Brigham Young University
BYU ScholarsArchive

Theses and Dissertations

2019-06-01

Toward Using Empirical Mode Decomposition to Identify Anomalies in Stream FlowData and Correlations with other Environmental Data

Saul Gallegos Ramirez
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Engineering Commons

BYU ScholarsArchive Citation

Ramirez, Saul Gallegos, "Toward Using Empirical Mode Decomposition to Identify Anomalies in Stream FlowData and Correlations with other Environmental Data" (2019). *Theses and Dissertations*. 7574.
https://scholarsarchive.byu.edu/etd/7574

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Toward Using Empirical Mode Decomposition to Identify Anomalies in Stream Flow Data and Correlations with Other Environmental Data

Saul Gallegos Ramirez

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Gustavious P. Williams, Chair
Daniel P. Ames
Rollin H. Hotchkiss

Department of Civil and Environmental Engineering

Brigham Young University

Copyright © 2019 Saul Gallegos Ramirez

All Rights Reserved

ABSTRACT

Toward Using Empirical Mode Decomposition to Identify Anomalies in Streamflow Data and Correlations with Other Environmental Data

Saul Gallegos Ramirez
Department of Civil and Environmental Engineering, BYU
Master of Science

I applied empirical mode decomposition (EMD) and the Hilbert-Herbert transforms, as tools to analyze streamflow data. I used the EMD method to extract and analyze periodic processes and trends in several environmental datasets including daily stream flow, daily precipitation, and daily temperature on data from the watersheds of two rivers in the Upper Colorado River Basin, the Yampa and the Upper-Green rivers. I used these data to identify forcing functions governing streamflow. Forcing functions include environmental factors such as temperature and precipitation and anthropogenic factors such as dams or diversions. The Green and Yampa Rivers have similar headwaters, but the Yampa has minimal diversions or controls while Flaming George Dam on the Green river significantly affects flow. This provides two different flow regimes with similar large watersheds. In addition to flow data, I analyzed several time series data sets, including temperature and precipitation from Northeast Utah, North Western Colorado, and Southern Wyoming. These data are from the area that defines the Yampa River and Green River watersheds, which stretch from Flaming Gorge Dam to Ouray Colorado.

The EMD method is a relatively new technique that allows any time series data set, including non-linear and non-stationary datasets that are common in earth observation data, to be decomposed into a small quantity of composite finite data series, called intrinsic mode functions (IMFs). The EMD method can decompose any complicated data into several IMFs that represent independent signals in the original data. These IMFs may represent periodic forcing functions, such as environmental conditions or dam operations, or they may be artifacts of the decomposition method and not have an associated physical meaning. This study attempts to assign physical meaning to some IMFs resulting from the decomposition of the Green and Yampa flows where possible. To assign physical meaning to the IMFs, I analyzed frequencies of each IMF using the Hilbert-Hung transform, part of the Empirical Mode Decomposition method, and then compared frequencies of the IMFs with the known frequencies of physical processes. I performed these calculations on both flow, temperature, and precipitation.

I found significant correlation between IMF components of flow, precipitation, and temperature data with El Niño Southern Oscillation (ENSO) events. The EMD process also extracts the long-term trend in non-linear data sets that can provide insights into the effects of climate change on the flow system. Though in preliminary stages of research, these analysis methods may lead to further understanding the availability of water within the upper Yampa and Green River Watersheds.

Keywords: Empirical Mode Decomposition, Hilbert Haung Transforms, El Niño, La Niña, Yampa River, Yampa, Flaming Gorge, Green River

ACKNOWLEDGEMENTS

They say that we stand on the shoulders of giants; I would not be where I am in life without the sacrifices of people that came before me. My family came with nothing to this country and sacrificed everything so that I could have the opportunity to obtain an education. In return for my family's sacrifice, I have put all my effort into obtaining an education. I don't in the slightest believe this work is a payment for their sacrifice, but this work is a symbol of my appreciation for my family for what they have done for me.

My second vote of appreciation goes out to Dr. Gus P. Williams, my advisor and mentor. This work required many long days and late nights. I would never have completed this work without his direction. Dr. Williams allowed me to experience discovery first hand by not giving me the answers. He let me bounce ideas off of him; then once I had direction, he let me wrestle with the problem. After the work was complete, Dr. Williams still gave me advice on how to achieve my goals. I can't imagine a professor that cares more about the success of his students.

Finally, I'd like to thank Brent Hargis from the Engineer Research and Development Center (ERDC) for sponsoring this research. More importantly, I would like to thank him for the great work he did during his time at Brigham Young University. His research was invaluable for establishing the direction this project would take.

TABLE OF CONTENTS

ABSTRACT iii
TABLE OF CONTENTS iv
LIST OF TABLES vi
LIST OF FIGURES vii
1 Introduction 1
1.1 Problem Statement 1
1.2 Gaps in Research 2
1.3 Objectives 4
2 Background 6
2.1 El Niño Southern Oscillation Data 6
2.2 Proposed Method of Time Series Analysis 9
2.2.1 Empirical Mode Decomposition 9
2.2.1 Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) 11
2.2.2 Hilbert-Haung Transform 13
2.3 Common Methods of Environmental Time Series Data Analysis 13
2.3.1 Fourier Transforms 13
2.3.2 Spectrogram Method 14
2.3.3 Wavelet Transforms 14
2.4 Summary of Signal Analysis Methods 15
3 Software Resources Used 16
3.1 MATLAB 16
3.2 ArcGIS Pro 16
3.3 Excel 16
4 Implementation 17
4.1 Study Area 17
4.2 Data Collection 19
4.3 Missing Data Analysis 20
5 Matlab Processing 25
5.1 MATLAB Code 25
5.1.1 Data Readers 25
5.1.2 Data Processors 26
5.1.3 Data Plotters 27
6 Characterizing the Results 29
6.1 Discussion of Results 29

6.2 Discussion of Stream Discharge Results 29
6.3 Discussion of Temperature Results 34
6.4 Discussion of Precipitation Results 37
7 Conclusions 42
8 Future Work 43
8.1 Recommendations 43
References 45
Appendix A. Matlab Code 48
A.1 Custom Plotter 48
A.2 Data Interpolator 60
A.3 Data Normalizing Code 62
A.4 Master Code 64
A.5 Read Flow Data 76
A.6 Reader Nino Data 78
A.7 Read ONI Data 79
A.8 Read Precipitation Quarterly 80
A.9 Read Precipitation Data Yearly 83
A.10 Reader Temperature Data 86
A.11 Residual Code 89
Appendix B. Processed Datasets by Location and Data 98
B.1 Flaming Gorge Flow Gage 98
B.2 Jensen Flow Gage 104
B.3 Little Snake- Lily Flow 109
B.4 Yampa Maybel Flow 113
B.5 Ourey Quarterly Precipitation 118
B.6 Jensen Quarterly Precipitation 122
B.7 Ouray Yearly Precipitation 126
B.8 Jensen Yearly Precipitation 131
B.9 Jensen Temperature 136
B.10 Vernal Temperature 139
B.11 Combination 143

LIST OF TABLES

Table 2.3-1: A Summary of Different Signal Processing Methods and Descriptions 15
Table 4.2-1: Summary of Data 20
Table 4.3-1: Table of Missing Values 21
Table 5.1.1-1: The First Matrix Contains a Column for Gage Name, Filename, and the 3-Letter Acronym for the Gage 26
Table 5.1.1-2: Matrix 2 is Organized as a Column of Gage Name, Data Location, and the 3-Letter Acronym for the Gage to Ease Coding 26
Table 5.1.1-3: Data Reader Input File and Structure 26

LIST OF FIGURES

Figure 2.1-1 Precipitation Patterns Caused by El Niño and La Niña (NOAA, 2018)..... 8
Figure 2.1-2 Precipitation Patterns Caused by El Niño in the United States (NOAA, 2018)..... 8
Figure 2.2.1-1: Sifting/Decomposition Process Visualized (Kim & Oh, 2009) 10
Figure 2.2.1-1 Sample Code Creating Data Comparing Overshoot and Undershoot of the Spline Method to the PCHIP Method Which Does Not Under or Overshoot the Data Envelope..... 12
Figure 2.2.1-2 Plotting of Sample Data Comparing Overshoot and Undershoot of the Spline Method and How the PCHIP Method Does Not Over or Undershoot the Data Envelope..... 13
Figure 4.1-1 Overview of Study Within the Yampa River and Green River Watersheds 18
Figure 4.3.1-1: Complete Ouray Precipitation Dataset 22
Figure 4.3.1-2: Complete Ouray Precipitation Dataset Decomposition Showing Missing Data Gaps Being Transferred from the Original Dataset to the IMFs and Then the Residual . 23
Figure 4.3.1-3: Shortened Ouray Precipitation Dataset Decomposition Ignoring the 9 Years of Missing Data and Three Years of Corresponding Data Before the Gap..... 24
Figure 5.1.2-1 Mastercode, Generates the Associated IMFs and HHTs and Creates a Series of Labels Used in Plotting..... 27
Figure 5.1.3-1 Customplotter Input Settings..... 28
Figure 6.2-1: Flaming Gorge Stream Discharge Data..... 30
Figure 6.2-2: Flaming Gorge Stream Discharge Complete EMD..... 31
Figure 6.2-3: Flaming Gorge Stream Discharge IMF 7, IMF 8, IMF 9 Super Imposed and Compared to El Niño ONI (Bottom Panel)..... 32
Figure 6.2-4: Jensen Streamflow Discharge IMF 7, IMF 8 Super Imposed and Compared to El Niño ONI (Bottom Panel)..... 33
Figure 6.3-1: Vernal Temperature Dataset..... 34
Figure 6.3-2: Decomposed Vernal Temperature Dataset..... 35
Figure 6.3-3: Vernal Temperature Decomposition Compared to the ONI Dataset (Bottom Panel). 36
Figure 6.3-4: Residual of Jensen and Vernal Datasets..... 37
Figure 6.4-1: Jensen Yearly Precipitation Dataset. Every Point Is the Total Yearly Precipitation Depth..... 38
Figure 6.4-2: Jensen Precipitation Decomposition 39
Figure 6.4-3: Jensen IMF 2 Compared to the ONI Dataset (Bottom Panel)..... 40
Figure 6.4-4: Ourey and Jensen Precipitation Residuals..... 41
Figure B.1-1:Flaming Gorge Flow Empirical Mode Decomposition 98
Figure B.1-2:Flaming Gorge Flow HHT..... 99

Figure B.1-3:Flaming Gorge Flow Empirical Mode Decomposition IMF 7,8,9 and 10 100
Figure B.1-4:Flaming Gorge Flow Empirical Mode Decomposition Residual Superimposed on IMF 7 8 and 9 101
Figure B.1-5:Flaming Gorge Flow Empirical Mode Decomposition IMF 7 8 and 9 102
Figure B.1-6:Flaming Gorge Flow 103
Figure B.2-1:Jensen Flow Empirical Mode Decomposition 104
Figure B.2-2:Jensen Flow Empirical Mode Decomposition HHT 105
Figure B.2-3:Jensen Empirical Mode Decomposition IMF 7 and 8 106
Figure B.2-4:Jensen Empirical Mode Decomposition IMF 8 107
Figure B.2-5:Jensen Flow Data 108
Figure B.3-1:Little Snake Lily Empirical Mode Decomposition 109
Figure B.3-2: Little Snake Lily Empirical Mode Decomposition HHT 110
Figure B.3-3: Little Snake Lily Empirical Mode Decomposition IMF 7 111
Figure B.3-4: Little Snake Lily Flow 112
Figure B.4-1: Yampa Maybel Empirical Mode Decomposition 113
Figure B.4-2: Yampa Maybel Empirical Mode Decomposition HHT 114
Figure B.4-3: Yampa Maybel Empirical Mode Decomposition IMF 8 and 9 115
Figure B.4-4: Yampa Maybel Empirical Mode Decomposition IMF 8 116
Figure B.4-5: Yampa Maybel Flow 117
Figure B.5-1: Ourey Quarterly Precipitation Empirical Mode Decomposition 118
Figure B.5-2: Ourey Quarterly Precipitation Empirical Mode Decomposition HHT 119
Figure B.5-3: Ourey Quarterly Precipitation Empirical Mode Decomposition IMF 1 120
Figure B.5-4: Ourey Quarterly Precipitation 121
Figure B.6-1: Jensen Quarterly Precipitation Empirical Mode Decomposition 122
Figure B.6-2: Jensen Quarterly Precipitation Empirical Mode Decomposition HHT 123
Figure B.6-3: Jensen Quarterly Precipitation Empirical Mode Decomposition IMF 1 124
Figure B.6-4: Jensen Quarterly Precipitation 125
Figure B.7-1: Ouray Yearly Precipitation Empirical Mode Decomposition 126
Figure B.7-2: Ouray Yearly Precipitation Empirical Mode Decomposition HHT 127
Figure B.7-3: Ouray Yearly Precipitation Empirical Mode Decomposition IMF 1 128
Figure B.7-4: Ouray Yearly Precipitation 129
Figure B.7-5: Ouray Yearly Precipitation Empirical Mode Decomposition Shortened to Account for Missing Data 130
Figure B.8-1: Jensen Yearly Precipitation Empirical Mode Decomposition 131
Figure B.8-2: Jensen Yearly Precipitation Empirical Mode Decomposition HHT 132

Figure B.8-3: Jensen Yearly Precipitation Empirical Mode Decomposition IMF 2 and 3 133
Figure B.8-4: Jensen Yearly Precipitation Empirical Mode Decomposition IMF 2 134
Figure B.8-5: Jensen Yearly Precipitation 135
Figure B.9-1: Jensen Average Temperature Empirical Mode Decomposition 136
Figure B.9-2: Jensen Average Temperature Empirical Mode Decomposition HHT 137
Figure B.9-3: Jensen Average Temperature 138
Figure B.10-1: Vernal Average Temperature Empirical Mode Decomposition 139
Figure B.10-2: Vernal Average Temperature Empirical Mode Decomposition HHT 140
Figure B.10-3: Vernal Average Temperature Empirical Mode Decomposition IMF 9 and 10 141
Figure B.10-4: Vernal Average Temperature 142
Figure B.11-1: Flow Residuals 143
Figure B.11-2: Quarterly Precipitation Residuals 144
Figure B.11-3: Yearly Precipitation Residuals 145
Figure B.11-4: Temperature Residuals 146

1 INTRODUCTION

1.1 Problem Statement

Some environmental processes create signals that can be observed in other processes that they influence. For example, analysis has shown that there is correlation of the El Niño Southern Oscillation (ENSO) process with precipitation and temperature datasets collected from gage data (Norden Eh Huang, 2014), that is the signal from the ENSO process can be observed in both precipitation and temperature signals. In the past to separate and identify such signals, scientists have analyzed data using wavelet analysis, and Fourier transforms (Norden E Huang et al., 1998). These methods, when applied to environmental data, have difficulties analyzing the non-linear, non-periodic signals that commonly occur in nature. Recently, researchers have extracted signals that correlate to the ENSO signal from precipitation data using a new analysis method called Empirical Mode Decomposition (EMD) (el-Askary, Sarkar, Chiu, Kafatos, & El-Ghazawi, 2004). This method works on signals that are non-linear and non-stationary (Norden E Huang et al., 1998). I applied Empirical Mode Decomposition (Norden E Huang et al., 1998) to extract and analyze trends and periodic processes in several environmental datasets including daily streamflow, daily precipitation, and daily temperature on two rivers in the Upper Colorado River Basin. I analyzed the ability of this technique to extract signals even after Flaming Gorge Dam, a very large dam on the Green River, significantly altered the processes.

The Empirical Mode Decomposition method is a relatively new empirical signal processing technique that allows any time series data set, including non-linear and non-stationary datasets that are common in earth observation data, to be decomposed into a small quantity of composite finite data series called intrinsic mode functions (IMFs) (Norden E Huang et al., 1998). The EMD process also extracts a long-term trend in non-linear data sets that can provide insights into the effects of climate change or other long-term processes on the flow system. These trends and individual IMFs may or may not represent any physical processes.

The algorithm EMD uses is simple and does not assume stationary or linear signals. First, all local extrema are identified and a function to define the upper and lower envelopes is generated by interpolating between the local maxima and minima. Next, a temporary signal is created by interpolating an average of the upper and lower envelopes at every data interval (Kim & Oh, 2009). Next, the temporary signal is subtracted from the original dataset to obtain a partial IMF. The process is repeated until the number of extrema and zero-crossings differ by a maximum of one, the local average of the resulting equation must be zero. Once both requirements are satisfied, the partial IMF becomes a complete IMF, and the remaining data becomes a remainder. The IMF is then subtracted from the data set, and the processes is repeated. This process is known as sifting. I include a more comprehensive description in section 2.2.

1.2 Gaps in Research

Time series data can be used to analyze and understand the behavior of a processes over time using various signal processing techniques. Sometimes signal processing can reveal

valuable information by decomposing time series dataset into subsets. These subsets can identify independent processes or isolate and characterize the effects of a smaller component.

Historically, researchers have commonly used several tools for processing periodic signals including spectrograms, wavelet analysis, and the Fourier analysis. The resulting components from these analyses can sometimes represent simple real processes that can be combined to create the original signal. In other cases, these components, which are simpler than the original signal, do not represent any real processes, but are just artifacts of the mathematical decomposition. Many of these methods, such as wavelets or Fourier transform methods, require a number of conditions be met for application. Common conditions are linearity and stationarity. Each of these methods make assumptions that are not always valid when analyzing physical data (Norden E Huang et al., 1998).

Some of the challenges that arise when attempting to analyze physical data is that most real periodic environmental data are non-linear and non-stationary. Non-linear signals are defined as signals described by differential equations with time varying coefficients. Linear signals have constant coefficients. For example, the El Niño Southern Oscillation (ENSO) events have a periodic structure, but the structure period changes from 5 to 7 years on a seeming random basis. Non-stationary datasets have averages, variances, or covariances that change over time. Most environmental data sets have an underlying trend that makes them non-stationary.

EMD is a signal processing method that can process non-linear and non-stationary data and exactly recreates the original signal when the resulting component IMFs are summed. This is in contrast to approaches such as wavelets or Fourier analysis that require linear and stationary signals and only approximate the original signal when summing the resulting components. These methods, wavelets or Fourier analysis, can be only used on non-linear and non-stationary signals

by using a large number of components to represent the non-linear or non-stationary components over some finite signal length. This representation is only approximate, meaning the recreated signal, usually created by sum of the decomposed signal components, will not reproduce the original signal exactly.

By using EMD, environmental data sets with non-linear and non-stationary data can be processed. Most environmental data sets are non-stationary, that is they exhibit trends, and do not always meet other statistical assumptions required for non-empirical analysis. Because of the relaxed assumptions, EMD is a more mathematically sound approach to process environmental signals. However, a gap in research lies in the fact that, due to the relatively young age of EMD, signal processing with this method is not widely applied to streamflow or precipitation data.

