An Exploration of the Teager Operator

Rebecca Fiebrink
MUMT 605, Autumn 2004
Final Project Report
Dr. Philippe Depalle, Professor

Abstract

The Teager energy operator is a simple function related to the “energy” of a sinusoidal signal. This paper explores the behavior and usefulness of the Teager energy operator in several contexts, drawing on the work of several researchers in signal processing and speech analysis. It also describes an implementation of Teager-based analysis using a filter bank and the DESA-2 algorithm to retrieve information regarding the spectral content of a multicomponent signal.

Introduction

In 1990, James F. Kaiser published “On a simple algorithm to calculate the 'energy' of a signal,” which derived a mathematically simple, real-time measure of the “energy” of a sinusoidal signal [1]. This function, commonly called the “Teager energy operator,” has been the subject of several subsequent studies with focus on its many uses in discrete and analog signal processing.

This project involves the review of several such studies and the verification of their results through computer simulation, and this paper serves as a companion to the project scripts listed in Appendix II. First, this paper provides an overview of the work of Kaiser and selected other researchers on the Teager operator and its use in various contexts. Then, after discussing the function implementation in both Max/MSP and MATLAB, its basic behavior is illustrated within these two environments. Subsequent sections explore the behavior of the Teager operator on noisy signals, the behavior and precision of the DESA-2 algorithm on various signals, the use of a generalized Teager operator in resolving two closely spaced tones, and the implementation of a filter bank system for Teager-based analysis of multicomponent signals.

The results of this project demonstrate the ease of implementation of the Teager operator, generally support and elucidate the results of several research papers on the subject, illustrate the effects of noise on Teager-based analysis in several different contexts, and provide an example of a new basic analysis system. These results suggest that the Teager operator should be considered as a potential analysis tool in many practical signal processing tasks, and that further research will likely continue to explore the application of the Teager operator to new problems and contexts.

Background

[1] outlines the motivation, derivation, and basic use of a simple algorithm to calculate the “energy” of a discrete, single-component sinusoidal signal. Using the equation for the total energy of a spring-mass system and the approximation $\sin(x) = x$ for small values of x,

Kaiser derives the following expression for the input energy of a discrete signal (where A and Ω are the instantaneous amplitude and frequency):

$$E_n = A^2\Omega^2 = x_n^2 - x_{n+1}*x_{n-1}$$

The paper continues by demonstrating the basic behavior of this operator on an exponentially-damped sinusoid, a chirp signal, and a two-component signal. It is of note that the non-linearity of the Teager operator makes it impractical for direct analysis of multicomponent signals, and [1] recommends the use of a filter bank in this situation.

Many subsequent studies have been done to analyze the properties of the Teager operator and its extensibility to other signal processing tasks. Much of this work was discussed in my in-class presentation and therefore will not be duplicated in this paper.

Of most relevance to the work presented in this paper are [5] and [4]. [5] offers two simple algorithms, DESA-1 and DESA-2, for separating the amplitude and frequency contributions to the Teager operator output. [4] describes a generalized Teager energy function, of which the operator described in [1] is a special case. This generalized function, which is calculated with a variable lag rather than a lag of one sample, can be adjusted to enhance the presence of the difference or summation frequency of two closely-spaced tones, thereby allowing more precise spectral analysis of a signal. In addition to [4] and [5], the reader is referred to [2], [3], [6], and [7] for other interesting discussion of the Teager operator's properties and applications.

Basic Implementation in Max/MSP

The first task of this project involved the writing of a Max/MSP patch to perform Teager operator analysis of a signal in real-time. The resulting patch (teag.mxb) is shown below in Fig. 1. Clearly, the Teager operator's implementation in Max/MSP is quite simple.

This patch was then used to analyze the same basic input signal types as in [1]. The patch simple_teager.mxb demonstrates that when a sine wave with time invariant amplitude and frequency is input to the “teag” object, the output of the Teager function is a DC signal. Furthermore, increasing the amplitude or frequency of the input sine wave results in an increased output of the Teager function.

The “teag” object was used in the same manner in three other patches to qualitatively verify the behavior of the Teager function on damped sine, chirp, and two-component signals, respectively. Each of these patches stores the signal and the Teager function value in a buffer so that they may be compared. The contents of these buffers for typical input signal parameters appear below in Figs. 2-4.

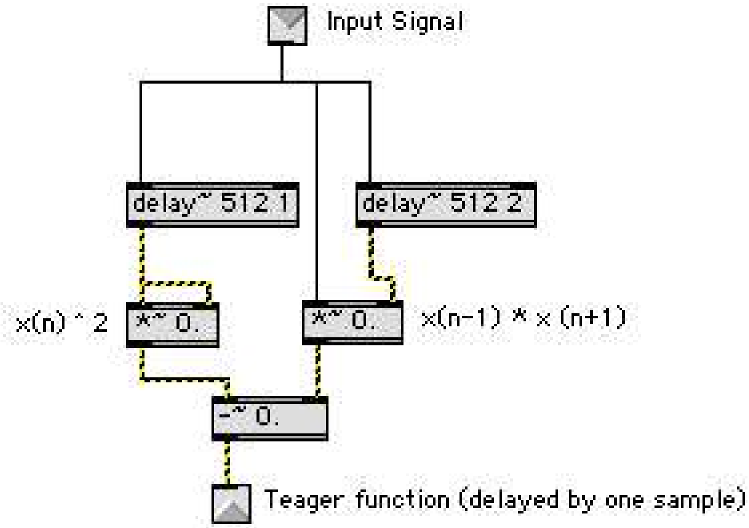


Figure 1: Max/MSP patch to calculate the Teager operator on a signal input

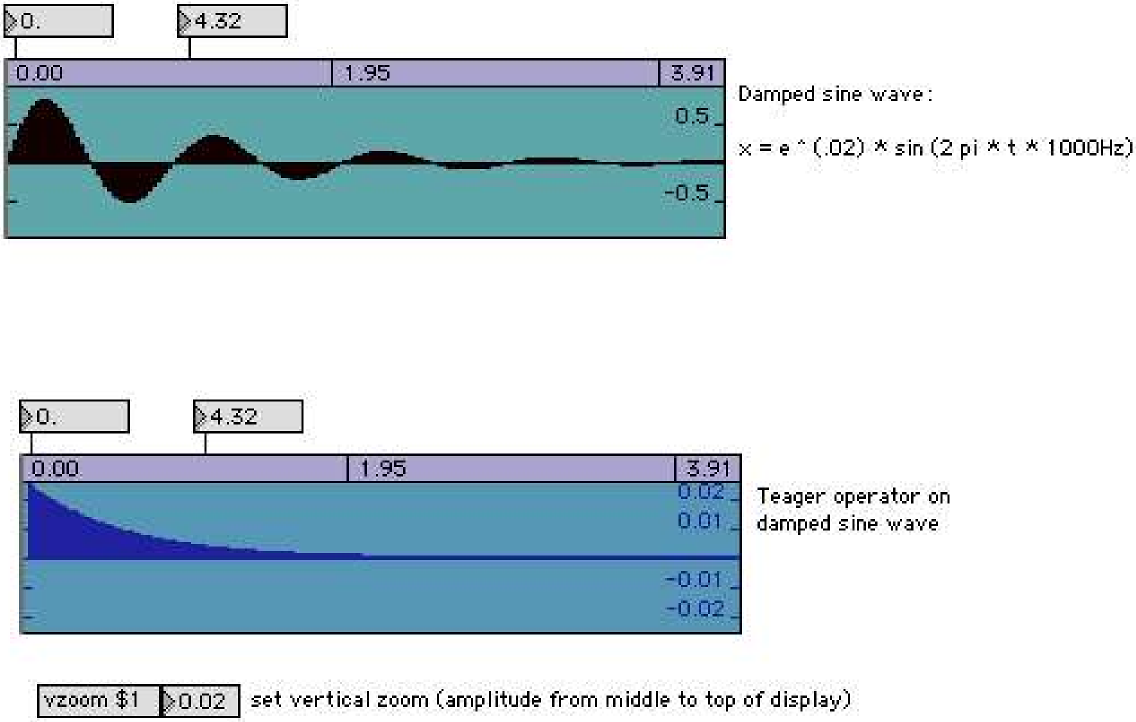
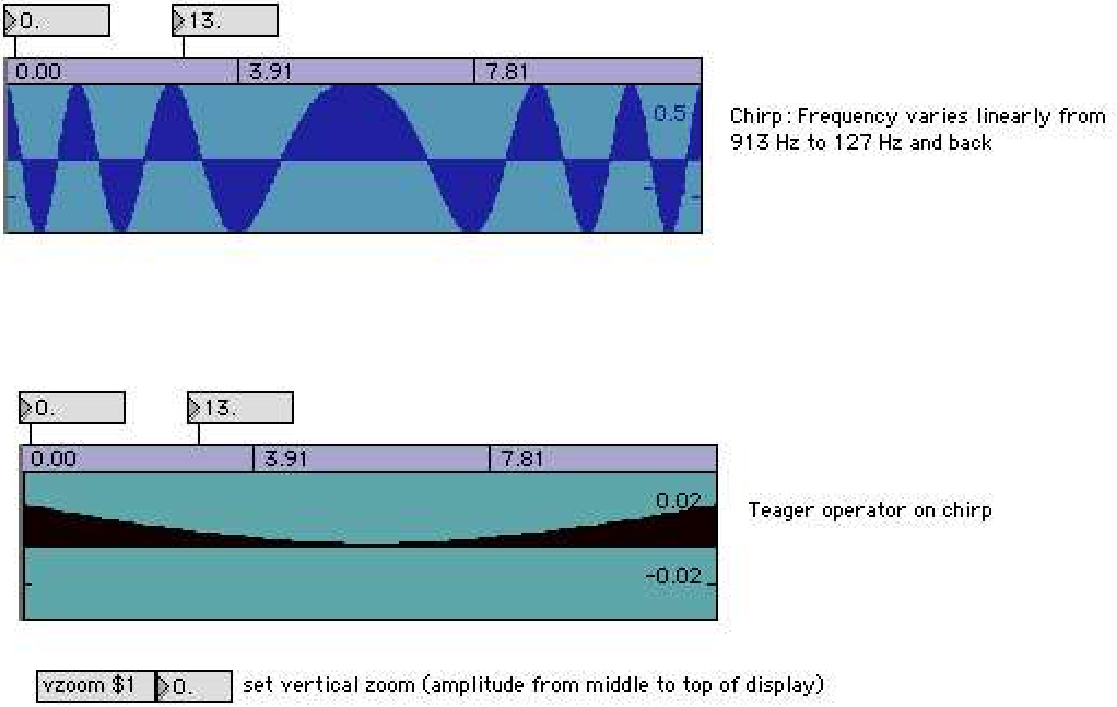


Figure 2: Max/MSP example of a damped sine wave and its Teager operator output

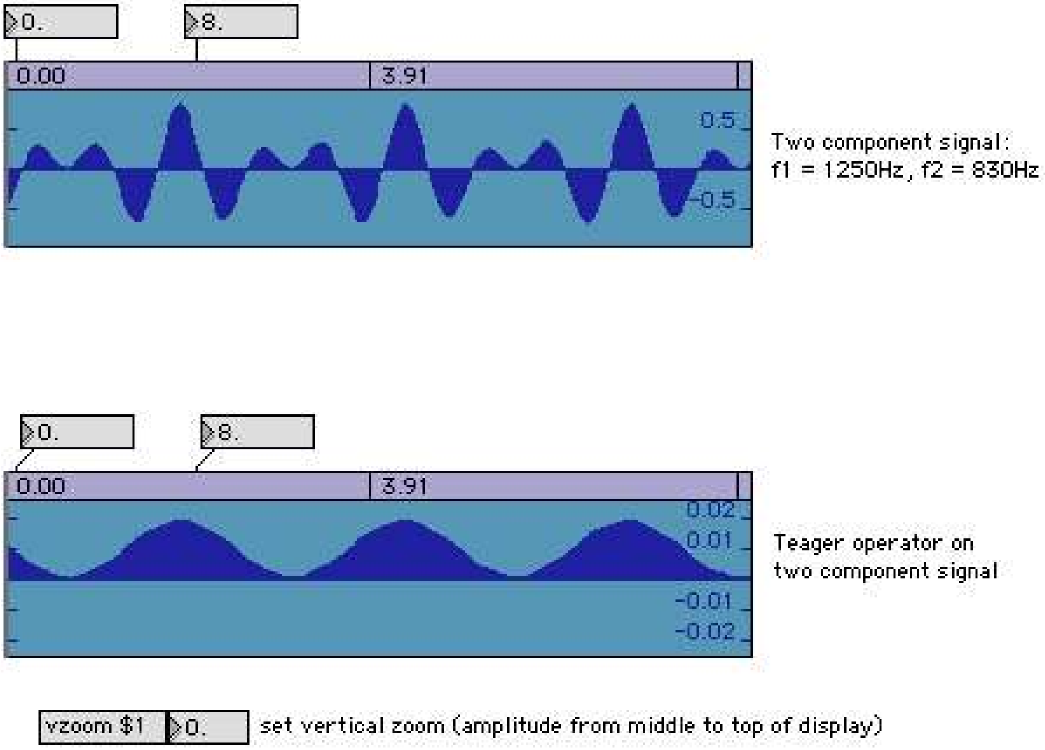


Chirp: Frequency varies linearly from 913 Hz to 127 Hz and back

Teager operator on chirp

set vertical zoom (amplitude from middle to top of display)

Figure 3: Max/MSP example of a chirp signal and its Teager operator output\



Two component signal:
f1 = 1250Hz, f2 = 830Hz

Teager operator on two component signal

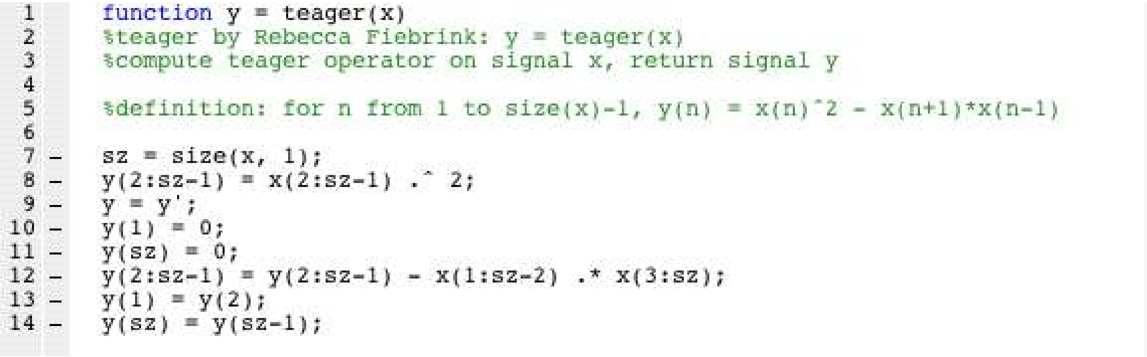
set vertical zoom (amplitude from middle to top of display)

Figure 4: Max/MSP example of a two-component sine wave and its Teager operator output

The buffer outputs demonstrate that the Teager operator behaves in the manner described by [1]. However, quantitative analysis of the operator output is not practical in Max/MSP, so no further observations were made in this environment. The above work simply demonstrates that Teager energy function can be computed directly in Max/MSP, making the function output value available as a signal that behaves precisely as expected on basic input signals. Furthermore, the function output need only be delayed by only one sample from the input signal for real-time analysis.

Implementation and Exploration in MATLAB

Although Max/MSP is an environment familiar to many musicians, MATLAB allowed greater precision, efficiency, and flexibility in further exploration of the Teager operator in this study. Using the same equation for the discrete Teager operator as in the MSP implementation, a short MATLAB function was written. The code for this function appears below in Fig. 5 to demonstrate the succinctness of implementation. Note that this function assigns the first and last vector values to be identical to the second and second-to-last values, respectively. The decision could be made to omit these values entirely, but this implementation allows the size of the Teager vector be identical to the size of the input vector while not introducing discontinuities in the Teager vector. This allowed for the simplest approach to analysis and graphing.



```
function y = teager(x)
%teager by Rebecca Fiebrink: y = teager(x)
%compute teager operator on signal x, return signal y

%definition: for n from 1 to size(x)-1, y(n) = x(n)^2 - x(n+1)*x(n-1)

sz = size(x, 1);
y(2:sz-1) = x(2:sz-1) .^ 2;
y = y';
y(1) = 0;
y(sz) = 0;
y(2:sz-1) = y(2:sz-1) - x(1:sz-2) .* x(3:sz);
y(1) = y(2);
y(sz) = y(sz-1);
```

Figure 5: MATLAB function for calculating the Teager operator

MATLAB Part 1: Basic behavior

The MATLAB component of this project is broken down into four parts, each of which focuses on a different facet of the operator and its use. Scripts with the code for each part appear in part1.m through part4.m, and the graphs output by all scripts appear in Appendix I. Part 1 of the MATLAB component of this project uses the function above to analyze the same set of basic input signals as were analyzed earlier in Max/MSP. This script also generates noisy versions of each of the signals, with a signal-to-noise ratio (SNR) of

25, to demonstrate the sensitivity of the Teager operator to a modest amount of noise. Finally, the effects of noise are further explored by applying the Teager operator to different versions of the same signal, each having a different SNR.

