Analysis of Synchrosqueezing Transforms for Time-Frequency Representation of Non-Stationary Signals

Project Report, EEE 505 Fall 2023

Victoria Lewis, Himanshu Parashar, Shantanu Patne

Abstract:

This project explores the application of Synchrosqueezing Transforms (SST) for improved time-frequency (TF) representation of non-stationary signals. Motivated by challenges in conventional TF analysis methods for such signals, our study systematically investigates the capabilities and limitations of SST, including its adaptive and second-order variants. Utilizing both Continuous Wavelet Transform (CWT) and Short-Time Fourier Transform (STFT) contexts, we assess SST's efficacy under various conditions, such as combined linear chirp signals and noisy variants. Our methodology involves code analysis, modification, and signal generation to compare conventional and adaptive SSTs. The major findings reveal that while SST offers enhanced representation, the adaptive counterparts require more investigation to properly optimize, presenting clearer and more consistent results across different signal variations, before demonstrating its potential for robust TF analysis of non-stationary signals.

1. Introduction:

This project delves into the exploration and assessment of SSTs as a powerful tool for the TF representation of non-stationary signals. Non-stationary signals, prevalent in diverse fields such as biomedical signal processing, communication systems, and environmental monitoring, pose challenges in conventional TF analysis methods. These challenges include issues such as poor resolution and sensitivity to signals within close range of each other. SSTs leverage adaptive TF analysis by concentrating signal energy in localized regions of the TF plane, thereby providing an enhanced representation of time-varying signal characteristics. This project systematically investigates the capabilities and limitations of SST in capturing the dynamic behavior of non-stationary signals, and the efficacy of its evolved versions, the adaptive SST and second-order SST.

The goal of this project is to determine whether or not SSTs are a suitable TFR for non-stationary signals under a variety of conditions. The main conditions we will be considering are combined linear chirp signals, combined linear signals that cross over each other, and noisy variants of the two. This is to determine under which conditions an SST is most suitable for signal processing, and to determine when performing an adaptive or second-order transformation might be necessary when working with the SST.

This report is organized as follows. In this section, we give a background on the technology used, namely the Synchrosqueezing operations, CWT, STFT, and their respective adaptive counterparts. Following that, in Section 2 we will discuss the different tasks each team member took the lead on and how they contributed to the overall completion of the project. Then the process the team followed to create their results will be presented in Section 3, followed by the results and a discussion of what they mean within the context of this project in Section 4. In Section 5 we will review our results and explain the conclusions we have reached for our project. We will then wrap up the paper by acknowledging those who contributed to the completion of our project in Section 6. The resources we referenced within this project will be seen at the end, all of which were vital to our understanding of the concepts and our processes to receive results, with our most heavily referenced sources being sources [1-3], helping with Short-Time Fourier transform (STFT) based SSTs and continuous wavelet transform (CWT) based SSTs respectively.

1.1. Background:

Synchrosqueezing is a technique used in signal processing to improve the TF representation of signals. It is particularly applied in the context of CWT and STFT due to. The main goal of synchrosqueezing is to obtain a more accurate and interpretable representation of the signal's TF content, especially for non-stationary signals. Synchrosqueezing was first introduced in the analysis of auditory signals [7] as a reassignment method applied on the CWT, hereon noted as Wavelet-based SST (WSST). It represents a specific instance of reallocation methods [6], which strive to enhance a TF representation R(t, ω) by redistributing its values to a different point (t, ω) in the TF plane. This reallocation is determined based on the local characteristics of R(t, ω) around the point (t, ω), aiming to refine and improve the representation. Its STFT variant, hereby known as Fourier-based SST (FSST), was defined in [8] and proved to be robust to noise and small perturbations, similar to the WSST. However, its ineffectiveness on frequency-varying signals [1] led to the investigation of a 2nd order SST [9, 10], which improved the concentration of the TF representation. Adaptive variants of the WSST and FSST have been recently proposed by the authors of [1-3] with a time-varying window. The authors select the time-varying window width by minimizing the Renyi entropy, achieving sharp SST representations. Their results show that the adaptive FSST is very promising in estimating the components' Instantaneous Frequencies (IF) in a well-separated multi-component signal.

1.2. CWT, and WSST:

The Continuous Wavelet Transform, or CWT, is a powerful signal processing technique that utilizes a wavelet, commonly a "bump" or a Morlet wavelet, to capture the signal characteristics at varying scales and positions. Given a signal $x(t)$ and a wavelet, $\psi(t)$, the CWT is defined by,

$$W_x(a,\ b)\ =\ \int_{-\infty}^{\infty} x(t)\frac{1}{\sqrt{a}}\overline{\psi(\frac{t-b}{a})}dt \qquad -(1)$$

The variables a and b are called the scale and time variables respectively. This transform is invertible and we can obtain the original signal $x(t)$ along with its components from $W_x(a,\ b)$.

$$x(t)\ =\ \frac{1}{C_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W_x(a,b)\psi_{a,b}(t)db\frac{da}{|a|}\ , \qquad -(2)$$

where C_ψ is the admissible condition for a continuous wavelet [2].

The result of the CWT is often represented as a scalogram, which is a 2D plot showing how the amplitude of the wavelet transform varies with time and scale. The parameters of the wavelet function, namely σ, in the Morlet wavelet, controls the window width of the TF localization and thus affects the resolution of the TF representation obtained from CWT. To achieve sharper results, we reassign the scale variable a to a frequency variable in an operation known as *Synchrosqueezing*.

To define SST, we must first obtain the phase transformation $\omega_x(a,\ b)$ (also known as the reference IF function) by taking the partial derivative of CWT with respect to b, defined as,

$$\omega_x(a,\ b)\ =\frac{1}{j2\pi}\ \frac{\partial}{\partial b}log\left|W_x(a,\ b)\right|\ =\ \frac{\frac{\partial}{\partial b}W_x(a,b)}{j2\pi W_x(a,b)},\quad for\, W_x(a,\ b)\ \neq\ 0.\qquad -(3)$$

Using this phase transformation, we map the time-scale information to the TF plane [6] using the SST, denoted as $T_x(\xi,\ b)$ where ξ is the frequency variable. It is defined as,

$$T_x(\xi,\ b)\ =\ \int_{a\in\Re_+:\ W_x(a,b)\neq 0} W_x(a,\ b)\delta(\omega_x(a,\ b)\ -\ \xi)\frac{da}{a}\qquad -(4)$$

This transformation is also invertible and $x(t)$ can be recovered from its SST similar to the recovery from the CWT. For multicomponent signals, if the components are overlapping, then the conventional SST defined above will be unable to separate the components [2]. Therefore, the authors of [2] propose a time-varying parameter $\sigma\ =\ \sigma(t)$ such that the associated SST gives a sharp resolution, provided that the well-separated conditions they define are satisfied. The well-separated conditions state that the components of the CWT in the time-scale plane need to be sufficiently separable with the upper bound of the k^{th} component greater than the lower bound of the $(k\ +\ 1)^{th}$ component.