1.3 Objectives

I evaluated whether I could use EMD methods to quantify and characterize signals from different environmental forcing functions and processes and determine if the signals were present in flow data from different locations that have similar watersheds but different amounts of anthropogenic control. Specifically, I will explore whether the EMD method can be used to analyze river flow data and identify various forcing functions such as dam operations, El Niño Southern Oscillation (ENSO), and also determine if the same periodic processes present in the flow data can be identified in local temperature or precipitation data. I evaluated whether processes like large reservoirs obscured or hid signals from nature processes, such as ENSO, or whether these signals were present in the flow data below the dam.

I selected the Green and Yampa Rivers as they have similar headwaters, but the Yampa has minimal diversions or controls while Flaming George Dam on the Green River significantly

affects flow. This provides flow data from two different river regimes with similar large watersheds.

The objective of this study is to process several datasets, including flow from two rivers, one controlled and one mostly wild, along with temperature and precipitation records from the area comprising the watersheds of these two rivers. I will extract long-term trends and attempt to identify signals from natural processes. I will explore whether this method can be used to separate the effects of dams and other man-made regulatory structures from environmental processes. Once I have decomposed the original signals into independent, periodic IMFs, I will attempt to associate physical meaning, or physical processes, with selected IMFs. This may be difficult, since periodic processes within individual IMFs may or may not have physical meaning (Norden E Huang et al., 1998). However, if signals can be identified across the two rivers or different data sets, it is more likely that the IMF represents an actual process, either natural or anthropogenic

2 BACKGROUND

2.1 El Niño Southern Oscillation Data

The El Niño Southern Oscillation is an environmental phenomenon caused by bands of warm water that occur in the Pacific Ocean between the International Date Line and 120°W (Service, 2019). Every few years, El Niño affects the likelihood of floods in the western coast of South America, as well as the likelihood of drought in South East Asia. This climate pattern repeats on an approximate five- to seven-year period, and affects precipitation, and thus river flow, in western North and South America.

To understand what El Niño is, one must understand the normal climate pattern in the Southern Pacific Ocean. The winds in the Pacific Ocean travel from South America in the direction of Australia, this is a phenomenon known as the South East Trade winds, which occur between 30° latitude and the equator. The area is known for consistent winds of about 11 to 13 miles per hour. The South East Trade Winds are named after the ability to quickly move trading ships across the ocean and are caused by the Coriolis effect. As wind moves from east to west, the shear force moves the warm water near the ocean surface piling it on the western side of Australasia. Near South America this creates an upwelling, the rising of seawater in this upwelling region causes the cooler water to rise to the surface and replace the warm water being transported. This creates a temperature gradient where warm surface water is in Australasia, and cool surface water is near South America. The warm water increases air temperature, resulting in

rising air, cloud formations, and ultimately more rainfall. This pattern reinforces the wind circulations because warm moist air will rise in the west and cool, drier, air will descend near the Eastern part of the Pacific Ocean causing a steady state scenario. This is the normal weather pattern, with moisture being drawn away from the Americans, creating drier climate in the Americas (Trenberth, 1997) (Pizarro & Lall, 2002).

El Niño occurs when the trade winds are weakened. As trade winds are weakened, there is less transport of warm water, and less upwelling of cool water. Therefore, the centroid of the warm water moves eastward toward South America until the temperature gradient is eliminated, ultimately changing the precipitation and wind patterns in the Pacific Ocean. This results in increased precipitation in western North and South America.

La Niña occurs when trade winds are strengthened, moving or containing the warm water at the western part of the tropical Pacific; this process causes an increase of upwell in the Pacific Ocean. Air will rise over this warm area, but will dump precipitation near the center of the ocean. La Niña is the opposite of El Niño, in the fact that it will increase drought probability in South America, and flood probability in Australasia as shown in Figure 2.1-1 (NOAA, 2018).

The change of wind pattern can affect the likelihood of floods due to precipitation within the United States as shown in Figure 2.1-2 (NOAA, 2018).

The goal of my research was to determine if I could use EMD to analyze the environmental datasets and extract the signals that characterize the El Niño or La Niña events from within these periodic environmental processes, river flow, rainfall, and temperature in the presence of man-made control structures such as a large dam.

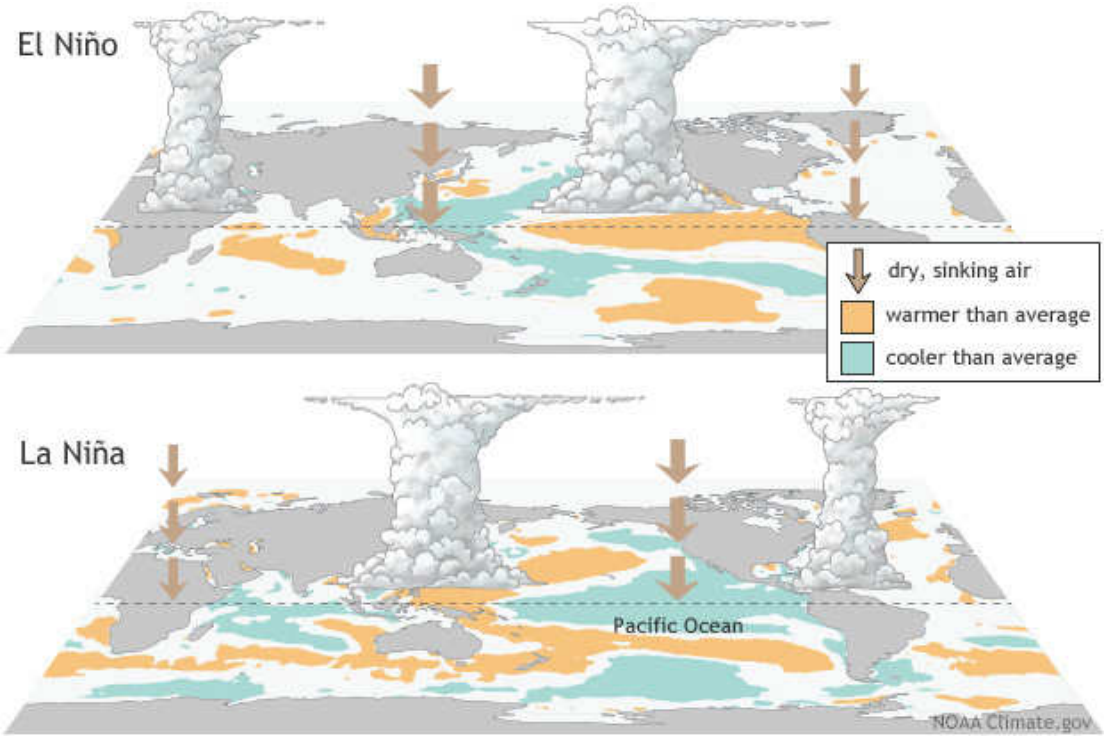

Figure 2.1-1 Precipitation Patterns Caused by El Niño and La Niña (NOAA, 2018)

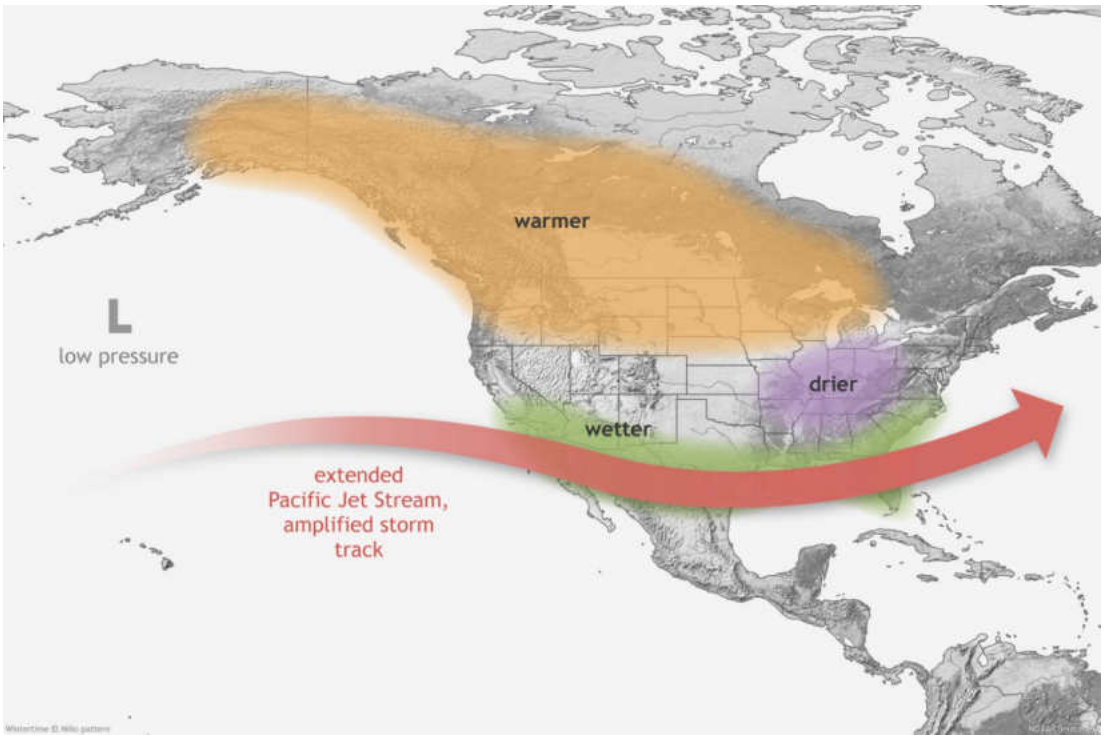

Figure 2.1-2 Precipitation Patterns Caused by El Niño in the United States (NOAA, 2018)

Quantitatively, El Niño and La Niña events are measured and standardized by the National Oceanic and Atmospheric Administration (NOAA) using the Oceanic Niño Index (ONI). The ONI is a running 3-month sea surface temperature (SST) mean in the Niño 3.4 region which is located at 5ºN-5ºS, 120º-170ºW. Values are based on a centered 30-year normal SST which is updated every 5 years. El Niño events are defined as 5 consecutive ONI periods with a value of +0.5º while La Nina events have a 5 consecutive ONI periods with a value of -0.5º. The SST measurements are obtained between 1 millimeter and 20 meters below the sea surface. The ONI measurements begin in December so that the first value will be centered in January (Service, 2019).

2.2 Proposed Method of Time Series Analysis

2.2.1 Empirical Mode Decomposition

The EMD method is a relatively new empirical technique that allows any time series data set, including non-linear and non-stationary datasets, that are common in earth observation data, to be decomposed into a small quantity of composite finite data series called IMFs (Norden E Huang et al., 1998). The EMD process also extracts a long-term trend in non-linear data sets that can provide insights into the effects of climate change or other long-term processes on the flow system. These trends may or may not represent any physical processes. The algorithm EMD uses is simple and does not assume stationary or linear signals.

In the EMD algorithm, first all local extrema are identified, both local maximums and minimums (Figure 2.2.1-1a). In the next step, the algorithm interpolates a curve through the local maxima or minima to create the upper and lower envelopes (Figure 2.2.1-1b). For this task I used the piecewise cubic hermite interpolating polynomial (PCHIP) algorithm to interpolate

the local maxima and minima. Next, the average of these two envelopes is computed, by interpolating the upper and lower envelopes at every data interval (Figure 2.2.1-1c). This average is subtracted from the original dataset to obtain a partial IMF (Figure 2.2.1-1d). The process is repeated until the number of extrema and zero-crossings differ by one and the local average is zero (Figure 2.2.1-1e). Once both requirements are satisfied, the partial IMF becomes a complete IMF, and the remaining data becomes a residual and the process starts again (Figure 2.2.1-1f). This process is known as sifting.

After an IMF is created, it is subtracted from the original data and the process is repeated on the remaining data until the stopping rule is satisfied. The stopping rule is a convergence criterion that stops sifting when the current relative tolerance is less than the specified tolerance or a maximum number of IMFs are created. Once the stopping rule is satisfied, the remainder becomes the residual, which describes the dataset's long-term trend.

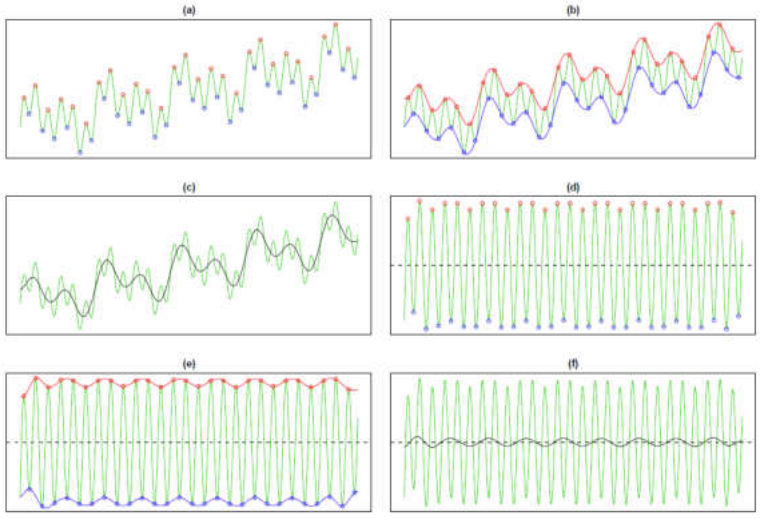

Figure 2.2.1-1: Sifting/Decomposition Process Visualized (Kim & Oh, 2009)

I applied this technique to flow, precipitation, and temperature gage datasets within the area of interest to highlight trends, and attempted to assign a physical explanation to constant frequencies within oscillations. To ensure accurate results, I included only data sets with over 60 years of data in these calculations.

When EMD is applied to environmental data without a boundary condition, there are a myriad of outcomes based on the sifting tolerance. Due to human influence causing drastic changes in river flow, I used data from the stream gage at Flaming Gorge to select a value of 0.2 to use as the sifting tolerance. This tolerance of 0.2 is also suggested by Dr. Haung (Norden E Huang et al., 1998). Setting the sifting tolerance too low will overdecompose the signal making it likely to lose any physical meaning, and making the tolerance too high will superimpose signals that potentially are different (Norden E Huang et al., 1998). Haung suggests this value normally be between 0.2 and 0.3, therefore the value I selected is in the recommended bounds. I selected this tolerance value because the resulting residual most accurately depicted historical markers such as the filling of Flaming Gorge Reservoir in 1963, the move from peak hour flooding to constant flow for power generation in 1992, and the move to a more natural flow pattern that increased river flow in 2006. I did not set a limit on the number of IMFs created, letting the algorithm run to completion.

2.2.1 Piecewise Cubic Hermite Interpolating Polynomial (PCHIP)

The EMD code requires an interpolation method to generate the data envelopes. I used PCHIP during IMF generation. Most EMD code uses cubic spline interpolation, but for environmental data, cubic spline methods can result in significant under and overshoot in the interpolated data. The results of PCHIP are a shape-preserving piecewise cubic interpolation

that does not overshoot or undershoot as other data interpolators. A simple example of why this interpolator was chosen over others is shown in Figure 2.2.1-1 and Figure 2.2.1-2.

```
1 x = -3:3;
2 y = [-1 -1 -1 0 1 1 1];
3 xq1 = -3:.01:3;
4 p = pchip(x,y,xq1);
5 s = spline(x,y,xq1);
6 plot(x,y,'o',xq1,p,'-',xq1,s,'-.')
7 legend('Sample Points','pchip','spline','Location','SouthEast')
```

Figure 2.2.1-1 Sample Code Creating Data Comparing Overshoot and Undershoot of the Spline Method to the PCHIP Method Which Does Not Under or Overshoot the Data Envelope

Line 1 of the code in Figure 2.2.1-1, sets a value of x from -3 to 3 by a step of 1, while line 2 is a set of random y values. The variable xq1 is a query point for our interpolation, or the x-coordinates at which Matlab will apply the interpolation function. Figure 2.2.1-2 shows overshoot from the spline interpolation from -3 to -2 and from 1 to 2. The spline interpolation experiences undershoot from -2 to -1 and from 2 to 3. However, the PCHIP interpolation connects our data without any visible overshoot or undershoot. Overshoot and undershoot not only introduce error, but can obscure physical meaning in datasets such as flow where negative flow has no meaning.

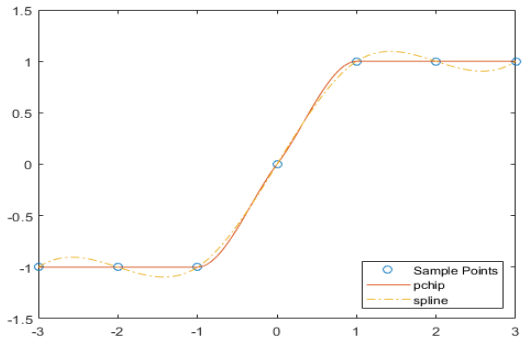

Figure 2.2.1-2 Plotting of Sample Data Comparing Overshoot and Undershoot of the Spline Method and How the PCHIP Method Does Not Over or Undershoot the Data Envelope

2.2.2 Hilbert-Haung Transform

The Hilbert-Haung Transform (HHT) is a computational method used to solve for the instantaneous frequency changes within one oscillation cycle (Norden Eh Huang, 2014). The HHT gives insight to the dominant signal frequency while allowing a non-constant phase. Paired with the EMD, the HHT analyzes data by processing the IMF subsets through the HHT algorithm. Like EMD, the HHT method can be used on non-linear data, which allows the method to be adaptive. However, the adaptive nature of the HHT makes it difficult to describe theoretically (Norden Eh Huang, 2014). The HHT can be used to generate frequency plots similar to spectrograms for IMF data.

2.3 Common Methods of Environmental Time Series Data Analysis

2.3.1 Fourier Transforms

The Fourier Transform is the most common transformation for time series data analysis. This transformation method assumes that any signal can be transformed into a summation of

sinusoidal functions (Bracewell & Bracewell, 1986). These sinusoidal functions have a stationary basic period with time independent frequency. This means that the Fourier Transform is only able to process stationary data. Perhaps its greatest limitation of the Fourier Transform is that approximations are used to model datasets, using sinusoidal functions. When the sinusoidal decompositions are used to reconstruct the original signal, the result contains approximation errors and does not exactly represent the original dataset.

2.3.2 Spectrogram Method

A spectrogram method of analysis uses the Fourier decomposition as its base. It implements a limited time window-width Fourier spectral analysis (Stankovic, 1994) (Norden E Huang et al., 1998). By continuously plotting frequencies along the time axis, you can compute and visualize a time-frequency distribution. Spectrograms are often used to visualize time series data, emphasizing amplitude, intensity, and overtones.

Spectrogram analysis relies on the same assumption of Fourier spectral analysis: that data are piecewise stationary. This implies that the window of analysis perfectly captures stationary time scales. However, this assumption is violated when analyzing non-stationary, non-linear signals.

2.3.3 Wavelet Transforms

The Wavelet Transform is another relatively recent signal processing method. Similar to a Spectrogram, the wavelet analysis uses an adjustable window that fits a repeating function to the data at different scales, similar to the way Fourier analysis fits sine functions at different scales (Labat, 2008). This approach is useful for analyzing gradual changes in frequency within data as, unlike Fourier analysis, wavelets can represent these frequencies changes. Most

applications of the Wavelet Transforms have been used in image compression and edge detection (Norden E Huang et al., 1998). Though there have been applications to river flow (Coulibaly & Burn, 2004; Melesse, Abtew, Dessalegne, & Wang, 2010; Smith, Turcotte, & Isacks, 1998).

