles. The centroid of each of these clusters would then be compared against the centroid poles similarly obtained from other aircraft. Although several poles of different aircraft might be similar, the set of poles belonging to an aircraft could form the basis for classification if that set was unique among

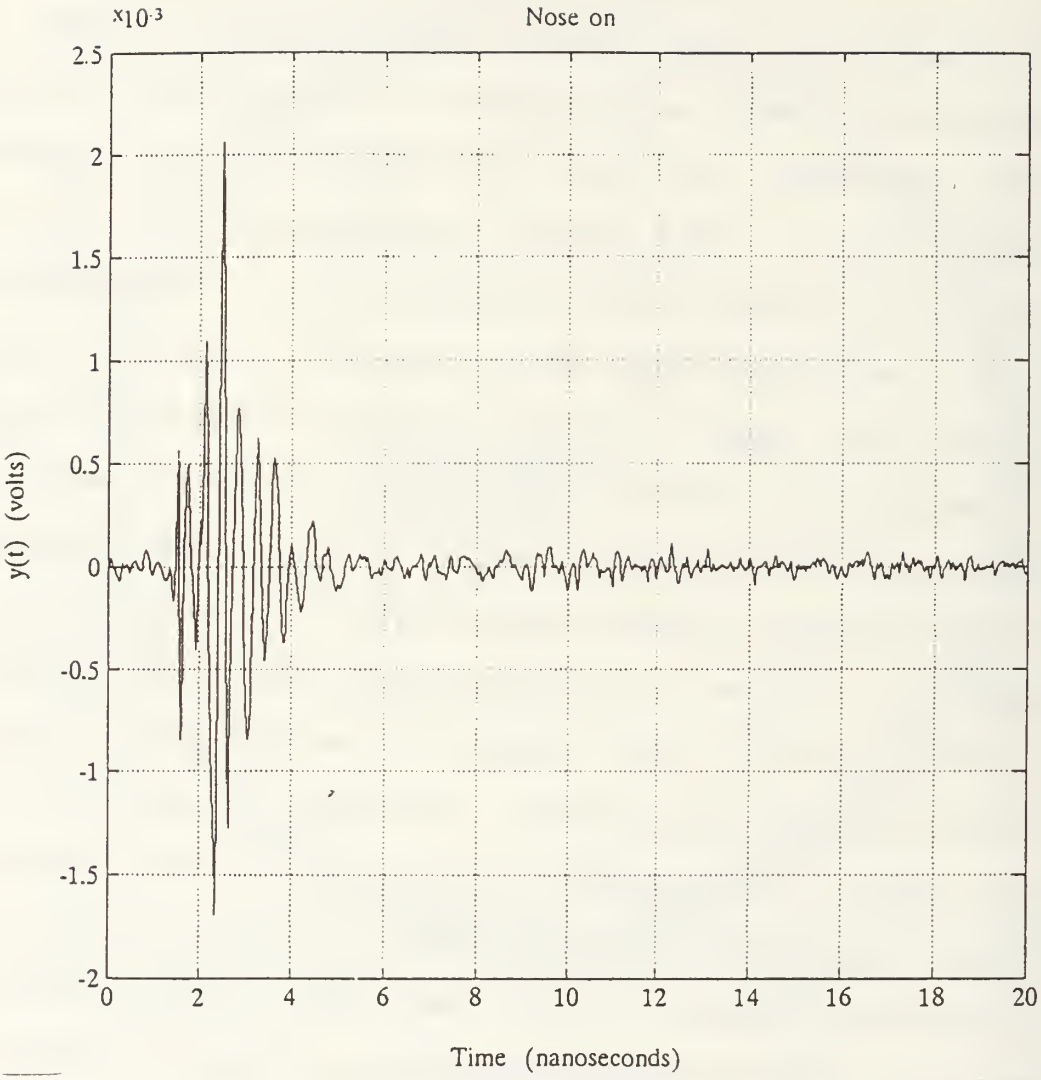

Figure 70. Target 3 Scattering, Nose-on

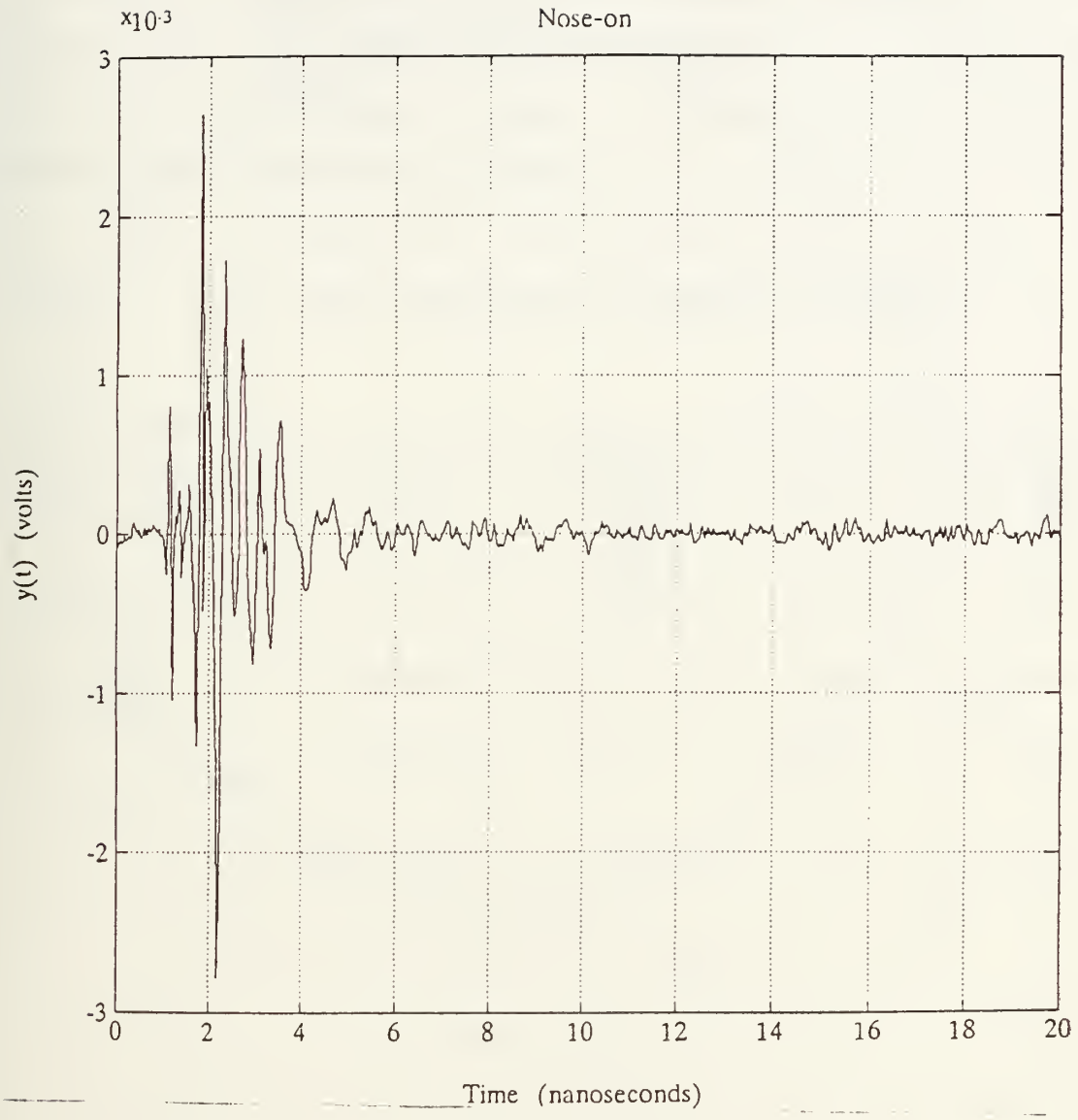

Figure 71. Target 4 Scattering, Nose-on

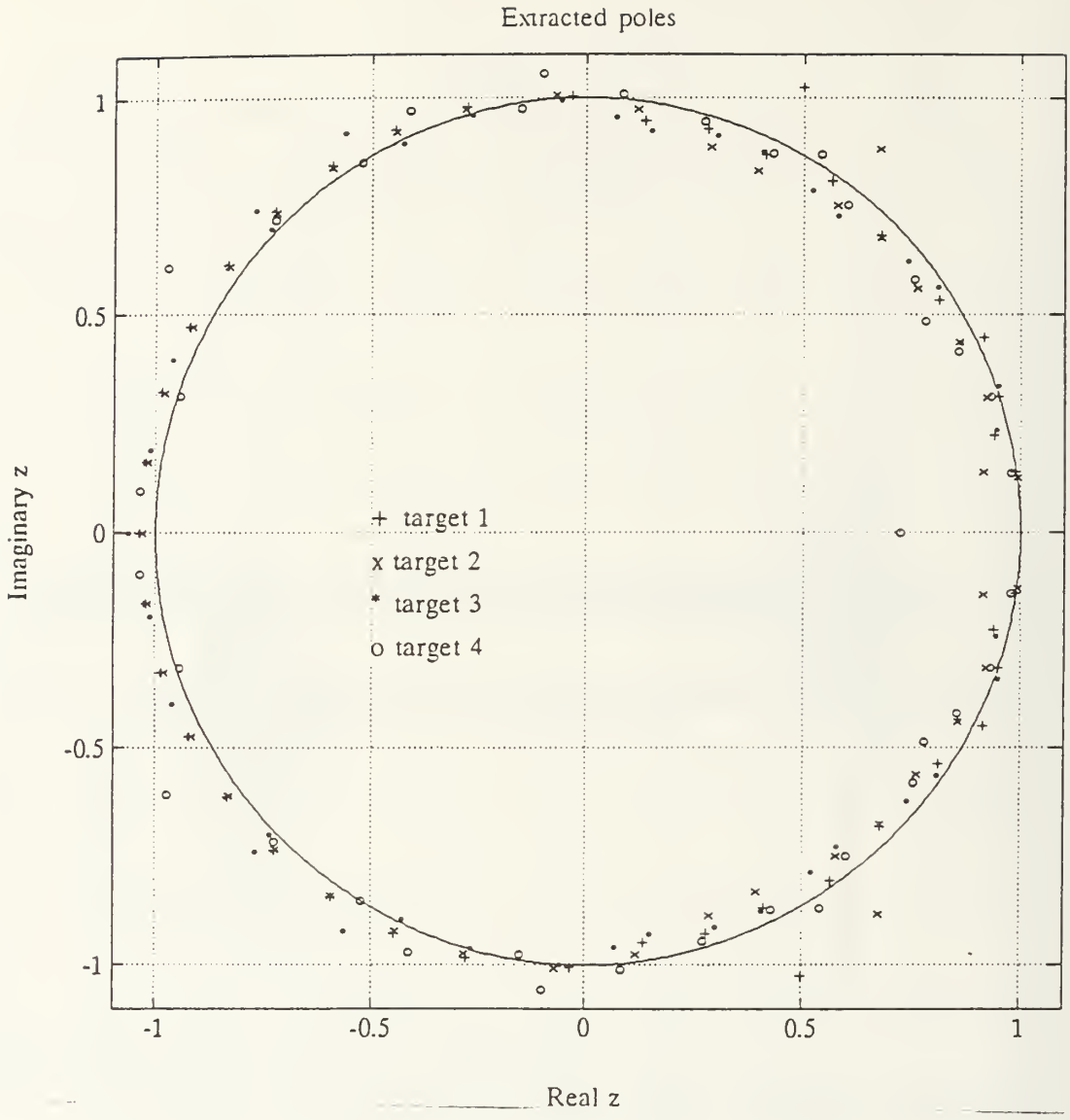

Figure 72. Cadzow- Solomon Pole Comparisons, 4 Targets, Nose-on

the sets belonging to all other measured aircraft. The results in Figure 72 demonstrate the possibility of using the Cadzow-Solomon pole extraction algorithm to aid in the classification of aircraft, perhaps by use of the extracted poles in constructing annihilation filters.

TABLE 1. FULL SIZE DIMENSIONS OF TARGETS RECORDED

Target number	1	2	3	4
Overall length (meters)	12.20	15.03	16.94	16.00
Overall height (meters)	3.35	5.09	4.51	4.80
Wingspan (meters)	10.96	10.00	11.43	13.95
Tailplane span (meters)	Unknown	5.58	6.92	5.75

III. SUMMARIES AND CONCLUSIONS

In this chapter, a step-by-step guide through each algorithm is presented. At each step, techniques and lessons learned are discussed together with general observations. Conclusions are presented at the end of the chapter.

A. KUMARESAN-TUFTS

The first step in processing a signal with the Kumaresan-Tufts algorithm is to determine the beginning of early-time. The objective is to pick the earliest possible starting point without entering into the latter part of early-time. If the starting point for processing is improperly chosen to include the early-time, the results will be completely unreliable since the signal no longer satisfies the late time model. If the starting point is chosen too late, the signal may not be sufficiently strong in the presence of measurement noise. Since the signal is the sum of exponentially damped sinusoids, the optimum starting point is at the precise instant of transaction into late-time. The key to determining the beginning of late-time is in determining the beginning of early time. Determining the first response of the target to excitation cannot usually be done by a simple visual inspection of measurement data. Unless the exact distance to

the target is known, the most accurate method attempted by the author for determining the beginning of early—time is to process the signal using the Cadzow-Solomon algorithm. This is discussed in the next section. However, the reliance of the Kumaresan-Tufts algorithm on information provided by the Cadzow-Solomon algorithm is an obvious disadvantage of the former method.

Once the starting point for processing has been selected, the next step is to determine the dimensions of the data matrix and, consequently, the number of points in the signal to be processed. In trials conducted on noiseless synthetic data, the accuracy of pole extraction increased steadily with the increase in the data matrix dimensions. These trials were conducted up to the limit of the array dimensions defined in the computer program of Appendix A. The number of points processed in measurement data should be as large as possible, while still meeting the following two constraints. First, incorporate as many cycles of the data as possible. Usually, visual inspection of the data reveals a repeating pattern which should be entirely incorporated into the window of points to be processed. When only portions of these patterns are selected, a disproportionate weighting tends to be placed on certain poles. Second, signal portions late in the response which are no longer distinguishable in the presence of measurement noise should not be selected.