The figures “Teager on Sine and Damped Sine Waves” and “Teager on Chirp and 2-Component Signals” (see Appendix I) demonstrate that the output of the Teager operator on each signal matches the behavior found in the Max/MSP analysis and described in [1]. Furthermore, it is apparent that the noise has a large impact on the Teager operator output. For the noisy single component sine wave, the Teager operator behavior deviates greatly from the steady line that is output without noise. However, basic behavior of the Teager operator on noisy damped sine, chirp, and two-component signals is still apparent.

A related question asks, how might the magnitude of the signal-to-noise ratio affect the Teager operator? The figure “Effect of Noise on the Teager Operator” demonstrates that, for a single component sinusoid, even a very low SNR of 200 results in noticeable changes in the Teager output. The quality of the Teager operator output significantly degrades as the SNR decreases, and the degradation of the operator output is noticeably more dramatic than that of the source signal. This is a logical result of the fact that the Teager operator is calculated on only three adjacent samples rather than on a large window.

MATLAB Part 2: Using DESA-2 to separate amplitude and frequency

Part 2 of the MATLAB component uses an implementation of the DESA-2 algorithm to separate amplitude and frequency contributions to the Teager operator behavior. First, DESA-2 is used to separate amplitude and frequency components for the set of basic input signals. The figures “Real and DESA-2 Amplitude and Frequency for Simple Sine,” “Real and DESA-2 Amplitude and Frequency for Damped Sine,” and “Real and DESA-2 Amplitude and Frequency for Chirp” display the DESA-2 calculations alongside the true amplitude and frequency values. Visual comparisons verify that DESA-2 is generally accurate and fast to respond to changes in the input.

There are, however, some notable discrepancies between the real values and the DESA-2 calculations. Most interesting is the behavior of the DESA-2 frequency measure on the damped sine wave. The initial frequency value of 980 for the first few samples is followed by a steady line which, at first glance, seems to coincide with the behavior of the true frequency (which remains steady). However, this steady frequency value is approximately 1003 Hz, while the true value of the frequency is 1000 Hz.

Furthermore, it is interesting to note that the DESA-2 values for the chirp signal are less smooth than those for the other signals. The abrupt drop in the calculated amplitude around sample 25 may not be a failure of DESA-2, because the amplitude of the input signal generated by the MATLAB chirp function is not really 1 at that point. However, the frequency and amplitude tracking for the chirp signal show a distinctly wavy behavior that is not as apparent for the other signals.

It is also notable that DESA-2 occasionally calculates complex values for frequency and amplitude on these simple noiseless signals, because of the square root operations. The MATLAB script for Part 2 handles this by only considering the real part of any frequency and amplitude values calculated, and this seems to work well.

After verifying that DESA-2 performed reasonably well on basic signal inputs, the performance of DESA-2 was examined with respect the frequency of the signal. [5] notes that the presence of an arcsine computation in the DESA-2 algorithm allows tracking of frequencies up to only one-fourth of the sampling frequency. While the authors assert that the performance of DESA-2 led to errors of 1% or less in general, no discussion is offered on the effect of frequency on error magnitude. Therefore, five frequency values were chosen that ranged from one-twentieth the sample rate to one-fourth the sample rate. DESA-2 was run on both simple and damped sine waves with these frequencies.

The average magnitude of the percentage of error for both DESA-2 amplitude and frequency appears in “% Error for DESA-2 Amplitude and Frequency, Damped and Simple Sine.” (In these graphs, the x-axis shows the frequency of each input signal in radians/sample.) These plots indicate that the magnitude of error is similar for frequency and amplitude estimation for a particular wave shape, and the magnitude of error is much higher for the damped sine wave input. However, the error remains very low for even the damped sine wave (less than 2%). Interestingly, the error decreases as the frequency approaches one-fourth the sample rate, and the shape of the error decay is similar for both frequency and amplitude, and for both the damped and simple sine waves. Overall, this analysis suggests that DESA-2 may be used to track the frequency and amplitude accurately, and that while frequencies close to one-fourth the sampling rate will have the least error, error is generally low for all input frequencies.

Finally, the effect of noise on DESA-2's ability to track frequency and amplitudes was examined. The effects of the noise are shown in the DESA-2 output in “Effect of Noise on DESA-2,” and it is apparent that even a SNR of 100 results in serious degradation of the algorithm's ability to track frequency. The script also outputs the mean percentage error for amplitude (1.0%) and frequency (157%) tracking on this signal, and it is apparent that a large difference appears between the magnitude of the error for amplitude and frequency. The percentage error for the amplitude remains tiny. This result suggests that DESA-2 might still be useful on noisy signals if only the amplitude measurement is desired.

The MATLAB analysis of the behavior of DESA-2 supports [5]'s finding that DESA-2 can track frequency and amplitude relatively accurately in the absence of noise. For this reason, DESA-2 is integrated into the filter bank-based analyzer at in Part 4. However, idiosyncrasies such as the bumpiness in the calculated frequency of the chirp, errors in misidentification of frequency such as for the damped sine wave, and high sensitivity of the frequency measure to noise indicate that DESA-2 may be more appropriate for some analysis contexts than for others.

MATLAB Part 3: Using the Teager operator to resolve two tones

In order to use the Teager energy function in the same manner as in [4], a new MATLAB function was written to allow for variable lag parameters. This function appears in gtkef.m. For a given lag m and input length N, the generalized Teager operator is undefined for samples 1 to m and samples (N-m+1) to N. As in the implementation of the original Teager operator, these values are set to the nearest defined samples (sample (m+1) and sample (N-m)). Again, this allowed the Teager output vector to be the same size as the input vector, easing the analysis process. Experimentation with cutting the output vector size by removing these undefined samples had no perceivable effect on the operator's use in peak detection.

The frequencies and lag parameter values chosen for analysis are the same as in [4]. The signal analyzed was made up of one 330Hz and one 300Hz sine wave. First, the output of the original Teager operator, with a lag parameter of 1, is shown with its FFT (“TKEO on noiseless two-component signal” and “FFT of TKEO on noiseless two-component signal”). As noted by [4], analysis of the original operator reveals a strong frequency component at the difference frequency of 30Hz, but no corresponding component at the summation frequency of 630Hz.

Next, noise was added to the signal, and it was analyzed with the original Teager operator. The Teager operator and its FFT appear in “TKEO on noisy two-component signal, SNR=25” and “FFT of TKEO on noisy two-component signal, SNR=25.” The spectral peak at 30Hz is less pronounced but still somewhat apparent, and there is no indication of a peak at 630Hz.

“GTKEO on noiseless signal, m=25” and “FFT of GTKEO on noiseless signal, m=25” demonstrate the use of the generalized Teager operator at a lag value that accentuates the summation frequency. The Teager operator output itself clearly shows a prominent high frequency component, especially when compared to the output when m=1 as in the original case. This is indeed reflected in its FFT, which shows a clear peak at 630Hz and none at 30Hz. When the generalized operator with a lag parameter of m=25 is applied to the noisy version of the signal, the behavior of the operator is much less clear from the time-domain output alone (“GTKEO on noisy signal, SNR=25, m=25.”) However, the spectrum of the generalized Teager operator still shows a very pronounced peak at 630Hz (“FFT of GTKEO on noisy signal, SNR=25, m=25”).

Next, the generalized Teager operator with a lag parameter of m=6 was used to analyze a noiseless signal and a noisy signal. [4] asserts that m=6 will enhance the difference frequency even more than m=1, and this is indeed the case. “GTKEO on noiseless signal, m=6” and “FFT of GTKEO on noiseless signal, m=6” appear very similar to the original Teager operator and its FFT with m=1; however, comparison of the two FFTs shows that the peak of the generalized operator at 30Hz is significantly higher due to the decibel scaling of the y-axis. Similarly, at first glance the generalized Teager operator output with m=6 and its FFT (in “GTKEO on noisy signal, SNR=25, m=6” and “FFT of GTKEO on noisy signal, SNR=25, m=6”) appear quite alike to those for the original operator on noisy input. However, again, the spectral peak at 30Hz is in fact more prominent in the case where m=6.

The above results support [4]'s finding that changing the lag parameter can result in enhancing the summation or difference frequencies of two component tones. Furthermore, noise with a SNR of 25 does not degrade the prominence of the summation or difference components in the FFT to the point where they are no longer easily distinguishable.

MATLAB Part 4: Using Teager with a filter bank

The final component of this project involved the use of a filter bank to allow Teager analysis of multicomponent signals. For this purpose a function (myfilters1.m) was written to implement a linear-phase response filter bank with a logarithmically scaled, variable frequency range. This function also applies the original (m=1) Teager operator on the output of each filter. “Teager analysis of filtered 440Hz signal”, “Teager analysis of filtered 440Hz + 1000Hz signal”, and “Teager analysis of filtered 440Hz + 1000Hz + 300Hz signal” show the output of the filter bank for 1-, 2-, and 3-component signals. In these graphs, the x-axis corresponds to the sample number, the y-axis corresponds to the filter number, and the z-axis shows magnitude. Each of these graphs demonstrates the expected steady output of each filter after an initial delay. As long as no more than one signal component is present in the frequency range of each filter, such a system can be used to obtain Teager energy measures for each frequency band.

This system was then used to analyze recorded musical sounds, which consisted of multiple frequency components. “Teager analysis of filtered piano C3” displays the Teager analysis of a piano note. When more than one partial is present in the frequency band for a single filter in the higher frequency range, the Teager output is periodic as expected.

Because of the ability to obtain Teager energy measures within each frequency band, it should also be possible to use DESA-2 to obtain the frequency and amplitude values within each band. The final portion of the Part 4 MATLAB script uses a second filtering function (myfilters2.m) to identify the primary frequency components in a multicomponent signal along with their amplitudes. This is accomplished by first isolating the portion of the signal after the initial delay period, then averaging the DESA-2 frequency calculation for each filter. (This assumes that each filter has at most one frequency component of the signal within its range.) Because of small amounts of leakage into adjacent filters, there exist many duplicate values among the set of frequency averages. Therefore, only the unique frequency values are saved, and the filter numbers corresponding to these frequencies are identified. The effects of leakage are minimized by zeroing the outputs of any filters not in this set. Finally, the mean amplitudes of the filters in the set are matched with the corresponding frequencies.

The result is a set of frequency and amplitude pairs identifying the signal components. This set is plotted in “Amplitude of Frequency Components, using DESA-2 and a filter bank.” It is clear from this graph that the algorithm correctly identified strong frequency components at 440Hz, 1000Hz, and 300Hz. The amplitudes for each component are slightly below the correct value of 10, due to the leakage effect.

The application of this algorithm in its present form to real signals is made problematic because of the inability of the DESA-2 function to handle silence correctly (a silent signal results in a division by 0). This problem could be easily overcome in an application in which silence is a possible input. A significant improvement could be made to this algorithm by using a moving window and applying DESA-2 over the output within this window, as the current implementation assumes that frequency and amplitude are unchanging over the course of the entire input signal. In any case, the above results clearly demonstrate that the Teager operator and DESA-2 can be used in conjunction with a filter bank to provide an alternative to traditional spectral analysis methods.

Conclusion

Past research has shown that the Teager operator is useful in signal analysis in several different contexts, and this project further demonstrates this fact. The Teager operator can be implemented easily in Max/MSP for real time musical control, but MATLAB offers a superior environment for exploring its behavior and extending it to new applications. This project elucidates and generally supports the findings of [1], [5], and [4]. Additionally, this study shows that much can be done to use the Teager operator in traditional signal processing tasks, such as the retrieval of information regarding spectral components and their magnitudes. These results suggest that the Teager operator has a well-deserved place on the palette of signal processing tools available for practical use in a variety of contexts.

References

[1] J. F. Kaiser, “On a simple algorithm to calculate the ‘energy’ of a signal,” in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’90), vol. 1, Apr. 1990, pp. 381–384.

[2] ——, “Some useful properties of teager’s energy operators,” in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’93), vol. 3, Apr. 1993, pp. 149–152.

[3] R. Kumaresan, A. G. Sadasiv, C. S. Ramalingam, and J. F. Kaiser, “Instantaneous non-linear operators for tracking multicomponent signal parameters,” in Proc. IEEE Sixth SP Workshop on Statistical Signal and Array Processing ’92), Oct. 1992, pp. 1041–1046.

[4] W. Lin, C. Hamilton, and P. Chitrapu, “A generalization to the teagerkaiser energy function and application to resolving two closely-spaced tones,” in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’95), vol. 3, May 1995, pp. 1637–1640.

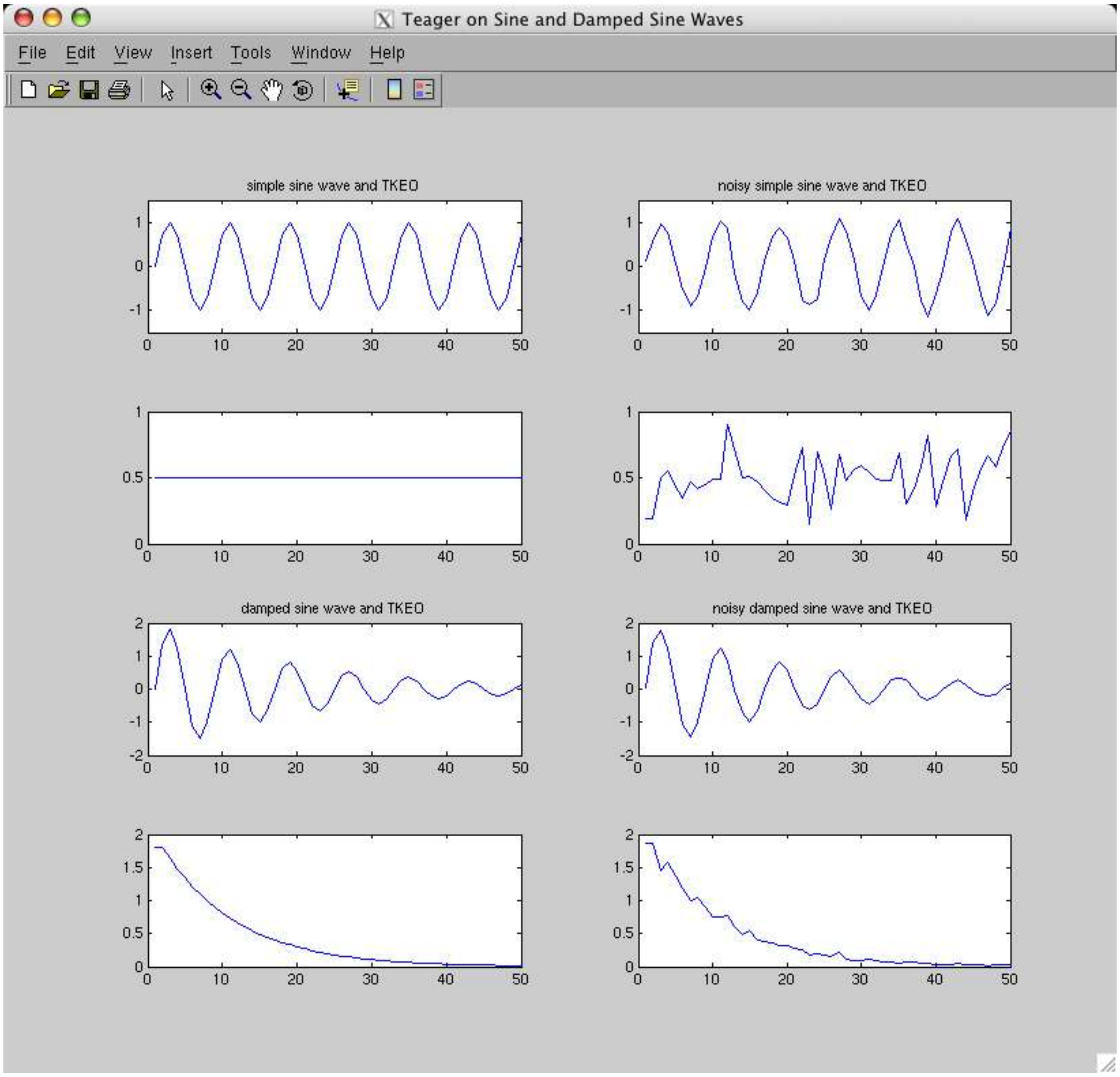
[5] P. Maragos, J. F. Kaiser, and T. F. Quatieri, “On separating amplitude from frequency modulations using energy operators,” in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’92), vol. 2, Mar. 1992, pp. 1–4.