Wavelet analysis has difficulty adapting over time for environmental data, as it is not designed for non-linear data. Because the wavelet analysis is based on a summation of functions, it inherits some of the shortcomings of the Fourier Transform Analysis.

2.4 Summary of Signal Analysis Methods

A summary of the different transform methods can be seen in Table 2.3-1 (Norden E Huang et al., 1998).

Table 2.3-1: A Summary of Different Signal Processing Methods and Descriptions

	Fourier	**Wavelet**	**Hilbert**
Basis	*a priori*	*a priori*	adaptive
Frequency	convolution: global uncertainty	convolution: regional uncertainty	differentiation: local, certainty
Presentation	energy-frequency	energy-time-frequency	energy-time-frequency
Nonlinear	No	No	Yes
Nonstationary	no	yes	yes
Feature Extraction	no	discrete: no; continuous: yes	yes
Theoretical base	theory complete	theory complete	empirical

3 SOFTWARE RESOURCES USED

3.1 MATLAB

MATLAB® is a high level programing language and numerical analysis environment developed by Mathworks (MathWorks, 2019). MATLAB is commonly used by engineers and scientists to develop algorithms to solve mathematics problems. I chose MATLAB as the primary software for developing the EMD code due to its built-in Empirical Mode Decomposition functions released with the MATLAB 2018R package.

3.2 ArcGIS Pro

ArcGIS is a geographic information system (GIS) developed by ESRI (ESRI, 2019). Its primary application in this research was to compile geospatial information such as gage locations and gage types, and visually present it as a map.

3.3 Excel

Excel is a spreadsheet program developed by Microsoft as part of the Microsoft Office Suite (Microsoft, 2019). The program is arranged in rows and columns that be used to input data and manipulate those data mathematically. I used Excel primarily to organize data before inputting the datasets into MATLAB.

4 IMPLEMENTATION

4.1 Study Area

For this project, I analyzed data from several stream gages along the Green and Yampa Rivers. Data were provided by the United States Geological Survey (USGS) (Survey, 2019). My original selection of gages was: Flaming Gorge, Green River near Jensen Utah, Green River near Ouray, Yampa near Deer Lodge Colorado, Little Snake near Lily Colorado, Yampa, near Maybel Colorado, Yampa near Craig Colorado, Yampa above Elk Head near Hayden Colorado and Yampa river near Elk Head (Figure 4.1-1). I selected these gages because they characterize flow in the Green and Yampa Rivers, and flows in the Green River after the confluence where the Yampa joins the Green. As noted, these rivers have similar headwaters, but the Yampa has minimal diversions or controls while Flaming George Dam on the Green River significantly affects flow. This provides two different flow regimes with similar large watersheds. The reason for selecting these gages is to evaluate if the rivers, before joining, exhibit different periodic processes, if these processes can be correlated with periodic processes extracted from the temperature or precipitation data, and if these processes can be extracted from the data after the confluence of the two rivers. I targeted the El Niño processes with a period of 5 to 7 years to determine if I could identify these processes in the data from these gages and if these signals could be tracked downstream.

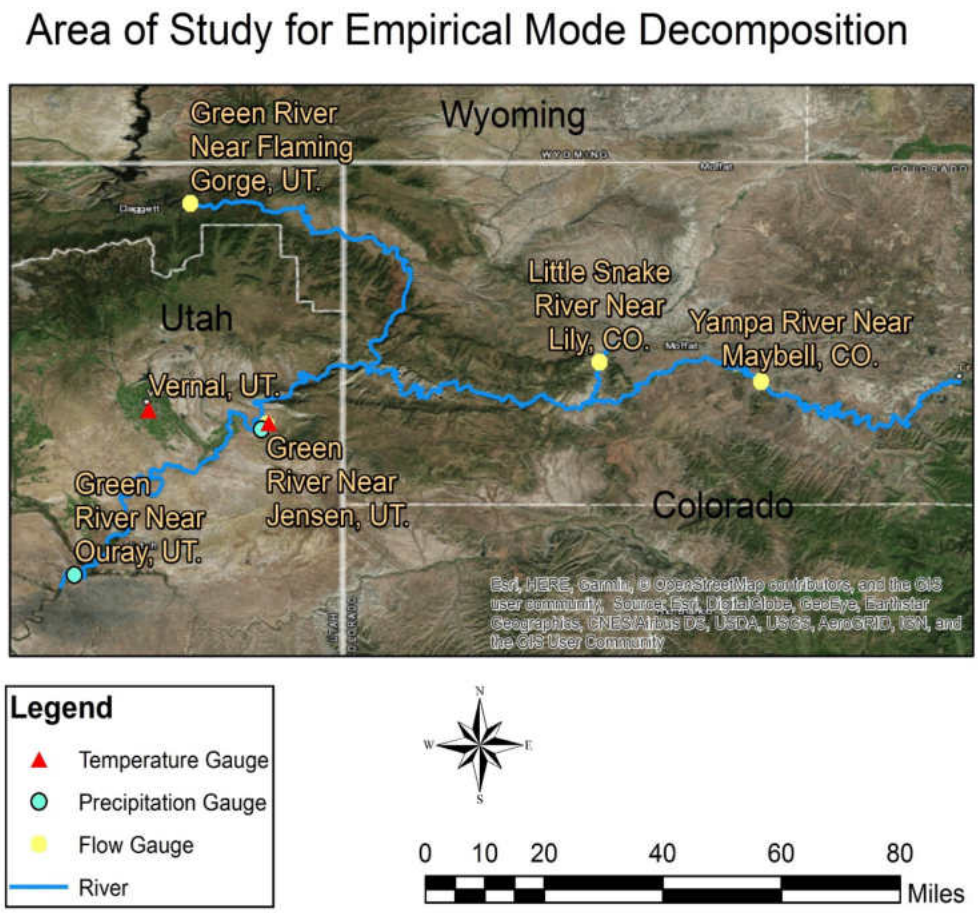

Figure 4.1-1 Overview of Study Within the Yampa River and Green River Watersheds

I also analyzed precipitation data from gages within the watershed at Flaming Gorge, Jensen, Ouray, Maybel, and Craig (Figure4.1-1). I selected precipitation data from meteorological stations as close to the flow area of study as possible. Precipitation data were obtained through the Utah Climate Center, which provided the Vernal and Ouray datasets (Center, 2019).

Another interesting set of data that I analyzed was air temperature. Temperature was more difficult to obtain because there are not many temperature gages located in my study area. I selected temperature gages at Vernal, Utah; Grand Junction, Colorado; Jensen, Utah; and Flaming Gorge Dam, Utah (Center, 2019).

4.2 Data Collection

Stream discharge data were obtained from the USGS database (Survey, 2019). Precipitation and Temperature data were obtained from the Utah Climate Center of Utah State University (Center, 2019). For a dataset to be included in my analysis, they needed to meet a set of criteria. I developed these criteria based on previous research to ensure data quality data and to minimize missing data (Hargis, 2014).

Criteria for dataset (e.g., site) selection: Each site must have

- Approximately 60 years or more of data available
- Daily data
- Must be from a valid or accredited source

Discharge gage data are presented as daily averages, in cubic feet per second. Temperature data were obtained as daily max and minimum and calculated as daily average in degrees centigrade. Precipitation data were obtained as daily average values, but I calculated yearly precipitation data in millimeters per year to provide mathematical continuity during decomposition. A summary of the datasets is provided in Table 4.2-1.

Managing and choosing a precipitation data reduction method was challenging. Before I chose total yearly precipitation depth as my method for processing, I used total quarterly precipitation to satisfy the continuity requirement of EMD. When I plotted and analyzed the data using EMD, this method did not show the expected results due to rapidly changing crests and troughs in the signal. This was because the precipitation data were so sparse and precipitation in a given quarter could be very low and the data can appear to be noisy. Results from the quarterly

EMD datasets are shown in Appendix B. Based on this preliminary effort, I used total yearly precipitation as the reduction basis for this data set.

Table 4.2-1: Summary of Data

Dataset	Gage Name	Beginning Date	Ending Date	Appendix
Stream Discharge	Flaming Gorge Flow	10/1/1946	3/15/2019	B.1
	Jensen Flow	10/1/1946	5/16/2018	B.2
	Little Snake Lily Flow	10/1/1921	5/2/2018	B.3
	Yampa Maybel Flow	5/1/1916	5/21/2018	B.4
Yearly Precipitation	Ouray Yearly Precipitation	6/1/1941	5/31/2018	B.7
	Jensen Yearly Precipitation	3/27/1925	5/31/2018	B.8
Temperature	Jensen Temperature	10/1/1939	6/15/2018	B.9
	Vernal Temperature	7/27/1900	4/30/2018	B.10

4.3 Missing Data Analysis

For the EMD algorithm to properly work, continuous data must be provided. However, there were several gaps from missing data in the temperature and precipitation datasets. Previous work, showed that missing values with gap periods up to a year have little impact on the overall EMD/IMF interpolation (Hargis, 2014). For missing daily values in the temperature dataset, I linearly interpolated between missing days. Missing daily values in the precipitation dataset were assumed to be zero, when entire years were missing, the yearly precipitation was linearly interpolated. Linear interpolation is the simplest and often least accurate method for replacing these data, however, previous research showed that these shorter gaps, less than 1-year, have

limited impact when filled with linear data. Because of these minimal impacts, I selected linear interpolation for the ease of implementation. Interpolation introduces error into the data sets because the interpolated data do not represent reality, however, since the signals I am attempting to analyze have a period of 5 to 7 years, small gaps in data do not affect the analysis. Table 4.3-1 describes the amount of missing data within this study.

Table 4.3-1: Table of Missing Values

Dataset	Gage Name	Beginning Date	Ending Date	Missing Values	Completeness
Stream Discharge	Flaming Gorge Flow	10/1/1946	3/15/2019	0	100%
	Jensen Flow	10/1/1946	5/16/2018	0	100%
	Little Snake Lily Flow	10/1/1921	5/2/2018	0	100%
	Yampa Maybel Flow	5/1/1916	5/21/2018	0	100%
Yearly Precipitation	Ouray Yearly Precipitation	6/1/1941	5/31/2018	5329	81%
	Jensen Yearly Precipitation	3/27/1925	5/31/2018	3659	89%
Temperature	Jensen Temperature	10/1/1939	6/15/2018	557	98%
	Vernal Temperature	7/27/1900	4/30/2018	4035	91%

The IMF's show the gap, with each subsequent IMF also showing the gap from the original dataset. Essentially, then analyzing the resulting IMFs, the values in the gap regions are not valid, these effects extend a little into the actual data.

I compared IMFs resulting from decomposing the full data set (large gap at the beginning) and a shortened data set (does not include the gaps). Figure 4.3.1-3 shows the IMFs that result from decomposing a shortened Ouray dataset that does not include the first three years of data or the 9 years of missing data. Figure 4.3.1-3 shows, by comparing the long dataset and the

shortened dataset, several differences. First, the shape of the IMF near the missing data is affected as seen in IMF 3 of the shortened dataset. Next, the magnitudes vary slightly, which can be seen by comparing IMF 2 and 3 of the long and short datasets. Additionally, the rate of change and concavity of the two residuals differ. The concavity of the long dataset residual may suggest that an event occurred causing a peak in the data. The short dataset would not mislead researchers to make that assumption.

Overall, I suggest that datasets not include early data if large gaps exist since the mathematical error introduced into the IMFs may not be worth the few more years of additional data. If gaps bisect a dataset, a method of interpolation could be a viable solution to save physical meaning (Nelsen, Williams, Williams, & Berrett, 2018). While I used linear interpolation, since my gaps were relatively small compared to the periods I was interested in, more advanced data imputation methods could be used.

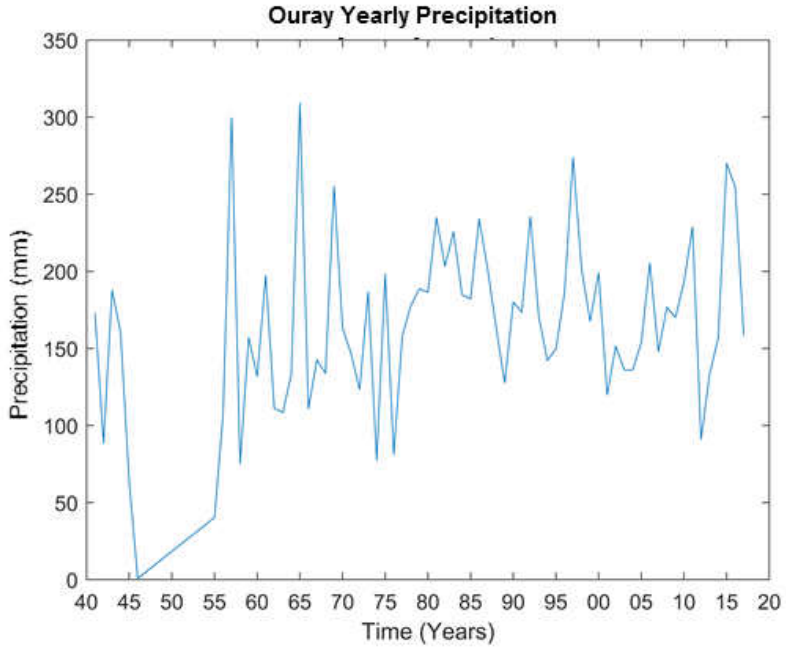

Figure 4.3.1-1: Complete Ouray Precipitation Dataset

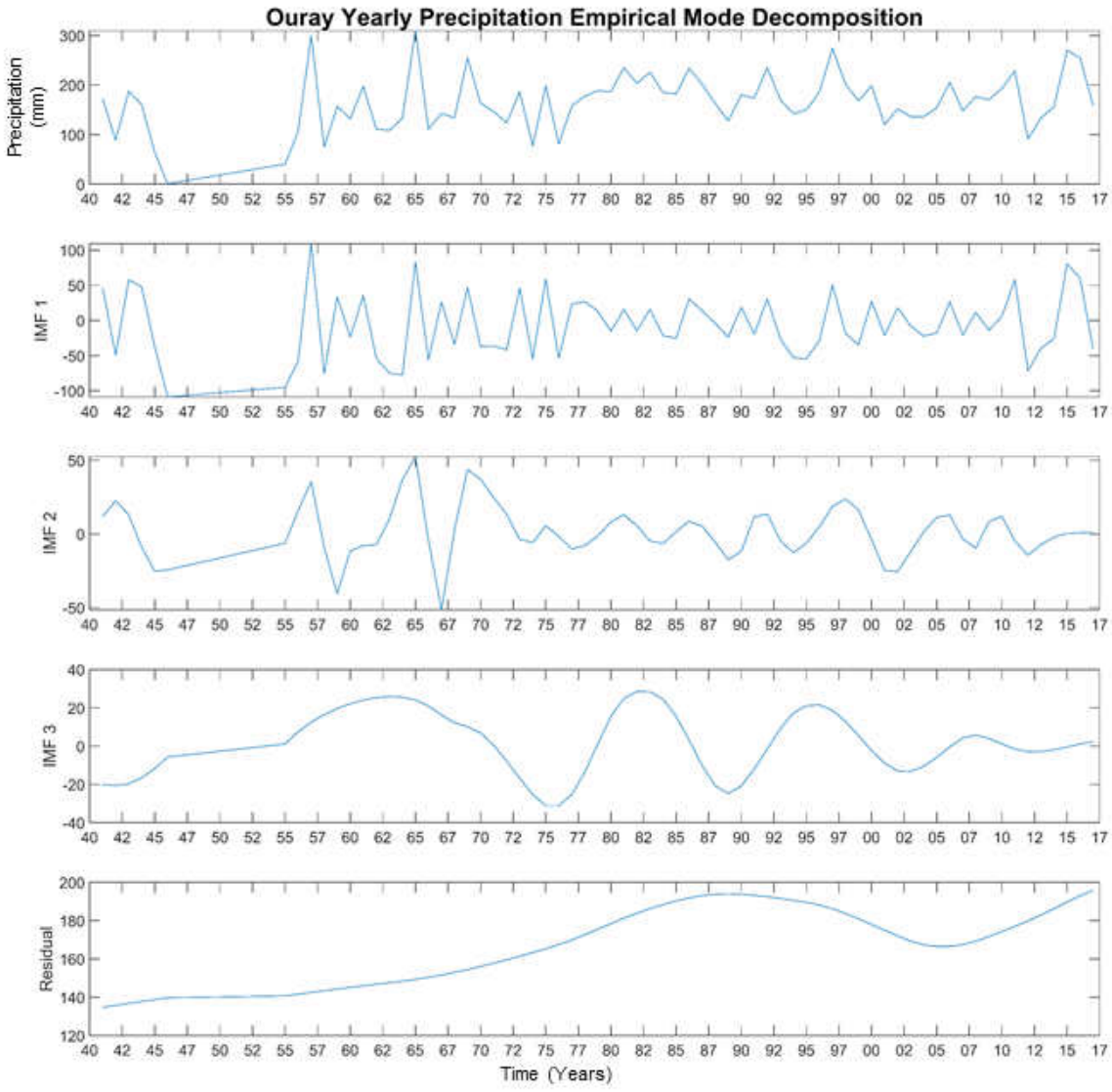

Figure 4.3.1-2: Complete Ouray Precipitation Dataset Decomposition Showing Missing Data Gaps Being Transferred from the Original Dataset to the IMFs and Then the Residual

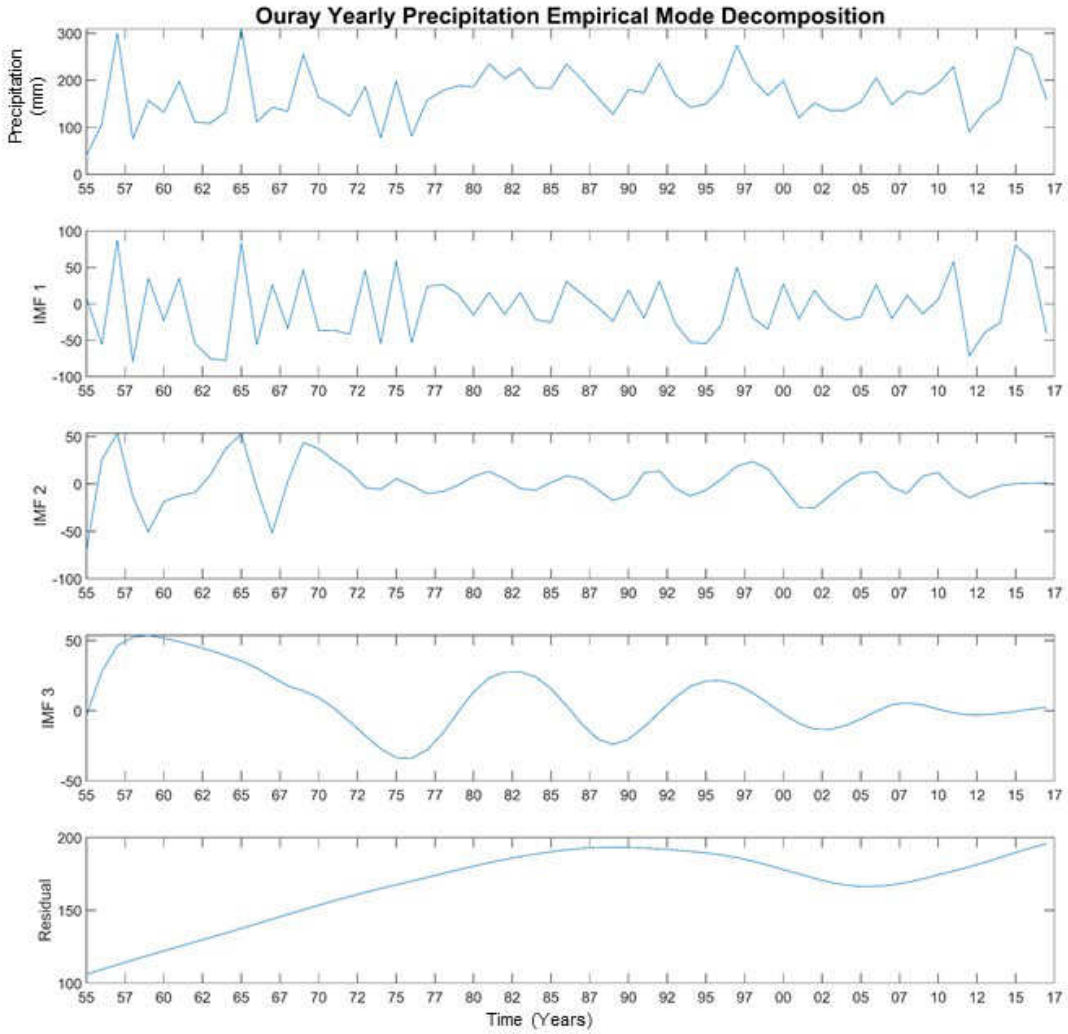

Figure 4.3.1-3: Shortened Ouray Precipitation Dataset Decomposition Ignoring the 9 Years of Missing Data and Three Years of Corresponding Data Before the Gap.