The final step involves determining the number of true poles in the system. The following approach has proven to be the most successful. First, process the signal without any eigenvalue compensation to establish an upper bound on the order of the system. In most cases, the number of poles outside the unit circle will be less than the overestimated order of the system. If not, increase the row dimension of the data matrix in order to increase the estimated order of the system, and repeat. When the number of poles is less than the estimated order of the system, then one should gradually increase the number of eigenvalues compensated in successive trials, while closely observing the effects induced on the poles outside the unit circle. As the number of eigenvalues compensated is steadily increased, noise poles and weak signal poles will move inside the unit circle. The programs in Appendix A and B allow the user to compare the results of successive trials, by generating overlays for each plot. If N poles are in the signal space, at least the first N eigenvalues must not be compensated, or true poles may be lost. As the actual order of the system is approached by compensation, the user will notice an orderly, even arrangement assumed by the noise poles. If certain poles still remain suspect after compensation, vary slightly the other parameters, such as the starting point and the

dimensions of the data matrix. Generally, only true signal poles will repeatedly assert themselves under varying parameters.

B. CADZOW-SOLOMON

The techniques and general observations offered in the preceding section apply equally to the Cadzow-Solomon algorithm. An important consideration in this method, not discussed above, is the selection of the beginning of early-time. Candidates for a starting point are usually at or near zero crossings within approximately thirty points of the object's first definite response to electromagnetic excitation. Begin processing at the chosen point while varying parameters in successive trials. Select the point whose successive results are the most consistent under varying parameters.

The selection of the starting point for beginning of early-time can be very critical. For example, not a single pole could be extracted in one trial wherein the starting point occurred only ten points after the actual starting point. Additionally, in most cases observed, the late-time start given by the selected early-time occurred within less than two points from a zero crossing. If this observation proves to be generally true in later research, it may serve

as a way to check the starting point selected for one algorithm in terms of the other.

C. CONCLUSIONS

Both the Kumaresan-Tufts and the Cadzow-Solomon algorithms can effectively extract poles from the scattering response of a radar target. Because both algorithms obtain a least-squares solution to the system model, both perform acceptably in the presence of noise. Although eigenvalue compensation is not analytically justified in the Cadzow-Solomon algorithm, the results obtained through eigenvalue compensation in this method were generally superior to those similarly obtained in the Kumaresan-Tufts method. The results demonstrated the inherent advantages of an algorithm capable of processing a target's strongest response in the early time. The Kumaresan-Tufts method compared favorably with the Cadzow-Solomon only in responses with a long late—time.