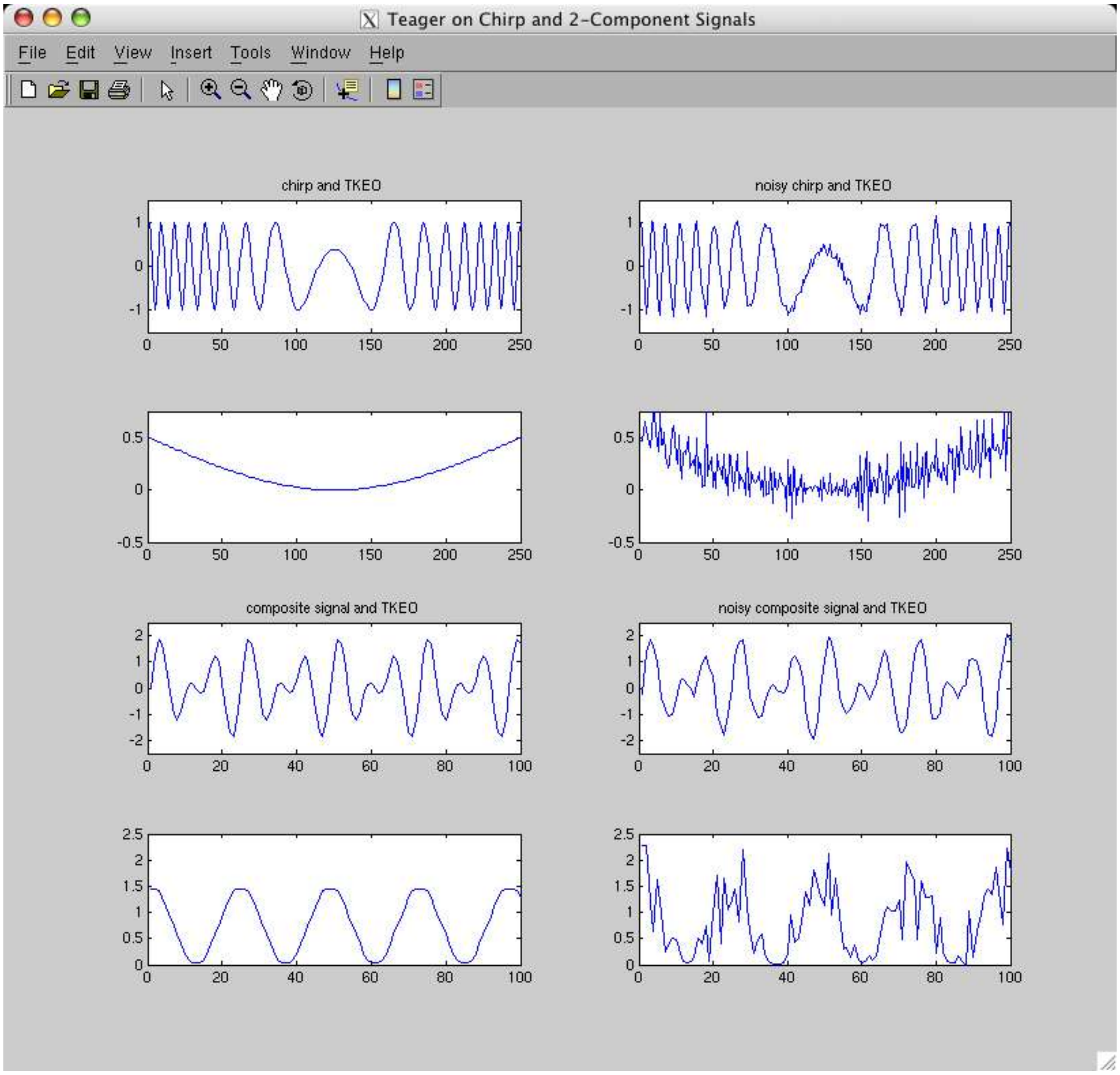
[6] S. Mukhopadhyay and G. C. Ray, “A new interpretation of nonlinear energy operator and its efficacy in spike detection,” IEEE Transactions on Biomedical Engineering, vol. 45, pp. 180–187, Feb. 1998.

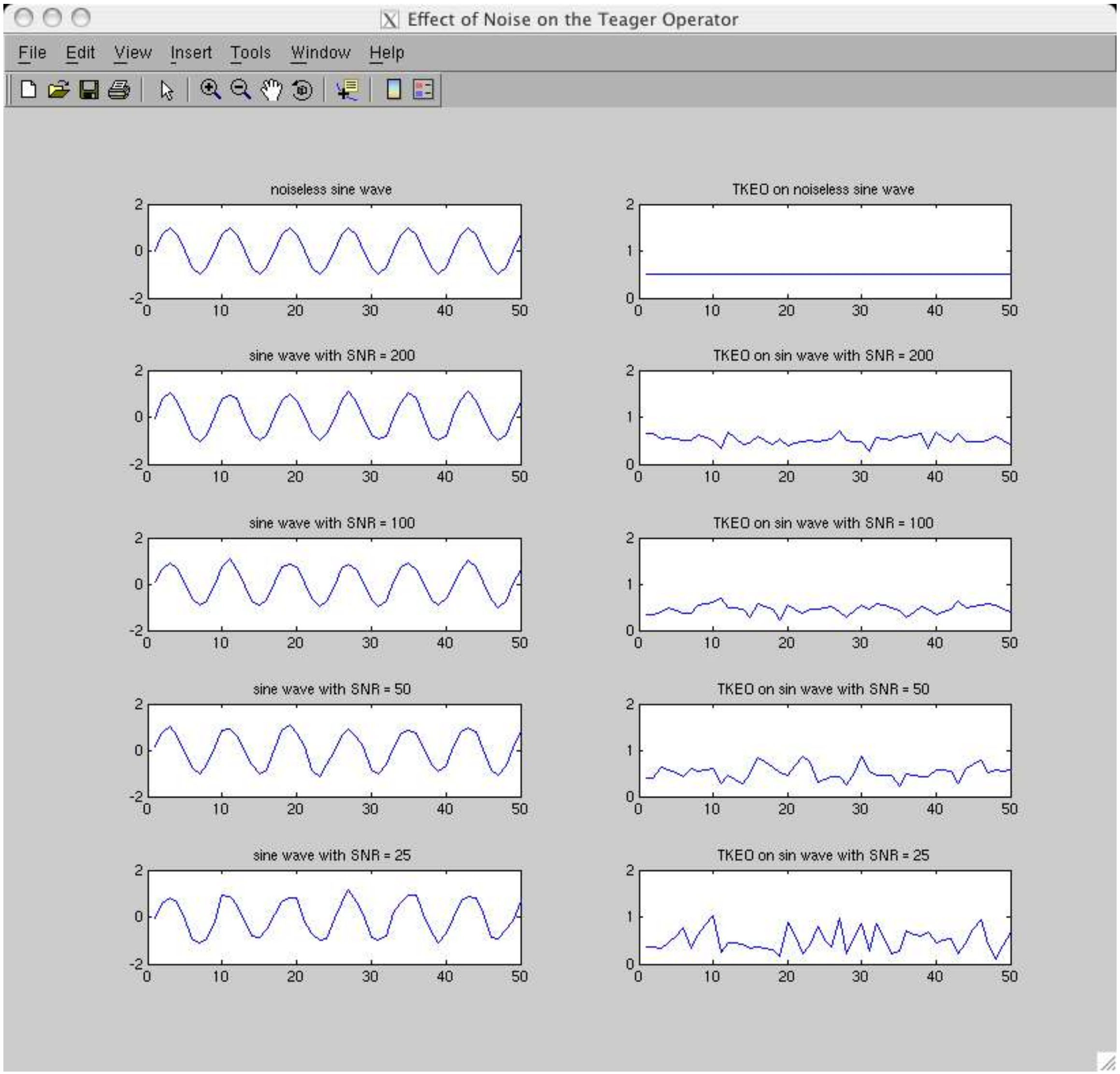
[7] G. Zhou, J. H. L. Hansen, and J. F. Kaiser, “Classification of speech under stress based on features derived from the nonlinear teager energy operator,” in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’98), vol. 1, May 1998, pp. 549–552.

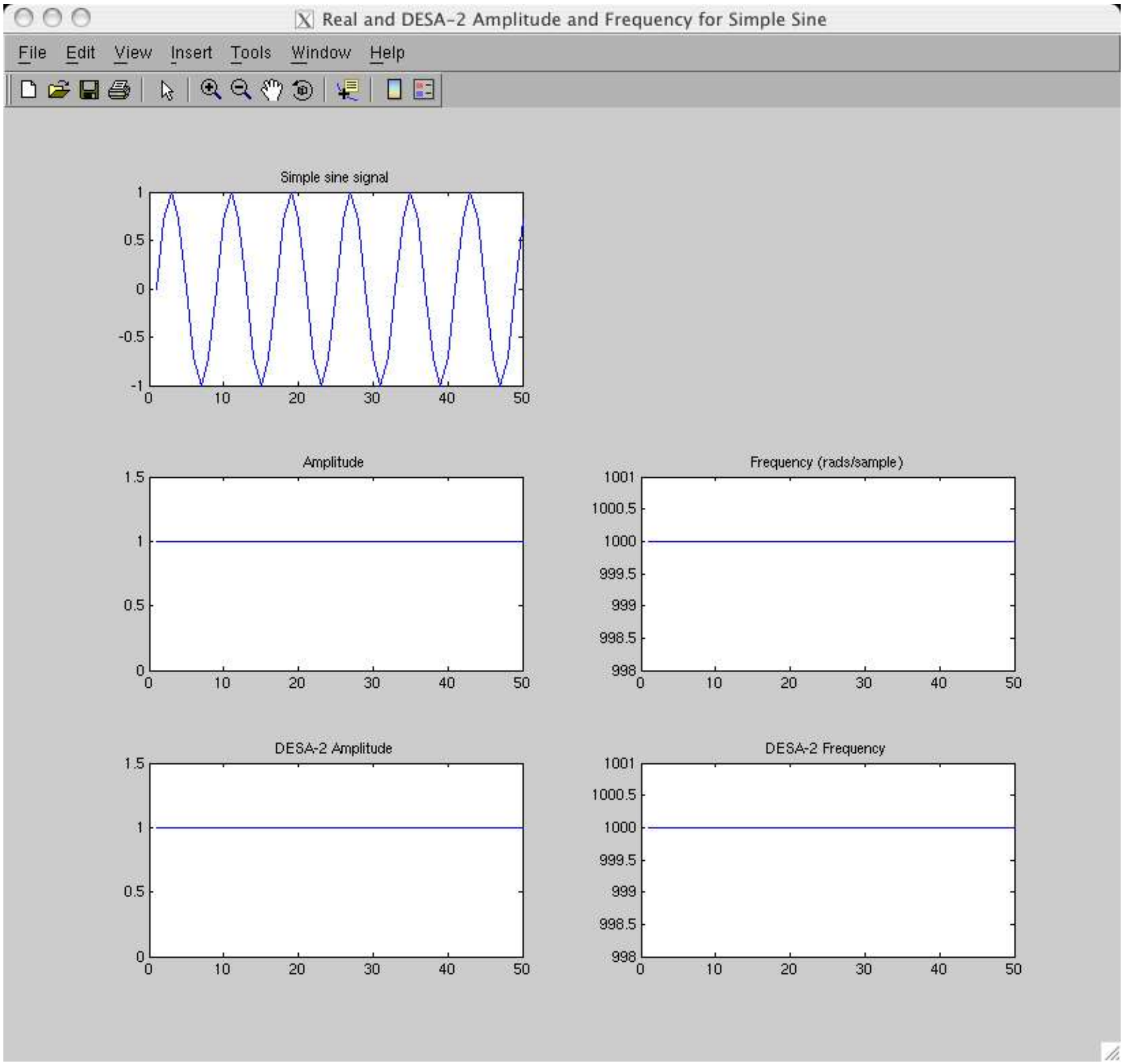
Appendix I: MATLAB Graphics

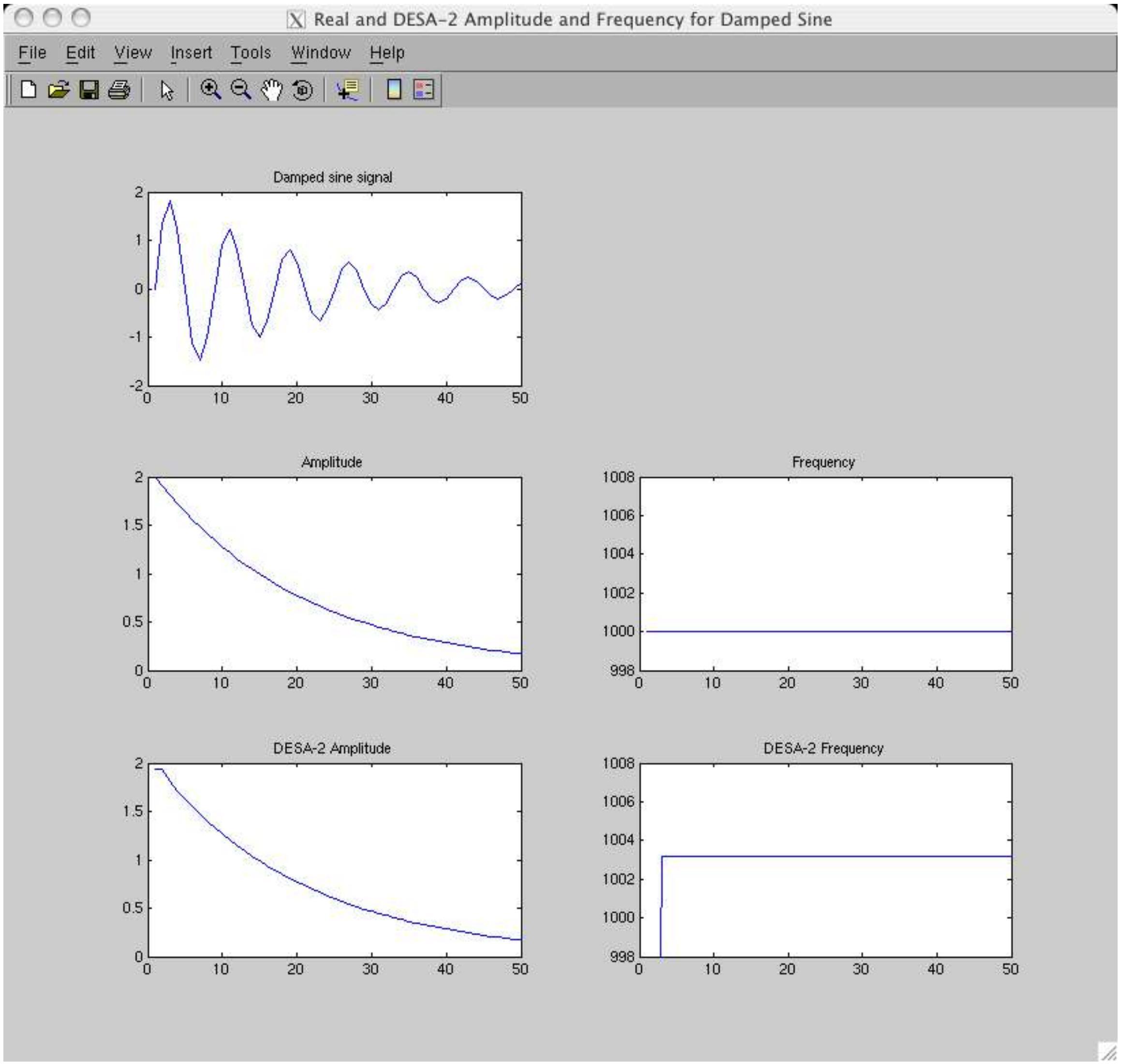
The following graphics are identical to those output by the MATLAB scripts for parts 1 through 4. They are provided here for the reader who does not wish to run the scripts directly, as they are necessary supplements to this paper.