The adaptive wavelet is defined as,

$$\widehat{\psi}_\sigma(\xi)\ =\ \overline{\widehat{g}(\sigma(\mu\ -\ \xi))},\qquad -(5)$$

where, $\mu\ >\ 0$, and g is a fast-decaying function. The definition of CWT does not change with the new wavelet. Let this adaptive CWT be denoted as $W_x^{adp}(a,\ b)$. Therefore, similar to the conventional SST, we first estimate the IF using the phase transformation $\omega_x^{adp}(a,b)$. To do so we first denote $g_2(t)\ =\ tg'(t)$ and calculate the CWT using $g_2(t)$ denoted as $W_{x,g_2}^{adp}(a,b)$. Then, the phase transformation is given as,

$$\omega_x^{adp}(a,b)\ =\ Re\{\frac{\partial_b(W_x^{adp}(a,b))}{j2\pi W_x^{adp}(a,b)}\}\ +\ \frac{\sigma'(b)}{\sigma(b)}\ Re\{\frac{W_{x,g_2}^{adp}(a,b)}{j2\pi W_x^{adp}(a,b)}\}\quad for\, W_x^{adp}(a,\ b)\ \neq\ 0\qquad -(6)$$

The time-varying SST (adaptive SST) is then defined by,

$$T_x^{adp}(\xi,\, b) \;=\; \int_{a\in\Re_+:\, W_x(a,b)\neq 0} W_x^{adp}(a,\, b)\delta(\omega_x^{adp}(a,\, b) \;-\; \xi)\frac{da}{a} \qquad -(7)$$

This can also be called the first-order adaptive SST. To calculate the second-order SST, we take the partial derivative of the CWT with respect to a. The derivation of $\omega_x^{2nd}(a,\, b)$ can be referred to in [2]. This is given as,

$$T_x^{2nd}(\xi,\, b) \;=\; \int_{a\in\Re_+:\, W_x(a,b)\neq 0} W_x(a,\, b)\delta(\omega_x^{2nd}(a,\, b) \;-\; \xi)\frac{da}{a} \qquad -(8)$$

for the conventional SST, and defined as follows for the adaptive counterpart.

$$T_x^{2adp}(\xi,\, b) \;=\; \int_{a\in\Re_+:\, W_x(a,b)\neq 0} W_x^{adp}(a,\, b)\delta(\omega_x^{2adp}(a,\, b) \;-\; \xi)\frac{da}{a}. \qquad -(9)$$

These transforms are also invertible and the original signal $x(t)$ can be recovered as in previous cases.

1.3. STFT, and FSST:

The Short-Time Fourier Transform (STFT) is a time-frequency analysis technique commonly used in signal processing to reveal the frequency content of a signal as it evolves. It provides a compromise between the time and frequency resolutions, making it suitable for analyzing non-stationary signals. The STFT uses a window to estimate the time-frequency characteristics of the signal. In [1, 3] the authors use a window with a time-varying width to implement the adaptive STFT and the consequent SSTs. Similar to the CWT and WSST, we first define the STFT of a signal $x(t)$ with a window $h_\sigma(t)$

$$S_x(t,\, f) \;=\; \int_{-\infty}^{\infty} x(t)\, \overline{h_\sigma(\tau \;-\; t)}\, e^{j2\pi f(\tau - t)} d\tau \qquad -(10)$$

where f is the frequency variable, σ is known as the window width and $h_\sigma(t)$ is defined as,

$$h_\sigma(t) \;=\; \frac{1}{\sigma} h(\frac{t}{\sigma}) \;...\; for\, \sigma \;>\; 0 \qquad -(11)$$

As with the Wavelet counterpart in Eqn. (3), we first define a phase transformation $\omega_x(t,\, f)$ by taking a partial derivate with respect to the time variable as,

$$\omega_x(t,f) \;=\; \frac{\frac{\partial}{\partial t}S_x(t,f)}{j2\pi S_x(t,f)} \;\; for\, S_x(t,\, f) \;\neq\; 0 \qquad -(12)$$

Using this phase transformation, similar to the WSST Eqn. (4), we reassign the frequency variable using the *synchrosqueezing* operation, denoted as $R_x(t,\, \xi)$, where ξ is the reassigned frequency variable. This is defined as,

$$R_x(t,\, \xi) \;=\; \int_{\varsigma:S_x(t,\,\varsigma)\neq 0} S_x(t,\, \varsigma)\delta(\omega_x(t,\, \varsigma) \,-\, \xi)d\varsigma \qquad -(13)$$

We follow the same procedure from Eqns. (5) - (9) to obtain the Adaptive STFT, the first and second-order adaptive FSST, and the second-order conventional FSST. The derivations can be referred to in [1,3]. The final SST equations are given as,

$$R_x^{adp}(t,\xi) \;=\; \int_{\eta\in\Re:S_x^{adp}(t,\,\eta)\neq 0} S_x^{adp}(t,\, \eta)\delta(\omega_x^{adp}(t,\, \eta) \,-\, \xi)d\eta \qquad -(14)$$

for the adaptive FSST, where $S_x^{adp}(t,\, \eta)$ is the adaptive STFT and $\omega_x^{adp}(t,\, \eta)$ is its corresponding phase transformation.

$$R_x^{2nd}(t,\xi) \;=\; \int_{\eta\in\Re:S_x(t,\,\eta)\neq 0} S_x(t,\, \eta)\delta(\omega_x^{2nd}(t,\, \eta) \,-\, \xi)d\eta \qquad -(15)$$

$$R_x^{2adp}(t,\xi) \;=\; \int_{\eta\in\Re:S_x^{adp}(t,\,\eta)\neq 0} S_x^{adp}(t,\, \eta)\delta(\omega_x^{2adp}(t,\, \eta) \,-\, \xi)d\eta \qquad -(16)$$

Equations. (15) and (16) give the second-order FSSTs for the conventional and the adaptive STFT, respectively. The transforms are all invertible and individual components of the original signal can be recovered as showcased in [3].