APPENDIX A. THE KUMARESAN-TUFTS POLE EXTRACTION ALGORTITHM

The following program implements the Kumaresan-Tufts algorithm as described in Chapter 2 of this thesis. The program is written in Fortran 77. The SVD and root-finding subroutines called by this program are found in the EISPACK library [18]. The SVD subroutine is a translation from ALGOL as given in [19]. The matrix multiplication and graphics subroutines, also called by this program, are found in Appendix C and D respectively.

```
      INTEGER IERR,Kd,M,MN,MAGPOL,NSTRTPT,DELTAY
      INTEGER IER,NCAUS,NMENU,L/1/
      INTEGER*2 KdPLT
      REAL*8 A(70,70),W(70),U(70,70),V(70,70),RV1(70)
      REAL*8 VS(70,70),UT(70,70),AINV(70,70),X(70)
      REAL*8 XP(70),B(70),SIGMA(70,70),SIG(70,70)
      REAL*8 COF(70),ROOTR(70),ROOTI(70)
      REAL*8 D(1024),AVG,MACHEP/1.0E-16/,Dy(140)
      COMPLEX*16 S(70)
      LOGICAL MATU/.TRUE./,MATV/.TRUE./,CAUSAL/.TRUE./,LONG/.TRUE./
      LOGICAL DSET/.FALSE./,NUFILE/.TRUE./
      CHARACTER TITLE*16,HEADER*64,YN*1,DC*1,TITLER*16,TITLEI*16
      CHARACTER TITL*16

C     Enter parameters for processing

11    IF (DSET) CLOSE(10)
      NOVERLAY=0
      OPEN(10,FILE='PLOT')
      IF (DSET) GO TO 85
      WRITE (*,*) 'Welcome to signal processing using the'
      WRITE (*,*) 'Kumaresan-Tufts method'
      WRITE (*,*) ' '
      WRITE (*,*) 'Do you want '
      WRITE (*,*) ' '
      WRITE (*,*) '1. The long version for beginners'
      WRITE (*,*) '2. The short version for pros'
      WRITE (*,*) ' '
```

```
15    WRITE (*,*) 'Please enter 1 or 2 '
      READ (*,*) N
      IF (N .EQ. 1) THEN
      LONG=.TRUE.
      ELSEIF (N .EQ. 2) THEN
      LONG=.FALSE.
      ELSE
      GO TO 15
      ENDIF

      WRITE (*,*) 'Session will begin with entry of parameters needed fo+r processing'
      WRITE (*,*)
      WRITE (*,*) 'Do you want to enter parameters from'
      WRITE (*,*) ' '
      WRITE (*,*) '1. The keyboard'
      WRITE (*,*) '2. A previously created file of parameters'
      WRITE (*,*) ' '
16    WRITE (*,*) 'Please enter 1 or 2 '
      READ (*,*) N
      IF (N .EQ. 1) THEN
      GO TO 1
      ELSEIF (N .EQ. 2) THEN
10    WRITE (*,*) 'Enter title of file containing parameters'
      READ (*,105) TITL
      OPEN(1,FILE=TITL)
      READ(1,105) TITLE
      READ(1,110) NPTS
      READ(1,110) NRT
      READ(1,110) Kd
      READ(1,110) M
      READ(1,110) DELTAY
      READ(1,110) NSTRTPT
      READ(1,110) NCAUS
      CLOSE(1)
      GO TO 85
      ELSE
      GO TO 16
      ENDIF
      WRITE (*,*) ' '

1     NUFILE=.TRUE.
      IF (.NOT. DSET) NSTRTPT=1
      WRITE (*,*) 'Enter title of data file to be read'
      READ (*,105) TITLE
      OPEN(1,FILE=TITLE)
      READ(1,105) HEADER
      READ(1,110) NPTS
      IF (NPTS .GT. 1024) THEN
      WRITE (*,*) 'Number of points in data file exceeds the dimension'
      WRITE (*,*) 'of the array used in the program to store the file'
```

```
      STOP
      ENDIF
      CLOSE(1)

      IF (DSET) THEN
      IF (NSTRTPT+(Kd+M-1)*DELTAY .LE. NPTS) GO TO 85
      ENDIF

3     IF (NUFILE) THEN
      WRITE (*,*) 'Enter Kd, >= the estimated order of the system '
      READ (*,*) Kd
      IF (Kd .GT. 69) THEN
      WRITE (*,*) 'Kd must be less than 70, or dimension statements'
      WRITE (*,*) 'in this program must changed by the user'
      GO TO 3
      ELSEIF (Kd .LT. 2) THEN
      WRITE (*,*) 'Kd must be at least 2'
      GO TO 3
        ENDIF
      IF (2*Kd .GT. NPTS) THEN
      WRITE (*,*) 'Kd must be less than or equal to ',NPTS/2
      GO TO 3
      ELSEIF (2*Kd .EQ. NPTS) THEN
      WRITE (*,*) 'Kd equals',Kd
      WRITE (*,*) 'M must be',Kd
      M=Kd
      WRITE (*,*) 'since there are a total of',NPTS
      WRITE (*,*) 'points in ',TITLE
      GO TO 45
      ENDIF
      GO TO 4
      ELSEIF (DSET) THEN
        N=M
20      IF (NSTRTPT+(N+M-1)*DELTAY .LE. NPTS) THEN
        WRITE (*,*) 'Given the other parameters chosen thus far,'
25      WRITE (*,*) 'Kd may range from                ',NRT
        WRITE (*,*) '                                to',N
        WRITE (*,*) 'Enter Kd'
        READ (*,*) Kd
        IF (Kd .GE. NRT .AND. Kd .LE. N) GO TO 85
        GO TO 25
        ELSE
        N=N-1
        GO TO 20
        ENDIF
      ENDIF

4     IF (NUFILE) THEN
      WRITE (*,*) 'Enter M, the row dimension of the data matrix'
      IF (.NOT. DSET .AND. LONG) THEN
```

```
      WRITE (*,*) ' '
      WRITE (*,*) 'Note: Kd+M points in ',title
      WRITE (*,*) '      will be processed '
      WRITE (*,*) ' '
      ENDIF
30    WRITE (*,*) 'M may range from',Kd
      IF (NPTS-Kd .GT. 69) THEN
      WRITE (*,*) '              to         69'
      ELSE
      WRITE (*,*) '              to',NPTS-Kd
      ENDIF
      READ (*,*) M
      IF (M .GT. 69) THEN
      WRITE (*,*) 'M must also be less than 70'
      GO TO 30
      ELSEIF (M .LT. Kd) THEN
      WRITE (*,*) 'M must be greater than or equal to Kd, Kd= ',Kd
      GO TO 30
      ELSEIF (Kd+M .GT. NPTS) THEN
      WRITE (*,*) 'Kd+M must be less than or equal to',NPTS,','
      WRITE (*,*) 'the number of data points in',TITLE
      WRITE (*,*) ' '
      GO TO 30
      ENDIF
      ELSE
      N=Kd
35     IF (NSTRTPT+(Kd+N-1)*DELTAY .LE. NPTS) THEN
        N=N+1
        GO TO 35
        ELSE
        N=N-1
        ENDIF
        IF (N .EQ. Kd) THEN
        WRITE (*,*) 'M must equal',Kd
        M=Kd
        GO TO 85
        ENDIF
        IF (N .GT. 69) N=69
40    WRITE (*,*) 'M may range from',Kd
      WRITE (*,*) '              to',N
      WRITE (*,*) 'Enter M'
      READ (*,*) M
      IF (M .GE. Kd .AND. M .LE. N) GO TO 85
      GO TO 40
      ENDIF

45    IF (.NOT. NUFILE) GO TO 85

5     N=1
50    IF (NSTRTPT+N*(Kd+M-1) .LE. NPTS) THEN
```

```
      N=N+1
      GO TO 50
      ELSE
      N=N-1
      ENDIF
      IF (N .EQ. 1) THEN
      WRITE (*,*) 'Given the other parameters chosen thus far,'
      WRITE (*,*) 'Spacing can only be 1'
      DELTAY=1
       IF (NUFILE) THEN
       GO TO 60
       ELSE
       GO TO 85
       ENDIF
      ENDIF
      IF (.NOT. DSET .AND. LONG) THEN
      WRITE (*,*) 'Enter spacing between the ',Kd+M
      WRITE (*,*) 'data points of ',TITLE
      WRITE (*,*) 'to be processed '
      WRITE (*,*) ' '
      WRITE (*,*) 'If, for example, one is chosen, then ',Kd+M
      WRITE (*,*) 'consecutive points in ',TITLE
      WRITE (*,*) 'will be processed '
      WRITE (*,*) ' '
      ENDIF
55    WRITE (*,*) 'Spacing may range from          1 '
      WRITE (*,*) '                      to',N
      READ (*,*) DELTAY
      IF (DELTAY .GE. 1 .AND. DELTAY .LE. N) THEN
       IF (NUFILE) THEN
       GO TO 60
       ELSE
       GO TO 85
       ENDIF
      ELSE
      GO TO 55
      ENDIF

60    WRITE (*,*) 'Do you wish to adjust eigenvalues? (y/n)'
      READ (*,120) YN
      IF (YN .EQ. 'N' .OR. YN .EQ. 'n') THEN
      IF (NUFILE) GO TO 6
      GO TO 85
      ENDIF
      IF (YN .NE. 'Y' .AND. YN .NE. 'y') GO TO 60
2     WRITE (*,*) 'Discard or compensate eigenvalues? (d/c)'
      READ (*,120) DC
      IF (DC .EQ. 'D' .OR. DC .EQ. 'd') GO TO 65
      IF (DC .NE. 'C' .AND. DC .NE. 'c') GO TO 2
      WRITE (*,*) 'Enter estimate of the actual order of the system'
```

```
      WRITE (*,*) ' '
      IF (LONG) THEN
      WRITE (*,*) 'This estimate will be used to determine the '
      WRITE (*,*) 'number of eigenvalues compensated or discarded '
      ENDIF
65    WRITE (*,*) 'the estimate may range from               2'
      WRITE (*,*) '                                  to',Kd-1
      READ (*,*) NRT
      IF (NRT .GT. Kd .OR. NRT .LT. 2) THEN
      GO TO 65
      ELSEIF (.NOT. NUFILE) THEN
      GO TO 85
      ENDIF

6     NSTRTPT=1
70    IF (NSTRTPT+(Kd+M-1)*DELTAY .LE. NPTS) THEN
      NSTRTPT=NSTRTPT+1