5 MATLAB PROCESSING

5.1 MATLAB Code

I created custom MATLAB functions for data processing and analysis. I created MATLAB scripts to customize figures and generate data reports. This code is contained in the appendix. The MATLAB scripts and code are generally divided into three groups: Data Readers, Data Processors, and Plotters.

5.1.1 Data Readers

The Data Readers are the first part the EMD analysis process. The Data Reader scripts convert previously prepared Excel files (.csv) into MATLAB Data (.mat) files. MATLAB data files, significantly speedup IMF and HHT processing since MATLAB can load a ".mat" file much faster than reading a ".csv" file.

First, the Data Reader creates two string matrices containing names of the data used in the labeling and storage of data; a sample of the organization of each vector is shown in figure 5.1.1-1 and figure 5.1.1-2. The first string matrix is containing names of gages and file names. The second matrix contains gage names and file locations used for loading and reading data.

Next, the Data Reader references the string vectors to create a data structure which stores dates and associated measurements (Figure 5.1.1-3). In addition to the measurement data, this

data structure will store the IMFs and Residual data after processing. The complete Data Reader code is located in the appendix.

Table 5.1.1-1: The First Matrix Contains a Column for Gage Name, Filename, and the 3-Letter Acronym for the Gage

String Vector 1		
Column1	**Column2**	**Column3**
Long Name	File name	Shortened Code Name

Table 5.1.1-2: Matrix 2 is Organized as a Column of Gage Name, Data Location, and the 3-Letter Acronym for the Gage to Ease Coding

String Vector 2		
Column1	**Column2**	**Column3**
Long Name	Exact Data Location	Shortened Code Name

Table 5.1.1-3: Data Reader Input File and Structure

Data Structure			
Column1	**Column2**	**Column3**	**Column4**
Dates	Associated Data	IMF	Residual

5.1.2 Data Processors

The Data Processor, located in a MATLAB script called “MasterCode”, implements the EMD processing and generates the associated IMFs and HHTs. The code loads the previously

made ".mat" files, and creates a series of labels used for plotting (lines 11-16). Line 24 creates a matrix containing labels for plotting. The code then creates a series of plots. These plots are used as a check in order to see that loaded data matches the data from the Excel files. The code then decomposed each time series into IMFs and a Residual. These results are stored in the Data Structure and associated with the gage site and data dates. Next, the code analyzes the IMF and Residual signals with the HHT to generate time-frequency plots. Lines 44-70 create the IMFs, and HHTs for discharge data. I repeated this three-step process for each of the four data sets: Stream Discharge, Temperature, Quarterly Precipitation, and Yearly Precipitation.

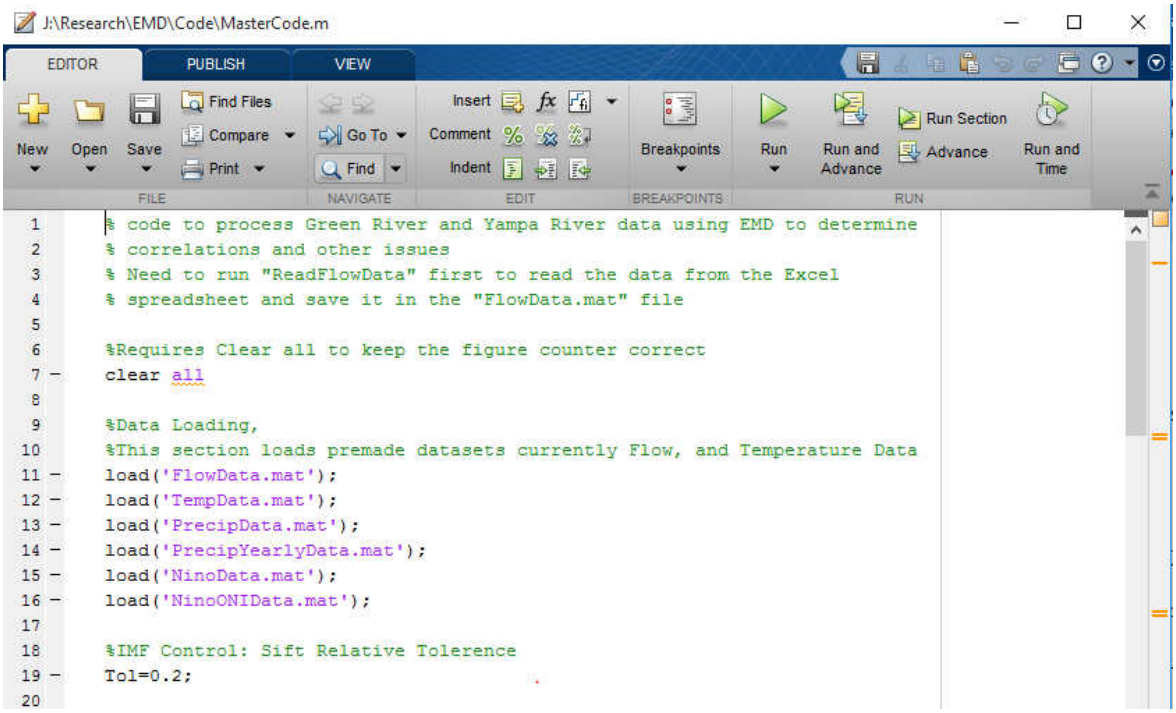

Figure 5.1.2-1 Mastercode, Generates the Associated IMFs and HHTs and Creates a Series of Labels Used in Plotting

5.1.3 Data Plotters

I wrote three Plotting Scripts to analyze data: CustomPloter, Residualcode, and DataNormalizer. I used each of these scripts to analyze data in different ways.

The CustomPlotter script, allows the user to select a data set, the IMFs of interest and plot just that part of the data. Some of the features of this script include:

- Ability to Superimpose IMFs, as well as Residuals
- Extract examined data and create a .mat file from it
- Compare selected data to El Niño using but the ONI method and Magnitude Method
- Save the figure

```
%Data Plotting Settings

V=[9,10];           %Which IMF datasets do you want to look at,
n=12;                    %Which area are  you interested in- Key above
SuperPosition=1;        %Superimpose IMFs of interest Y(1)/N(0)
Res=0;                  %Residual visible Y(1)/ N(0)
Dataextract=0;          %Saves Superimposed Vector
NinoRef=1;              %Displays barchart showing known El Nino Years
NinoMethod=0;           %Custom Plotter using ONI (0) or Nino intensity (1)
ResidualAdd=0;          %Add Residual to superimposed Y(1)/ N(0)
SaveFigure=1;           %Saves the Current Figure
```

Figure 5.1.3-1 Customplotter Input Settings

The Residualcode allows for the plotting of data residuals from respective datasets. It also has the ability to compare residuals across the four datasets. The DataNormalizer is a companion script to Residualcode that normalizes data between -1 and 1 in order to give better idea of the rate of change that is occurring in the residuals since flow residuals tend to be much greater than the other dataset's residuals

6 CHARACTERIZING THE RESULTS

6.1 Discussion of Results

I applied EMD to all the listed datasets to qualitatively identify a relationship between IMFs and the El Niño ONI dataset. In these results, I present specific data sets which show extracted signals from the El Niño weather pattern in the Discharge, Temperature, and Precipitation datasets within the study area.

The standard presentation of EMD components is to present the original signal at the top of a figure, with subsequent IMFs below, and finally the long-term residual at the bottom in vertical format. In this paper, the original signal will be presented in the figure, followed by the complete decomposition in the next figure, and selected IMFs of interest in the final figure. This pattern will be repeated for different data types and locations.

6.2 Discussion of Stream Discharge Results

Figure 6.2-1 shows discharge data from Flaming Gorge Dam. From examining the dataset, several changes in the reservoir operations can be seen. The downstream stream gauge shows the highly varied natural flow of the Green River. Construction of the dam began in 1958 and the Green River's diversion tunnel was completed by November 19, 1959. The Flaming Gorge Dam began operation after its completion in 1964. The period after the dam was completed until the mid-1990 represent a period where the dam was operated primarily to provide peak power,

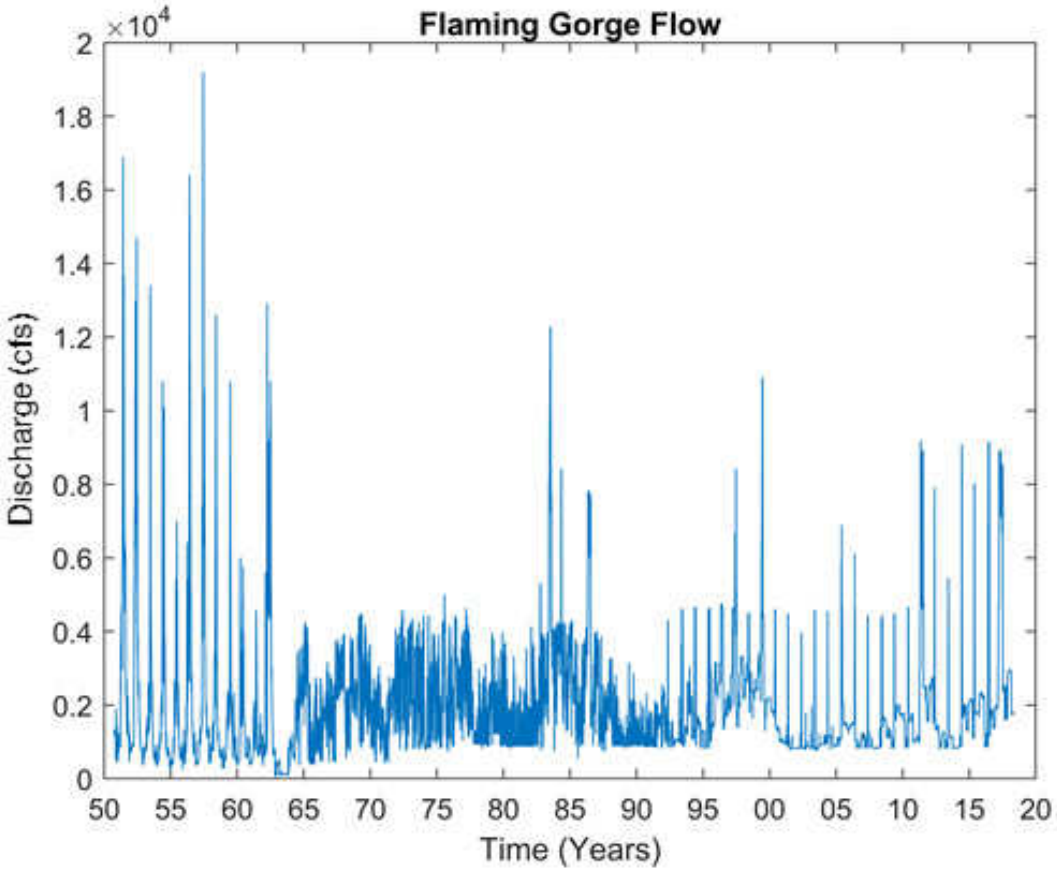

Figure 6.2-1: Flaming Gorge Stream Discharge Data

Figure 6.2-2 shows the EMD decomposition of the Flaming Gorge Dam discharge dataset that contains 11 IMFs and the residual. IMFs 7, 8 and 9 showed temporal patterns similar to the ONI. I summed these three IMFs and compared the resulting signal to the El Niño ONI dataset. The second to the bottom panel of Figure 6.2-3 shows the summed IMFs, while the bottom panel provides a visual comparison between the summed IMF's and the El Niño ONI dataset. Qualitatively the ONI dataset matches the summed IMF.

Visually there is good correlation between peaks in the ONI dataset and peaks in the composite IMF (Figure 6.2-3). The ONI often peaks before the IMF. This intuitively makes

sense, if a wet year is caused by El Niño, more water will come into the reservoir, therefore, more water will be discharged from the dam the following year. This reasoning suggesting that the El Niño signal within the streamflow IMFs is present despite the structure, and operational influences. La Niña is not as easily seen in the composite IMF, though traces are present. A possible explanation for this is that La Niña is often a drier year and the reservoir is discharging minimum flow, or being filled, partially hiding the presence of La Niña signal in the release data and thus in the composite IMF.

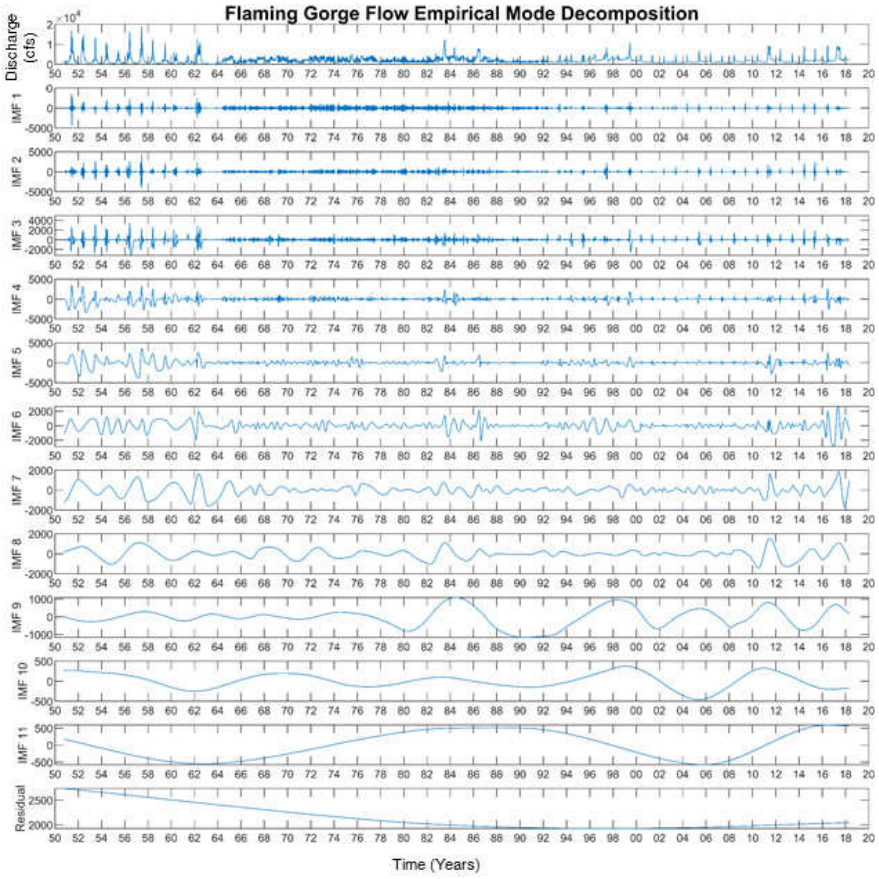

Figure 6.2-2: Flaming Gorge Stream Discharge Complete EMD

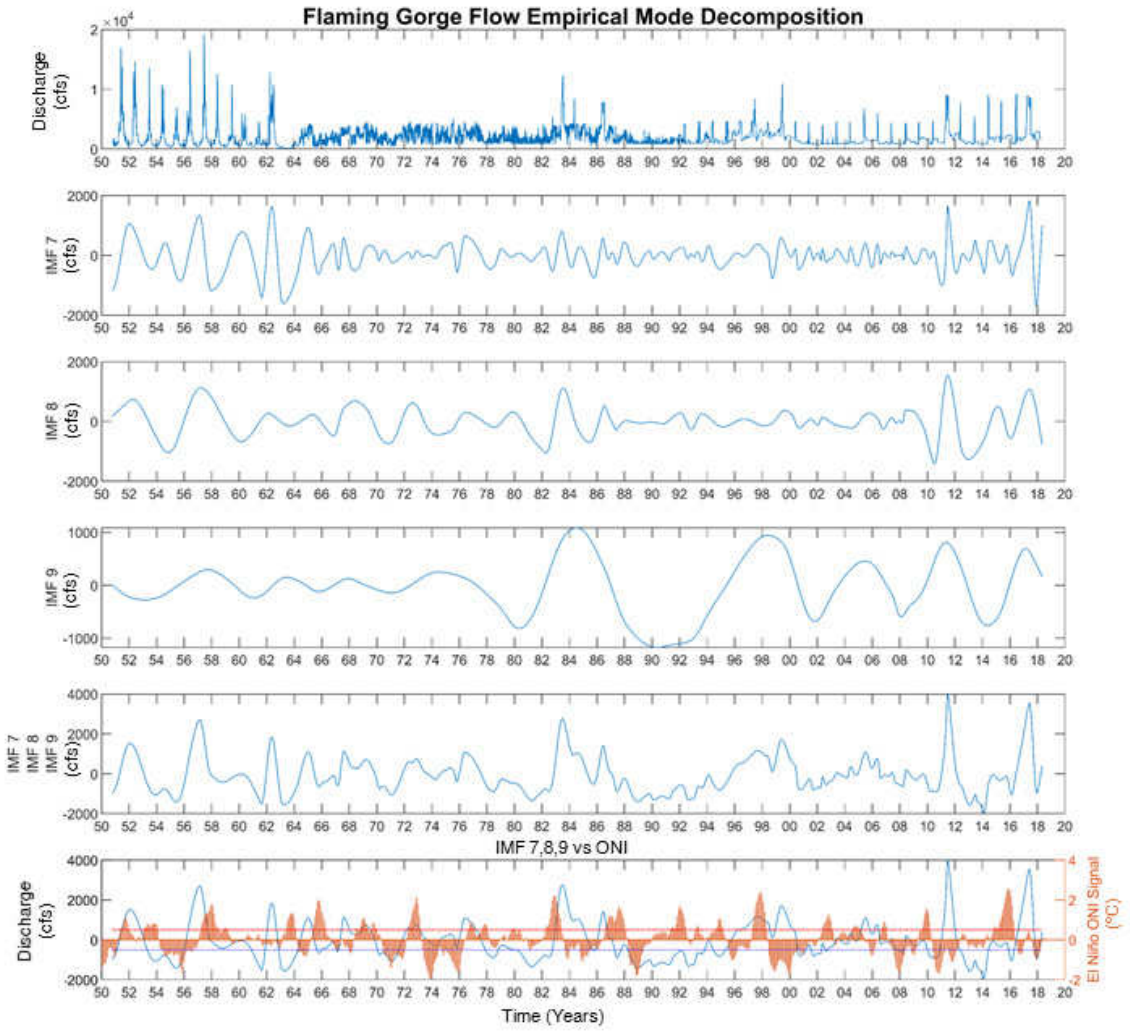

Figure 6.2-3: Flaming Gorge Stream Discharge IMF 7, IMF 8, IMF 9 Super Imposed and Compared to El Niño ONI (Bottom Panel)

The signal can be tracked downstream of Flaming Gorge Dam. One example of this is at the Jensen gage. This gage is important because it is approximately 40 miles south of the confluence of the Green River and the Yampa River and over 50 miles downstream of Flaming Gorge Dam. I created a composite IMF from IMF 7 and 8 (Figure 6.2-4). This composite IMF is similar to the ONI dataset suggesting that the El Niño signal that is passed through Flaming

Gorge Dam is preserved, even after mixing with the Yampa River influx. It may also be slightly strengthened, as the Yampa River is not impacted from dam operations and the signal is less attenuated.