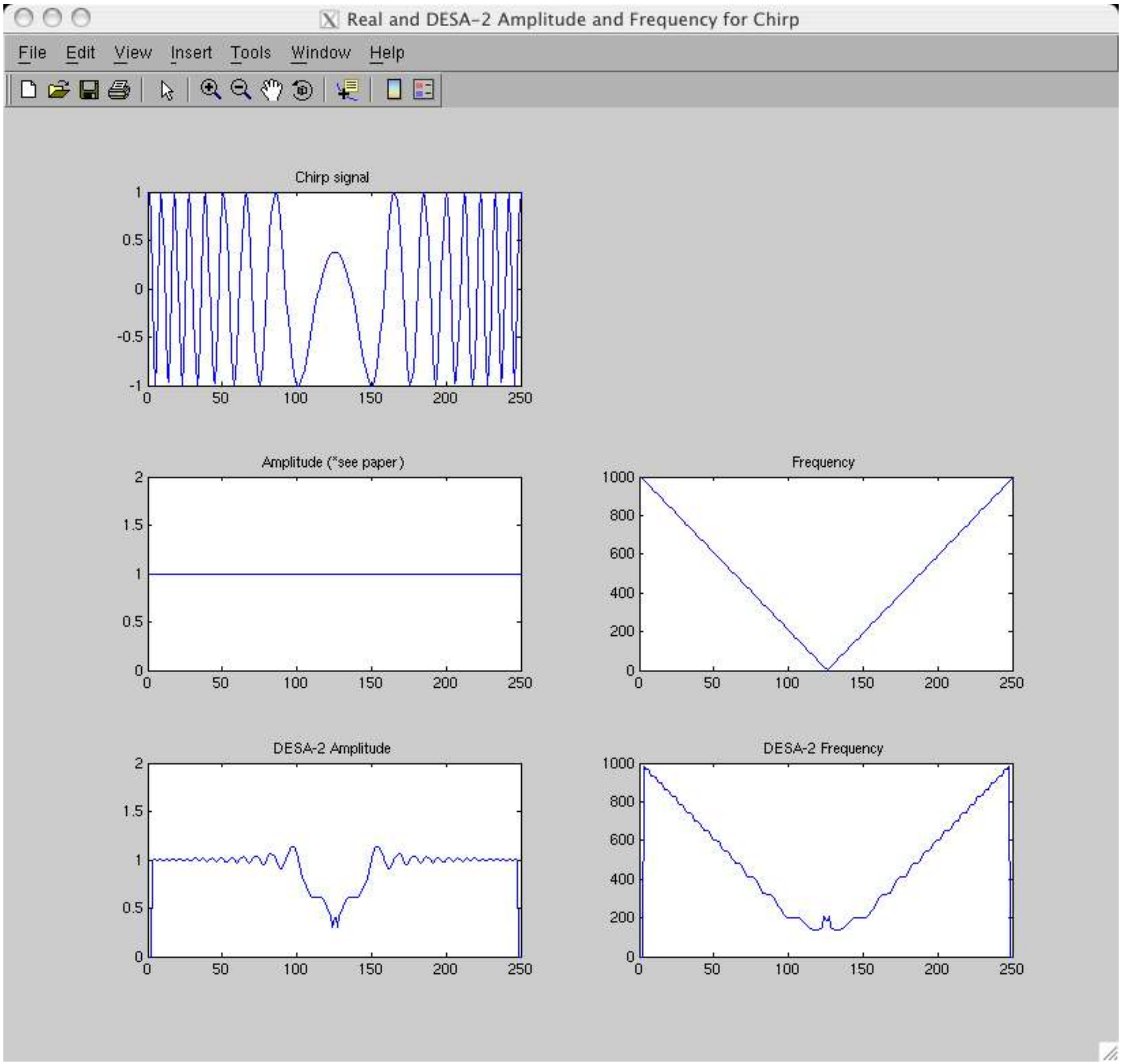

Teager on Sine and Damped Sine Waves
File
Edit
View
Insert
Tools
Window
Help
simple sine wave and TKEO
noisy simple sine wave and TKEO
damped sine wave and TKEO
noisy damped sine wave and TKEO
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
0
10
20
30
40
50

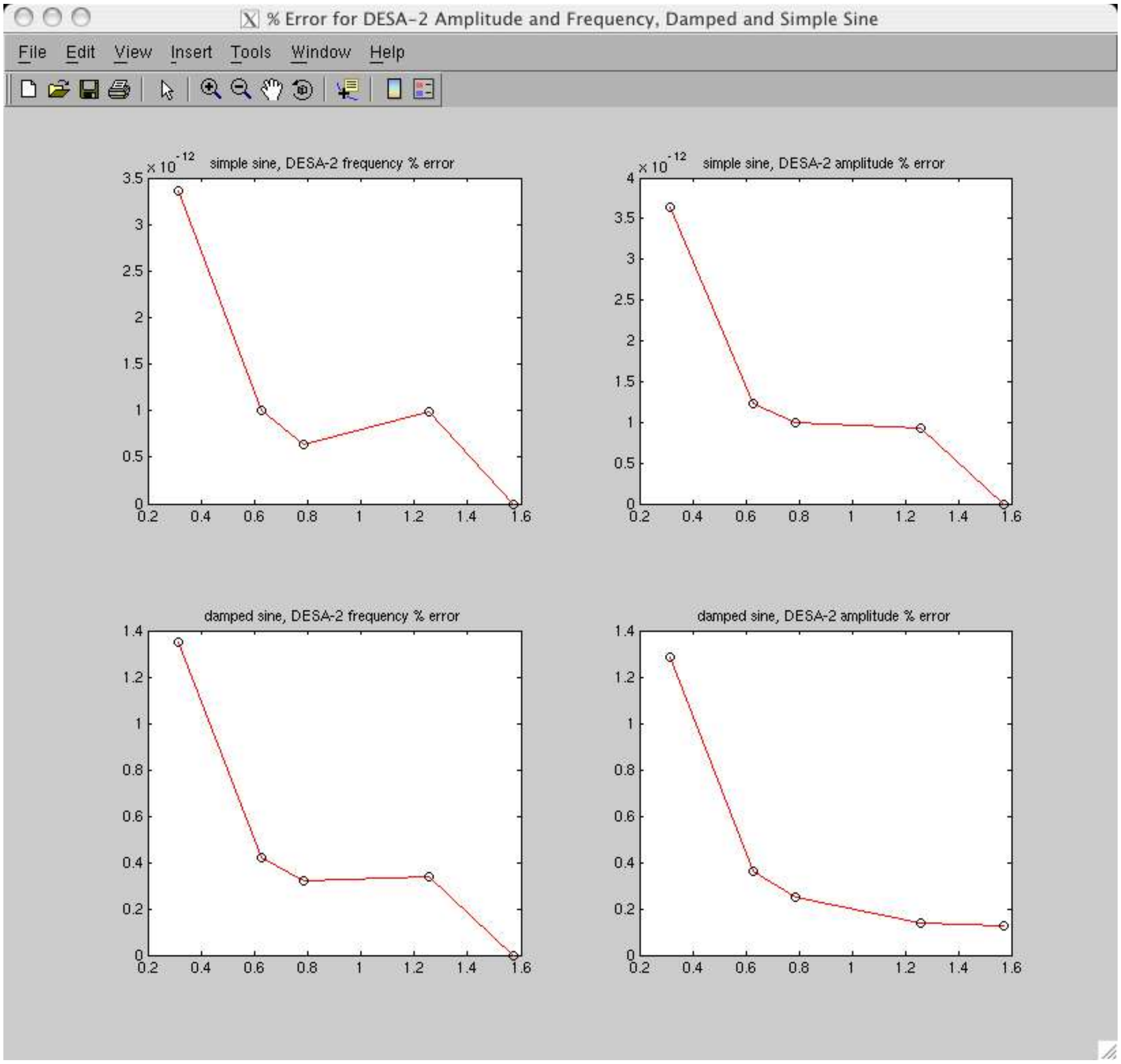

Teager on Chirp and 2-Component Signals
File Edit View Insert Tools Window Help
chirp and TKEO
noisy chirp and TKEO
1
0
-1
0 50 100 150 200 250
0.5
0
-0.5
0 50 100 150 200 250
composite signal and TKEO
noisy composite signal and TKEO
2
1
0
-1
-2
0 20 40 60 80 100
2.5
2
1.5
1
0.5
0
0 20 40 60 80 100

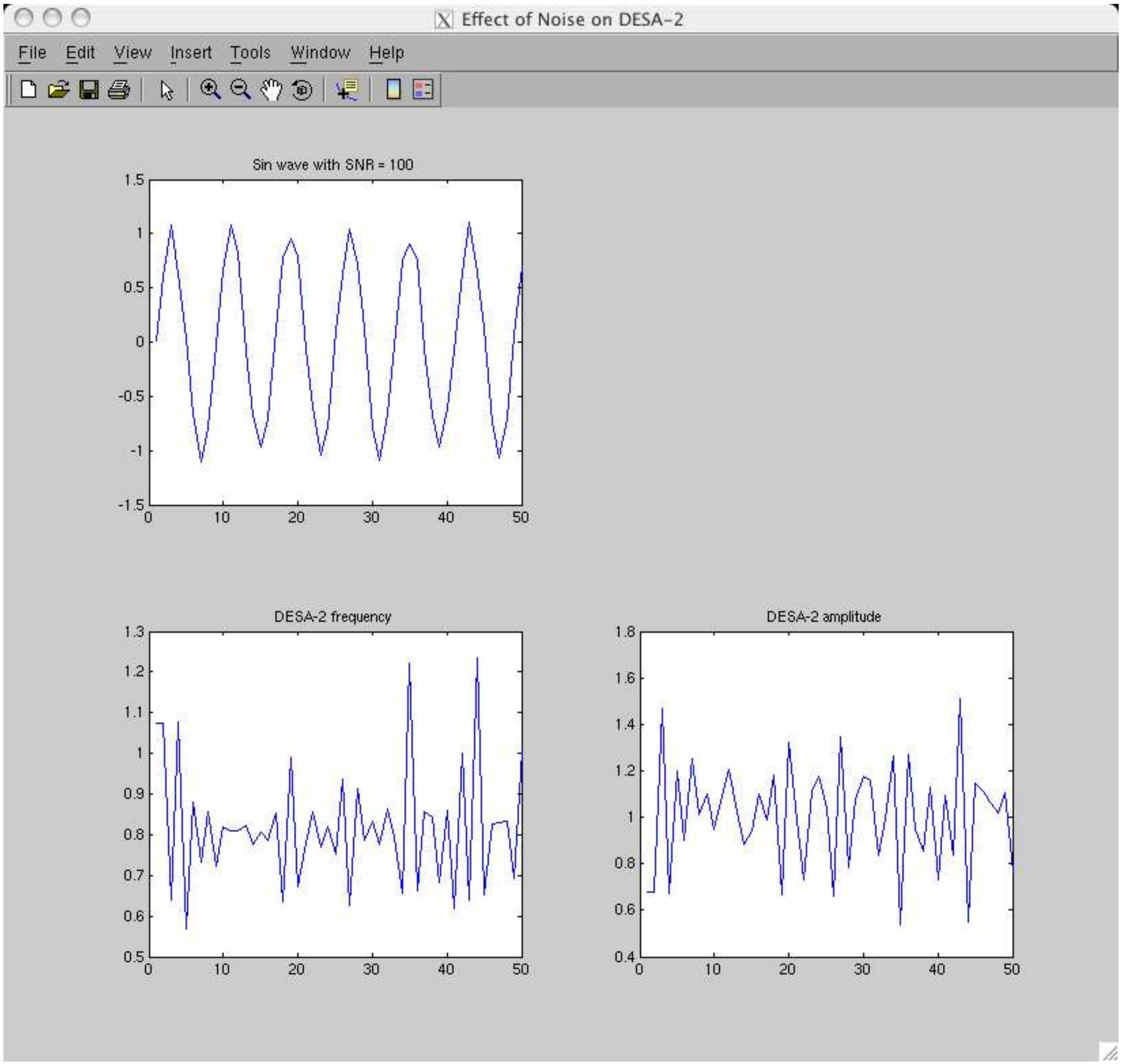

Effect of Noise on the Teager Operator
File
Edit
View
Insert
Tools
Window
Help
noiseless sine wave
TKEO on noiseless sine wave
sine wave with SNR = 200
TKEO on sin wave with SNR = 200
sine wave with SNR = 100
TKEO on sin wave with SNR = 100
sine wave with SNR = 50
TKEO on sin wave with SNR = 50
sine wave with SNR = 25
TKEO on sin wave with SNR = 25
2
0
-2
1
0
10
20
30
40
50

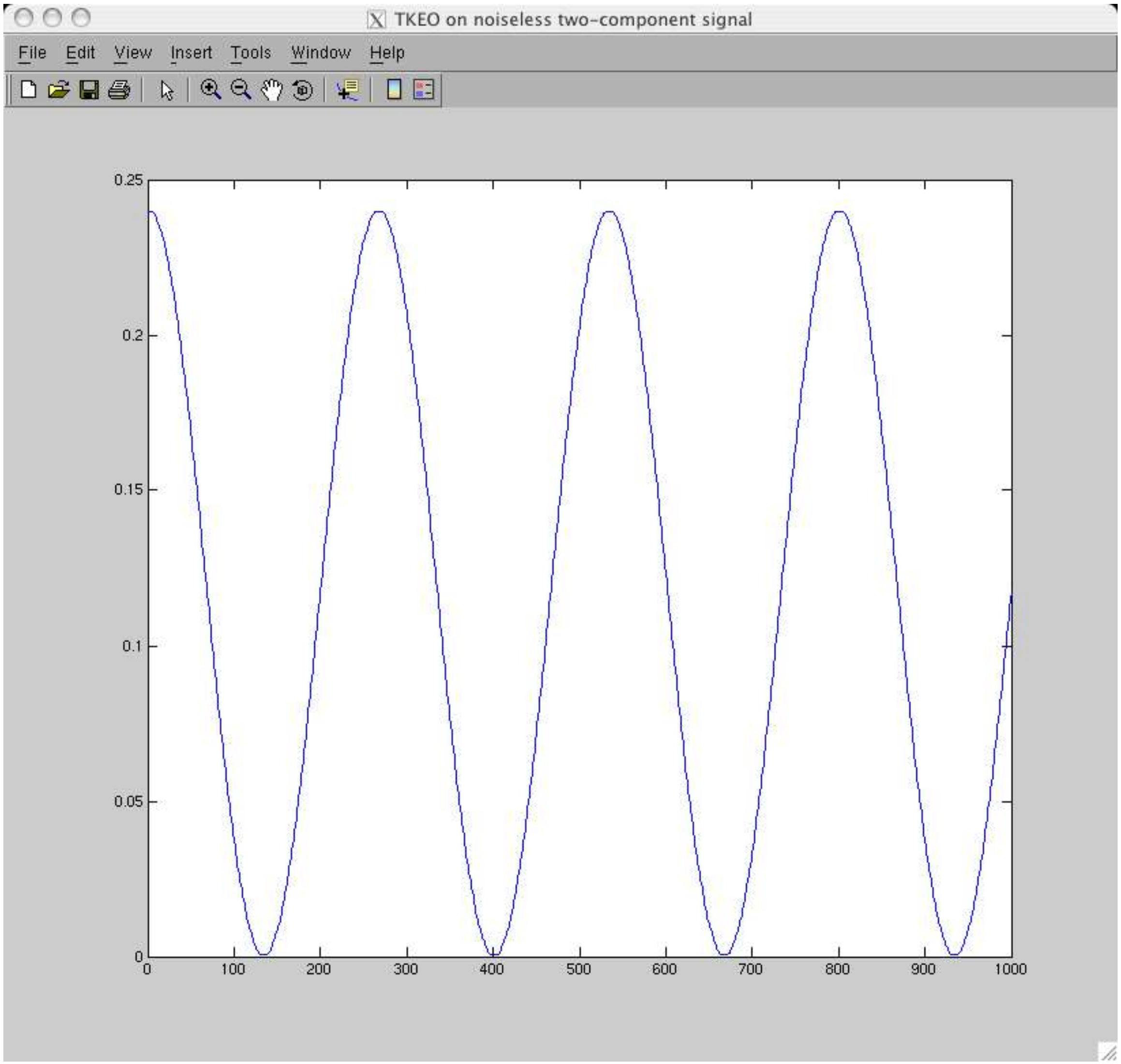

Real and DESA-2 Amplitude and Frequency for Simple Sine
File
Edit
View
Insert
Tools
Window
Help
Simple sine signal
1
0.5
0
-0.5
-1
0
10
20
30
40
50
Amplitude
1.5
1
0.5
0
Frequency (rads/sample)
1001
1000.5
1000
999.5
999
998.5
998
DESA-2 Amplitude
DESA-2 Frequency