      GO TO 70
      ELSE
      NSTRTPT=NSTRTPT-1
      ENDIF
      IF (NSTRTPT .EQ. 1) THEN
      WRITE (*,*) 'Given the other parameters chosen thus far,'
      WRITE (*,*) 'the starting point for processing the data'
      WRITE (*,*) 'must be the first point in the data file'
      GO TO 85
      ENDIF
      WRITE (*,*) 'Enter desired starting point in data file'
      IF (.NOT. DSET .AND. LONG) THEN
      WRITE (*,*) '1 indicates the first point in the data file '
      ENDIF
      WRITE (*,*) ' '
      WRITE (*,*) 'Given the other parameters chosen thus far,'
75    WRITE (*,*) 'the starting point may range from         1'
      WRITE (*,*) '                                  to',NSTRTPT
      READ (*,*) N
      IF (N .GE. 1 .AND. N .LE. NSTRTPT) THEN
      NSTRTPT=N
      ELSE
      WRITE (*,*) 'Enter starting point again'
      WRITE (*,*) ' '
      GO TO 75
      ENDIF
      IF (.NOT. NUFILE) GO TO 85

7     WRITE (*,*) 'Do you want the data matrix arrangement to be'
      WRITE (*,*) ' '
      WRITE (*,*) '1. Causal'
      WRITE (*,*) '2. Non-causal'
      WRITE (*,*) ' '
```

```
80    WRITE (*,*) 'Please enter 1 or 2 '
      READ (*,*) NCAUS
      IF (NCAUS .EQ. 1) THEN
      CAUSAL=.TRUE.
      ELSEIF (NCAUS .EQ. 2) THEN
      CAUSAL=.FALSE.
      ELSE
      GO TO 80
      ENDIF
      GO TO 85
9     WRITE (*,*) 'Enter title of file to contain parameters'
      READ (*,105) TITL
      OPEN(1,FILE=TITL)
      WRITE(1,105) TITLE
      WRITE(1,110) NPTS
      WRITE(1,110) NRT
      WRITE(1,110) Kd
      WRITE(1,110) M
      WRITE(1,110) DELTAY
      WRITE(1,110) NSTRTPT
      WRITE(1,110) NCAUS
      CLOSE(1)
      IF (DSET) GO TO 85

12    IF (DSET) THEN
      CLOSE(2)
      CLOSE(3)
      CALL SUBPLT(NOVERLAY)
      ENDIF

85    DSET=.TRUE.
      NUFILE=.FALSE.
      WRITE(*,*) ' '
      WRITE(*,*) '1. Data file to be processed                    ',T
     +ITLE
      WRITE(*,*) '   Number of data points in data file          ',NPTS
      WRITE(*,*) '2. Estimated order of the system               ',NRT
      WRITE(*,*) '3. Kd, the number of columns in the data matrix',Kd
      WRITE(*,*) '4.  M, the number of  rows   in the data matrix',M
      WRITE(*,*) '5. Spacing between data points being processed ',DELTA
     +Y
      WRITE(*,*) '6. First point in the data file to be processed',NSTRT
     +PT
      WRITE(*,*) '    Last point in the data file to be processed',NSTRT       IF

     +PT+Kd+M-1
      IF (NCAUS .EQ. 1) THEN
      WRITE(*,*) '7. Data matrix arrangement for processing          CA
     +USAL   '
      ELSE
```

```
      WRITE(*,*) '7. Data matrix arrangement for processing       NON-CA
     +USAL   '
      ENDIF
      WRITE(*,*) ' '
      WRITE(*,*) '8. Begin processing using above settings'
      WRITE(*,*) '9. Store parameters 1-7 in a file'
      WRITE(*,*) '10. Retrieve parameters 1-7 from a previously created
     +file'
      WRITE(*,*) '11. Reset overlays'
      WRITE(*,*) '12. Re-plot overlays'
      WRITE(*,*) '13. End this session of Kumaresan-Tufts signal process
     +ing'
      WRITE(*,*) ' '
      WRITE(*,*) 'Enter an integer from 1 to 12 to make changes as often
     + as you desire'
90    READ (*,*) NMENU
      IF (NMENU .LT. 1 .OR. NMENU .GT. 13) THEN
      WRITE(*,*) 'Enter an integer from 1 to 13'
      GO TO 90
      ENDIF

      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13),NMENU

8     OPEN(1,FILE=TITLE)
      READ(1,105) HEADER
      READ(1,110) NPTS
      READ(1,115) XQ
      READ(1,115) XQ
      DO 95 I=1,NPTS
      READ(1,115) D(I)
95    CONTINUE
      CLOSE(1)
      KdPLT=Kd
      WRITE(*,*) 'Enter title of file to contain real part of poles'
      READ(*,105) TITLER
      OPEN(2,file=TITLER)

      WRITE(*,*)'Enter title of file to contain imaginary part of poles'
      READ(*,105) TITLEI
      OPEN(3,file=TITLEI)
      WRITE(10,100) (KdPLT)
      WRITE(10,105) TITLER
      WRITE(10,105) TITLEI
100   FORMAT(I2)

      MN=MAX(M,Kd)

105   FORMAT(A)
110   FORMAT(I5)
115   FORMAT(E12.6)
```

```
120   FORMAT(A1)

C     Form data matrix

      DO 125 I=1,Kd+M
      Dy(I)=D((I-1)*DELTAY+NSTRTPT)
125   CONTINUE

130   DO 140 I=1,M
      DO 135 J=1,Kd
      A(I,J)=Dy(I+J)
135   CONTINUE
140   CONTINUE

      B(1)=Dy(1)
      DO 145 I=2,M
      B(I)=A(I-1,1)
145   CONTINUE

C     Begin singular value decomposition

      CALL SVD(MACHEP,M,Kd,MN,A,W,MATU,U,MATV,V,IERR,RV1)

C     Errors in SVD?
      IF (IERR .GT. 0.0) THEN
      WRITE (*,*) 'Error in singular value number ',IERR,STOP
      ENDIF
      IF (YN .EQ. 'N') GO TO 190

      DO 150 I=1,Kd
      XP(I)=0.0
150   CONTINUE

C     Discard or compensate eigenvalues
C     Order singular values

      XP(1)=W(1)
      DO 165 I=2,Kd
      DO 160 J=1,I
      IF (W(I) .GT. XP(J)) THEN
      DO 155 K=I+1,J,-1
155   XP(K)=XP(K-1)
      XP(J)=W(I)
      GO TO 165
      ENDIF
160   CONTINUE
      XP(I+1)=W(I)
165   CONTINUE

C     XP( ) now contains ordered singular values-XP(1) is the largest
```

```
C     Discard eigenvalues
      IF (DC .EQ. 'D') THEN
      DO 170 J=NRT+1,Kd
170   W(J)=(0.0)
      ELSE
C     Compensate eigenvalues
      AVG=0.0
      DO 175 J=NRT+1,Kd
      AVG=AVG+XP(J)**2
175   CONTINUE
      IF (Kd .GT. NRT) AVG=AVG/DBLE(FLOAT(Kd-NRT))

      DO 185 J=1,Kd
      DO 180 K=1,Kd
      IF ( W(J) .EQ. XP(K) ) THEN
         IF ( K .GT. NRT ) THEN
         W(J)=0.0
         ELSE
         W(J)=DSQRT(DABS( W(J)*W(J)-AVG))
         ENDIF
         GO TO 185
         ENDIF
180   CONTINUE
185   CONTINUE
      ENDIF

190   DO 200 I=1,M
      DO 195 J=1,M
      UT(I,J)=(U(J,I))
195   CONTINUE
200   CONTINUE

c     Form SIGMA+ (KdxM)
      DO 210 I=1,Kd
      DO 205 J=1,M
      SIGMA(I,J)=0.0
      IF (I .EQ. J .AND. W(J) .NE. 0.0) THEN
      SIGMA(I,J)=1.0D0/W(J)
      ELSE
      SIGMA(I,J)=0.0d0
      ENDIF
205    CONTINUE
210    CONTINUE

C     Form SIGMA (MxKd)
      DO 220 I=1,M
      DO 215 J=1,Kd
      SIG(I,J)=0.0
      IF (I .EQ. J) SIG(I,J)=W(J)
```

```
215    CONTINUE
220    CONTINUE

C      V=KdxKd,SIGMA+=KdxM,VS=KdxM
       CALL MXMUL(V,SIGMA,Kd,Kd,M,VS)

C      VS=KdxM,UT=MxM,AINV=KdxM
       CALL MXMUL(VS,UT,Kd,M,M,AINV)

C      Calculate matrix multiplication of AINV x B, where
C      AINV=KdxM,B=Mx1,XP=Kdx1
       CALL MXMUL(AINV,B,Kd,M,L,XP)

C      Calculate autoregressive coefficients from prediction coefficients
       IF (XP(Kd) .EQ. 0.0) THEN
       WRITE (*,*) 'ERROR, avoiding division by zero'
       STOP
       ELSE
       B(Kd)=1.0d0/XP(Kd)
       ENDIF
       DO 225 I=2,Kd
       B(I-1)=-B(Kd)*XP(Kd-I+1)
225    CONTINUE

       DO 230 I=1,Kd
       X(I)=-B(Kd-I+1)
       IF (NCAUS .EQ. 1) X(I)=-XP(Kd-I+1)
230    CONTINUE
       X(Kd+1)=1.0

C      Compute the roots of the polynomial in z

       CALL POLRT(X,COF,KD,ROOTR,ROOTI,IER)

       IF (IER .NE. 0) WRITE (*,*) 'ERROR with POLRT, IER=',IER,STOP

       DO 235 I=1,Kd
       WRITE(2,115) ROOTR(I)
       WRITE(3,115) ROOTI(I)
       S(I)=DCMPLX(ROOTR(I),ROOTI(I))
235    CONTINUE

       MAGPOL=0
       DO 240 I=1,Kd
       IF (CDABS(S(I)) .GE. 1.0d0) MAGPOL=MAGPOL+1
240    CONTINUE

       WRITE(*,*) '# of poles with magnitude <= 1',Kd-MAGPOL
       WRITE (*,*) 'HIT ANY KEY TO CONTINUE'
       READ (*,105) HEADER
```

```
C     Plot poles

      NOVERLAY=NOVERLAY+1
      CLOSE(2)
      CLOSE(3)
      CALL SUBPLT(NOVERLAY)

      J=0
      K=0

      DO 245 I=1,Kd
      IF (CDABS(S(I)) .LT. 1.0) THEN
      J=J+1
      K=K+1
      WRITE (*,*) S(I),CDABS(S(I))
      ENDIF
      IF (J .EQ. 20) THEN
      WRITE (*,*) 'Enter any key to continue'
      READ (*,105) HEADER
      J=0
      ENDIF
245   CONTINUE

      WRITE(*,*) 'Poles with magnitude less than one: ',K

      GO TO 85
13    STOP
      END
```