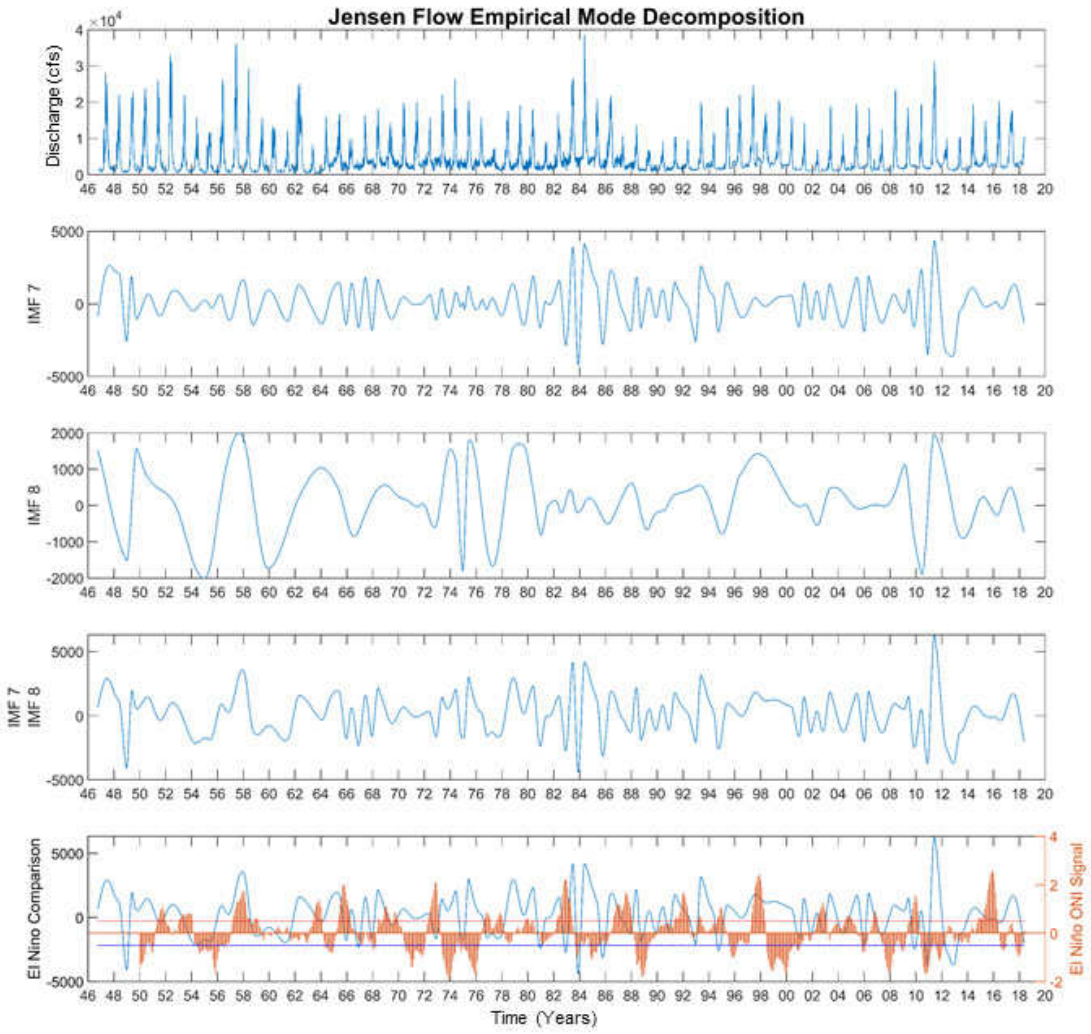

Figure 6.2-4: Jensen Streamflow Discharge IMF 7, IMF 8 Super Imposed and Compared to El Niño ONI (Bottom Panel)

6.3 Discussion of Temperature Results

The temperature dataset at Vernal was the longest dataset analyzed (Figure 6.3-1). In preparation for the EMD process, I arithmetically averaged daily high and low temperatures. I then decomposed these data as shown in Figure 6.3-2. Figure 6.3-3 shows a correlation between temperature variations and the ONI index. This relationship is interesting since both datasets correspond to variations of temperature. The figure shows that an increase in the ONI corresponds with increased temperatures in the composite dataset of IMF 9 and 10.

Figure 6.3-4 is a comparison of the Vernal and Jensen temperature dataset residuals. It shows that the residuals represent an increase of approximately 1 degree centigrade over the last 35 years and evidence of a cooling during the mid-70's –80's.

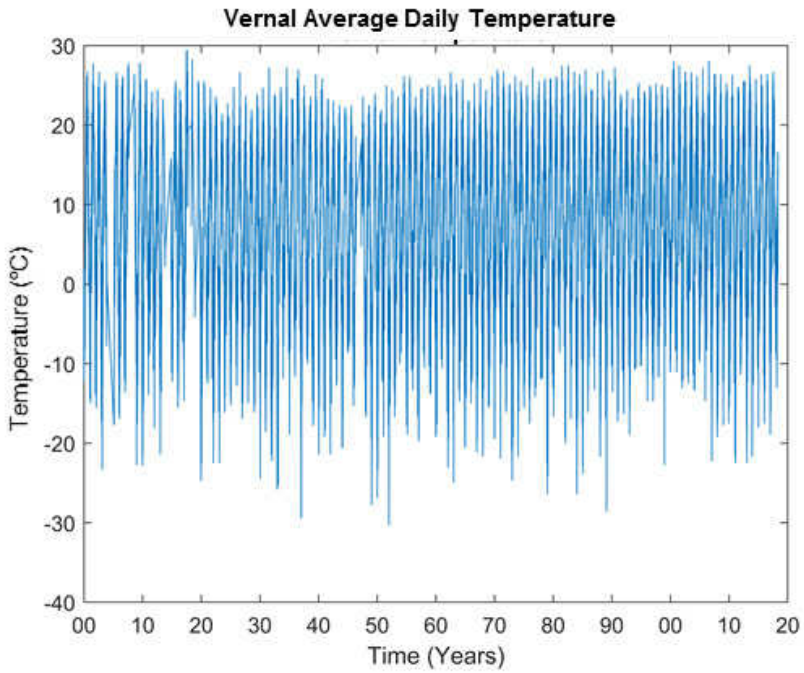

Figure 6.3-1: Vernal Temperature Dataset

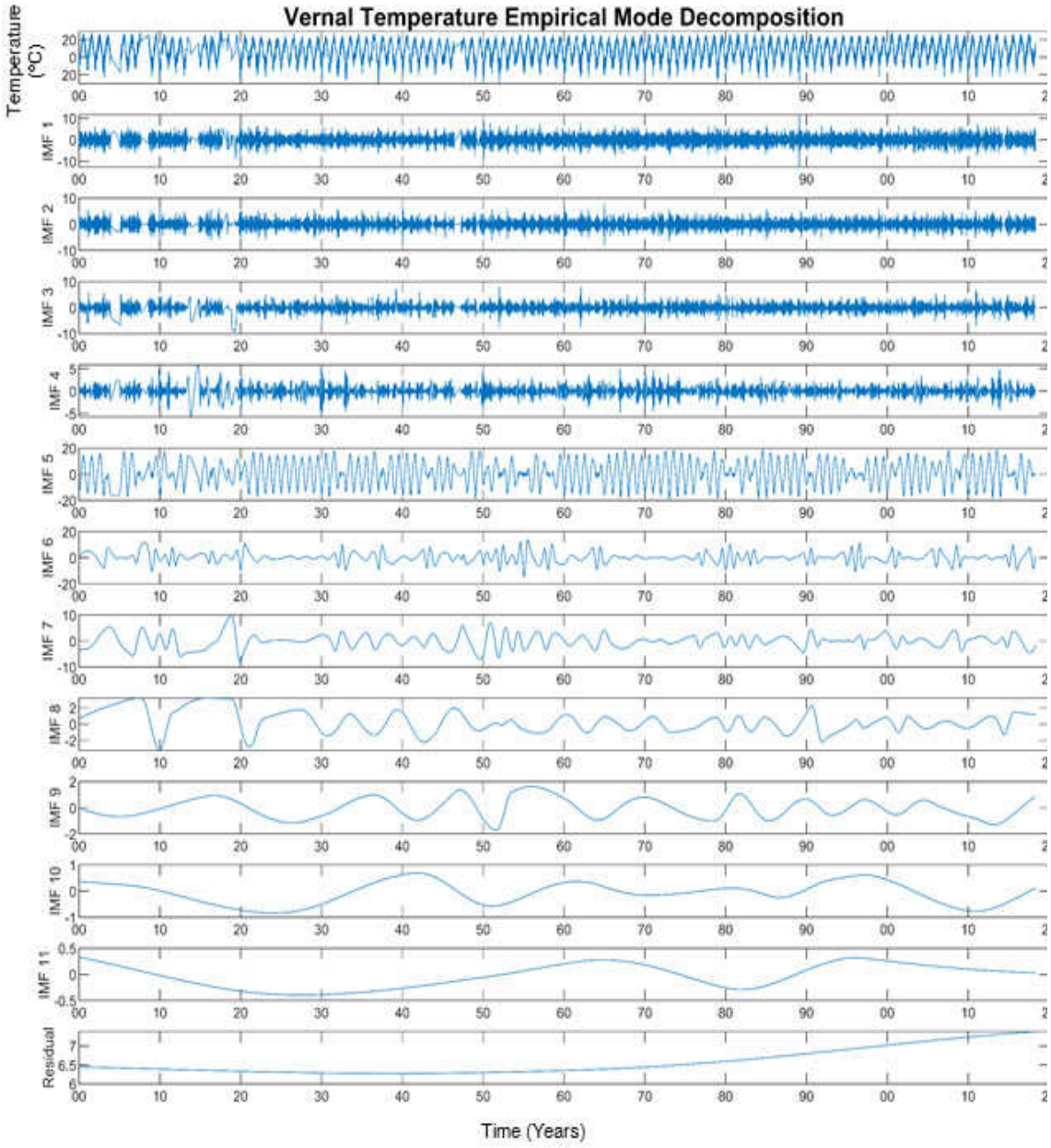

Figure 6.3-2: Decomposed Vernal Temperature Dataset

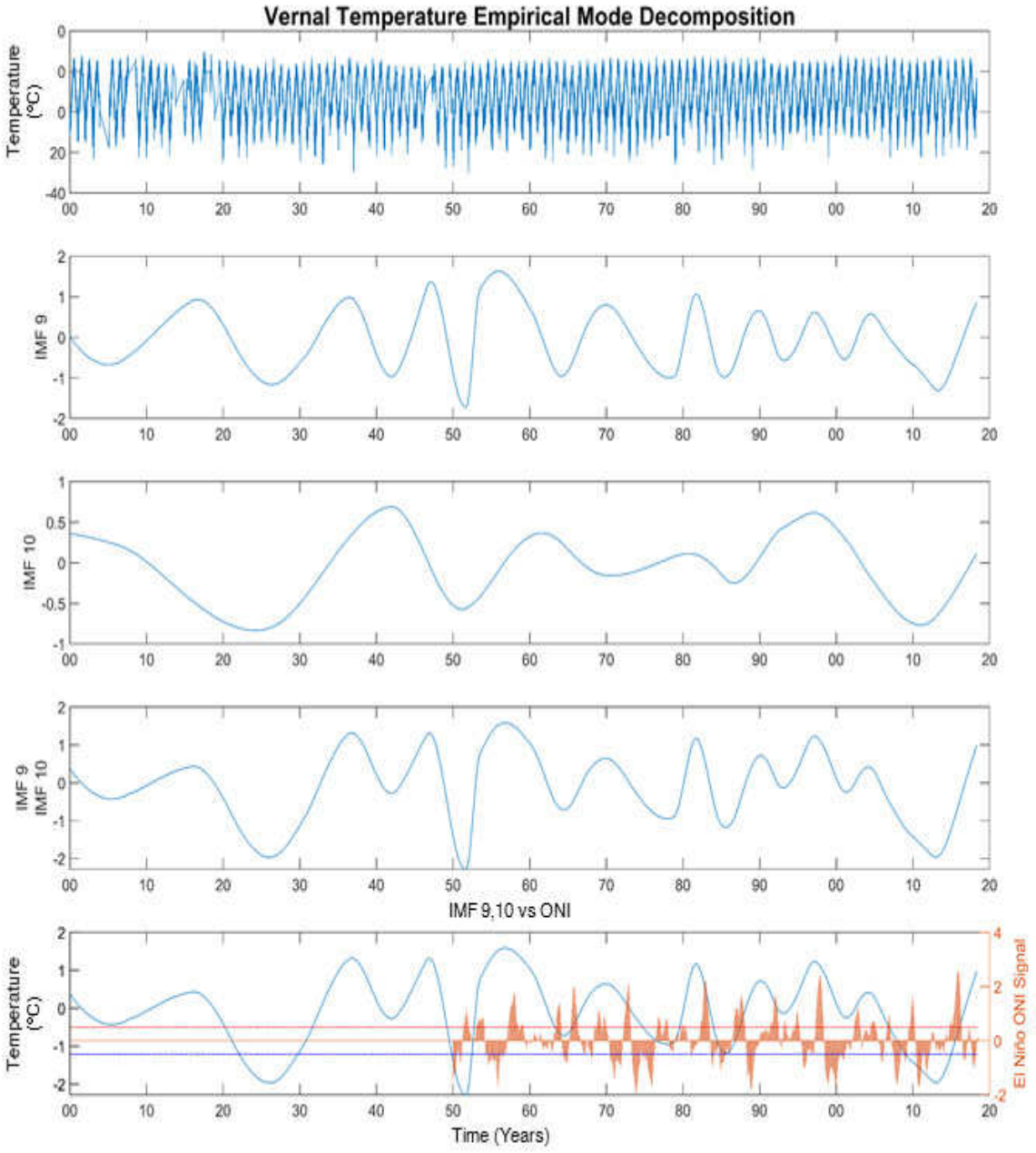

Figure 6.3-3: Vernal Temperature Decomposition Compared to the ONI Dataset (Bottom Panel).

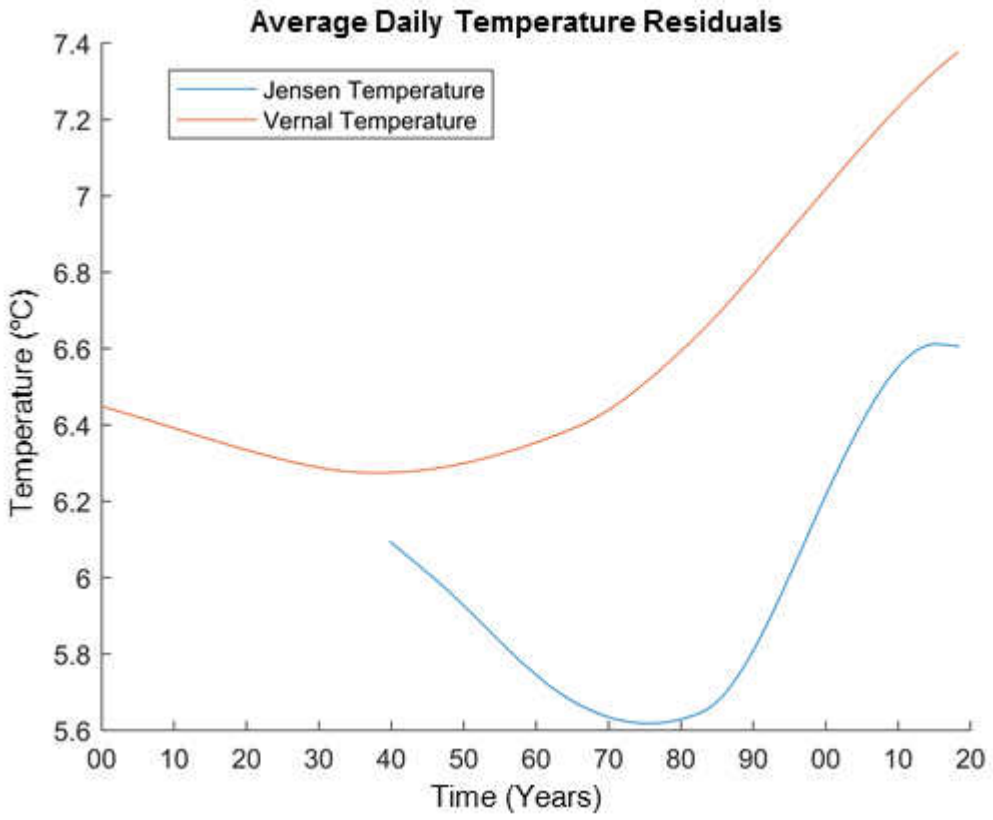

Figure 6.3-4: Residual of Jensen and Vernal Datasets

6.4 Discussion of Precipitation Results

Figure 6.4-1 shows the Jensen yearly precipitation dataset. Figure 6.4-2 shows the decomposed dataset, with Figure 6.4-3 showing the IMF/ONI comparison and the residuals (Figure 6.4-4).