2. Task Management:

Each of our team members worked hard on this project and contributed a good amount of effort to it. Much of the work done on the project was done as a group, with the team frequently meeting in person and discussing the tasks at hand. Due to this, the team members have similar contributions to each other in regards to the project. Additionally, the initial vision for this project was slightly different, also including an analysis of an Adaptive array of Liènard-type Intermittence (ALI) in addition to the SSTs which has since been removed from the project scope due to its differences from SSTs making comparison difficult between the two. Some of the tasks completed by the team members early on in the project were concerning this, which while not included in the final scope of the project, is still work that they should be credited for as without it the project scope would not have been altered to its current state.

Victoria began working primarily with code provided by resources [1-3], focusing on the WSST code. She spent time going through the code to figure out what each line of the code did, leaving comments where needed. With her team members, she helped to determine which parts of the code were necessary to satisfy the scope of the project, and which parts of the code could be omitted. She helped with the organization of the final report, and she contributed primarily to the writing of sections 1, 4, and 6. She also helped coordinate some meeting times for the team throughout the semester and helped to arrange the workloads of different team members so their tasks didn't cause additional stress during busy times with other classes.

Shantanu worked on the FSST code provided by [1, 3] to understand the algorithm presented. He investigated the initial ALI approach of signal estimation [4] and found it to be unfeasible for the sake of this project. He went through the FSST and WSST theorized in [1-3] to understand the underlying principles behind the techniques and analyze the results obtained from both. He has worked primarily on Sections 1, 4, and 5. to summarize the theoretical equations in FSST and WSST and discuss the simulation results. He also collated the necessary code into a common repository for everyone to access and combined the required plots into a script for the ease of simulation.

Himanshu worked on a signal generation script for MATLAB to integrate with the synchrosqueezing codes provided by [1-3], which allows the team to quickly create various signals to test while ensuring that the same signals are being used for both CWT and STFT synchrosqueezing method. He added varying levels of Additive White Gaussian Noise (AWGN) to these signals to understand how synchrosqueezing methods handle noisy conditions. His contribution extended to organizing the final report, with a primary focus on Sections 3, 4, and 5 along with the Abstract section. He actively participated in project group meetings whether conducted in person or online within the team.

3. Methods

The completion of this project was achieved primarily by analyzing and altering the code used in sources [1-3]. The code implements the equations defined in Sections 1.1 and 1.2, most prominently Eqn. (4), (7) - (9), and Eqn. (13) to (16). The code initially included more results than were needed for this project and did not include a conventional second-order SST for either the FSST or the WSST. The code was adjusted to run the adaptive portion of the code twice, once by removing the time variation that makes the SST adaptive to create both first and second-order conventional SSTs, and the second time by putting the time variation back in place to compute the adaptive variation of both SSTs. Several portions of the code were also deemed unnecessary for the scope of this project and removed, mostly where the code computed results for a second time-variation value to compare different adaptive results, as our project scope is to compare first and second-order conventional and adaptive SSTs to each other. This altered code was then run using various signals generated under different circumstances to get the results discussed in the following section.

4. Results and Discussion:

For our project, we ran two separate signals through the WSST and FSST code the team created based on the code provided by sources [1-3]. The first was a multicomponent non-overlapping signal with no noise, which was then repeated with a signal-to-noise ratio (SNR) of 10dB. The second was an overlapping multicomponent signal with no noise, which was also repeated with a signal-to-noise ratio (SNR) of 10dB.

The multicomponent non-overlapping signal was the sum of two linear chirp signals preprogrammed into the MATLAB code signal_gen.m, visible in Appendix A. The signal's time domain representation was plotted at the start so the team could see what information would go through the different transformations for a more accurate understanding of the results. This plot is visible here in Figure 1.

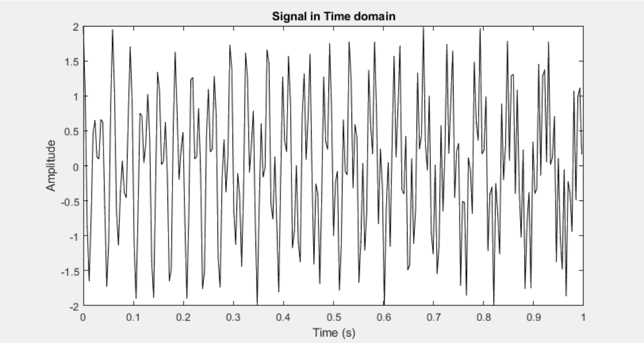

Figure 1: Plot of Noiseless Linear Chirp Non-Overlapping Multicomponent Signal in the Time Domain

Once this signal was generated, it was run through the WSST code to generate the signal's CWT, conventional first and second-order WSSTs, and adaptive first and second-order WSSTs. The resulting plots from these transformations are visible below in Figure 2.

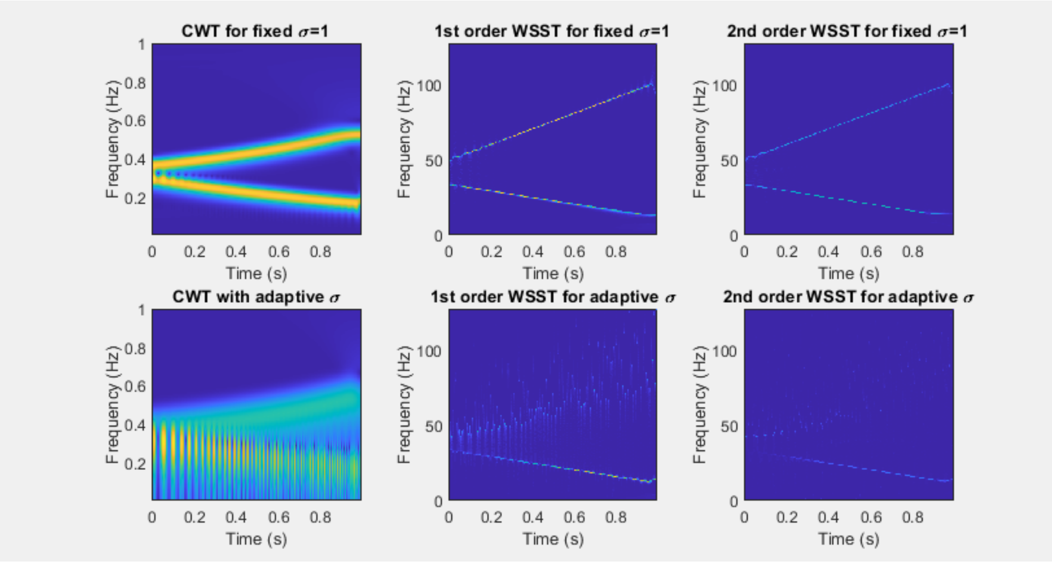

Figure 2: Conventional CWT of the non-overlapping signal with no noise (Top-left); First order conventional WSST of the non-overlapping signal with no noise (Top-center); Second order conventional WSST of the non-overlapping signal with no noise (Top-right); Adaptive CWT of the non-overlapping signal with no noise (Bottom-left); First

order adaptive WSST of the non-overlapping signal with no noise (Bottom-center); Second order adaptive WSST of the non-overlapping signal with no noise (Bottom-right)