APPENDIX B: THE CADZOW-SOLOMON POLE EXTRACTION ALGORITHM

The following program implements the Cadzow-Solomon algorithm as described in Chapter 2 of this thesis. The program is written in Fortran 77. The SVD and root-finding subroutines called by this program are found in the EISPACK library [18]. The SVD subroutine is a translation from ALGOL as given in [19]. The matrix multiplication and graphics subroutines, also called by this program, are found in Appendix C and D respectively.

```
$LARGE
      INTEGER IERR,Kd,Kn,M,MN,MAGPOL,NSTRTPT,DELTAY
      INTEGER IER,NCAUS,NMENU,INSTRTPT
      INTEGER*2 KdPLT
      REAL*8 A(70,70),W(70),U(70,70),V(70,70),RV1(70)
      REAL*8 VS(70,70),UT(70,70),AINV(70,70),X(70)
      REAL*8 XP(70),B(70),SIGMA(70,70),SIG(70,70)
      REAL*8 COF(70),ROOTR(70),ROOTI(70)
      REAL   MAG
      REAL*8 D(1024),AVG,MACHEP/1.0E-16/,Dy(140),Dx(1024)
      COMPLEX*16 S(70)
      LOGICAL MATU/.TRUE./,MATV/.TRUE./,CAUSAL/.TRUE./,LONG/.TRUE./
      LOGICAL DSET/.FALSE./,NUFILE/.TRUE./
      CHARACTER TITLE*16,HEADER*64,YN*1,DC*1,TITLER*16,TITLEI*16
      CHARACTER TITL*16,TITLD*16

C     Enter parameters for processing

14    IF (DSET) CLOSE(10)
      NOVERLAY=0
      OPEN(10,FILE='PLOT')
      IF (DSET) GO TO 215
      WRITE (*,*) 'Welcome to signal processing using the'
      WRITE (*,*) 'Cadzow-Solomon method'
      WRITE (*,*) ' '
      WRITE (*,*) 'Do you want '
```

```
      WRITE (*,*) ' '
      WRITE (*,*) '1. The long version for beginners'
      WRITE (*,*) '2. The short version for pros'
      WRITE (*,*) ' '
25    WRITE (*,*) 'Please enter 1 or 2 '
      READ (*,*) N
      IF (N .EQ. 1) THEN
      LONG=.TRUE.
      ELSEIF (N .EQ. 2) THEN
      LONG=.FALSE.
      ELSE
      GO TO 25
      ENDIF

      WRITE (*,*) 'Session will begin with entry of parameters needed fo
     +r processing'
      WRITE (*,*)
      WRITE (*,*) 'Do you want to enter parameters from'
      WRITE (*,*) ' '
      WRITE (*,*) '1. The keyboard'
      WRITE (*,*) '2. A previously created file of parameters'
      WRITE (*,*) ' '
35    WRITE (*,*) 'Please enter 1 or 2 '
      READ (*,*) N
      IF (N .EQ. 1) THEN
      GO TO 8
      ELSEIF (N .EQ. 2) THEN
13    WRITE (*,*) 'Enter title of file containing parameters'
      READ (*,100) TITL
      OPEN(1,FILE=TITL)
      READ(1,100) TITLE
      READ(1,110) NPTS
      READ(1,110) NRT
      READ(1,110) Kd
      READ(1,110) M
      READ(1,110) DELTAY
      READ(1,110) NSTRTPT
      READ(1,110) NCAUS
      READ(1,100) TITLD
      READ(1,110) NDPTS
      READ(1,110) Kn
      READ(1,110) INSTRTPT
      CLOSE(1)
      GO TO 215
      ELSE
      GO TO 35
      ENDIF
      WRITE (*,*) ' '

8     WRITE (*,*) 'Enter title of file containing excitation waveform'
```

```
      READ (*,100) TITLD
      OPEN(8,FILE=TITLD)
      READ(8,100) HEADER
      READ(8,110) N
      IF (N .GT. 1024) THEN
      WRITE (*,*) 'Number of points in data file exceeds the dimension'
      WRITE (*,*) 'of the array used in the program to store the file'
      STOP
      ENDIF
      CLOSE(8)
      IF ((N .GE. NDPTS) .AND. DSET) THEN
      NDPTS=N
      GO TO 215
      ENDIF
      NDPTS=N

9     WRITE (*,*) 'Enter estimated order of waveform'
      IF (DSET) THEN
      MAXIMUM=NDPTS-M
      IF (MAXIMUM .GT. M-Kd-1) MAXIMUM=M-Kd-1
      IF (MAXIMUM .GT. NDPTS-INSTRTPT-Kn-M+1) THEN
      MAXIMUM=NDPTS-INSTRTPT-Kn-M+1
      ENDIF
      ELSE
      MAXIMUM=66
      ENDIF
      IF (MAXIMUM .EQ. 1) THEN
       WRITE (*,*) 'The estimated order of the waveform can only be 1'
      IF (DSET) GO TO 215
      GO TO 10
      ELSE
      IF (DSET) THEN
      WRITE (*,*) 'Given the other parameters chosen thus far,'
      ENDIF
45    WRITE (*,*) 'the order may range from                  1'
      WRITE (*,*) '                                 to',MAXIMUM
      READ (*,*) Kn
      IF (Kn .GE. 1 .AND. Kn .LE. MAXIMUM) THEN
      IF (DSET) GO TO 215
      GO TO 10
      ENDIF
      WRITE (*,*) 'Enter estimated order again'
      WRITE (*,*) ' '
      GO TO 45
      ENDIF
      IF (DSET) GO TO 215

10    INSTRTPT=1
55    IF (INSTRTPT+Kn+M-1 .GT. NDPTS) THEN
      INSTRTPT=INSTRTPT-1
```

```
      ELSE
      INSTRTPT=INSTRTPT+1
      GO TO 55
      ENDIF
      MSTRT=INSTRTPT

      IF (INSTRTPT .EQ. 1) THEN
      WRITE (*,*) 'The first point can only be 1'
      GO TO 215
      ELSE
      WRITE (*,*) 'Enter first point in waveform file to be processed'
65    WRITE (*,*) 'Given the other parameters chosen thus far,'
      WRITE (*,*) 'the starting point may range from          1'
      WRITE (*,*) '                                    to',MSTRT
      READ (*,*) INSTRTPT
      IF (INSTRTPT .GE. 1 .AND. INSTRTPT .LE. MSTRT) THEN
      IF (DSET) GO TO 215
      GO TO 1
      ENDIF
      WRITE (*,*) 'Enter starting point again'
      WRITE (*,*) ' '
      GO TO 65
      ENDIF
      IF (DSET) GO TO 215

1     IF (.NOT. DSET) NUFILE=.TRUE.
      IF (.NOT. DSET) NSTRTPT=1
      WRITE (*,*) 'Enter title of data file to be read'
      READ (*,100) TITLE
      OPEN(12,FILE=TITLE)
      READ(12,100) HEADER
      READ(12,110) NPTS
      IF (NPTS .GT. 1024) THEN
      WRITE (*,*) 'Number of points in data file exceeds the dimension'
      WRITE (*,*) 'of the array used in the program to store the file'
      STOP
      ENDIF
      CLOSE(12)

      IF (NUFILE) THEN
      GO TO 3
      ELSEIF (NSTRTPT+(Kd+M-1)*DELTAY .LE. NPTS) THEN
      GO TO 215
      ELSE
      GO TO 6
      ENDIF


3     IF (NUFILE) THEN
      MAXIMUM=69-Kn-1
```

```
      IF (MAXIMUM .GT. NPTS-69) MAXIMUM=NPTS-69
      MIN=2
      IF (MIN .EQ. MAXIMUM) THEN
      Kd=MIN
      WRITE (*,*) 'Given the other parameters chosen thus far,'
      WRITE (*,*) 'Kd must be ',MIN
      GO TO 4
      ENDIF

      WRITE (*,*) 'Enter Kd, >= the estimated order of the system '

      WRITE (*,*) 'Given the other parameters chosen thus far,'
75    WRITE (*,*) 'Kd may range from',MIN
      WRITE (*,*) '               to',MAXIMUM
      READ (*,*) Kd
      IF (Kd .GE. MIN .AND. Kd .LE. MAXIMUM) GO TO 4
      GO TO 75

      ELSEIF (DSET) THEN
      MAXIMUM=M-Kn-1
      IF (MAXIMUM .GT. NPTS-M) MAXIMUM=NPTS-M
      MIN=2
      N=MAXIMUM
85    IF (NSTRTPT+(N+M-1)*DELTAY .LE. NPTS) THEN
      MAXIMUM=N
      IF (MIN .EQ. MAXIMUM) THEN
      Kd=MIN
      GO TO 215
      ELSEIF (MAXIMUM .LT. MIN) THEN
      DELTAY=1
      IF (1+(2+M-1)*DELTAY .LE. NPTS) THEN
      Kd=2
      GO TO 135
      ENDIF
      WRITE (*,*) 'Error. Kd must be less than 2'
      Kd=2
      GO TO 215
      ENDIF
        WRITE (*,*) 'Given the other parameters chosen thus far,'