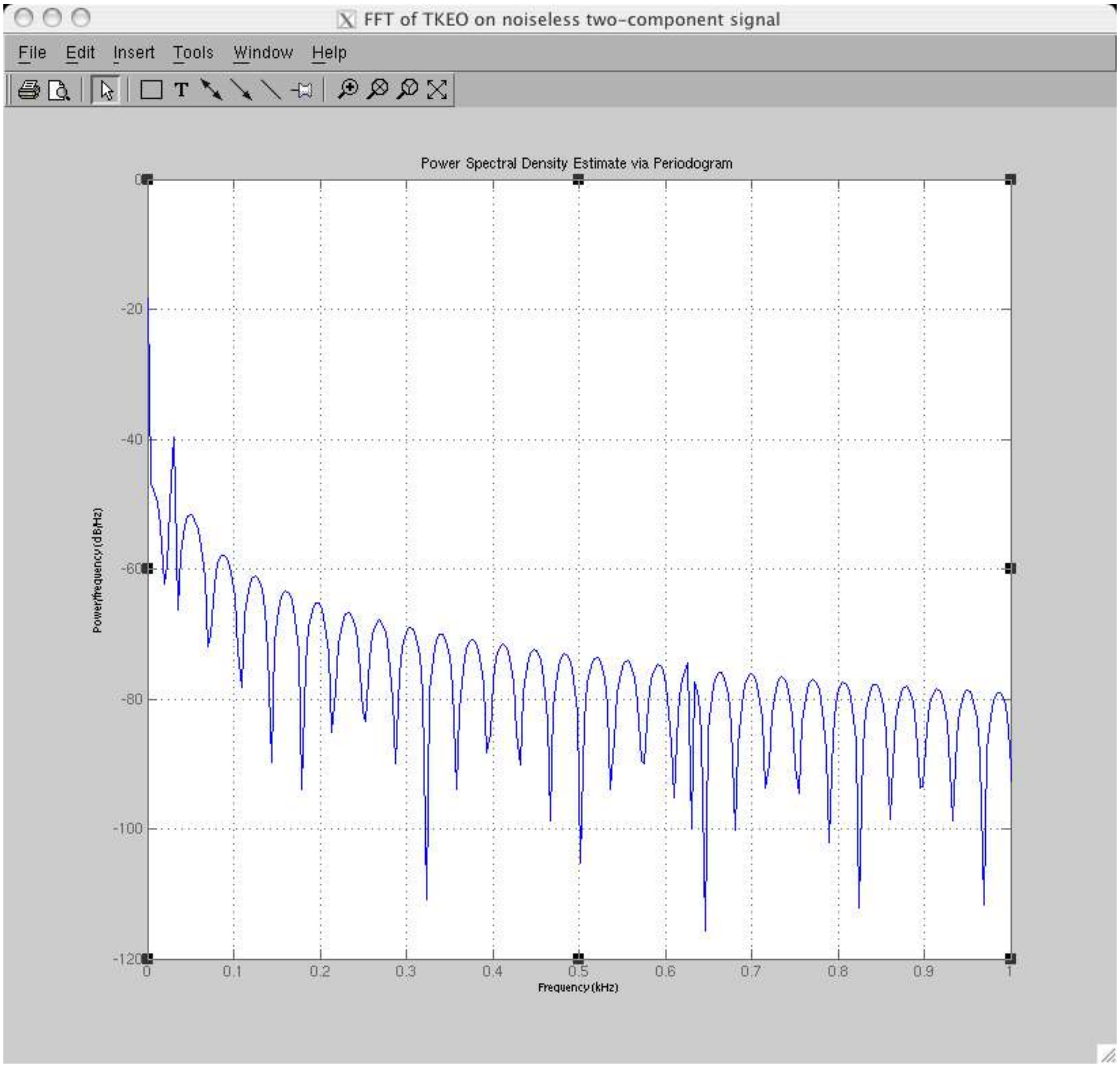

Real and DESA-2 Amplitude and Frequency for Damped Sine
File
Edit
View
Insert
Tools
Window
Help
Damped sine signal
2
1
0
-1
-2
0
10
20
30
40
50
Amplitude
2
1.5
1
0.5
0
0
10
20
30
40
50
Frequency
1008
1006
1004
1002
1000
998
0
10
20
30
40
50
DESA-2 Amplitude
2
1.5
1
0.5
0
0
10
20
30
40
50
DESA-2 Frequency
1008
1006
1004
1002
1000
998
0
10
20
30
40
50

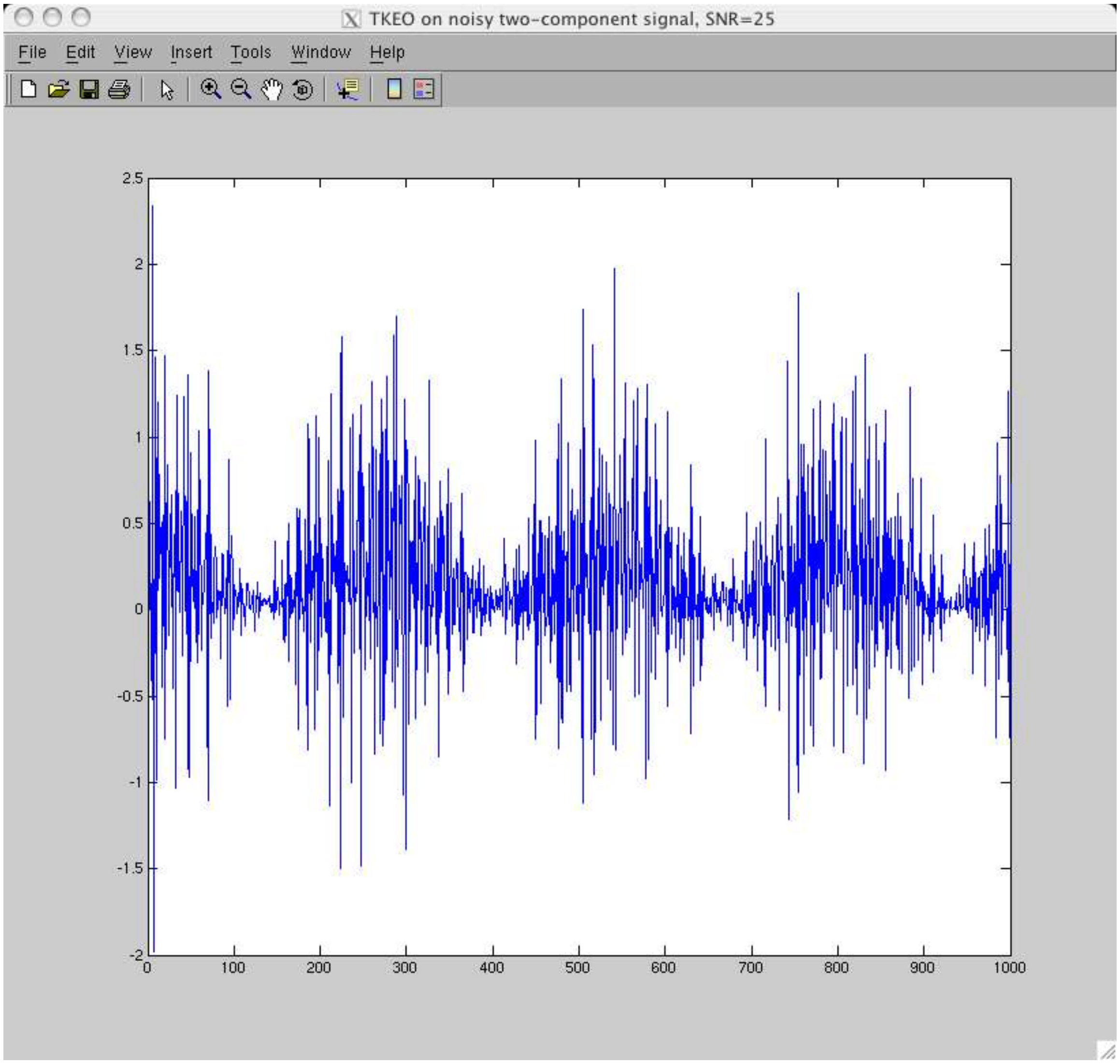

Real and DESA-2 Amplitude and Frequency for Chirp
File Edit View Insert Tools Window Help
Chirp signal
1
0.5
0
-0.5
-1
0 50 100 150 200 250
Amplitude (*see paper)
2
1.5
1
0.5
0
0 50 100 150 200 250
Frequency
1000
800
600
400
200
0
0 50 100 150 200 250
DESA-2 Amplitude
2
1.5
1
0.5
0
0 50 100 150 200 250
DESA-2 Frequency
1000
800
600
400
200
0
0 50 100 150 200 250

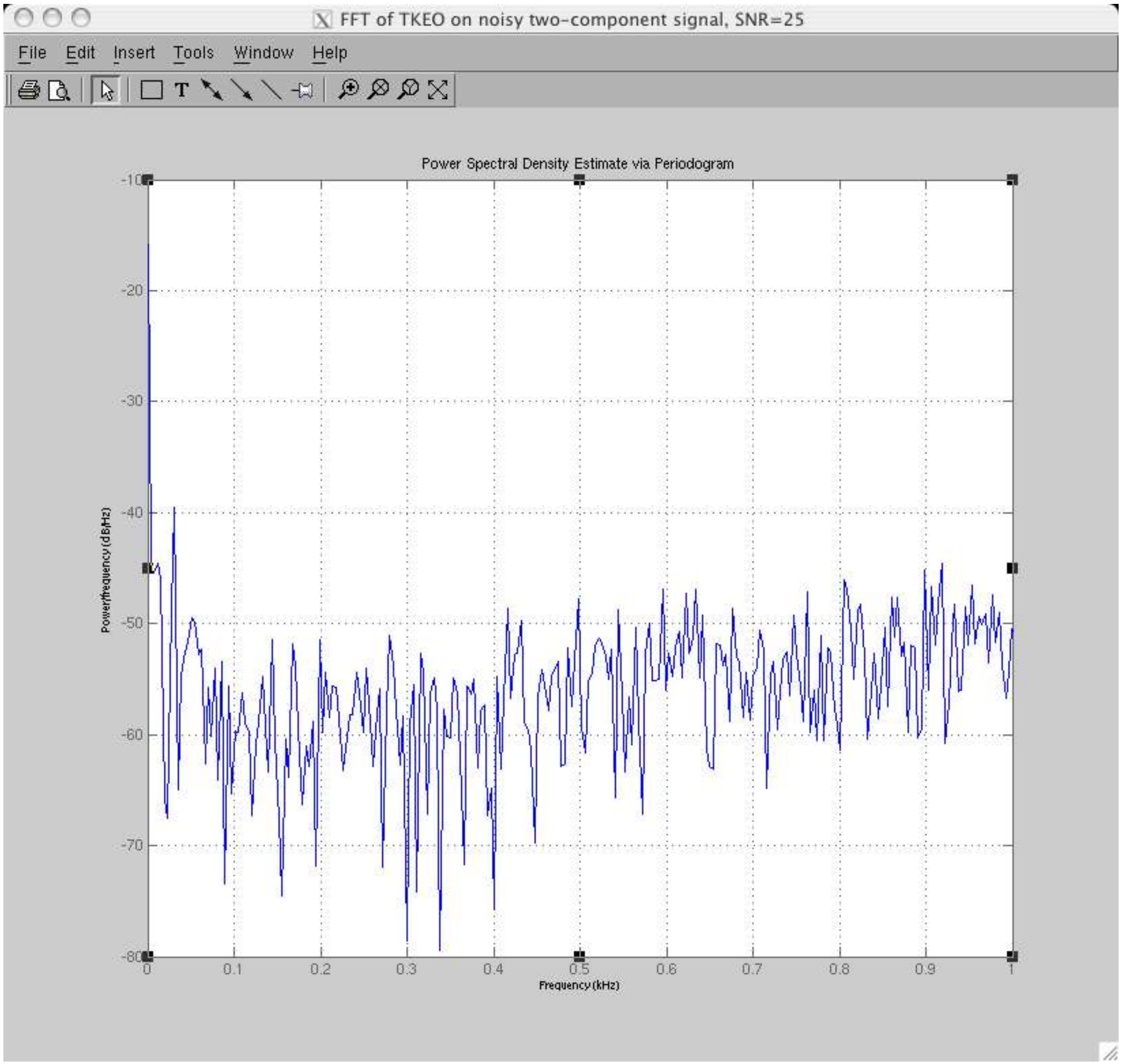

% Error for DESA-2 Amplitude and Frequency, Damped and Simple Sine
File
Edit
View
Insert
Tools
Window
Help
simple sine, DESA-2 frequency % error
× 10^-12
3.5
3
2.5
2
1.5
1
0.5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
simple sine, DESA-2 amplitude % error
× 10^-12
4
3.5
3
2.5
2
1.5
1
0.5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
damped sine, DESA-2 frequency % error
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
damped sine, DESA-2 amplitude % error
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6

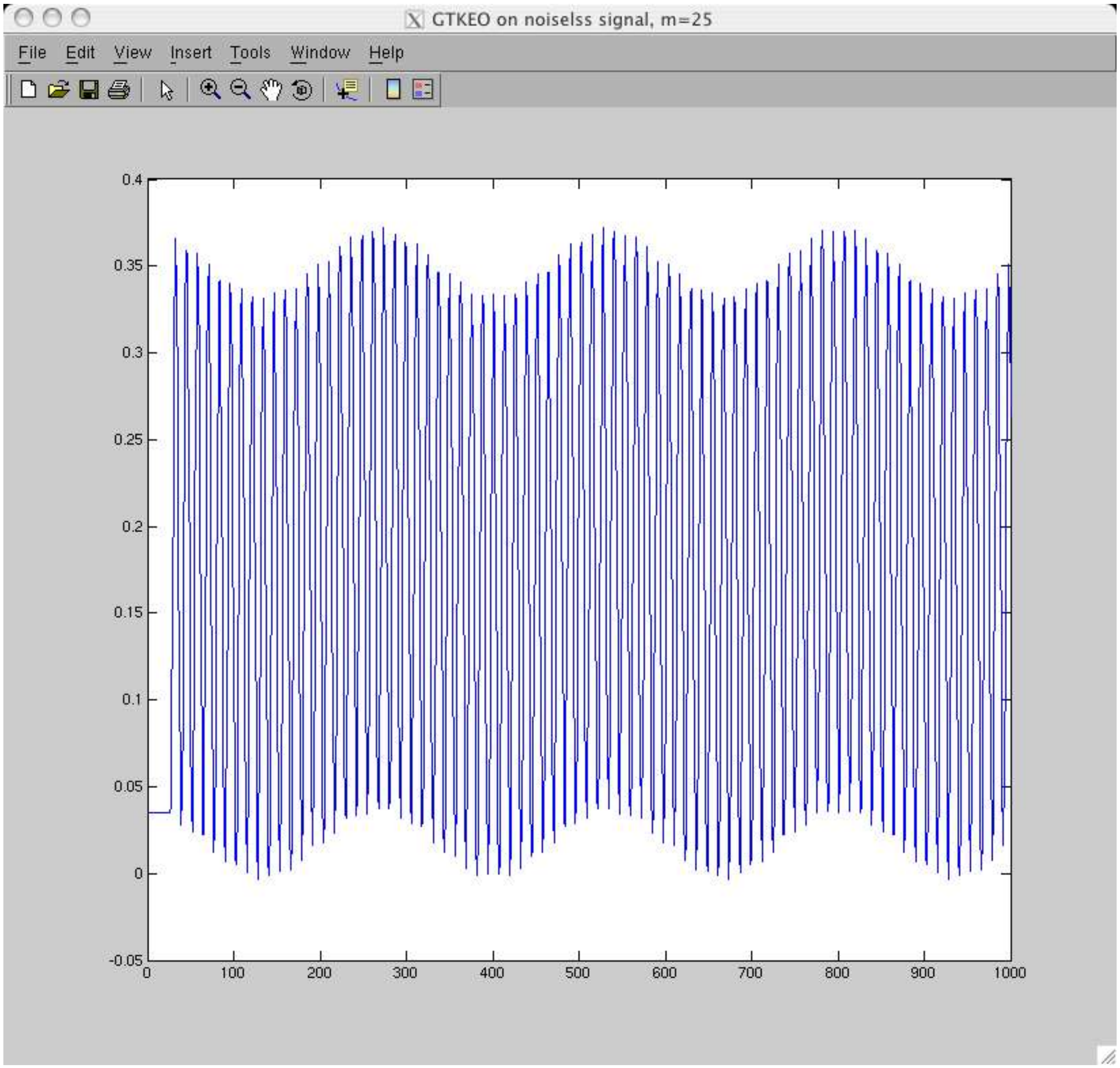

Effect of Noise on DESA-2
File
Edit
View
Insert
Tools
Window
Help
Sin wave with SNR = 100
1.5
1
0.5
0
-0.5
-1
-1.5
0
10
20
30
40
50
DESA-2 frequency
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0
10
20
30
40
50
DESA-2 amplitude
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0
10
20
30
40
50