Then the same process was repeated with the FSST, finding the STFT transforms for both conventional and adaptive methods, then taking both the first and second-order SST using the respective STFT results for each method. This created 6 additional plots of a similar nature to the WSST, which can be seen in Figure 3.

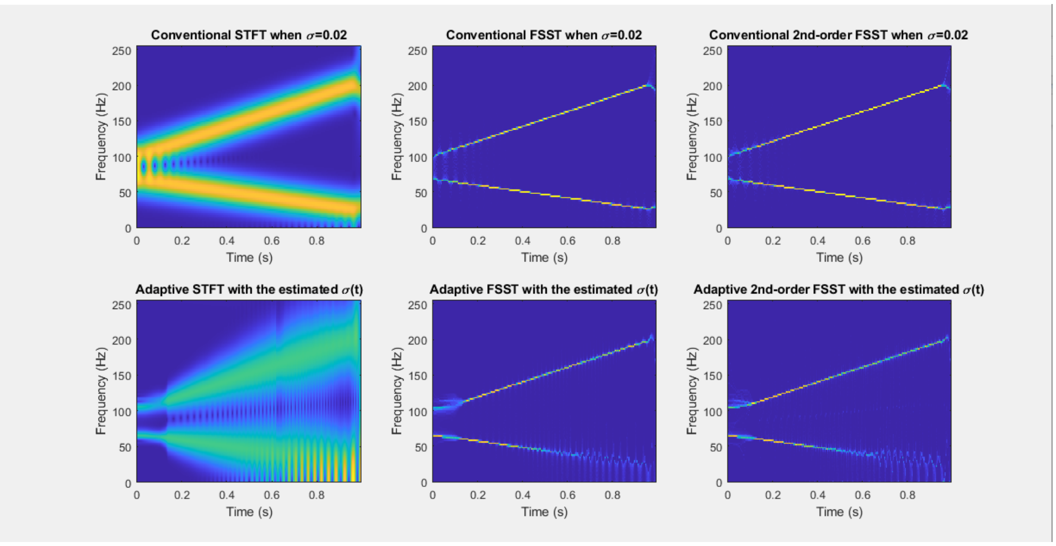

Figure 3: Conventional STFT of the non-overlapping signal with no noise (Top-left); First order conventional FSST of the non-overlapping signal with no noise (Top-center); Second order conventional FSST of the non-overlapping signal with no noise (Top-right); Adaptive STFT of the non-overlapping signal with no noise (Bottom-left); First order adaptive FSST of the non-overlapping signal with no noise (Bottom-center); Second order adaptive FSST of the non-overlapping signal with no noise (Bottom-right)

The results of both the WSST and the FSST have similar patterns, with the TFR of the signals being visible to some degree in conventional and adaptive first and second-order transforms for both types of SSTs. Within the WSST from Figure 2, the clarity of the results remains largely the same when going from first order to second order, but shows a drastic degradation when going from the conventional to the adaptive WSST. The degradation in signal quality when using an adaptive transformation is likely caused by an inaccurate calculation of the time-varying parameter σ. The estimation algorithm provided by the authors of [1-3] falls short and cannot correctly estimate the σ required to get a sharp representation. The authors also vary other wavelet parameters σ to estimate the most accurate σ_{est}. In our simulations, however, we only vary the σ parameters which might be the cause of our incorrect results.

When comparing the WSST results to the FSST results, the FSST results in Figure 3 are much clearer and more consistent between all variations, not experiencing the degradation the WSST does between orders and methods, with almost no differences between first and second-order transformations, and a significantly smaller difference between the conventional and adaptive FSST than there was with the WSST. While the difference was smaller, the FSST's adaptive transforms did perform worse than the conventional. This is likely due to a similar cause as that in the WSST, i.e. the other window parameters need to also be varied to obtain the most accurate result. While this seems like a trial-and-error method, it needs further investigation to optimize all necessary parameters.

The next signal analyzed used the same non-overlapping multicomponent signal as previously analyzed, using Additive White Gaussian Noise (AWGN) with an SNR of 10dB to determine how the transforms handled noisy signals. To maintain consistent results, the procedure for this signal was kept the same despite the only modification to the signal being the addition of noise. Being so, the analysis of this signal began with plotting the signal in the time domain, as can be seen in Figure 4.

Once this signal was generated, it was run through the WSST code just like its noiseless version to generate the signal's CWT, conventional first and second-order WSSTs, and adaptive first and second-order WSSTs. These results are visible below in Figure 5.

Again in the same manner as this signal's Noiseless version, the same process was repeated with the FSST. The STFT transforms for both conventional and adaptive methods were plotted, and then both the first and second-order SST for each method were plotted as well. These plots can be seen in Figure 6.