95      WRITE (*,*) 'Kd may range from               ',MIN
        WRITE (*,*) '                              to',MAXIMUM
        WRITE (*,*) 'Enter Kd'
        READ (*,*) Kd
        IF (Kd .GE. MIN .AND. Kd .LE. MAXIMUM) GO TO 215
        GO TO 95
      ELSE
        N=N-1
        GO TO 85
      ENDIF
      ENDIF
```

```
C     Determine M
4     IF (NUFILE) THEN
      WRITE (*,*) 'Enter M, the row dimension of the data matrix'
      IF (.NOT. DSET .AND. LONG) THEN
      WRITE (*,*) ' '
      WRITE (*,*) 'Note: Kd+M points in ',title
      WRITE (*,*) '      will be processed '
      WRITE (*,*) ' '
      ENDIF
105   WRITE (*,*) 'M may range from',Kd
      IF (NPTS-Kd .GT. 69) THEN
      WRITE (*,*) '              to        69'
      ELSE
      WRITE (*,*) '              to',NPTS-Kd
      ENDIF
      READ (*,*) M
      IF (M .GT. 69) THEN
      WRITE (*,*) 'M must also be less than 70'
      GO TO 105
      ELSEIF (M .LT. Kd) THEN
      WRITE (*,*) 'M must be greater than or equal to Kd, Kd= ',Kd
      GO TO 105
      ELSEIF (Kd+M .GT. NPTS) THEN
      WRITE (*,*) 'Kd+M must be less than or equal to',NPTS,','
      WRITE (*,*) 'the number of data points in',TITLE
      WRITE (*,*) ' '
      GO TO 105
      ENDIF
C     Begin part for data already set
      ELSE
      N=Kd
115     IF (NSTRTPT+(Kd+N-1)*DELTAY .LE. NPTS) THEN
        N=N+1
        GO TO 115
        ELSE
        N=N-1
        ENDIF
        IF (N .EQ. Kd) THEN
        WRITE (*,*) 'M must equal',Kd
        M=Kd
        GO TO 215
        ENDIF
        MAXIMUM=N
        IF (MAXIMUM .GT. 69) MAXIMUM=69
        IF (Kd+Kn+1 .EQ. MAXIMUM) THEN
        M=Kd+Kn+1
        GO TO 215
        ELSEIF (Kd+Kn+1 .GT. MAXIMUM) THEN
        WRITE (*,*) 'Kd must be reduced'
```

```
        GO TO 3
        ELSE
        MIN=Kd+Kn+1
        ENDIF
      IF (MIN .LT. Kn+Kd+1) MIN=Kn+Kd+1
125    WRITE (*,*) 'M may range from',MIN
      WRITE (*,*) '              to',MAXIMUM
      WRITE (*,*) 'Enter M'
      READ (*,*) M
      IF (M .GE. MIN .AND. M .LE. MAXIMUM) GO TO 215
      GO TO 125
      ENDIF

c     Determine DELTAY
135   IF (.NOT. NUFILE) GO TO 215
5     N=1
145    IF (NSTRTPT+N*(Kd+M-1) .LE. NPTS) THEN
      N=N+1
      GO TO 145
      ELSE
      N=N-1
      ENDIF
      IF (N .EQ. 1) THEN
      WRITE (*,*) 'Given the other parameters chosen thus far,'
      WRITE (*,*) 'Spacing can only be 1'
      DELTAY=1
       IF (NUFILE) THEN
       GO TO 165
       ELSE
       GO TO 215
       ENDIF
      ENDIF
      IF (.NOT. DSET .AND. LONG) THEN
      WRITE (*,*) 'Enter spacing between the ',Kd+M
      WRITE (*,*) 'data points of ',TITLE
      WRITE (*,*) 'to be processed '
      WRITE (*,*) ' '
      WRITE (*,*) 'If, for example, one is chosen, then ',Kd+M
      WRITE (*,*) 'consecutive points in ',TITLE
      WRITE (*,*) 'will be processed '
      WRITE (*,*) ' '
      ELSE
      WRITE (*,*) 'Enter spacing '
      WRITE (*,*) ' '
      ENDIF
155   WRITE (*,*) 'Spacing may range from          1 '
      WRITE (*,*) '                      to',N
      READ (*,*) DELTAY
      IF (DELTAY .GE. 1 .AND. DELTAY .LE. N) THEN
       IF (NUFILE) THEN
```

```
       GO TO 165
       ELSE
       GO TO 215
       ENDIF
      ELSE
      GO TO 155
      ENDIF

165   WRITE (*,*) 'Do you wish to adjust eigenvalues? (y/n)'
      READ (*,150) YN
      IF (YN .EQ. 'N' .OR. YN .EQ. 'n') THEN
      IF (NUFILE) GO TO 6
      GO TO 215
      ENDIF
      IF (YN .NE. 'Y' .AND. YN .NE. 'y') GO TO 165
2     WRITE (*,*) 'Discard or compensate eigenvalues? (d/c)'
      READ (*,150) DC
      IF (DC .EQ. 'D' .OR. DC .EQ. 'd') THEN
      NRT=Kd
      GO TO 175
      ENDIF
      IF (DC .NE. 'C' .AND. DC .NE. 'c') GO TO 2
      WRITE (*,*) 'Enter estimate of the actual order of the system'
      WRITE (*,*) ' '
      IF (LONG) THEN
      WRITE (*,*) 'This estimate will be used to determine the '
      WRITE (*,*) 'number of eigenvalues compensated or discarded '
      ENDIF
175   WRITE (*,*) 'the estimate may range from               2'
      WRITE (*,*) '                                 to',Kd+Kn+1
      READ (*,*) NRT
      IF (NRT .GT. Kd+Kn+1 .OR. NRT .LT. 2) THEN
      GO TO 175
      ELSEIF (.NOT. NUFILE) THEN
      GO TO 215
      ENDIF

6     NSTRTPT=1
185   IF (NSTRTPT+(Kd+M-1)*DELTAY .LE. NPTS) THEN
      NSTRTPT=NSTRTPT+1
      GO TO 185
      ELSE
      NSTRTPT=NSTRTPT-1
      ENDIF
      IF (NSTRTPT .EQ. 1) THEN
      WRITE (*,*) 'Given the other parameters chosen thus far,'
      WRITE (*,*) 'the starting point for processing the data'
      WRITE (*,*) 'must be the first point in the data file'
      GO TO 215
      ENDIF
```

```
      WRITE (*,*) 'Enter desired starting point in data file'
      IF (.NOT. DSET .AND. LONG) THEN
      WRITE (*,*) '1 indicates the first point in the data file '
      ENDIF
      WRITE (*,*) ' '
      WRITE (*,*) 'Given the other parameters chosen thus far,'
195   WRITE (*,*) 'the starting point may range from         1'
      WRITE (*,*) '                                  to',NSTRTPT
      READ (*,*) N
      IF (N .GE. 1 .AND. N .LE. NSTRTPT) THEN
      NSTRTPT=N
      ELSE
      WRITE (*,*) 'Enter starting point again'
      WRITE (*,*) ' '
      GO TO 195
      ENDIF
      IF (.NOT. NUFILE) GO TO 215

7     IF (DSET) THEN
      IF (NCAUS .EQ. 1) THEN
      NCAUS=2
      GO TO 215
      ELSE
      NCAUS=1
      GO TO 215
      ENDIF
      ENDIF
      WRITE (*,*) 'Do you want the data matrix arrangement to be'
      WRITE (*,*) ' '
      WRITE (*,*) '1. Causal'
      WRITE (*,*) '2. Non-causal'
      WRITE (*,*) ' '
205   WRITE (*,*) 'Please enter 1 or 2 '
      READ (*,*) NCAUS
      IF (NCAUS .EQ. 1) THEN
      CAUSAL=.TRUE.
      ELSEIF (NCAUS .EQ. 2) THEN
      CAUSAL=.FALSE.
      ELSE
      GO TO 205
      ENDIF
      GO TO 215


12    WRITE (*,*) 'Enter title of file to contain parameters'
      READ (*,100) TITL
      OPEN(1,FILE=TITL)
      WRITE(1,100) TITLE
      WRITE(1,110) NPTS
      WRITE(1,110) NRT
```

```
      WRITE(1,110) Kd
      WRITE(1,110) M
      WRITE(1,110) DELTAY
      WRITE(1,110) NSTRTPT
      WRITE(1,110) NCAUS
      WRITE(1,100) TITLD
      WRITE(1,110) NDPTS
      WRITE(1,110) Kn
      WRITE(1,110) INSTRTPT
      CLOSE(1)
      IF (DSET) GO TO 215

15    IF (DSET) THEN
      CLOSE(2)
      CLOSE(3)
      CALL SUBPLT(NOVERLAY)
      ENDIF

215   DSET=.TRUE.
      NUFILE=.FALSE.
      WRITE(*,*) ' '
      WRITE(*,*) '1. Data file to be processed                        ',T
     +ITLE
      WRITE(*,*) '   Number of data points in data file          ',NPTS