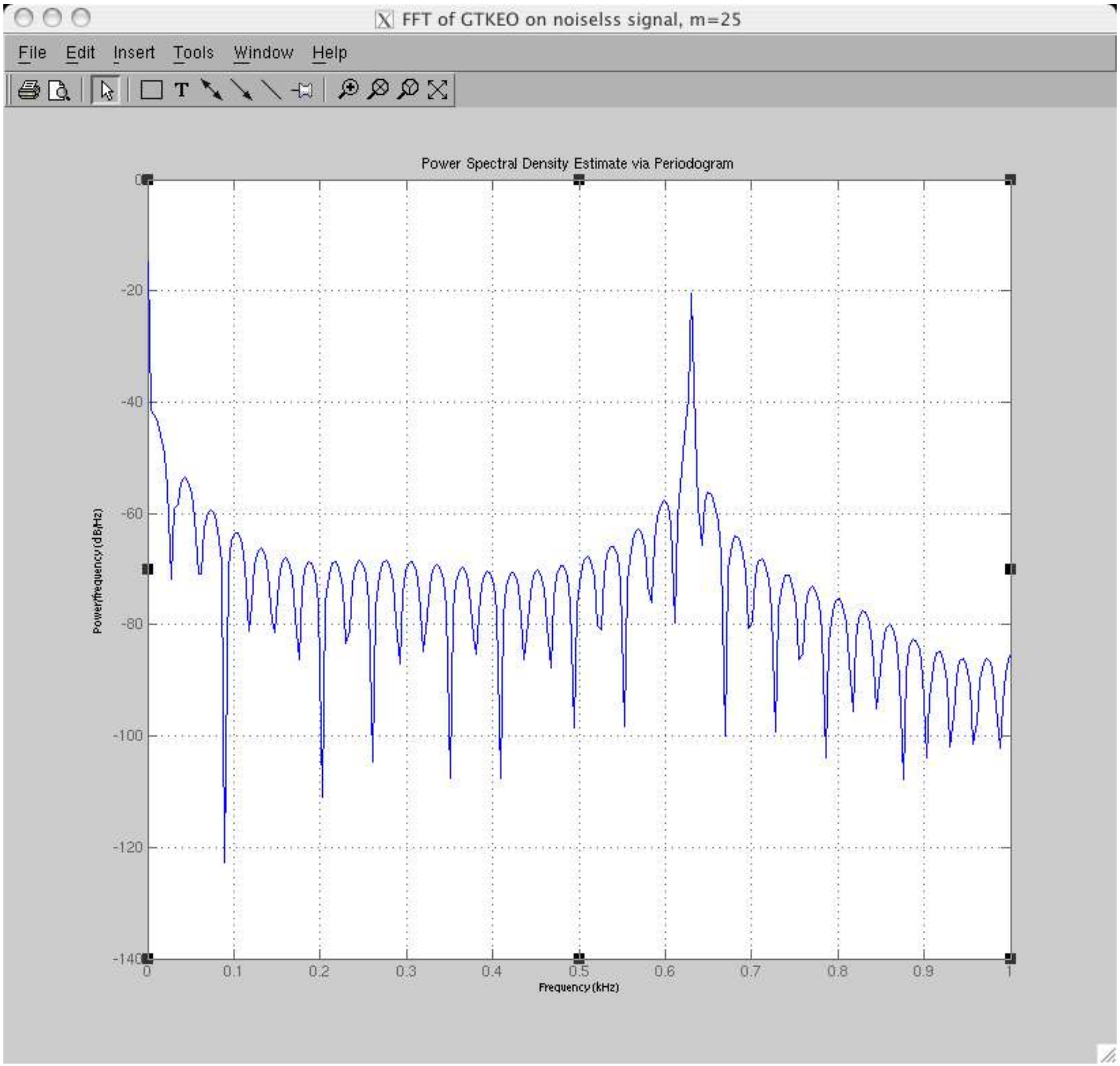

TKEO on noiseless two-component signal
File
Edit
View
Insert
Tools
Window
Help
0.25
0.2
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
900
1000

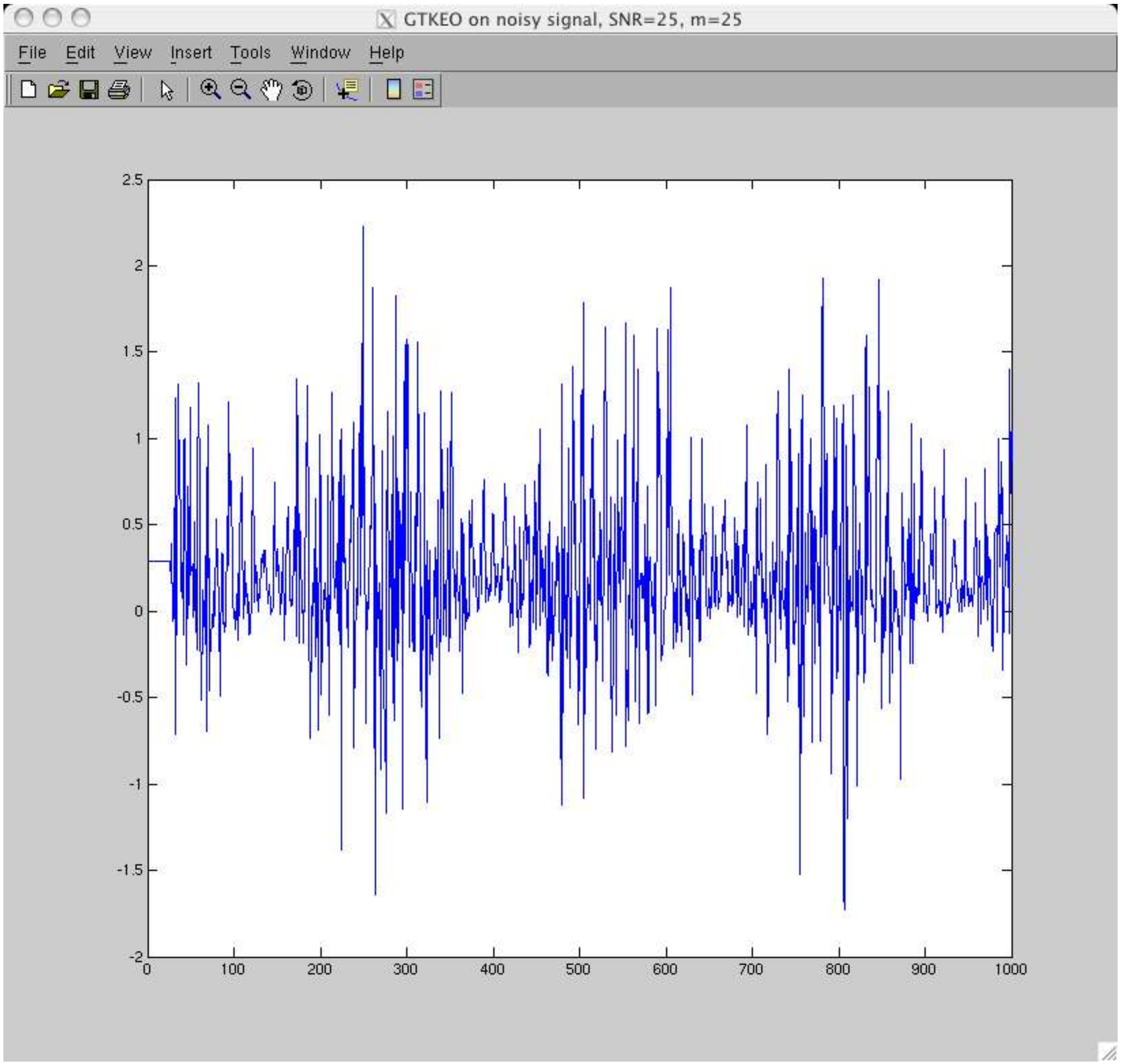

FFT of TKEO on noiseless two-component signal
File
Edit
Insert
Tools
Window
Help
Power Spectral Density Estimate via Periodogram
Power/frequency (dB/Hz)
0
-20
-40
-60
-80
-100
-120
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Frequency (kHz)

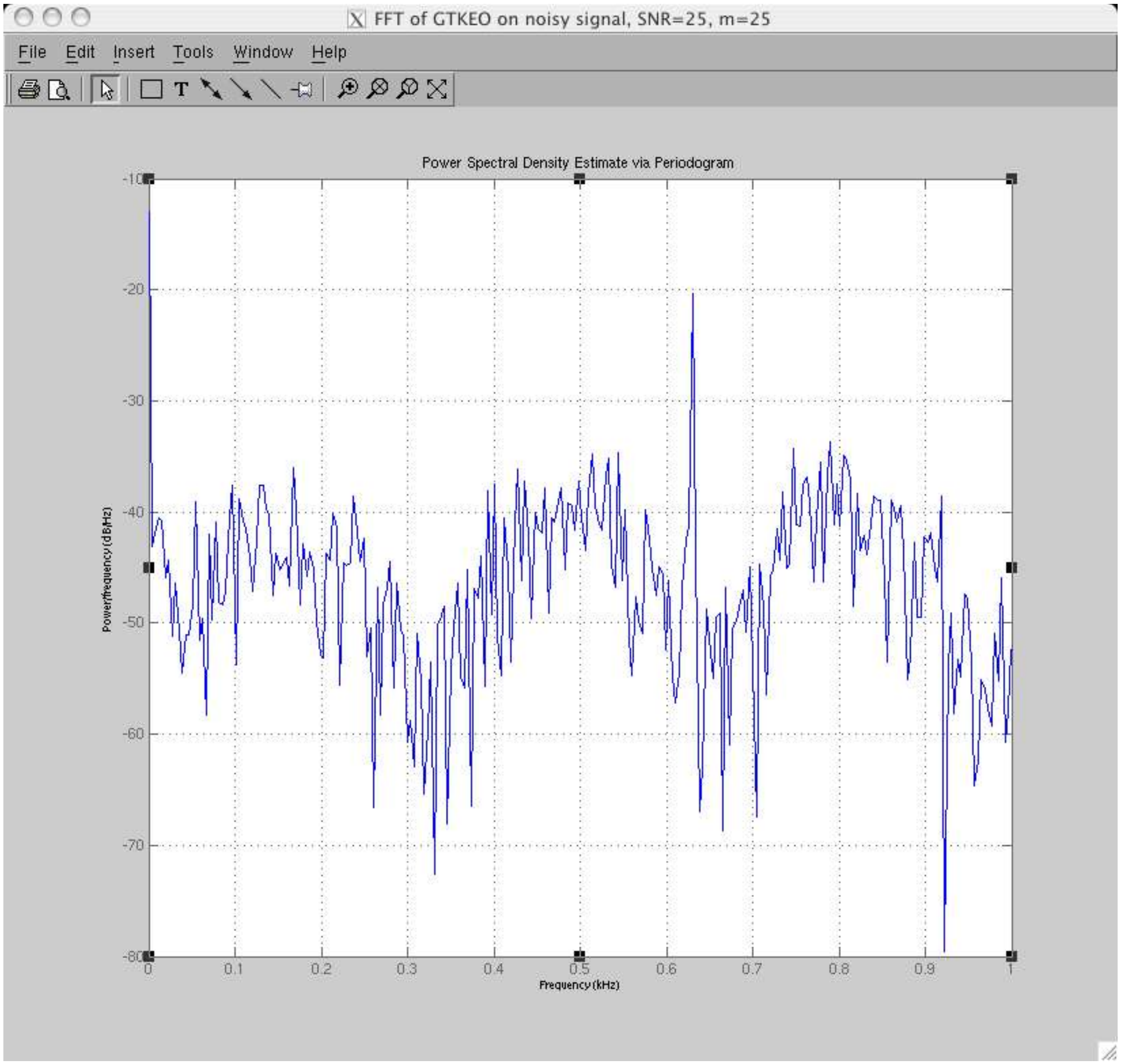

TKEO on noisy two-component signal, SNR=25
File
Edit
View
Insert
Tools
Window
Help
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
100
200
300
400
500
600
700
800
900
1000

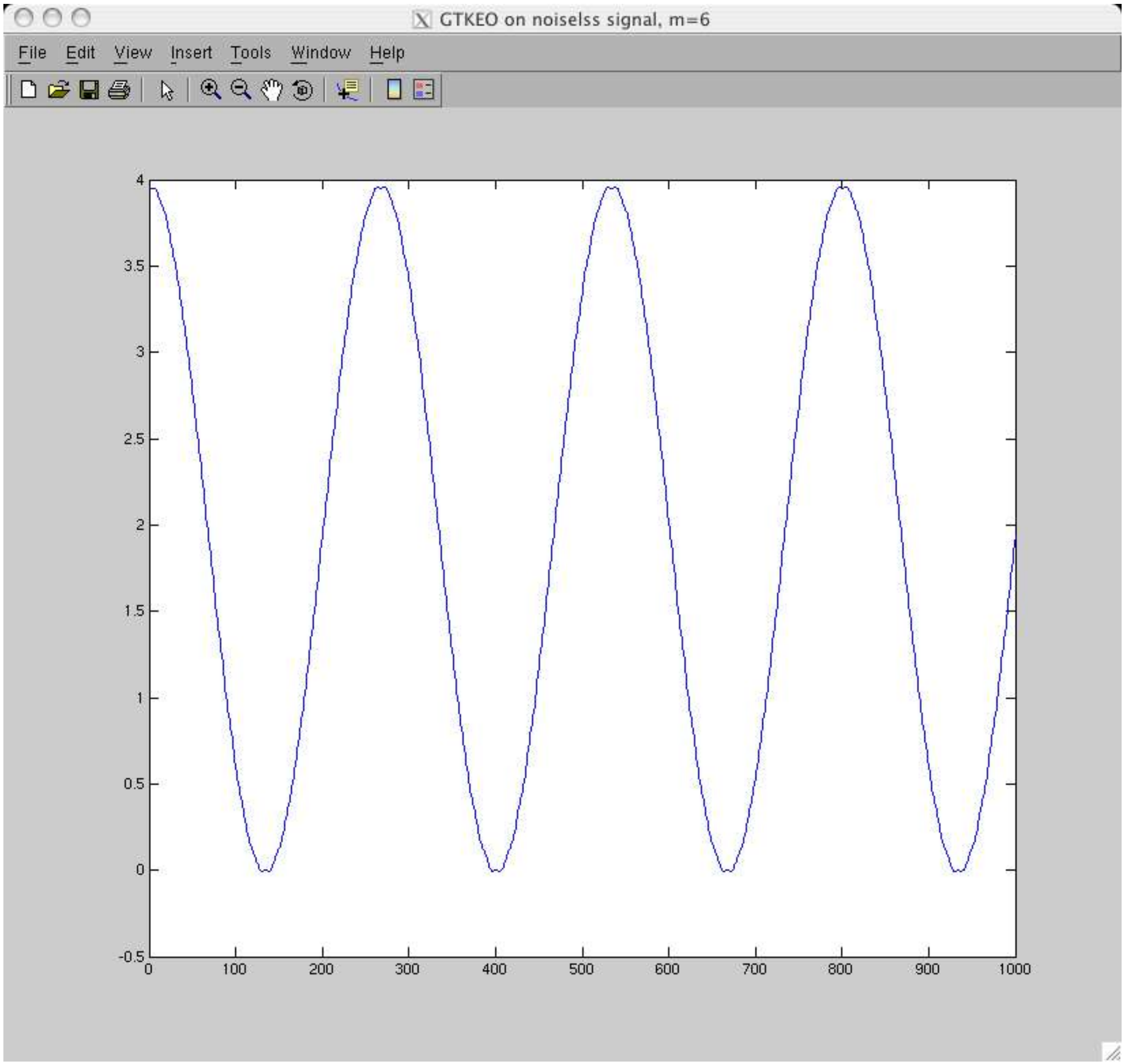

FFT of TKEO on noisy two-component signal, SNR=25
File
Edit
Insert
Tools
Window
Help
Power Spectral Density Estimate via Periodogram
-10
-20
-30
-40
-50
-60
-70
-80
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Frequency (kHz)
Power/frequency (dB/Hz)