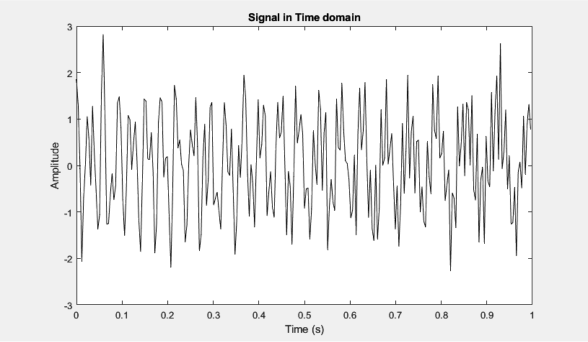

Figure 4: Plot of Linear Chirp Multicomponent Non-Overlapping Signal with Signal-to-Noise-Ratio 10dB in the Time Domain

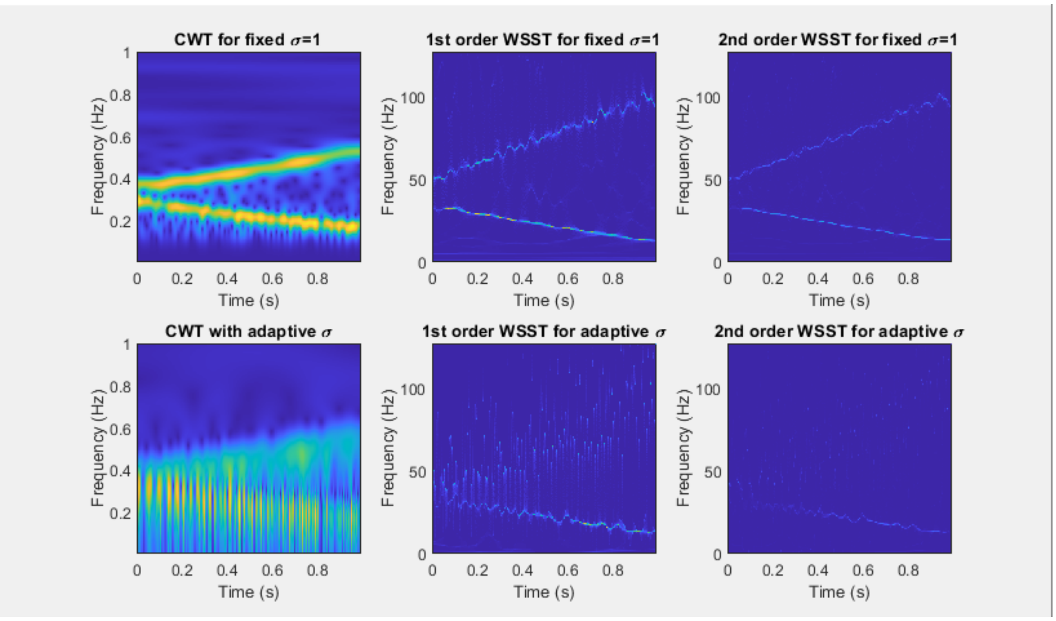

Figure 5: Conventional CWT of the noisy non-overlapping signal (Top-left); First order conventional WSST of the noisy non-overlapping signal (Top-center); Second order conventional WSST of the noisy non-overlapping signal(Top-right); Adaptive CWT of the noisy non-overlapping signal (Bottom-left); First order adaptive WSST of the noisy non-overlapping signal with (Bottom-center); Second order adaptive WSST of the noisy non-overlapping signal (Bottom-right)

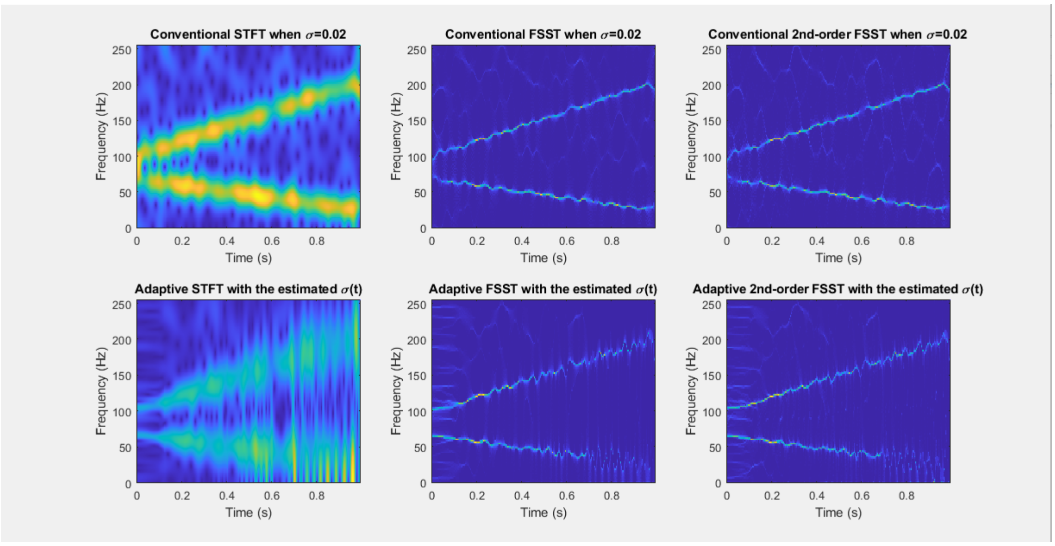

Figure 6: Conventional STFT of the noisy non-overlapping signal (Top-left); First order conventional FSST of the noisy non-overlapping signal (Top-center); Second order conventional FSST of the noisy non-overlapping signal (Top-right); Adaptive STFT of the noisy non-overlapping signal (Bottom-left); First order adaptive FSST of the noisy non-overlapping signal (Bottom-center); Second order adaptive FSST of the noisy non-overlapping signal (Bottom-right)

The results seen in Figure 5 and Figure 6 can be easily compared to those in Figure 2 and Figure 3, which is expected as the two pairs are derived from the same signals and only with and without noise. Due to this, the noisy non-overlapping signal has the same differences between its signals as was the case for the noiseless version. The real comparison to be made with these results comes from comparing the noisy signals to the noiseless ones. Compared to the noiseless results, the noise is very evident in all these plots. The FSST results in Figure 6 all appear to react to the noise in the same way, while the WSST results in Figure 5 seem to be affected less by the noise in their second-order transformations than in their first, showing a slight level of noise resistance.

A multicomponent overlapping signal with no noise was analyzed next, where the signal was the sum of two linear chirp signals, which are also preprogrammed into the MATLAB code visible in Appendix A. The signal's time domain representation was plotted just like with the prior signals for a consistent comparison of results. This plot is visible here in Figure 7.

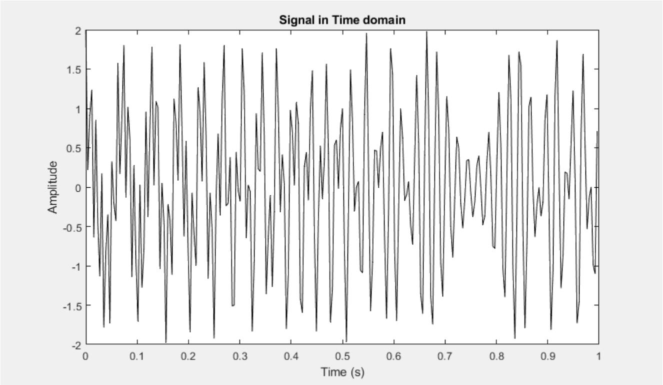

Figure 7: Plot of Noiseless Linear Chirp Overlapping Multicomponent Signal in the Time Domain

This signal was then run through the WSST code to generate the CWT, conventional first and second-order WSSTs, and adaptive first and second-order WSSTs. The plots created from these transformations can be seen as follows in Figure 8.