The IMF/ONI comparison (Figure 6.4-3) shows IMF 2 compared to the ONI dataset. The data comparison suggests that ONI peaks resulting from El Niño correlate to increases in precipitation near the Jensen area. The data also suggests that La Niña does not necessarily forecast a drier year, however, El Niño can be an indicator of a wetter year, especially if it is a very strong event.

The residuals shown below (Figure 6.4-4) suggest that yearly trend in total precipitation depth has increased over the 93-year span, though the area. However, annual averages may not exhibit this trend as several of the period IMFs are in the minimum part of their cycle in recent years, overriding the long-term trend.

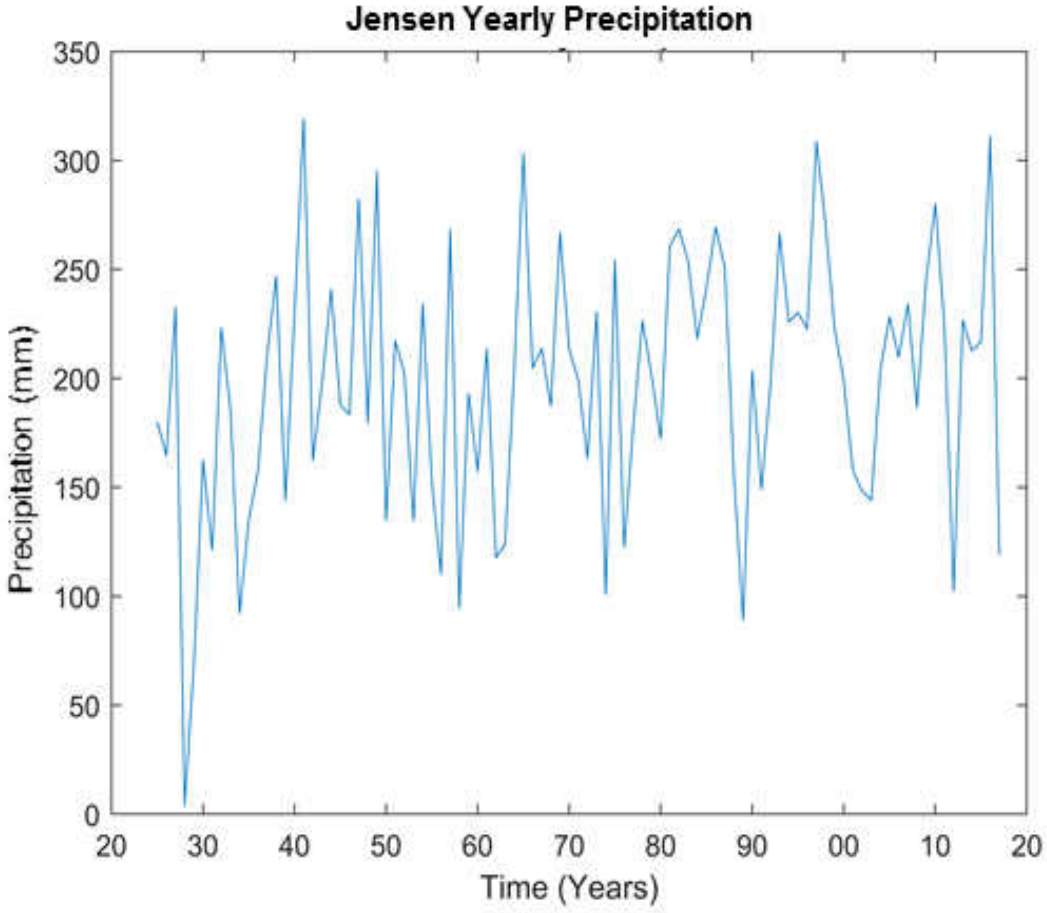

Figure 6.4-1: Jensen Yearly Precipitation Dataset. Every Point Is the Total Yearly Precipitation Depth

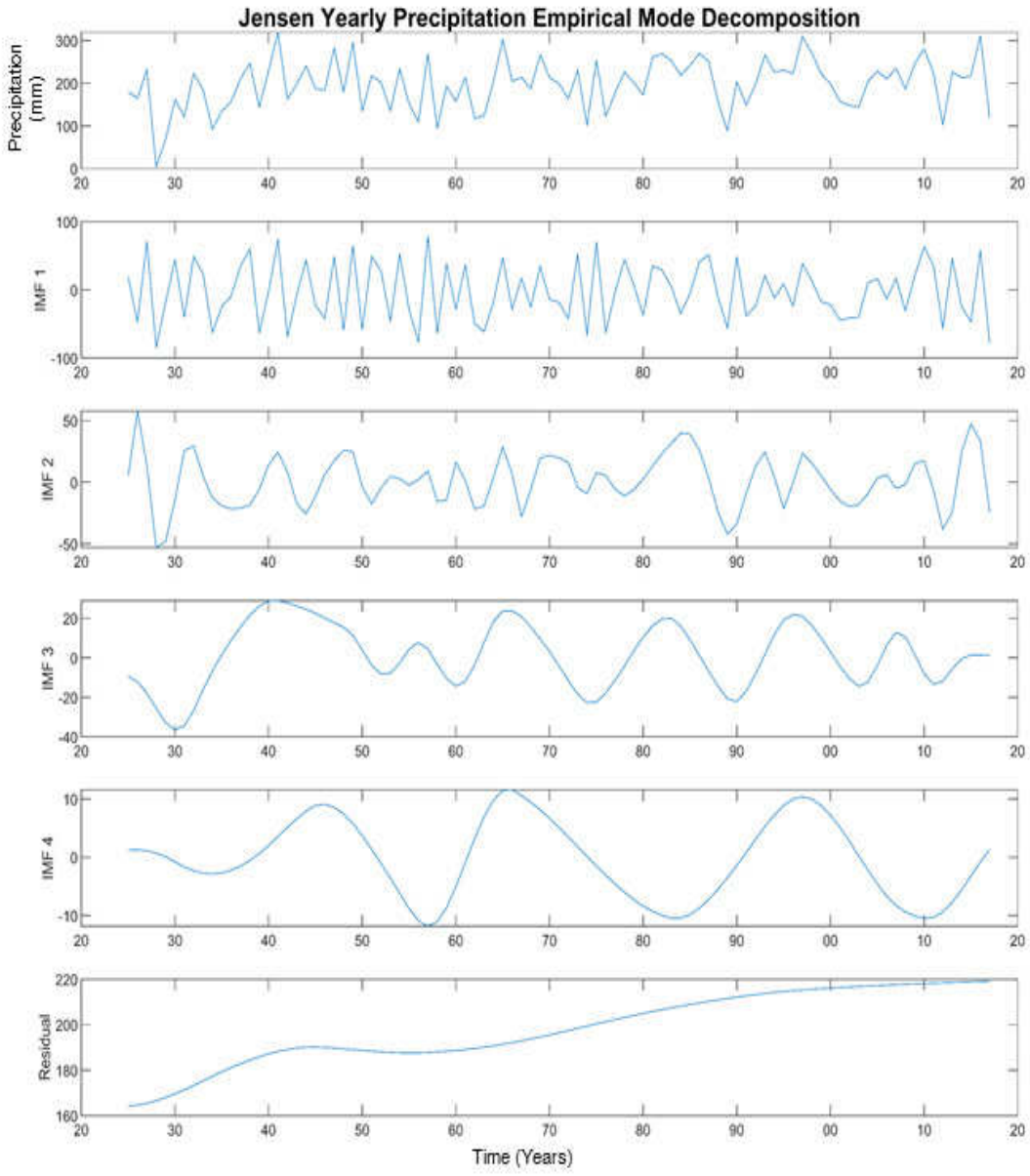

Figure 6.4-2: Jensen Precipitation Decomposition

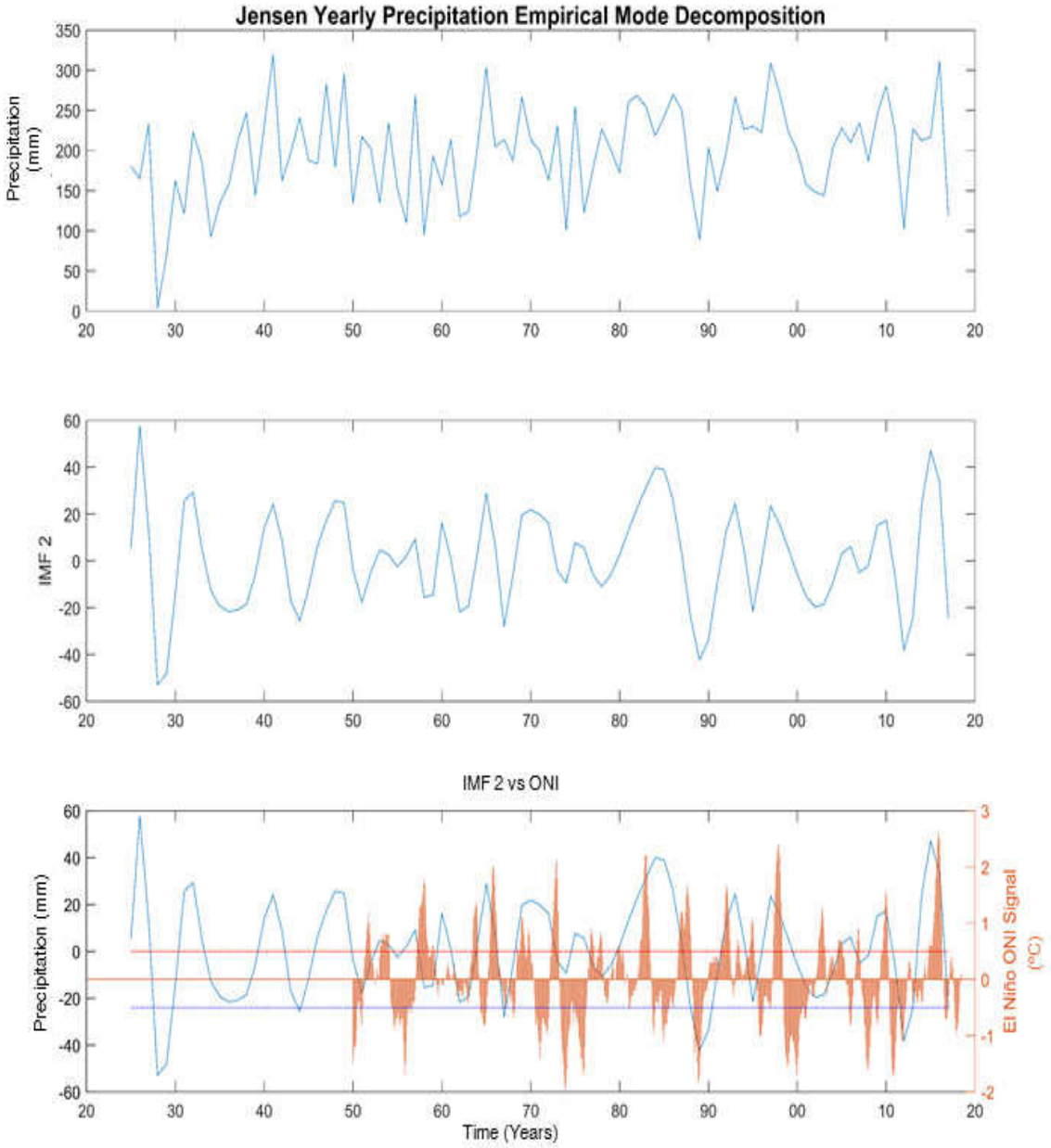

Figure 6.4-3: Jensen IMF 2 Compared to the ONI Dataset (Bottom Panel)

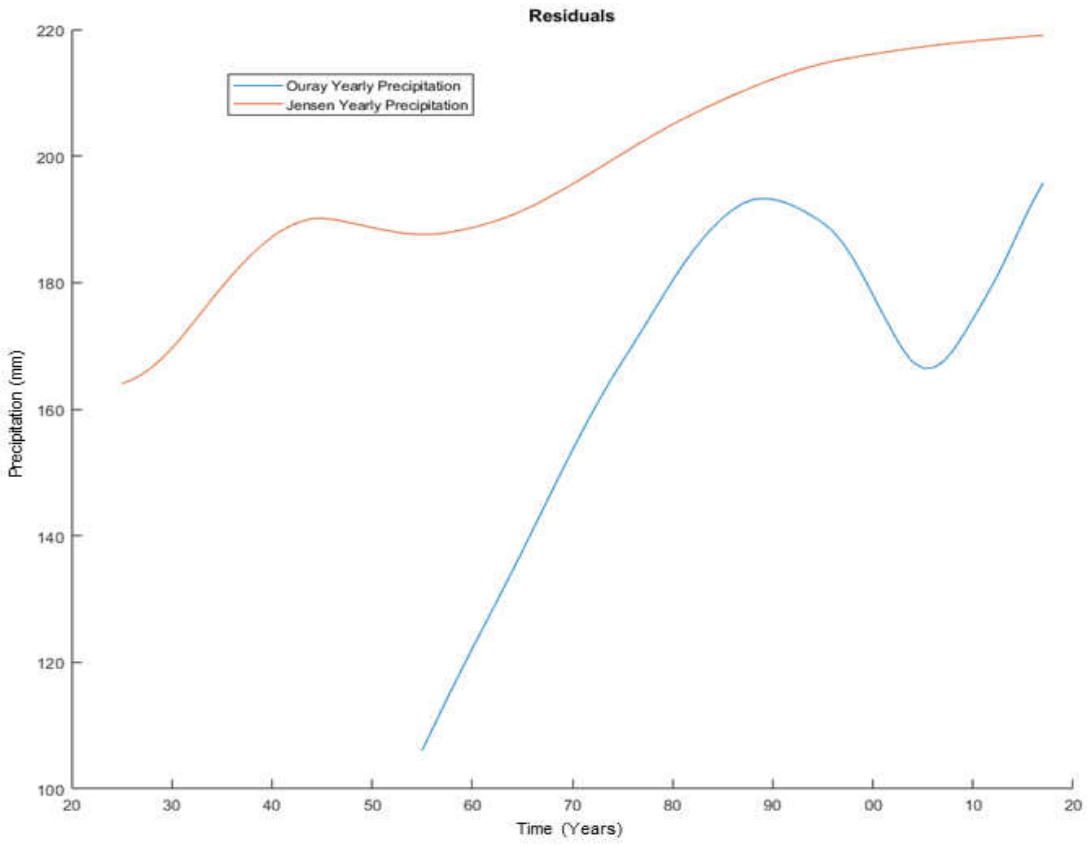

Figure 6.4-4: Ourey and Jensen Precipitation Residuals

7 CONCLUSIONS

By applying the EMD method to an area with a control group of stream gages (little anthropogenic influence) and an experimental group of stream gages (heavily regulated) gave much insight into how dam operations affect signals from natural phenomenon. By using EMD, I was able to extract the El Niño signal from streamflow discharge data and compare it to the ONI dataset. Additionally, I tracked the El Niño signal through the dam and downstream. This signal was preserved despite the mixing of rivers, the detention structure (dam), and dam discharge operations.

Using EMD analysis I showed a a strong positive correlation between yearly precipitation, and daily average temperature IMFs to El Niño demonstrating how this global process (El Niño) can affect local temperature patterns. The residual or long-term trend in the temperature data represents an increase of 1 degree centigrade over the last 35 years and evidence of a cooling during the mid-70's –80's. The long-term trend in precipitation suggests the region that precipitation is increasing. This is not seen in the composite data set (measured data) as a number of IMFs are in the minimum part of their cycles, masking the trend. It does suggest that precipitation may increase in the future if the long-term trend continues and the IMFs move towards the maximums in their cycles.

8 FUTURE WORK

8.1 Recommendations

One of the limitations for this study is that there are not many gages in the area with the criteria of quality data as defined in section 4.2.

- Must have ~60 years or more of data available
- Must have daily data
- Must be a valid or accredited source

Trends were not easily seen in data sets smaller than 50 years long due to edge effects (boundary error). Though it is necessary to have continuous data, the guidelines above are based on previous research and not hard rules. It's necessary to research the effects of boundary errors on IMFs and residuals. Additionally, it is will be of interest to further research how missing data affects IMFs and residuals. By understanding edge effects and missing data, the criteria above can be scientifically founded.

As more data becomes available, ONI and gage, it may be worthwhile to apply this method to the same area of study and record the differences.

Since the issue is not having sufficient long data sets, using Palaeohydrology to infer historical hydrological data and applying the EMD method may reveal interesting trends. Climate models could also be used to generate past climate data in order to estimate ONI values.

Another recommendation is the use of machine learning to recognize patterns within the IMFs and fill missing data gaps. This is more feasible than attempting to fill the original data gaps since IMFS are more likely to contain recognizable patterns.

REFERENCES

Bracewell, R. N., & Bracewell, R. N. (1986). *The Fourier transform and its applications* (Vol. 31999): McGraw-Hill New York.

Center, U. S. C. (2019). *Climate Data* [Precipitation, Temperature]. Retrieved from: https://climate.usu.edu/mapGUI/mapGUI.php

Coulibaly, P., & Burn, D. H. (2004). Wavelet analysis of variability in annual Canadian streamflows. *Water Resources Research, 40*(3). doi:10.1029/2003wr002667

el-Askary, H., Sarkar, S., Chiu, L., Kafatos, M., & El-Ghazawi, T. (2004). Rain gauge derived precipitation variability over Virginia and its relation with the El Nino southern oscillation. *Advances in space research, 33*(3), 338-342.

ESRI. (2019). Documentation. Retrieved from http://desktop.arcgis.com/en/documentation/

Hargis, B. H. (2014). Analysis of Long-Term Utah Temperature Trends Using Hilbert-Haung Transforms.

Huang, N. E. (2014). *Hilbert-Huang transform and its applications* (Vol. 16): World Scientific.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., . . . Liu, H. H. (1998). *The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis.* Paper presented at the Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences.

Kim, D., & Oh, H.-S. (2009). EMD: a package for empirical mode decomposition and Hilbert spectrum. *The R Journal, 1*(1), 40-46.

Labat, D. (2008). Wavelet analysis of the annual discharge records of the world's largest rivers. *Advances in Water Resources, 31*(1), 109-117. doi:https://doi.org/10.1016/j.advwatres.2007.07.004

MathWorks. (2019, 2019). R2018b. Retrieved from https://www.mathworks.com/help/documentation-center.html

Melesse, A., Abtew, W., Dessalegne, T., & Wang, X. (2010). Low and high flow analyses and wavelet application for characterization of the Blue Nile River system. *Hydrological Processes, 24*(3), 241-252. doi:10.1002/hyp.7312

Microsoft. (2019). Excel 2019. *What's new in Excel 2019 for Windows.* Retrieved from https://support.office.com/en-us/article/what-s-new-in-excel-2019-for-windows-5a201203-1155-4055-82a5-82bf0994631f

Nelsen, B., Williams, D., Williams, G., & Berrett, C. (2018). An Empirical Mode-Spatial Model for Environmental Data Imputation. *Hydrology, 5*(4), 63.