      WRITE(*,*) '2. Estimated order of the system               ',NRT
      WRITE(*,*) '3. Kd, the number of columns in the data matrix',Kd
      WRITE(*,*) '4.  M, the number of  rows   in the data matrix',M
      WRITE(*,*) '5. Spacing between data points being processed ',DELTA
     +Y
      WRITE(*,*) '6. First point in the data file to be processed',NSTRT
     +PT
      WRITE(*,*) '    Last point in the data file to be processed',NSTRT
     +PT+Kd+M-1
      IF (NCAUS .EQ. 1) THEN
      WRITE(*,*) '7. Data matrix arrangement for processing          CA
     +USAL   '
      ELSE
      WRITE(*,*) '7. Data matrix arrangement for processing       NON-CA
     +USAL   '
      ENDIF
      WRITE(*,*) ' '

      WRITE(*,*) '8. File containing excitation waveform             ',T
     +ITLD
      WRITE(*,*) '   Number of data points in above file         ',NDPTS
      WRITE(*,*) '9. Estimated order of the waveform             ',Kn

      WRITE(*,*) '10. First point in the file to be   '
      WRITE(*,*) '    input into the data matrix                 ',INSTR
     +TPT
```

```
      WRITE(*,*) ' '

      WRITE(*,*) '11. Begin processing using above settings'
      WRITE(*,*) '12. Store parameters 1-10 in a file'
      WRITE(*,*) '13. Retrieve parameters 1-10 from a previously created
     + file'
      WRITE(*,*) '14. Reset overlays'
      WRITE(*,*) '15. Re-plot overlays'
      WRITE(*,*) '16. End this session of Cadzow-Solomon signal processi
     +ng'
      WRITE(*,*) ' '
      WRITE(*,*) 'Enter an integer from 1 to 16 to make changes as often
     + as you desire'
225   READ (*,*) NMENU
      IF (NMENU .LT. 1 .OR. NMENU .GT. 16) THEN
      WRITE(*,*) 'Enter an integer from 1 to 16'
      GO TO 225
      ENDIF

      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),NMENU

11    OPEN(12,FILE=TITTLE)
      READ(12,100) HEADER
      READ(12,110) NPTS
      READ(12,120) XQ
      READ(12,120) XQ
      DO 235 I=1,NPTS
      READ(12,120) D(I)
235   CONTINUE
      CLOSE(12)

      OPEN(8,FILE=TITTLD)
      READ(8,100) HEADER
      READ(8,110) NDPTS
      READ(8,120) XQ
      READ(8,120) XQ
      DO 245 I=1,NDPTS
      READ(8,120) Dx(I)
245   CONTINUE
      CLOSE(8)

      KdPLT=Kd
      WRITE(*,*) 'enter title of file to contain real part of poles'
      READ(*,100) TITLER
      OPEN(2,FILE=TITTLER)


      WRITE(*,*)'enter title of file to contain imaginary part of poles'
      READ(*,100) TITLEI
      OPEN(3,FILE=TITTLEI)
```

```
      WRITE(10,130) (KdPLT)
      WRITE(10,100) TITLER
      WRITE(10,100) TITLEI
130   FORMAT(I2)

      MN=MAX(M,Kd+Kn+1)

100   FORMAT(A)
110   FORMAT(I5)
120   FORMAT(E12.6)
150   FORMAT(A)


      DO 255 I=1,Kd+M
      Dy(I)=D((I-1)*DELTAY+NSTRTPT)
255   CONTINUE

265   DO 285 I=1,M
      DO 275 J=1,Kd+Kn+1
      A(I,J)=Dy(I+J)
      IF (J .GE. Kd+1) A(I,J)=Dx(I+J+INSTRTPT-2-Kd)
275    CONTINUE
285    CONTINUE

      B(1)=Dy(1)
      DO 295 I=2,M
      B(I)=A(I-1,1)
295   CONTINUE

      N=Kd+Kn+1

C     Begin singular value decomposition

      CALL SVD(MACHEP,M,N,MN,A,W,MATU,U,MATV,V,IERR,RV1)

C     Errors in SVD?
      IF (IERR .GT. 0.0) THEN
      WRITE (*,*) 'Error in singular value number ',IERR,STOP
      ENDIF
      IF (YN .EQ. 'N') GO TO 385

      DO 305 I=1,Kd+Kn+1
      XP(I)=0.0
305   CONTINUE

C     Discard or compensate eigenvalues
c     Order singular values

      XP(1)=W(1)
```

```
      DO 335 I=2,Kd+Kn+1
      DO 325 J=1,I
      if (W(I) .GT. XP(J)) THEN
      DO 315 K=I+1,J,-1
315   XP(K)=XP(K-1)
      XP(j)=W(i)
      GO TO 335
      ENDIF
325   CONTINUE
      XP(I+1)=W(I)
335   CONTINUE

C     XP( ) now contains ordered singular values: XP(1) is the largest

C     Discard eigenvalues
      IF (DC .EQ. 'D') THEN
      DO 345 J=NRT+1,Kd+Kn+1
345   W(J)=(0.0)
      ELSE
C     Compensate eigenvalues
      AVG=0.0
      DO 355 J=NRT+1,Kd+Kn+1
      AVG=AVG+XP(J)**2
355   CONTINUE
      IF (Kd+Kn+1 .GT. NRT) AVG=AVG/DBLE(FLOAT(Kd+Kn+1-NRT))

      DO 375 J=1,Kd+Kn+1
      DO 365 K=1,Kd+Kn+1
      IF ( W(J) .EQ. XP(K) ) THEN
         IF ( K .GT. NRT ) THEN
         W(J)=0.0
         ELSE
         W(J)=DSQRT(DABS( W(J)*W(J)-AVG))
         ENDIF
         GO TO 375
         ENDIF
365   CONTINUE
375   CONTINUE
      ENDIF


385   DO 405 I=1,M
      DO 395 J=1,M
      UT(I,J)=(U(J,I))
395   CONTINUE
405   CONTINUE

C     Form SIGMA+ (Kd+Kn+1 x M)
      DO 425 I=1,Kd+Kn+1
      DO 415 J=1,M
```

```
      SIGMA(I,J)=0.0
      IF (I .EQ. J .AND. W(J) .NE. 0.0) THEN
      SIGMA(I,J)=1.0d0/W(J)
      ELSE
      SIGMA(I,J)=0.0D0
      ENDIF
415    CONTINUE
425    CONTINUE

C     Form SIGMA (M x Kd+Kn+1)
      DO 445 I=1,M
      DO 435 J=1,Kd+Kn+1
      SIG(I,J)=0.0
      IF (I .EQ. J) SIG(I,J)=W(J)
435    CONTINUE
445    CONTINUE

C     V=Kd+Kn+1xKd+Kn+1,SIGMA+=Kd+Kn+1xM,VS=Kd+Kn+1xM
      CALL MXMUL(V,SIGMA,Kd+Kn+1,Kd+Kn+1,M,VS)

C     VS=Kd+Kn+1xM,UT=MxM,AINV=Kd+Kn+1xM
      CALL MXMUL(VS,UT,Kd+Kn+1,M,M,AINV)

C     Calculate matrix multiplication of AINV x B, where
C     AINV=Kd+Kn+1xM,B=Mx1,XP=Kd+Kn+1x1
      CALL MXMUL(AINV,B,Kd+Kn+1,M,L,XP)

C     Compute autoregressive coefficients from prediction coefficients

      IF (XP(Kd) .EQ. 0.0) THEN
      WRITE (*,*) 'ERROR, avoiding division by zero'
      STOP
      ELSE
      B(Kd)=1.0d0/XP(Kd)
      ENDIF
      DO 455 I=2,Kd
      B(I-1)=-B(Kd)*XP(Kd-i+1)
455   CONTINUE

      DO 465 i=1,Kd
      X(I)=-B(Kd-I+1)
      IF (NCAUS .EQ. 1) X(I)=-XP(Kd-I+1)
465   CONTINUE
      X(Kd+1)=1.0

C     Compute the roots of the polynomial in z

      CALL POLRT(X,COF,KD,ROOTR,ROOTI,IER)

      IF (IER .NE. 0) WRITE (*,*) 'ERROR with POLRT, IER=',IER,STOP
```

```
      DO 475 I=1,Kd
      WRITE(2,120) ROOTR(I)
      WRITE(3,120) ROOTI(I)
      S(I)=DCMPLX(ROOTR(I),ROOTI(I))
475   CONTINUE

      MAGPOL=0
      DO 485 I=1,Kd
      IF (CDABS(S(I)) .GE. 1.0D0) MAGPOL=MAGPOL+1
485   CONTINUE

      WRITE(*,*) '# of poles with magnitude <= 1',Kd-MAGPOL
      WRITE (*,*) 'HIT ANY KEY TO CONTINUE'
      READ (*,100) HEADER

C     Plot poles
      NOVERLAY=NOVERLAY+1
      CLOSE(2)
      CLOSE(3)
      CALL SUBPLT(NOVERLAY)

      J=0
      K=0

      DO 495 I=1,Kd
      IF (CDABS(S(I)) .LT. 1.0) THEN
      WRITE (*,*) S(I),CDABS(S(I))
      J=J+1
      K=K+1
      ENDIF
      IF (J .EQ. 20) THEN
      WRITE (*,*) 'HIT ANY KEY TO CONTINUE'
      READ (*,100) HEADER
      J=0
      ENDIF
495   CONTINUE

      WRITE (*,*) 'Poles with magnitude less than one ',K
      WRITE (*,*) 'HIT ANY KEY TO CONTINUE'
      READ (*,100) HEADER

      GO TO 215

16    STOP
      END
^Z
```