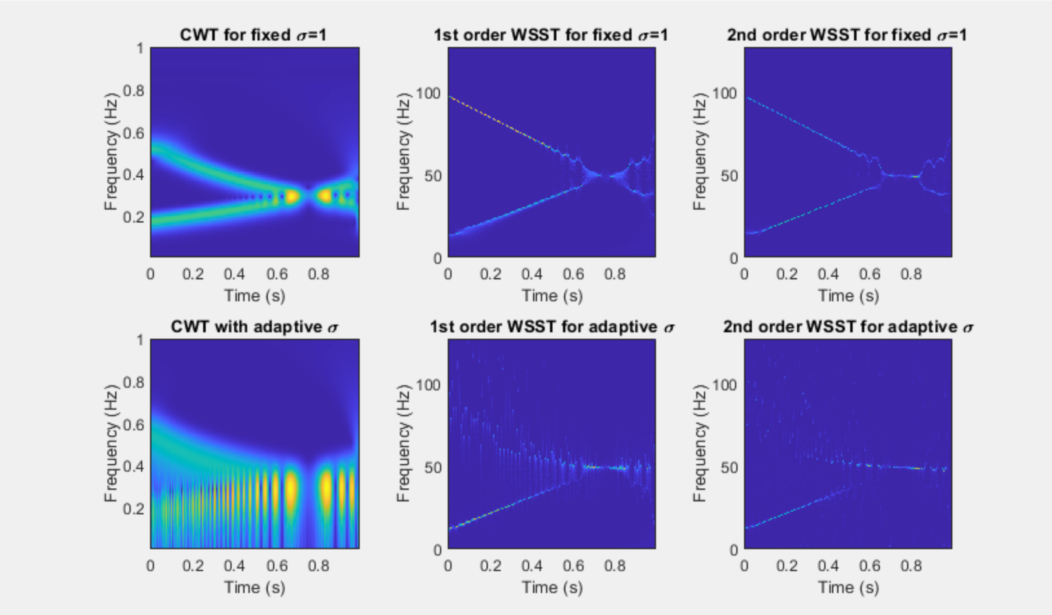

Figure 8: Conventional CWT of the overlapping signal with no noise (Top-left); First order conventional WSST of the overlapping signal with no noise (Top-center); Second order conventional WSST of the overlapping signal with no noise (Top-right); Adaptive CWT of the overlapping signal with no noise (Bottom-left); First order adaptive WSST of the overlapping signal with no noise (Bottom-center); Second order adaptive WSST of the overlapping signal with no noise (Bottom-right)

As before, the signal undergoes the same process with the FSST, finding the STFT transforms for both conventional and adaptive methods, and then getting both the first and second-order SST for each method. These plots can be seen in Figure 9.

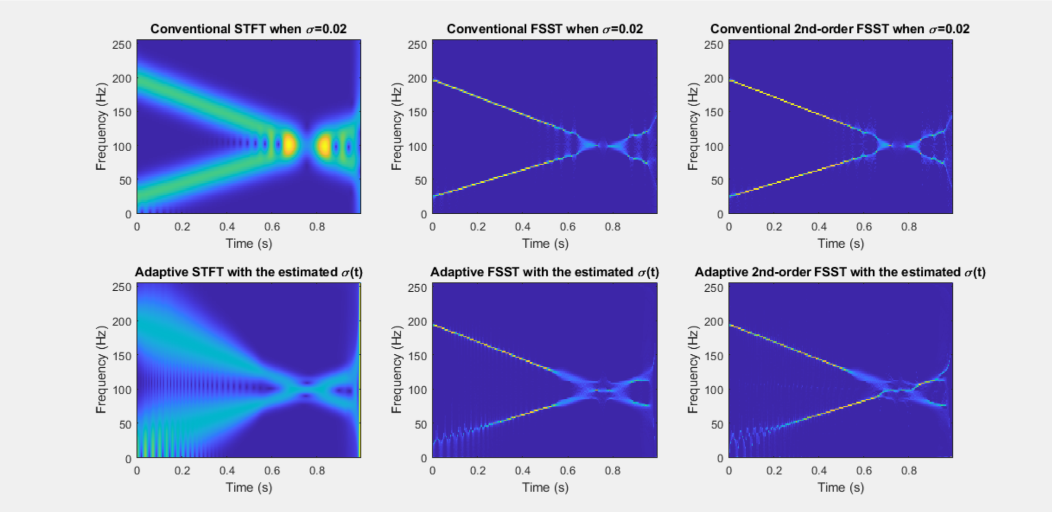

Figure 9: Conventional STFT of the overlapping signal with no noise (Top-left); First order conventional FSST of the overlapping signal with no noise (Top-center); Second order conventional FSST of the overlapping signal with no noise (Top-right); Adaptive STFT of the overlapping signal with no noise (Bottom-left); First order adaptive FSST of the overlapping signal with no noise (Bottom-center); Second order adaptive FSST of the overlapping signal with no noise (Bottom-right)

The results of both the WSST and the FSST again have similar patterns in their TFRs. The WSST from Figure 8 experiences some distortion from the overlapping signals in all of its variations and again displays consistent clarity in the results when going from first order to second order but shows degradation when going from a conventional to adaptive WSST. The conventional WSST shows good results when using a fixed σ, while the adaptive WSST is unable to estimate the time-varying $\sigma(t)$ properly. When comparing the WSST results to the FSST results, the FSST results don't experience the level of distortion that the WSST does with the overlapping signals, with an especially clearer result in the adaptive transforms. There are, however, some oscillations at the beginning of the plot due to an inaccurate σ estimation. The adaptive transformation results do slightly differ from the conventional results, which is also for the same reasons as this difference was with the non-overlapping signals.

The final signal analyzed was the multicomponent overlapping signals using AWGN with an SNR of 10dB to determine how the transforms performed with an overlapping noisy signal. Once again following the same procedure, the time-domain representation of the multicomponent overlapping signal with noise is shown below in Figure 10.