NOAA. (2018). El Niño & La Niña (El Niño-Southern Oscillation). Retrieved from https://www.climate.gov/enso

Pizarro, G., & Lall, U. (2002). El Niño-induced flooding in the U.S. West: What can we expect? *Eos, Transactions American Geophysical Union, 83*(32), 349-352. doi:doi:10.1029/2002EO000255

Service, N. W. (2019). *Cold & Warm Episodes by Season*. El Nino ONI. Retrieved from: https://origin.cpc.ncep.noaa.gov/products/analysis_monitoring/ensostuff/ONI_v5.php

Smith, L. C., Turcotte, D. L., & Isacks, B. L. (1998). Stream flow characterization and feature detection using a discrete wavelet transform. *Hydrological Processes, 12*(2), 233-249. doi:10.1002/(sici)1099-1085(199802)12:2<233::aid-hyp573>3.0.co;2-3

Stankovic, L. (1994). A method for time-frequency analysis. *IEEE Transactions on signal processing, 42*(1), 225-229.

Survey, U. S. G. (2019). *USGS Water Data for the Nation* [Flow Data]. Retrieved from: https://waterdata.usgs.gov/nwis?

Trenberth, K. E. (1997). The Definition of El Niño. *Bulletin of the American Meteorological Society, 78*(12), 2771-2778. doi:10.1175/1520-0477(1997)078<2771:tdoeno>2.0.co;2

APPENDIX A. MATLAB CODE

A.1 Custom Plotter

```
1  %Variable key for selecting "n" or "m".
2  %n is a variable used to select a specific dataset out of the data
3  %structure. "n" has been used to distinguish flow data from "m" which is
4  %temperature data
5
6  %Flow Data Key
7  %{
8  Max of 10 sets
9  n=1 "Flaming Gorge";
10 n=2 "Jensen";
11 n=3 "Little Snake Lily";
12 n=4 "Yampa Maybel";
13 Commented Out
14 n=5 "Yampa Deerlodge";
15 n=6 "Yampa Craig";
16 n=7 "Green River Ouray"
17 %}
```

```
18
19  %Temperature Data Key
20  %{
21  10 offset
22  n=11 "Jensen";
23  n=12 "Vernal";
24  Commented Out
25  n=13 "Grand Junction";
26  n=14 "Flaming Gorge";
27  n=15 "Maybel";
28  n=16 "Craig";
29  %}
30
31  %Precipitation Quarterly Data Key
32  %{
33  20 offset
34  n=21 "Ourey";
35  n=22 "Jensen";
36  Commented Out
37  n=23 "Maybel";
38  n=24 "Flaming Gorge";
39  n=25 "Craig";
40  %}
```

```
41
42  %Precipitation Yearly Data Key
43  %{
44  30 offset
45  n=31 "Ourey";
46  n=32 "Jensen";
47  Commented Out
48  n=33 "Maybel";
49  n=34 "Flaming Gorge";
50  n=35 "Craig";
51  %}
52
53  %Custom Plot creator
54  %Must run Mastercode first
55  %The user must input n,V, and SuperPosition=1/0 Res=1/0
56  %When working with flow data select n value associated with the data of
57  %interest, enter the value of V as a vector with the imf datasets of
58  %interest and if user desires to impose the datasets on eachother and plot
59  %make the value of SuperPosition 1, otherwise SuperPosition should be 0
60  %For best practice, only use one of the codes at a time.
61
62
63  %Data Plotting Settings
```

```
64
65  V=[9,10];      %Which IMF datasets do you want to look at,
66  n=12;          %Which area are you interested in- Key above
67  SuperPosition=1;    %Superimpose IMFs of interest Y(1)/N(0)
68  Res=0;         %Residual visible Y(1)/ N(0)
69  Dataextract=0;     %Saves Superimposed Vector
70  NinoRef=1;       %Displays barchart showing known El Nino Years
71  NinoMethod=0;      %Custom Plotter using ONI (0) or Nino intensity (1)
72  ResidualAdd=0;     %Add Residual to superimposed Y(1)/ N(0)
73  SaveFigure=1;      %Saves the Current Figure
74
75
76  %bar(Ninodata.date,Ninodata.flow)
77  %When Extracting Data, import by hand and rename workspace name
78  %figure
79  %plot(ExtractDataFlow(:,1),ExtractDataFlow(:,2),ExtractDataTemp(:,1),..
80  %ExtractDataTemp(:,2))
81  %datetick('x',11)
82  if NinoMethod==0;
83  %_____________________________Using ONI Data_____________________________
84  if n<=10
85    Comparex=NinoONIdata.date;
86    Comparey=NinoONIdata.flow;
```

```
87    Cplotimf(V,Gagedata(n).date,Gagedata(n).flow,Gagedata(n).imf,...
88      Gagedata(n).resid,...
89    Gage1(n,1),Lab,SuperPosition,Res,NinoRef,Comparex,Comparey,...
90    Dataextract,ResidualAdd,SaveFigure)
91  elseif n>10 && n<=20
92    n=n-10;
93    Comparex=NinoONIdata.date;
94    Comparey=NinoONIdata.flow;
95    Cplotimf(V,Tempdata(n).date,Tempdata(n).flow,Tempdata(n).imf,...
96      Tempdata(n).resid,...
97    Temp1(n,1),Lab,SuperPosition,Res,NinoRef,Comparex,Comparey,...
98    Dataextract,ResidualAdd,SaveFigure)
99  elseif n>20 && n<=30
100   n=n-20;
101     Comparex=NinoONIdata.date;
102     Comparey=NinoONIdata.flow;
103     Cplotimf(V,Precipdata(n).date,Precipdata(n).flow,Precipdata(n).imf,...
104       Precipdata(n).resid,...
105     Precip1(n,1),Lab,SuperPosition,Res,NinoRef,Comparex,Comparey,...
106     Dataextract,ResidualAdd,SaveFigure)
107   elseif n>30 && n<=40
108   n=n-30;
109     Comparex=NinoONIdata.date;
```

```
110    Comparey=NinoONIdata.flow;
111    Cplotimf(V,PrecipYearlydata(n).date,PrecipYearlydata(n).flow,...
112      PrecipYearlydata(n).imf,PrecipYearlydata(n).resid,...
113    PrecipY1(n,1),Lab,SuperPosition,Res,NinoRef,Comparex,Comparey,...
114    Dataextract,ResidualAdd,SaveFigure)
115
116  end
117  else
118    NinoMethod==1;
119  %_____Nino
Magnitude_____
120
121  if n<=10
122    Comparex=Ninodata.date;
123    Comparey=Ninodata.flow;
124    Cplotimf(V,Gagedata(n).date,Gagedata(n).flow,Gagedata(n).imf,...
125  Gagedata(n).resid,...
126    Gage1(n,1),Lab,SuperPosition,Res,NinoRef,Comparex,Comparey,...
127    Dataextract,ResidualAdd,SaveFigure)
128  elseif n>10 && n<=20
129    n=n-10;
130    Comparex=Ninodata.date;
131    Comparey=Ninodata.flow;
```

```
132    Cplotimf(V,Tempdata(n).date,Tempdata(n).flow,Tempdata(n).imf,...
133    Tempdata(n).resid,...
134    Temp1(n,1),Lab,SuperPosition,Res,NinoRef,Comparex,Comparey,...
135    Dataextract,ResidualAdd,SaveFigure)
136  elseif n>20 && n<=30
137  n=n-20;
138    Comparex=Ninodata.date;
139    Comparey=Ninodata.flow;
140    Cplotimf(V,Precipdata(n).date,Precipdata(n).flow,Precipdata(n).imf,...
141    Precipdata(n).resid,...
142    Precip1(n,1),Lab,SuperPosition,Res,NinoRef,Comparex,Comparey,...
143    Dataextract,ResidualAdd,SaveFigure)
144  elseif n>30 && n<=40
145  n=n-30;
146    Comparex=Ninodata.date;
147    Comparey=Ninodata.flow;
148    Cplotimf(V,PrecipYearlydata(n).date,PrecipYearlydata(n).flow,...
149    PrecipYearlydata(n).imf,PrecipYearlydata(n).resid,...
150    PrecipY1(n,1),Lab,SuperPosition,Res,NinoRef,Comparex,Comparey,...
151    Dataextract,ResidualAdd,SaveFigure)
152
153  end
154
```

```
155 end
156
157
158
%_____________________FUNCTION_____________________________________
159
160 function Cplotimf(A,Date,Flow,IMF,Residual,Name,Label,SuperPosition,...
161   Res,Nino,Compx,Compy,extract,ResidualAdd,SaveFigure)
162 figure1=figure('Position',[0,0,1024,1200]);
163 imfrows=length(A);
164 r=Res;
165 Super=SuperPosition;
166 SP=0;
167
168
169 %This section Plots the original time series signal
170 subplot(imfrows+1+Super+r+Nino,1,1);
171 plot(Date,Flow);
172 dateFormat=11;
173 datetick('x',dateFormat)
174 title(strcat('\fontsize{14}',Name+Label(1,3)));
175 ylabel('Signal');
176
```

```
177
178  %This section sets the size and plots the IMFs
179  for i=1:imfrows
180    j=A(1,i);
181    subplot(imfrows+1+Super+r+Nino,1,i+1);
182    plot(Date,IMF(:,j));
183    ylabel(strcat('IMF '+ string(j)));
184    dateFormat=11;
185    datetick('x',dateFormat);
186  end
187
188  %Option for displaying Residual
189  if r==1
190    subplot(imfrows+1+Super+r+Nino,1,imfrows+1+r);
191    plot(Date,Residual);
192    ylabel('Residual');
193    dateFormat=11;
194    datetick('x',dateFormat);
195  end
196
197  %Option for adding IMFs
198  if SuperPosition == 1
199    %Adding the Residual to the IMF
```

```
200  if ResidualAdd==0
201    for i=1:imfrows
202    j=A(1,i);
203    SP=SP+IMF(:,j);
204    end
205  %Saving Superimposed Datasets
206  if extract==1
207    ExtractData=[Date,SP];
208    save('J:\Research\EMD\Extracts\'+Name+' IMF '+mat2str(A)+'.mat',...
209      'ExtractData')
210  end
211
212  subplot(imfrows+1+Super+r+Nino,1,imfrows+1+r+Super);
213  plot(Date,SP);
214  subplot(imfrows+1+Super+r+Nino,1,imfrows+1+r+Super);
215  plot(Date,SP);
216  subplot(imfrows+1+Super+r+Nino,1,imfrows+1+r+Super);
217  plot(Date,SP);
218  ylabel(strcat('IMF '+ string(A)));
219  dateFormat=11;
220  datetick('x',dateFormat);
221
222  elseif ResidualAdd==1
```

```
223     for i=1:imfrows
224     j=A(1,i);
225     SP=SP+IMF(:,j);
226     end
227     SP=SP+Residual;
228
229     if extract==1
230     ExtractData=[Date,SP];
231     save('J:\Research\EMD\Extracts\'+Name+' IMF '+mat2str(A)+'.mat',...
232       'ExtractData')
233     end
234
235   subplot(imfrows+1+Super+r+Nino,1,imfrows+1+r+Super);
236   plot(Date,SP);
237
238   ylabel(strcat('IMF '+ string(A)+' Residual'));
239   dateFormat=11;
240   datetick('x',dateFormat);
241   end
242
243 else
244
245 end
```

```
246
247  if Nino==1
248    if SP==0
249      SP=IMF(:,A);
250    end
251    subplot(imfrows+1+Super+r+Nino,1,imfrows+1+r+Super+Nino);
252    plot(Date,SP);
253    ylabel('El Nino Comparison');
254    datetick('x',dateFormat);
255    %yyaxis left
256    hold on
257    yyaxis right
258    bar(Compx,Compy,'FaceAlpha',.80);
259    %plot(Compx,Compy);
260    %El Nino Reference
261    %El Nino event is efined by ONI of +/- 0.5
262      TimeRef=[min(Date);max(Date)];
263      ELNINO=[0.5;0.5];
264      LANINA=[-0.5;-0.5];
265      plot(TimeRef,ELNINO,'-.','Color','red','MarkerSize',0.1);
266      plot(TimeRef,LANINA,'-.','Color','blue','MarkerSize',0.1);
267    ylabel('El Niño ONI Signal')
268    dateFormat=11;
```

```
269    datetick('x',dateFormat);
270    hold off
271   end
272
273   if SaveFigure==1
274    Folder='J:\Research\Writing\Figures\Custom Plots\';
275    Tar=strcat(Folder,Name,' IMF ',mat2str(A));
276    print(gcf,Tar,'-dpng','-r1000')
277   end
278
279   end
280
281
282
```

A.2 Data Interpolator

```
1   clear all
2
3   %load('FlowData.mat');
4   load('TempData.mat');
5   %load('PrecipData.mat');
```

```
%load('NinoData.mat');

St=Tempdata;
t=1;
n=1;

%for i=1:length(Gage1)
for i=5:6

%for i=1:1

  ax=St(i).date(1);
  b=length(St(i).date);
  cx=St(i).date(b);
  Intx=ax:cx;
  Inty=interp1(St(i).date,St(i).flow,ax:t:cx,'linear');

  Data1=[St(i).date,St(i).flow];
  Data2=[Intx.',Inty.'];
  [DataFinal(i).original,DataFinal(i).Int]=ReadGage(Data1,Data2);

  %figure(i)=figure
  plot(St(i).date,St(i).flow,"o")
```

```
29      title("Original"+( i));
30    hold on
31
32    plot(Intx,Inty,"*")
33      title("Interpolate"+( i));
34    hold off
35
36  %Precipdata(i).date=x
37  %Precipdata(i).flow=y
38    %}
39  end
40
41
42  %Flow and Date importation
43  function [days, flow]=ReadGage(Data1,Data2)
44  days=Data1; %Importing Date data
45  flow=Data2; %Importing Flow Data
46  end
```

A.3 Data Normalizing Code

```
1   %This example normalize is used to compare trends between datasets
```

```
2   %In this case Yampa Deerlodge flow IMF [12] and Grand Junction IMF [11]
3   SaveFigure=0;
4
5   figure
6   x1=ExtractDataFlow(:,1)
7   y1=ExtractDataFlow(:,2)
8   y1Max=max(ExtractDataFlow(:,2))
9   y1Min=min(ExtractDataFlow(:,2))
10  y1Range=y1Max-y1Min
11  y1Norm=(y1-y1Min)/y1Range
12
13  x2=ExtractDataTemp(:,1)
14  y2=ExtractDataTemp(:,2)
15  y2Max=max(ExtractDataTemp(:,2))
16  y2Min=min(ExtractDataTemp(:,2))
17  y2Range=y2Max-y2Min
18  y2Norm=(y2-y2Min)/y2Range
19
20
21  plot(x1,y1Norm,x2,y2Norm)
22  datetick('x',11)
23
24  if SaveFigure==1
```

```
25    Folder='J:\Research\Writing\Figures\Custom Plots\';
26    Tar=strcat(Folder,Name,' IMF ',mat2str(A));
27    print(gcf,Tar,'-dpng','-r1000')
28  end
```