APPENDIX C. MATRIX MULTIPLICATION

```
      SUBROUTINE MXMUL(A,B,RA,CA,CB,AB)
      INTEGER RA,CA,CB
      REAL*8 A(70,70),B(70,70),AB(70,70)

C     Calculates matrix multiplication of A x B=AB, where
C     A=RAxCA,B=CAxCB,AB=RAxCB

      DO 30 I=1,RA
      DO 20 J=1,CB
      AB(I,J)=0.0
      DO 10  K=1,CA
      AB(I,J)=AB(I,J)+A(I,K)*B(K,J)
10    CONTINUE
20    CONTINUE
30    CONTINUE
      RETURN

      END
```

APPENDIX D. GRAPHICS ROUTINE

```
      SUBROUTINE SUBPLT(NOVERLAY)


C
C     MS-FORTRAN Program using "Grafmatic" Library Subroutines.
C     Plots a Solid Line and Optional Overlay Plot for Comparison.
C     Written by M.A. Morgan with Latest Update August 1989.
C
C     Default Printer is "IBM Graphics" (e.g. Epson, Okidata, IBM)
C     With Plot Rotated 90 degrees From the Vertical.  "GrafPlus.Com"
C     May be Run to Rotate Plot Upright on Paper and to Use a Variety
C     of Impact Printers.  "GrafLaser.Com" May be Run to Use a Laser
C     Printer.   See GrafPlus/Laser Manual From Jewell Technology.
C
C
      CHARACTER*1 YN,YN1,DUM,YN2,SYMBOL,BELL,FEED,FFYN
       CHARACTER*4 LINE
       CHARACTER*7 SYMB
       CHARACTER*16 LTIT,CTIT,FNAME,TITLER,TITLEI
       CHARACTER*64 TITLE,HCOPY
      REAL      CRTR(70),CRTI(70),NRTR(70),NRTI(70)
       INTEGER*2 N,JROW,JCOL,ISYM1,ISYM2,ITYPE1,ITYPE2,NSCRN
       INTEGER*2 CYAN,GREEN,WHITE,YELLOW,RED,BLACK,BLUE,NTWO
      INTEGER*2 JROW1,JROW2,JCOL1,JCOL2,CROSS,KdPLT,I
      INTEGER*2 PURPLE,RUST
      EXTERNAL XFUN,YFUNP,YFUNN
       LINE='---- '
       WHITE=7
       GREEN=10
       CYAN=11
       YELLOW=14
       RED=12
       BLACK=0
       BLUE=1
      NTWO=2
      PURPLE=5
      RUST=6
       BELL=CHAR(7)
      FEED=CHAR(12)

C      Clear Screen and Put Up Introduction - on Blue Backgound for EGA
C      Only;  Another Background Color is Possible by Changing "BLUE"
C      in the Calls to QPREG and QOVSCN.
       CALL QSMODE(NTWO)
        CALL QPREG(0,BLUE)
```

```
      CALL QOVSCN(BLUE)
      WRITE(*,*) BELL
       NS=1
       NSCRN =16
      ITYPE2=0

C     Calling GRAFMATIC Routines and Plotting F1 Solid Line Graph
      ITYPE1=0
      ISYM1=-1
      NDOTS1=0
       JROW1=1
       JROW2=350
       JCOL1= 75
       JCOL2= 565
       XMIN=-1.2
       XMAX=1.2
       YMIN=-1.2
       YMAX=1.20
       YOVERX=1.115
       XORG=0.0
       YORG=0.0
     XST=-1.1
     XFIN=1.1
     YST=-1.1
     YFIN=1.1

25    CALL QSMODE(NSCRN)
       CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XMIN,XMAX,YMIN,YMAX,XORG,YORG,
     +1,YOVERX,1.5)
       CALL QSETUP(NDOTS1,CYAN,ISYM1,RED)
     IF(XFIN-XST .LE. 9.0) XMAJOR=0.6
     IF(XFIN-XST .LE. 6.0) XMAJOR=0.4
     IF(XFIN-XST .LE. 3.3) XMAJOR=0.2
     IF(XFIN-XST .GE. 9.0) XMAJOR=(XFIN-XST)/10.0

     MINOR=0
     LABEL=1
     NDEC=2
      CALL QXAXIS(XST,XFIN,XMAJOR,MINOR,LABEL,NDEC)
     YMAJOR=XMAJOR
      CALL QYAXIS(YST,YFIN,YMAJOR,MINOR,LABEL,NDEC)
c    Plot unit circle
     A=-1.0
     B=1.0

     CALL QCURV(XFUN,YFUNP,A,B)
     CALL QCURV(XFUN,YFUNN,A,B)

     IF (NOVERLAY-1 .LT. 1) THEN
      IF (NOVERLAY-1 .EQ. 0) THEN
```

```
       WRITE (*,3) NOVERLAY-1
       ELSE
       NZERO=0
       WRITE (*,3) NZERO
       ENDIF
      ELSEIF (NOVERLAY-1 .GT. 1) THEN
      WRITE (*,3) NOVERLAY-1
      ELSE
      WRITE (*,4) NOVERLAY-1
      ENDIF
3     FORMAT (I3,' OVERLAYS')
4     FORMAT (I3,' OVERLAY ')
      REWIND(10)

      DO 20 I=1,NOVERLAY
      READ (10,110) KdPLT
      READ (10,100) TITLER
      READ (10,100) TITLEI
      OPEN(2,FILE=TITLER)
      OPEN(3,FILE=TITLEI)
      NKd=KdPLT
      DO 27 J=1,KdPLT
      READ (2,120) NRTR(J)
      READ (3,120) NRTI(J)
      IF  (DSQRT(NRTR(J)**2+NRTI(J)**2) .GT. 1.1) THEN
      NKd=NKd-1
      NRTR(J)=0.0
      NRTI(J)=0.0
      ENDIF
27    CONTINUE
      PURPLE=5
      RUST=6
       WHITE=7
       GREEN=10
       CYAN=11
       YELLOW=14
       RED=12
       BLUE=1
      IF (I .EQ. 1) THEN
       CALL QSETUP(NDOTS1,CYAN,ISYM1,RED)
      ELSEIF (I .EQ. 2) THEN
       CALL QSETUP(NDOTS1,CYAN,ISYM1,GREEN)
      ELSEIF (I .EQ. 3) THEN
       CALL QSETUP(NDOTS1,CYAN,ISYM1,YELLOW)
      ELSEIF (I .EQ. 4) THEN
       CALL QSETUP(NDOTS1,CYAN,ISYM1,BLUE)
      ELSEIF (I .EQ. 5) THEN
       CALL QSETUP(NDOTS1,CYAN,ISYM1,WHITE)
      ELSEIF (I .EQ. 6) THEN
       CALL QSETUP(NDOTS1,CYAN,ISYM1,PURPLE)
```

```
      ELSEIF (I .EQ. 7) THEN
       CALL QSETUP(NDOTS1,CYAN,ISYM1,RUST)
      ELSE
       CALL QSETUP(NDOTS1,CYAN,ISYM1,RED)
      ENDIF
      CALL QTABL(ITYPE1,KdPLT,NRTR,NRTI)
20    CONTINUE

      READ (*,100) DUM
      GO TO 40
      HCOPY='HARDCOPY---> ENTER P OR p'
      CALL QPTXT(30,HCOPY,RED,25,1)
      CALL QCMOV(55,1)
      HCOPY='                                          '
      CALL QPTXT(40,HCOPY,BLACK,25,1)
      IF (DUM .NE. 'P' .AND. DUM .NE. 'p') GO TO 40
      CALL QPSCRN
      OPEN (1,FILE='PRN')
      WRITE (1,160) FEED
100   FORMAT(A)
110   FORMAT(I2)
120   FORMAT(E12.6)
160   FORMAT(' ',A,\)
40    CONTINUE
      CALL QSMODE(NTWO)
      CALL QPREG(0,BLUE)
      CALL QOVSCN(BLUE)
      WRITE(*,*) NKd,'points were plotted'
       RETURN
       END

      REAL FUNCTION XFUN(T)
      XFUN=T
      RETURN
      END

      REAL FUNCTION YFUNP(T)
      YFUNP=SQRT(1.0-T*T)
      RETURN
      END

      REAL FUNCTION YFUNN(T)
      YFUNN=-SQRT(1.0-T*T)
      RETURN
      END
```

LIST OF REFERENCES

1. C. E. Baum, On the Singularity Expansion Method for the Solution of Electromagnetic Interaction Problems, Air Force Weapons Laboratory Interaction Note 88, December 1971.

2. D. L. Moffatt and R. K. Mains, "Detection and discrimination of radar targets," IEEE Trans. on Antennas and Propagation, AP-23, May 1975, pp 358-367.

3. Michael A. Morgan, "Singularity expansion representations of fields and currents in transient scattering," IEEE Trans. on Antennas and Propagation , AP-23, May 1984, pp 466-473.

4. Ramdas Kumaresan, and D.W. Tufts, "Estimating the parameters of exponentially damped sinusoids, and pole-zero modeling in noise," IEEE Trans. Acoustics, Speech, and Sig. Proc., December 1982, pp 833-840.

5. James A. Cadzow and Otis M. Solomon Jr., "Algebraic approach to system identification," IEEE Trans. Acoustics, Speech and Sig. Proc., ASSP-34, 3, June 1986, pp 462-469.

6. E.M. Kennaugh, "The K-pulse concept," IEEE Trans. on Antennas and Propagation, AP-29, March 1981.

7. James B. Dunavin, Identification of Scatterers Based Upon Annihilation of Complex Natural Resonances, Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1985.

8. Michael A. Morgan and James B. Dunavin, "Discrimination of scatterers using natural resonance annihilation," Abstracts of 1986 National Radio Science Meeting, Philadelphia, PA, June 1986.

9. K. M. Chen, D. P. Nyquist, E.J. Rothwell, L.L. Webb, and B. Drachman, "Radar target discrimination by convolution of radar returns with extinction-pulses and single-mode extraction signals." IEEE Trans. on Antennas and Propagation, AP-34, July 1986, pp 896-904.

10. Michael A. Morgan, "Scatterer discrimination based upon natural resonance annihilation," Journal of Electromagnetic Waves and Applications, 2,2, 1987, pp 155-176.

11. Michael A. Morgan, Signal Generator Program, Naval Postgraduate School, Monterey, CA, 1989.

12. Michael A. Morgan, Time-Domain Thin-Wire Integral Equation Program, Naval Postgraduate School, Monterey, CA, 1989.

13. Choong Y. Chong, Investigation of Non-linear Estimation of Natural Resonances in Target Identification, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1983.

14. S. Lawrence Marple, Jr., Digital Spectral Analysis with Applications, (Englewood Cliffs, NJ: Prentice-Hall, Inc.) 1987.

15. Ramdas Kumaresan, Estimation the Parameters of Exponentially Damped or Undamped Sinusoidal Signals in Noise, Ph.D. Thesis, University of Rhode Island, Kingston, RI, October 1982.

16. S. A. Norton, "Radar target classification by natural resonances: signal processing algorithms," Engineer's Thesis, Naval Postgraduate School, Monterey, CA, Mar 1988.

17. Norman J. Walsh, "Bandwidth and Signal to Noise Ratio Enhancement of the NPS Transient Scattering Laboratory," Master's Thesis, Naval Postgraduate School, Monterey, CA, Dec 1989.

18. J.M. Boyle, EISPACK Subroutine Library, Argonne National Laboratory, 1973.

19. G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," Numer. Math., 14, 1970, pp 403-420.

INTIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0412 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Defense Logistic Studies Information Exchange U.S. Army Logistics Management College Fort Lee, Virginia 23801-6043	1
4.	Department Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5002	1
5.	Professor Michael A. Morgan, Code 62Mw Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5002	10
6.	Professor Ralph Hippenstiel, Code 62Hi Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5002	1
7.	Commandant of the Marine Corps Code TE 06 Headquarters, U.S. Marine Corps Washington, D.C. 20380-0001	1
8.	Electronic Material Officer USS Blue Ridge (LCC-19) FPO San Francisco, California 96628-3300	2
9.	Dr. John N. Entzminger Director, Tactical Technology Office Defense Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, Virginia 22209	2

10. Mr. D. Jiglio 1
Tactical Technology Office
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, Virginia 22209

11. Dr. Arthur Jordan 1
Code 1114SE
Office of Naval Research
800 N. Quincy St.
Arlington, Virginia 22209

12. Dr. Rabiner Madan 1
Code 1114SE
Office of Naval Research
800 N. Quincy St.
Arlington, Virginia 22209

13. Professor K. M. Chen 1
Department of Electrical Engineering and
Systems Science
Michigan State University
East Lansing, Michigan 48824

14. Dr. M. L. VanBlaricum 1
Toyon Research Corporation
75 Aero Camino, Suite A
Goleta, California 93160

15. Capt. Peter D. Larison 1
1145 Lenox Court
Cape Coral, Florida 33904

16. Dr. Carl Baum 1
Air Force Weapons Laboratory
Kirtland Air Force Base, New Mexico 87117

17. Mr. Daniel Carpenter 1
TRW Military Electronics Division
One Rancho Carmel
San Diego, California 92128

18. Dr. C. Ray Smith 1
AMSMI-RD-AS-RA
Redstone Arsenal, Alabama 35898

19. Professor Herbert Uberall 1
Physics Department
Catholic University of America
Washington, D.C. 20064