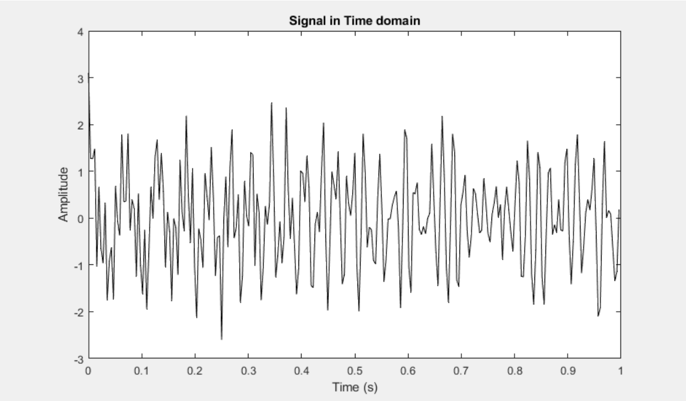

Figure 10: Plot of Linear Chirp Multicomponent Overlapping Signal with Signal-to-Noise-Ratio 10dB in the Time Domain

This signal was then run through the WSST code to generate the signal's CWT in the same manner as its noiseless version to get the conventional first and second-order WSSTs and adaptive first and second-order WSSTs. These results are visible below in Figure 11.

The FSST for the noisy signal was then calculated like its noiseless counterpart, plotting the STFT transforms for both conventional and adaptive methods, and then plotting both the first and second-order SST for each method. These plots can be seen in Figure 12.

The results seen in Figure 11 and Figure 12 can be easily compared to those in Figure 8 and Figure 9, much like their non-overlapping counterparts. The noisy overlapping signal has the same differences between its signals in both the WSST and the FSST as the noiseless versions have. When comparing to the noiseless results, the noise is very evident in all these plots, with the FSST plots showing good resolution in the higher-order SSTs due to the high energy concentration. The second-order WSST plots seem to be less affected by the noise than the first-order WSST in the conventional section for the same reasons as FSST, which aligns well with what was observed in the noisy non-overlapping version of these signals.

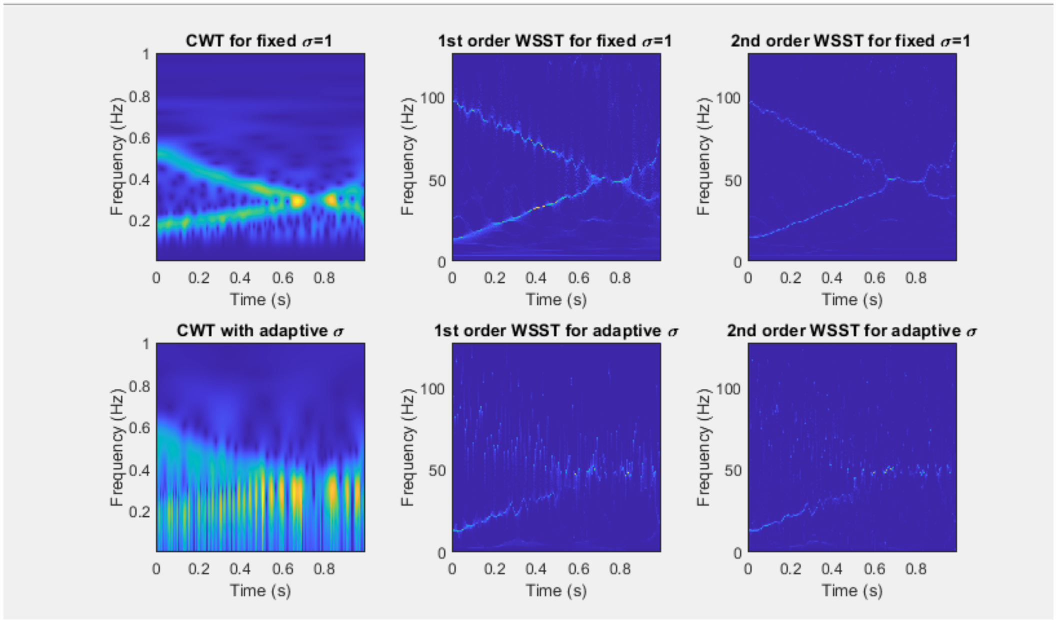

Figure 11: Conventional CWT of the noisy overlapping signal (Top-left); First order conventional WSST of the noisy overlapping signal (Top-center); Second order conventional WSST of the noisy overlapping signal(Top-right); Adaptive CWT of the noisy overlapping signal (Bottom-left); First order adaptive WSST of the noisy overlapping signal with (Bottom-center); Second order adaptive WSST of the noisy overlapping signal (Bottom-right)

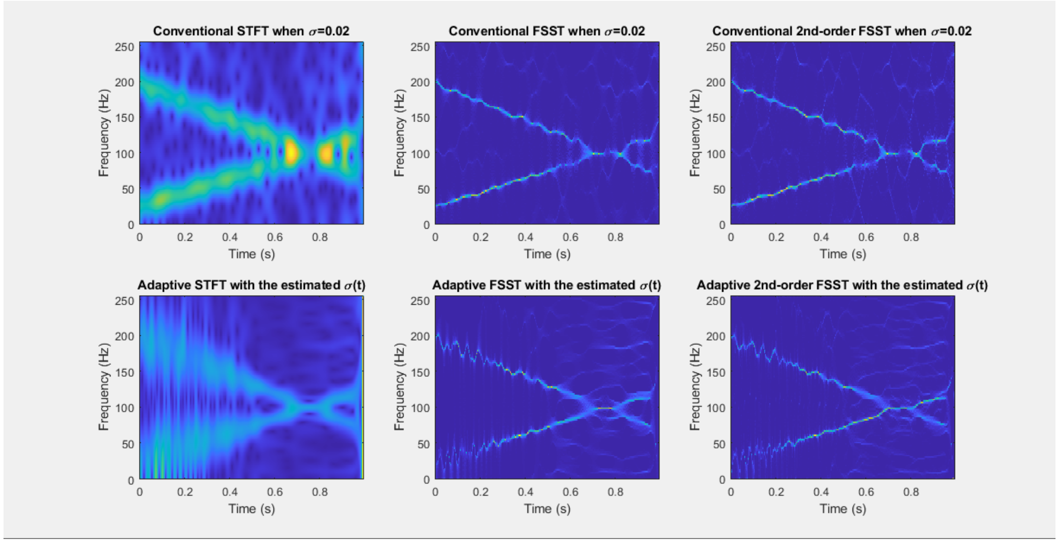

Figure 12: Conventional STFT of the noisy overlapping signal (Top-left); First order conventional FSST of the noisy overlapping signal (Top-center); Second order conventional FSST of the noisy overlapping signal (Top-right); Adaptive STFT of the noisy overlapping signal (Bottom-left); First order adaptive FSST of the noisy overlapping signal (Bottom-center); Second order adaptive FSST of the noisy overlapping signal (Bottom-right)