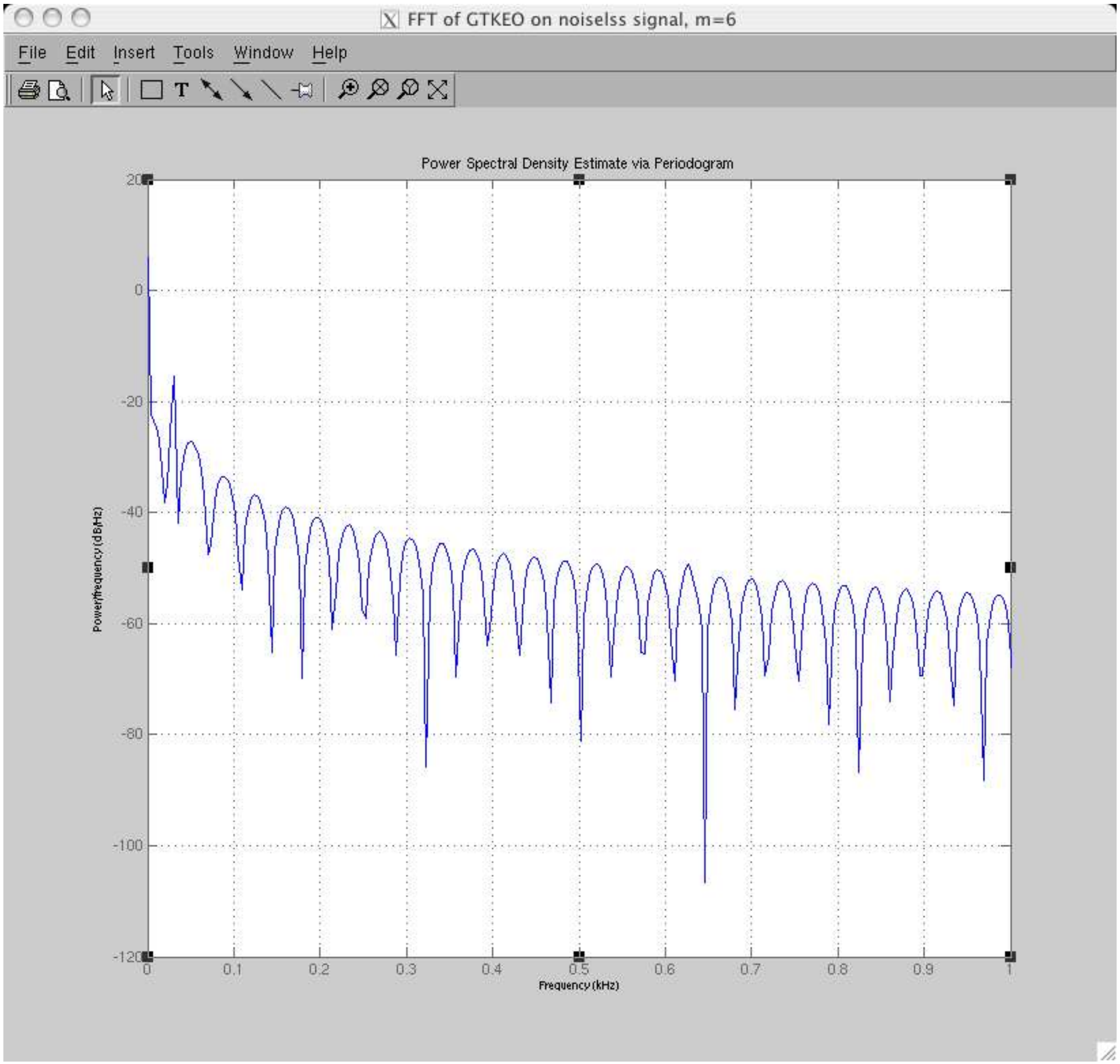

GTKEO on noiselss signal, m=25
File
Edit
View
Insert
Tools
Window
Help
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
0
100
200
300
400
500
600
700
800
900
1000

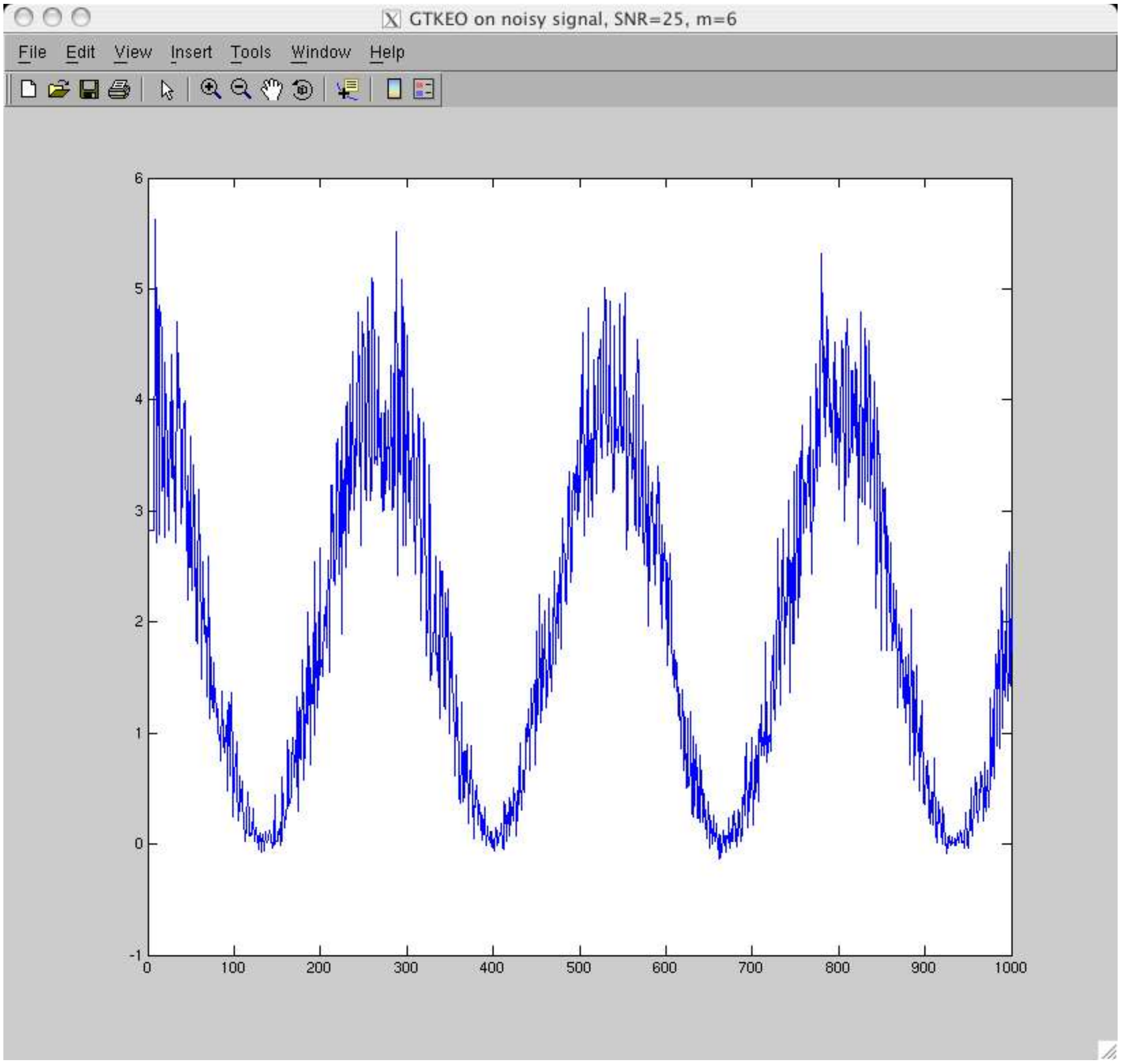

FFT of GTKEO on noiselss signal, m=25
File
Edit
Insert
Tools
Window
Help
Power Spectral Density Estimate via Periodogram
Power/frequency (dB/Hz)
0
-20
-40
-60
-80
-100
-120
-140
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Frequency (kHz)

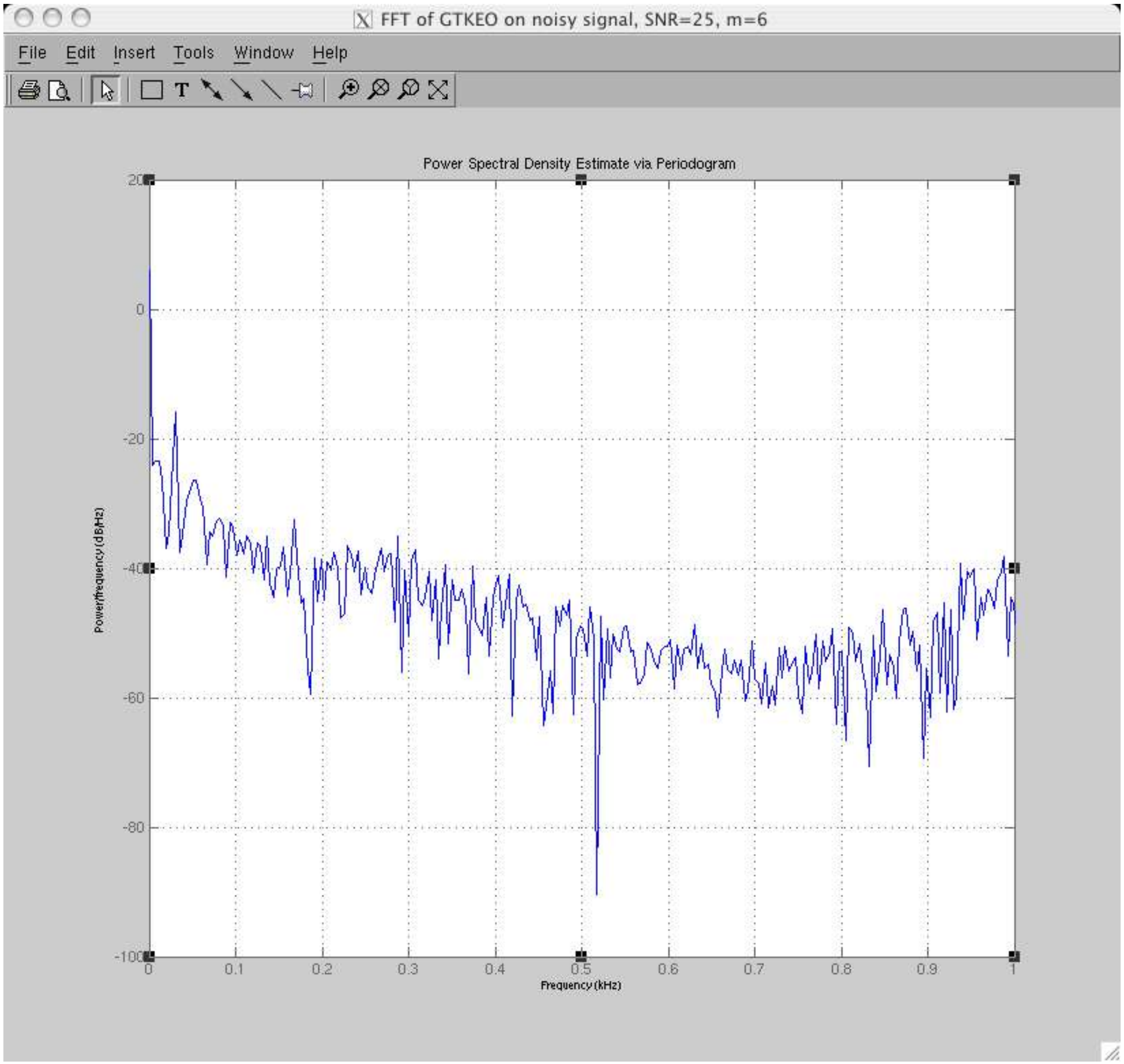

GTKEO on noisy signal, SNR=25, m=25
File
Edit
View
Insert
Tools
Window
Help
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
100
200
300
400
500
600
700
800
900
1000

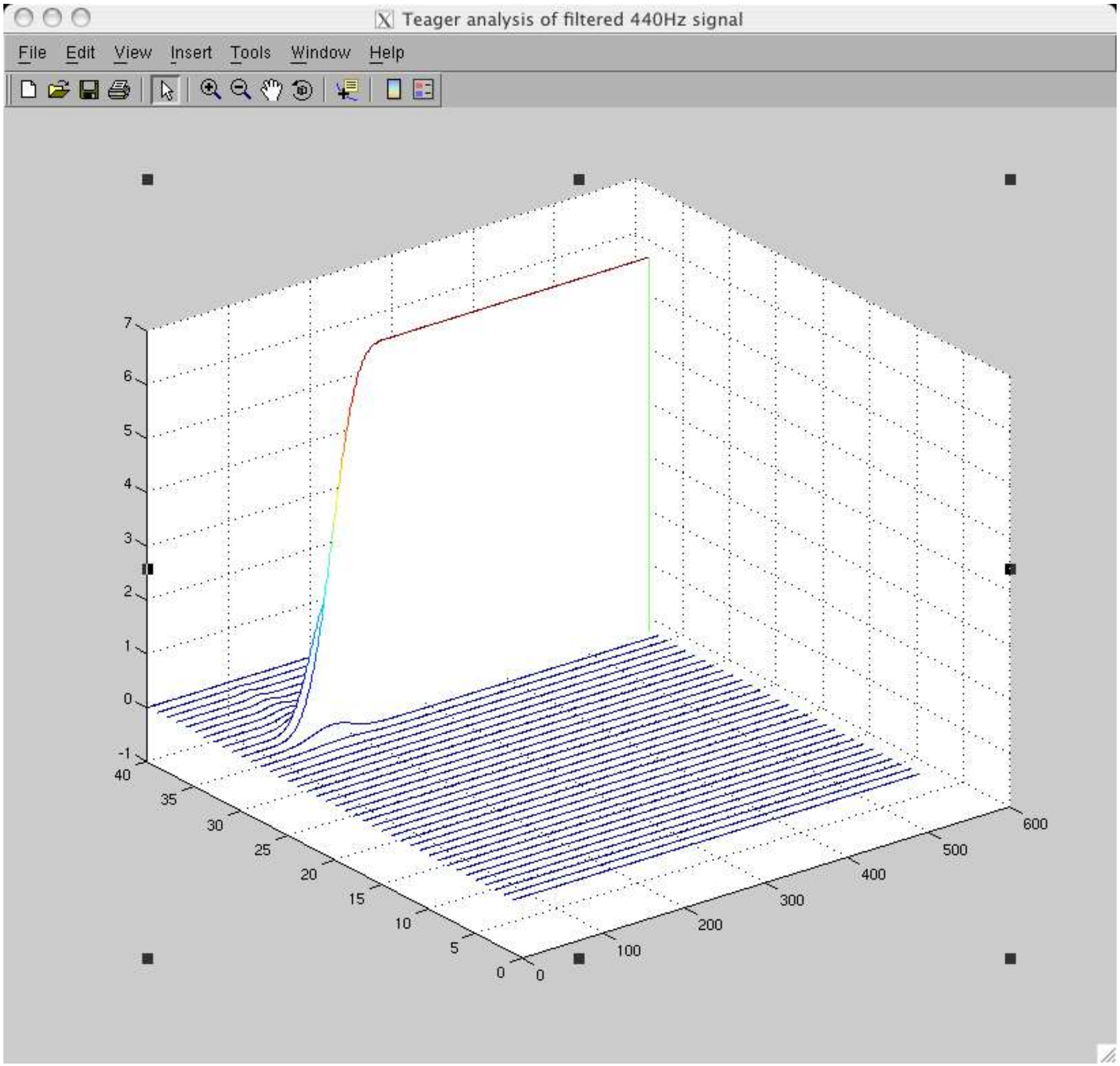

FFT of GTKEO on noisy signal, SNR=25, m=25
File
Edit
Insert
Tools
Window
Help
Power Spectral Density Estimate via Periodogram
-10
-20
-30
-40
-50
-60
-70
-80
Power/frequency (dB/Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Frequency (kHz)

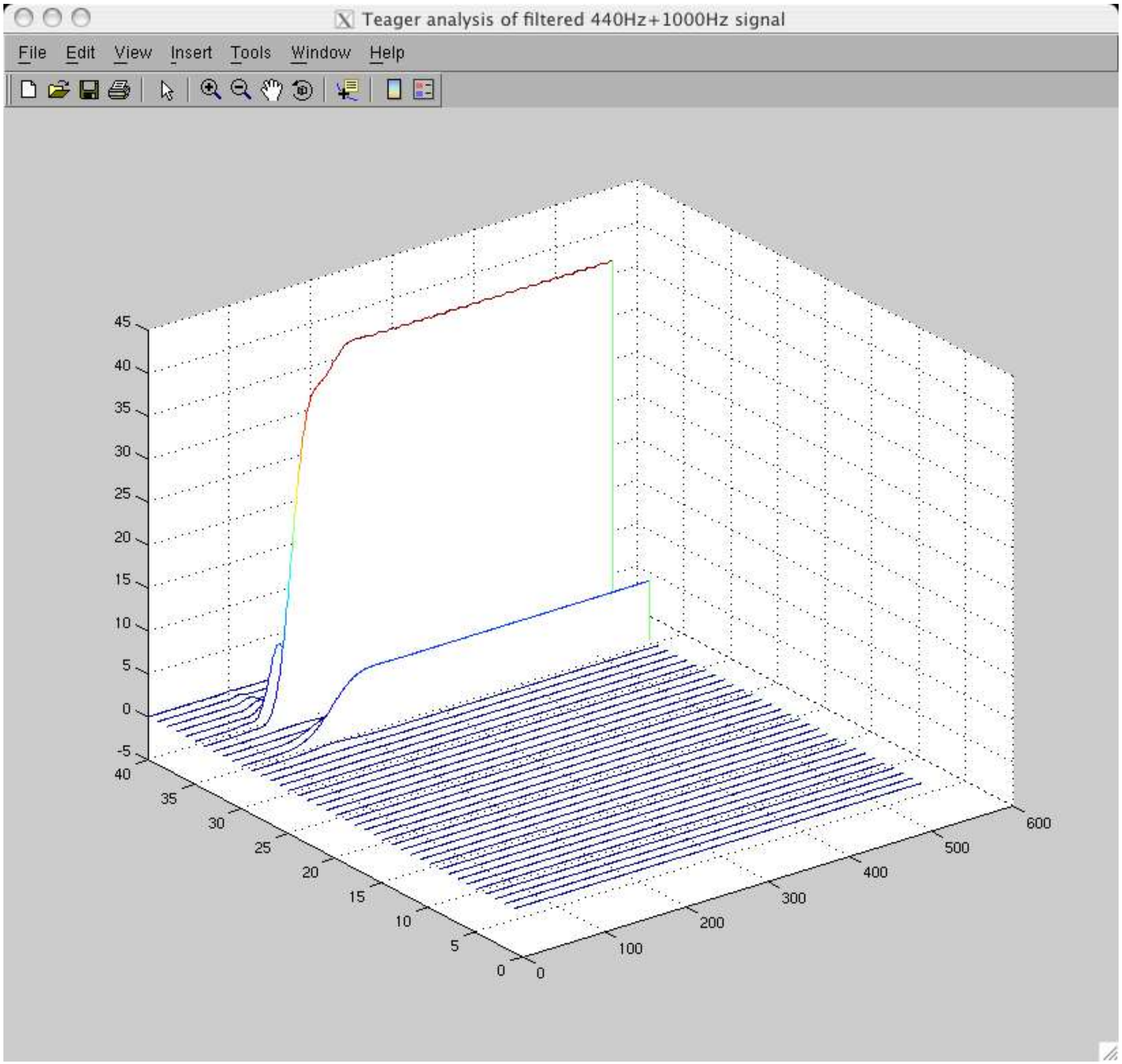

GTKEO on noiselss signal, m=6
File
Edit
View
Insert
Tools
Window
Help
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
100
200
300
400
500
600
700
800
900
1000