5. Conclusions:

Our project has provided an analysis of SST and their adaptive variants for improved time-frequency representation of non-stationary signals. Through the systematic exploration of both CWT and STFT contexts, we explored SST's capabilities and limitations under various conditions, including combined linear chirp signals and noisy variants. Our findings reveal that the Fourier-based transforms perform significantly better than the CWT, in noisy as well as clean conditions. However, that can also be attributed to an incomplete implementation of the algorithms provided by the original authors. The code provided for the STFT-based transforms is more recent and seems to be better maintained, while there is a requirement for additional efforts in analyzing and optimizing the CWT-based codes and conducting thorough investigations into the various parameters that can be modified.

In a nutshell, the project not only contributes to our understanding of signal processing techniques but also emphasizes the significance of adaptability and second-order analysis in SST. As technology continues to evolve, the findings from this study can guide further research and applications of SST in real-world scenarios when accurate time-frequency representation of non-stationary signals is paramount. Our team's collaborative efforts, along with the insights gained from this project can be built upon in the field of SSTs, and their adaptive and higher-order variations.

6. Acknowledgments:

The completion of the project was a group effort among the team, but also could not have been accomplished without the prior work done in other studies. The team especially wants to thank sources [1-3] for their work on the code the team based their project on. We also want to thank Arizona State

University for supplying the licenses to access some of the referenced articles and for the use of MATLAB. Finally, we want to thank our professor, Antonia Papandreou-Suppappola, for her consultation on our project topic and scope, her classroom instruction providing us with information we could use to aid in our understanding of the concepts used in this report, and for implementing a class allowing us the ability to complete a project of this nature where we had the freedom to choose our topic.

References:

[1] L. Li, H. Cai, H. Han, Q. Jiang, and H. Ji, "Adaptive short-time Fourier transform and synchrosqueezing transform for non-stationary signal separation", *Signal Processing*, vol. 166, pp. 107-231, Jul. 2019

[2] L. Li, H. Cai, and Q. Jiang, "Adaptive synchrosqueezing transform with a time-varying parameter for non-stationary signal separation," *Applied and Computational Harmonic Analysis*, vol. 49, no. 3, pp. 1075-1106, Nov. 2020

[3] H. Cai, Q. Jiang, L. Li, and B. Suter, "Analysis of adaptive short-time Fourier transform-based synchrosqueezing" *Analysis and Applications,* vol 19, no 01, May 2019

[4] P. Pancóatl-Bortolotti, A.H. Costa, R.A. Enríquez-Caldera, J.F. Guerrero-Castellanos, and M. Tello-Bello, "Time-frequency high-resolution for weak signal detection using chaotic intermittence", *Digital Signal Processing*, vol. 141, pp. 104-160, Jul. 2023

[5] Y. Hu, X. Tu, and F. Li, “High-order synchrosqueezing wavelet transform and application to planetary gearbox fault diagnosis.” *Mechanical Systems and Signal Processing*, vol 131, pp. 126-151, June 2019

[6] I. Daubechies, J. Lu, and H.-T. Wu, "Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 243-261, 2011.

[7] I. Daubechies and S. H. Maes, "A Nonlinear Squeezing of the Continuous Wavelet Transform Based on Auditory Nerve Models," 2017.

[8] G. Thakur, H.T. Wu, J. Math, Synchrosqueezing based recovery of instantaneous frequency from nonuniform samples, *SIAM Journal on Mathematical Analysis*, vol. 43, no. 5, pp. 2078–2095, 2011.

[9] R. Behera, S. Meignen, and T. Oberlin, Theoretical analysis of the 2nd-order synchrosqueezing transform, Appl. Comput. Harmon. Anal., 45 (2018), pp. 374–404.

[10] T. Oberlin and S. Meignen, The 2nd-order wavelet synchrosqueezing transform, in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), March 2017, New Orleans, LA, USA.

Appendix A

signal_gen.m

```
function [s, f, r, c] = signal_gen(type, noise, N)
%   Generate signals for use in the EEE505 project
%   Can generate 3 different signals: linear chirp, quadratic chirp,
%   multicomponent (linear + quadratic) chirp signal
%
%   INPUTS
%   N = sample length/sampling frequency
%   t = signal time vector
%   type = signal type, (DEFAULT = 'n')
%         ('n'/'o' = non-overlapping/overlapping multicomponent signal)
%   noise = set if generated signal to have AWGN, (DEFAULT = false)
%
%   OUTPUT
%   signal = generated signal of specified type and noise
%

    % Set default arguments
    arguments
        type = 'n'
        noise = false
        N = 256
    end

    t = linspace(0, 1-1/N, N);
    snr = 10;
    switch type
        case 'n'
            % Generate real linear chirp
            c1 = 50; % Start
            c2 = 102; % End
            x1 = chirp(t, c1, 1, c2);
            r1 = (c2 - c1);
            f1 = c1 + r1*t;
            % Generate real linear chirp
            d1 = 34; % Start
            d2 = 12; % End
            x2 = chirp(t, d1, 1, d2);
            r2 = (d2 - d1);
            f2 = d1 + r2*t;
            % Combine both and add noise
            s = x1 + x2;
            if (noise)
                s = awgn(s, snr, 'measured');
            end
            r = [r1; r2];
            c = [c1; d1];
            f = [f1; f2];
        case 'o'
            % Generate real linear chirp
            c1 = 12; % Start
            c2 = 62; % End
            x1 = chirp(t, c1, 1, c2);
            r1 = (c2 - c1);
            f1 = c1 + r1*t;

            % Generate real linear chirp
            d1 = 98; % Start
            d2 = 34; % End
            x2 = chirp(t, d1, 1, d2);
            r2 = (d2 - d1);
            f2 = c1 + r2*t;
            % Combine both and add noise
            s = x1 + x2;
            if (noise)
                s = awgn(s, snr, 'measured');
            end
            r = [r1; r2];
            c = [c1; d1];
            f = [f1; f2];
        otherwise
            msg = sprintf("ERROR: Not a valid signal type.\nCheck help for valid arguments.");
            error(msg);
    end
end
```