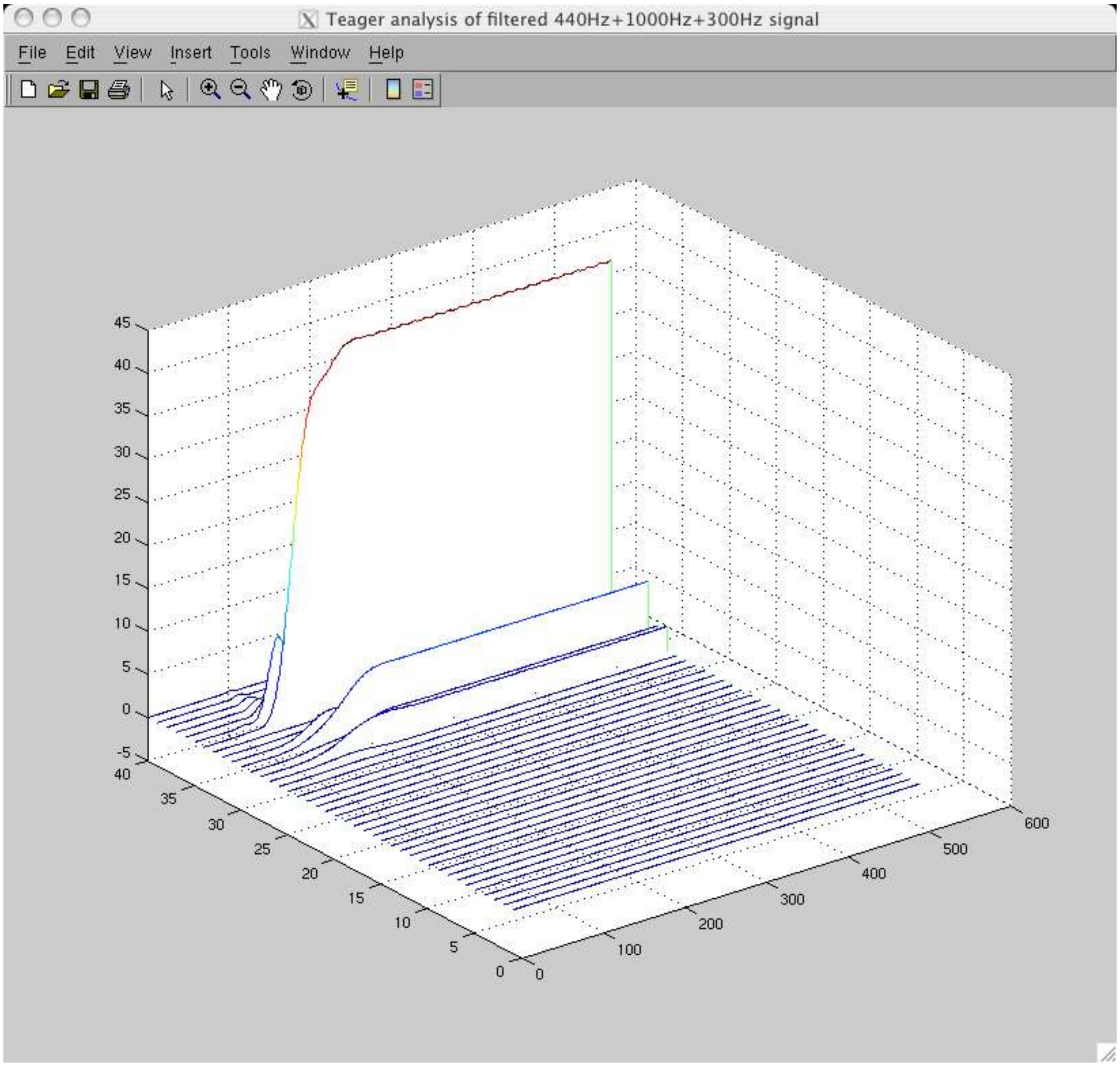

FFT of GTKEO on noiselss signal, m=6
File
Edit
Insert
Tools
Window
Help
Power Spectral Density Estimate via Periodogram
20
0
-20
-40
-60
-80
-100
-120
Power/frequency (dB/Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Frequency (kHz)

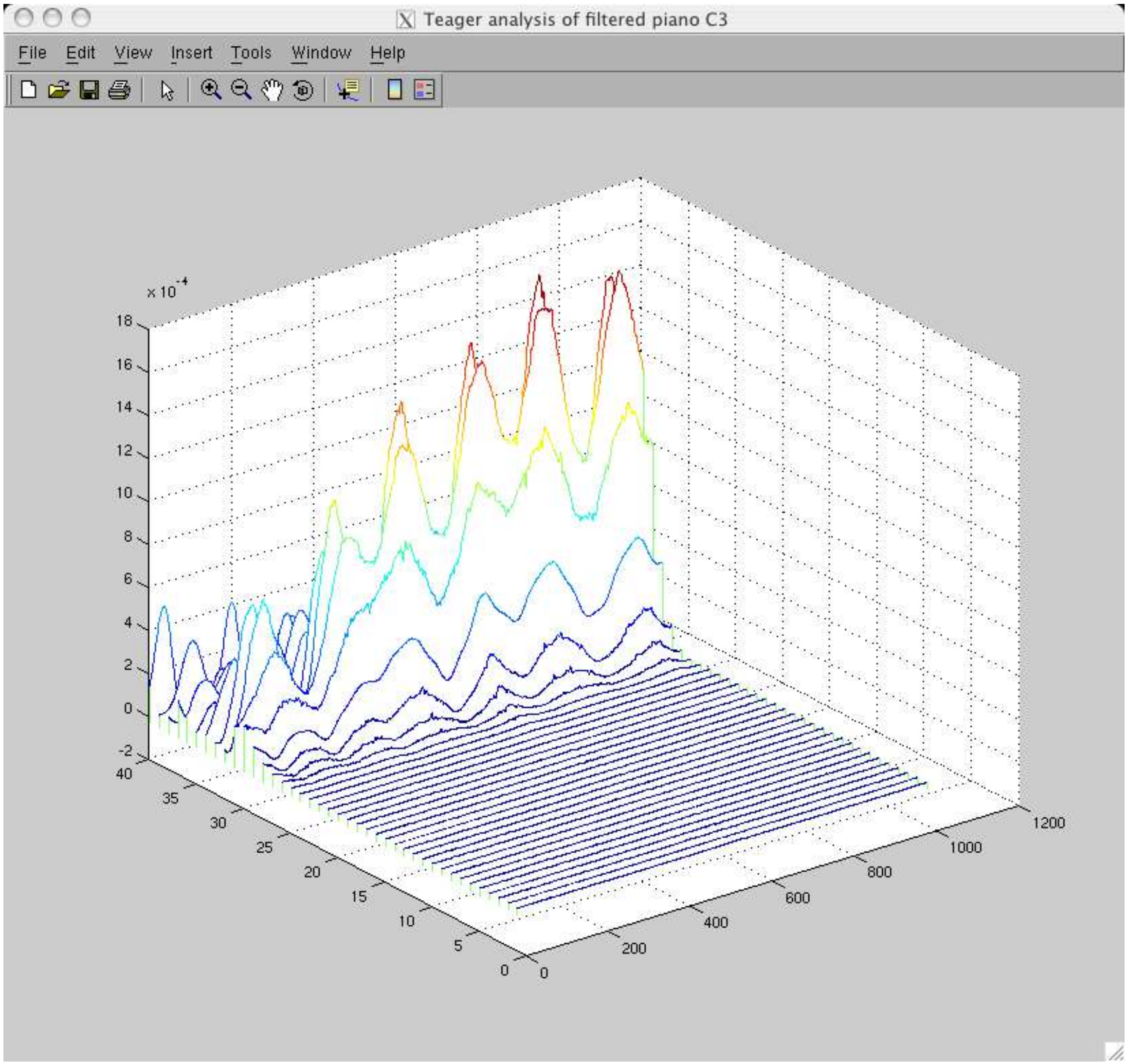

GTKEO on noisy signal, SNR=25, m=6
File
Edit
View
Insert
Tools
Window
Help
6
5
4
3
2
1
0
-1
0
100
200
300
400
500
600
700
800
900
1000

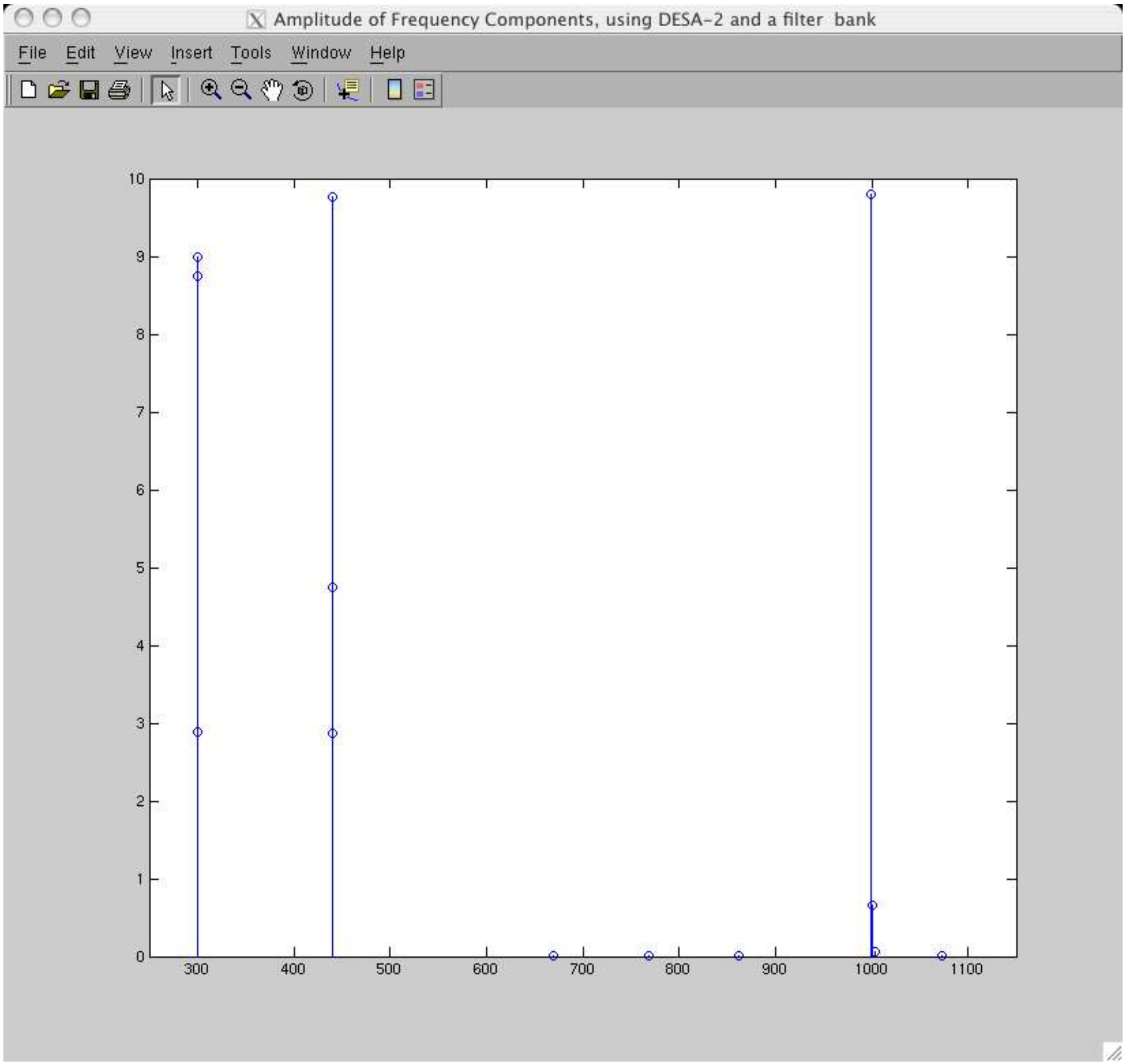

FFT of GTKEO on noisy signal, SNR=25, m=6
File
Edit
Insert
Tools
Window
Help
Power Spectral Density Estimate via Periodogram
Power/frequency (dB/Hz)
20
0
-20
-40
-60
-80
-100
Frequency (kHz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Teager analysis of filtered 440Hz signal
File
Edit
View
Insert
Tools
Window
Help
7
6
5
4
3
2
1
0
-1
40
35
30
25
20
15
10
5
0
0
100
200
300
400
500
600


Teager analysis of filtered 440Hz+1000Hz signal
File
Edit
View
Insert
Tools
Window
Help
45
40
35
30
25
20
15
10
5
0
-5
40
35
30
25
20
15
10
5
0
0
100
200
300
400
500
600


Teager analysis of filtered 440Hz+1000Hz+300Hz signal
File
Edit
View
Insert
Tools
Window
Help
45
40
35
30
25
20
15
10
5
0
-5
40
35
30
25
20
15
10
5
0
0
100
200
300
400
500
600


Teager analysis of filtered piano C3
File
Edit
View
Insert
Tools
Window
Help
x 10^-4
18
16
14
12
10
8
6
4
2
0
-2
40
35
30
25
20
15
10
5
0
0
200
400
600
800
1000
1200


Amplitude of Frequency Components, using DESA-2 and a filter bank
File
Edit
View
Insert
Tools
Window
Help
10
9
8
7
6
5
4
3
2
1
0
300
400
500
600
700
800
900
1000
1100

Appendix II: Associated Max/MSP and MATLAB Files

Max/MSP:

teag.mxb:	Simple patch taking the signal through an inlet and passing the real-time calculated Teager operator through an outlet.
simple_teager.mxb:	A simple patch illustrating how the teag object can be used.

MATLAB:

C3loudmonoshort.wav:	An 8-bit, 44.1Khz, 1 second, mono .wav clip of a piano C3 being struck loudly.
desa2.m:	Calculate the frequency and amplitude of a signal using DESA-2.
disp_spec2.m:	Display spectrum of signal using periodogram.
find_filt.m:	Given a number of filters in a logarithmic filter bank and an overall frequency range, find the filter number corresponding to a frequency (used in myfilters2.m for Part 4).
gen_noise.m:	Generate a noisy signal with a given SNR from a noiseless input image.
gtkef.m:	Implement [4]'s variable lag generalized Teager energy operator.
makemonoshort.m:	Generate a 1 second, 8 bit mono .wav file from a larger stereo .wav file, assuming a 44.1KHz sample rate. Used to process the piano recording before analysis in Part 4.
myfilters1.m:	Implement a basic linear phase response, logarithmic filter bank, and apply the Teager operator to the output of each filter.
myfilters2.m:	Implement a basic linear phase response, logarithmic filter bank. Use DESA-2 to analyze the frequency content of each filter output, then identify the primary frequency components of the signal and their amplitudes.
part1.m:	Demonstrate basic Teager functionality with basic signal inputs, with and without noise.
part2.m:	Examine performance of DESA-2 on different inputs, with and without noise.
part3.m:	Demonstrate the generalized Teager in resolving two tones.
part4.m:	Show how Teager operator can be used on multi-component signals using a filter bank.
teager.m:	Compute the Teager operator on a signal.