A.4 Master Code

```
1   % code to process Green River and Yampa River data using EMD to determine
2   % correlations and other issues
3   % Need to run "ReadFlowData" first to read the data from the Excel
4   % spreadsheet and save it in the "FlowData.mat" file
5
6   %Requires Clear all to keep the figure counter correct
7   clear all
8
9   %Data Loading,
10  %This section loads premade datasets currently Flow, and Temperature Data
11  load('FlowData.mat');
12  load('TempData.mat');
13  load('PrecipData.mat');
14  load('PrecipYearlyData.mat');
15  load('NinoData.mat');
```

```
16  load('NinoONIData.mat');
17
18  %IMF Control: Sift Relative Tolerence
19  Tol=0.2;
20
21  % Array for Figure Lables
22  %This matrix will contain the labels used for any figures
23
24  Lab=["Date","Flow (cfs)"," Empirical Mode Decomposition",...
25  " Hilbert Spectrum","Temp Figure"];
26
27  % Comptue IMFs using EMD
28  % Best Tolerances are 0.02 and .001, unstable afterwards (Crash)
29
30  % _____Flow Gage
Data_____
31  %Ploting Flow Data
32  %This Plots the initial flow time series data
33  %Changing the x-axis to a year format
34
35  %{
36  for i=1:length(Gage1)
37  %for i=1:3
```

```
38    PlotFlow(Gagedata(i).date,Gagedata(i).flow,Gage1(i,1),i)
39    dateFormat = 'yy';
40  end
41  %}
42
43  %EMD for flow data
44  for i=1:length(Gage1)
45  %for i=1:3
46    [imf, resid, info]=emd(Gagedata(i).flow,'Interpolation','pchip',...
47      'SiftRelativeTolerance',Tol,'MaxNumIMF',500,'SiftmaxIterations',...
48      900);%,'Display',0);
49    Gagedata(i).imf=imf;
50    Gagedata(i).resid=resid;
51  end
52
53
54  %Plotting Flow Data IMF
55  %Calls the User defined 'PlotIMF' function and applies it to Flow Data
56  for i=1:length(Gage1)
57  %for i=1:3
58    PlotIMF(Gagedata(i).date,Gagedata(i).flow,Gagedata(i).imf,...
59      Gagedata(i).resid,...
60    Gage1(i,1),Lab);
```

```
end


%Ploting HHT
for i=1:length(Gage1)
%for i=1:3
  h = findobj('type','figure');
  n = length(h);
  PlotHHT(Gagedata(i).imf,Gage1(i,1),Lab)
end

%}


%_____Temperature_____
%Ploting Flow Data
%This Plots the initial flow time series data
%Changing the x-axis to a year format
%{
for i=1:size(Temp1,1)
%for i=1:3
  PlotFlow(Tempdata(i).date,Tempdata(i).flow,Temp1(i,1),i);
  dateFormat = 'yy';
```

```
83  end
84  %}
85  %EMD for Temperature Data
86
87
88  for i=1:size(Temp1,1)
89    [imf, resid, info]=emd(Tempdata(i).flow,'Interpolation','pchip',...
90      'SiftRelativeTolerance',Tol,'MaxNumIMF',500,'SiftmaxIterations',...
91      900);%,'Display',0);
92    Tempdata(i).imf=imf;
93    Tempdata(i).resid=resid;
94  end
95
96
97  %Plotting Temperature based IMF
98  %for i=1:length(Temp1)
99  for i=1:2
100   PlotIMF(Tempdata(i).date,Tempdata(i).flow,Tempdata(i).imf,...
101     Tempdata(i).resid,...
102   Temp1(i,1),Lab);
103  end
104
105
```

```
%Ploting HHT
for i=1:size(Temp1,1)
%for i=1:3
  h = findobj('type','figure');
  n = length(h);
  PlotHHT(Tempdata(i).imf,Temp1(i,1),Lab)
end

%}


%_____Precipitation_Quarterly_____
%Ploting Precipitation Data
%This Plots the initial flow time series data
%Changing the x-axis to a year format
%{
for i=1:size(Precip1,1)
%for i=1:3
  PlotFlow(Precipdata(i).date,Precipdata(i).flow,Precip1(i,1),i);
  dateFormat = 'yy';
end
%}
```

```
129
130  %EMD for Quarterly Precipitation Data
131
132  %for i=1:length(Precip1)
133  for i=1:size(Precip1,1)
134    [imf, resid, info]=emd(Precipdata(i).flow,'Interpolation','pchip',...
135      'SiftRelativeTolerance',Tol,'MaxNumIMF',500,'SiftmaxIterations',...
136    900);%'Display',0);
137    Precipdata(i).imf=imf;
138    Precipdata(i).resid=resid;
139  end
140
141
142  %Plotting Quarterly Precip based IMF
143  %for i=1:length(Precip1)
144  for i=1:size(Precip1,1)
145   PlotIMF(Precipdata(i).date,Precipdata(i).flow,Precipdata(i).imf,...
146     Precipdata(i).resid,...
147   Precip1(i,1),Lab);
148  end
149
150
151  %Ploting HHT
```

```
for i=1:size(Precip1,1)
%for i=1:3
  h = findobj('type','figure');
  n = length(h);
  PlotHHT(Precipdata(i).imf,Precip1(i,1),Lab)
end

%}


%_____Precipitation_Yearly_____
%Ploting Precipitation Data
%This Plots the initial flow time series data
%Changing the x-axis to a year format
%{
for i=1:size(PrecipY1,1)
%for i=1:3
  PlotFlow(PrecipYearlydata(i).date,PrecipYearlydata(i).flow,...
PrecipY1(i,1),i);
  dateFormat = 'yy';
end
%}
```

```
%EMD for Yearly Precip Data
for i=1:size(PrecipY1,1)
  [imf, resid, info]=emd(PrecipYearlydata(i).flow,'Interpolation',...
   'pchip','SiftRelativeTolerance',Tol,'MaxNumIMF',500,...
   'SiftmaxIterations',900);%'Display',0);
  PrecipYearlydata(i).imf=imf;
  PrecipYearlydata(i).resid=resid;
end


%Plotting Yearly Precip Data based IMF
for i=1:size(PrecipY1,1)
 PlotIMF(PrecipYearlydata(i).date,PrecipYearlydata(i).flow,...
   PrecipYearlydata(i).imf,PrecipYearlydata(i).resid,...
 PrecipY1(i,1),Lab);
end


%Ploting Yearly Precip Data HHT
for i=1:size(PrecipY1,1)
  h = findobj('type','figure');
  n = length(h);
  PlotHHT(PrecipYearlydata(i).imf,PrecipY1(i,1),Lab)
```

```
197 end
198
199 %}
200
201
202
203
204
%_____Functions_____
205
206 %Ploting Flow vs Date
207 function PlotFlow(days,flow,Name,number)
208 figure('Name',strcat(Name,'Flow Data'),'NumberTitle','off');
209 plot(days, flow);
210 %This can be changed to 'dd/mm/yyy' in order to plot data of specific dates
211 dateFormat = 'yy';
212 datetick('x',dateFormat);
213   xlabel('Time (Years)');
214   ylabel('Signal');
215   title(strcat(Name));
216   %%Comment to stop file Creation
217   Tar=strcat('J:\Research\Writing\Figures\',Name);
218 %print(gcf,Tar,'-dpng','-r1000')
```

```
219  end
220
221  %Plotting all IMFs
222  function PlotIMF(Date,Flow,IMF,Residual,Name,Label)
223  %Dimensioning Matrix
224  %x1=1:length(Flow);
225  imfrows=size([IMF],2);
226  figure1=figure('Position',[0,0,1024,1200],'Name',Name);
227
228
229  %Plots the signal (Flow as time series)
230  subplot(imfrows+2,1,1);
231  plot(Date,Flow);
232  dateFormat=11;
233  datetick('x',dateFormat)
234  title(strcat('\fontsize{14}',Name+Label(1,3)));
235  ylabel('Signal');
236
237  for i=1:imfrows
238    subplot(imfrows+2,1,i+1);
239    plot(Date,IMF(:,i));
240    ylabel(strcat('IMF '+ string(i)));
241    dateFormat=11;
```

```
242    datetick('x',dateFormat)
243  end
244  subplot(imfrows+2,1,imfrows+2);
245  plot(Date,Residual);
246  ylabel('Residual');
247  dateFormat=11;
248  datetick('x',dateFormat)
249    %%Comment to stop file Creation
250    Tar=strcat('J:\Research\Writing\Figures\',Name,Label(1,3));
251  %print(gcf,Tar,'-dpng','-r1000')
252  end
253
254  %Plotting HHTs
255  function PlotHHT(IMF,Name,Label) %ActiveFigureNumber,
256  figure('name',strcat(Name+Label(1,4)))
257  hht(IMF);
258  title(strcat(Name+Label(1,4)));
259    %%Comment to stop file Creation
260    Tar=strcat('J:\Research\Writing\Figures\',Name,Label(1,4));
261  %print(gcf,Tar, '-dpng','-r1000')
262  end
```

A.5 Read Flow Data

```
1  % ReadFlowData
2  % Reads flow data from Excel Spreadsheet and stores it in a .mat file
3
4  %Scraping data from Spreadsheets
5  %Data has been historically imported from .csv format spreadsheets with
6  %dates in 'excel serial number' format in column A and
7  %flow values in column B
8
9  clear all
10
11 % Gage is matrix of meta data, column 1 is gage name, Column 2 is File
12 % name, Column 3 is the 3 letter acronym for the gage
13 Gage(1,1)="Flaming Gorge Flow";     Gage(1,2)="FG1.csv"; Gage(1,3)="FG1";
14 Gage(2,1)="Jensen Flow";         Gage(2,2)="GJ2.csv"; Gage(2,3)="GJ2";
15 Gage(3,1)="Little Snake Lily Flow";   Gage(3,2)="YL4.csv"; Gage(3,3)="LS1";
16 Gage(4,1)="Yampa Maybel Flow";      Gage(4,2)="YM5.csv"; Gage(4,3)="YM5";
17 %Gage(5,1)="Yampa Maybel Flow E";     Gage(5,2)="YME5.csv"; Gage(5,3)="YME5";
18 %Gage(5,1)="Yampa Deerlodge Flow";    Gage(3,2)="YD3.csv"; Gage(3,3)="YD3";
19 %Gage(6,1)="Yampa Craig Flow";      Gage(6,2)="YC6.csv"; Gage(6,3)="YC6";
20 %Gage(7,1)="Green River Ouray Flow";   Gage(7,2)="GO9.csv"; Gage(7,3)="GR9";
21
22 % Gage1 is matrix of meta data, column 1 is gage name, Column 2 is File
```

```
% name, Column 3 is the 3 letter acronym for the gage
Gage1=["Flaming Gorge Flow",   'J:\Research\EMD\Flow_Data\FG1.csv', "FG1"; ...
    "Jensen Flow",       'J:\Research\EMD\Flow_Data\GJ2.csv', "GJ2"; ...
    "Little Snake Lily Flow", 'J:\Research\EMD\Flow_Data\YL4.csv', "LS1"; ...
    "Yampa Maybel Flow",    'J:\Research\EMD\Flow_Data\YM5.csv', "YM5";];
    %"Yampa Maybel Flow E",   'J:\Research\EMD\Flow_Data\YME5.csv', "YME5";];
    %"Yampa Deerlodge Flow",  'J:\Research\EMD\Flow_Data\YD3.csv', "YD3"; ...
    %"Yampa Craig Flow",    'J:\Research\EMD\Flow_Data\YC6.csv', "YC6";];
    %"Green River Ouray Flow", 'J:\Research\EMD\Flow_Data\GO9.csv', "GR9"];



for i=1:length(Gage1)
%for i=1:3
  [Gagedata(i).date, Gagedata(i).flow]=ReadGage(Gage1(i,2));
end



save('FlowData.mat');


%Functions below
```

```
46  %-----------------------------------------------------------------------

47

48  %Flow and Date importation

49  function [days, flow]=ReadGage(filename)

50  days=xlsread(filename, 'A:A'); %Importing Date data

51  flow=xlsread(filename, 'B:B'); %Importing Flow Data

52  end
```

A.6 Reader Nino Data

```
1   clear all

2

3   Nino(1,1)="El Nino";

4   Nino(1,2)="J:\Research\EMD\El_Nino_Events\Nino.csv";

5   Gage(1,3)="Nino";

6

7   Nino1=["Nino", 'J:\Research\EMD\El_Nino_Events\Nino.csv', "Nino";];

8

9   [Ninodata(1).date, Ninodata(1).flow]=ReadGage(Nino1(1,2));

10

11  save('NinoData.mat');

12
```

```
13
14
15  %Functions below
16  %------------------------------------------------------------------------
17
18  %Flow and Date importation
19  function [days, flow]=ReadGage(filename)
20  days=xlsread(filename, 'A:A'); %Importing Date data
21  flow=xlsread(filename, 'B:B'); %Importing Flow Data
22  end
```

A.7 Read ONI Data

```
1   clear all
2
3   NinoONI(1,1)="El Nino ONI";
NinoONI(1,2)="J:\Research\EMD\El_Nino_Events\NOAA.csv";  Gage(1,4)="Nino ONI";
4
5   NinoONI=["Nino ONI", 'J:\Research\EMD\El_Nino_Events\NOAA.csv', "Nino ONI";];
6
7   [NinoONIdata(1).date, NinoONIdata(1).flow]=ReadGage(NinoONI(1,2));
8
```

```
9   save('NinoONIData.mat');
10
11
12
13  %Functions below
14  %--------------------------------------------------------------------
15
16  %Flow and Date importation
17  function [days, flow]=ReadGage(filename)
18  days=xlsread(filename, 'A:A'); %Importing Date data
19  flow=xlsread(filename, 'E:E'); %Importing Flow Data
20  end
```

A.8 Read Precipitation Quarterly

```
1   % ReadFlowData
2   % Reads flow data from Excel Spreadsheet and stores it in a .mat file
3
4   %Scraping data from Spreadsheets
5   %Data has been historically imported from .csv format spreadsheets with
6   %dates in 'excel serial number' format in column A and
7   %flow values in column B
```

```
8
9   %Clear all used to clear cache of matlab to not save unneeded excess
10  clear all
11
12  % Gage is matrix of meta data, column 1 is gage name, Column 2 is File
13  % location, Column 3 is the 3 letter acronym for the gage
14
15
16  Precip(1,1)="Ouray Precipitation";
Precip(1,2)='J:\Research\EMD\Precipitation\OGP.csv'; Precip(1,3)="OGP";
17  Precip(2,1)="Jensen Precipitation";
Precip(2,2)='J:\Research\EMD\Precipitation\JGP.csv'; Precip(2,3)="JGP";
18  %Precip(3,1)="Maybel Precipitation";
Precip(3,2)='J:\Research\EMD\Precipitation\MGP.csv'; Precip(3,3)="MGP";
19  %Precip(4,1)="Flaming Gorge Precipitation";
Precip(4,2)='J:\Research\EMD\Precipitation\FGP.csv'; Precip(4,3)="FGP";
20  %Precip(5,1)="Craig Precipitation";
Precip(5,2)='J:\Research\EMD\Precipitation\CGP.csv'; Precip(5,3)="CGP";
21
22  % Gage1 is matrix of meta data, column 1 is gage name, Column 2 is File
23  % name, Column 3 is the 3 letter acronym for the gage
24  %This Matrix will used to Scrape the data from the spreadsheets
25
```

```
26
27  Precip1=["Ourey Precipitation",      'J:\Research\EMD\Precipitation\OGP.csv', "OGP";...
28       "Jensen Precipitation",     'J:\Research\EMD\Precipitation\JGP.csv', "JGP";];
29  %{
30       "Maybel Precipitation",     'J:\Research\EMD\Precipitation\MGP.csv', "MGP";...
31       "Flaming Gorge Precipitation",  'J:\Research\EMD\Precipitation\FGP.csv', "FGP";...
32       "Craig Precipitation",      'J:\Research\EMD\Precipitation\CGP.csv', "CGP";];
33    %}
34
35  %The length function will only work if the matrix has more rows than
36  %columns
37
38  %for i=1:length(Precip1)
39  for i=1:2
40    [Precipdata(i).date, Precipdata(i).flow]=ReadGage(Precip1(i,2));
41  end
42
43
44
45  save('PrecipData.mat');
46
47
48
```

```
%Functions below
%------------------------------------------------------------------------


%Flow and Date importation
function [days, temp]=ReadGage(filename)
days=xlsread(filename, 'A:A'); %Importing Date data
temp=xlsread(filename, 'B:B'); %Importing Flow Data
end

```

A.9 Read Precipitation Data Yearly

```
% ReadFlowData
% Reads flow data from Excel Spreadsheet and stores it in a .mat file

%Scraping data from Spreadsheets
%Data has been historically imported from .csv format spreadsheets with
%dates in 'excel serial number' format in column A and
%flow values in column B

%Clear all used to clear cache of matlab to not save unneeded excess
```

```
10  clear all
11
12  % Gage is matrix of meta data, column 1 is gage name, Column 2 is File
13  % location, Column 3 is the 3 letter acronym for the gage
14
15
16  PrecipY(1,1)="Ouray Yearly Precipitation";
PrecipY(1,2)='J:\Research\EMD\PrecipitationYearly\OGYP.csv'; PrecipY(1,3)="OGYP";
17  PrecipY(2,1)="Jensen Yearly Precipitation";
PrecipY(2,2)='J:\Research\EMD\PrecipitationYearly\JGYP.csv'; PrecipY(2,3)="JGYP";
18  %Precip(3,1)="Maybel Precipitation";
Precip(3,2)='J:\Research\EMD\Precipitation\MGP.csv'; Precip(3,3)="MGP";
19  %Precip(4,1)="Flaming Gorge Precipitation";
Precip(4,2)='J:\Research\EMD\Precipitation\FGP.csv'; Precip(4,3)="FGP";
20  %Precip(5,1)="Craig Precipitation";
Precip(5,2)='J:\Research\EMD\Precipitation\CGP.csv'; Precip(5,3)="CGP";
21
22  % Gage1 is matrix of meta data, column 1 is gage name, Column 2 is File
23  % name, Column 3 is the 3 letter acronym for the gage
24  %This Matrix will used to Scrape the data from the spreadsheets
25
26
27  PrecipY1=["Ouray Yearly Precipitation",
```

```
'J:\Research\EMD\PrecipitationYearly\OGYP.csv', "OGYP";...
28      "Jensen Yearly Precipitation",     'J:\Research\EMD\PrecipitationYearly\JGYP.csv',
"JGYP";];
29  %{
30      "Maybel Precipitation",     'J:\Research\EMD\Precipitation\MGP.csv', "MGP";...
31      "Flaming Gorge Precipitation",  'J:\Research\EMD\Precipitation\FGP.csv', "FGP";...
32      "Craig Precipitation",      'J:\Research\EMD\Precipitation\CGP.csv', "CGP";];
33    %}
34
35  %The length function will only work if the matrix has more rows than
36  %columns
37
38  %for i=1:length(Precip1)
39  for i=1:2
40    [PrecipYearlydata(i).date, PrecipYearlydata(i).flow]=ReadGage(PrecipY1(i,2));
41  end
42
43
44
45  save('PrecipYearlyData.mat');
46
47
48
```

```
49  %Functions below
50  %------------------------------------------------------------------
51
52
53  %Flow and Date importation
54  function [days, temp]=ReadGage(filename)
55  days=xlsread(filename, 'A:A'); %Importing Date data
56  temp=xlsread(filename, 'B:B'); %Importing Flow Data
57  end
58
```

A.10 Reader Temperature Data

```
1   % ReadFlowData
2   % Reads flow data from Excel Spreadsheet and stores it in a .mat file
3
4   %Scraping data from Spreadsheets
5   %Data has been historically imported from .csv format spreadsheets with
6   %dates in 'excel serial number' format in column A and
7   %flow values in column B
8
9   %Clear all used to clear cache of matlab to not save unneeded excess
```

```
clear all

% Gage is matrix of meta data, column 1 is gage name, Column 2 is File
% location, Column 3 is the 3 letter acronym for the gage
%This is just a means of organization, not necessary


Temp(1,1)="Jensen Temperature";
Temp(1,2)='J:\Research\EMD\Temperature\JGT.csv'; Temp(1,3)="JGT";
Temp(2,1)="Vernal Temperature";
Temp(2,2)='J:\Research\EMD\Temperature\VGT.csv'; Temp(2,3)="VGT";
%Temp(3,1)="Grand Junction Temperature";
Temp(3,2)='J:\Research\EMD\Temperature\GJT.csv'; Temp(3,3)="GJT";
%Temp(4,1)="Flaming Gorge Temperature";
Temp(4,2)='J:\Research\EMD\Temperature\FGT.csv'; Temp(4,3)="FGT";
%Temp(5,1)="Maybel Temperature";
Temp(5,2)='J:\Research\EMD\Temperature\MGT.csv'; Temp(5,3)="MGT";
%Temp(6,1)="Craig Temperature";
Temp(6,2)='J:\Research\EMD\Temperature\CGT.csv'; Temp(6,3)="MGT";

% Gage1 is matrix of meta data, column 1 is gage name, Column 2 is File
% name, Column 3 is the 3 letter acronym for the gage
%This Matrix will used to Scrape the data from the spreadsheets
```

```
27
28
29  Temp1=["Jensen Temperature",     'J:\Research\EMD\Temperature\JGT.csv', "JGT";...
30     "Vernal Temperature",     'J:\Research\EMD\Temperature\VGT.csv', "VUT";];
31    %{
32     "Grand Junction Temperature", 'J:\Research\EMD\Temperature\GJT.csv', "GJT";...
33     "Flaming Gorge Temperature",  'J:\Research\EMD\Temperature\FGT.csv', "FGT";...
34     "Maybel Temperature",     'J:\Research\EMD\Temperature\MGT.csv', "MGT";...
35     "Craig Temperature",      'J:\Research\EMD\Temperature\CGT.csv', "CGT";];
36    %}
37
38  %The length function will only work if the matrix has more rows than
39  %columns
40
41  %for i=1:length(Temp1)
42  for i=1:2
43    [Tempdata(i).date, Tempdata(i).flow]=ReadGage(Temp1(i,2));
44  end
45
46
47
48  save('TempData.mat');
49
```

```
50
51
52  %Functions below
53  %------------------------------------------------------------------------
54
55
56  %Flow and Date importation
57  function [days, temp]=ReadGage(filename)
58  days=xlsread(filename, 'A:A'); %Importing Date data
59  temp=xlsread(filename, 'B:B'); %Importing Flow Data
60  end
61
```

A.11 Residual Code

```
1   %plot(Gagedata(1).date,Gagedata(1).resid)
2
3   Normalized=0;
4
5   Flow=1;
6   Temperature=1;
7   PrecipitationQuarter=1;
8   PrecipitationYearly=1;
```

```
9   SaveFigure=0;
10    if SaveFigure==1
11      Name='';
12    end
13
14  Color=1;
15
16  figure(700);
17    xlabel('Time (days)');
18    ylabel('Signal');
19    title('Residuals');
20
21  hold on
22  if Normalized==1
23  if Flow==1
24  for i=1:length(Gage1)
25
26    minF=min(Gagedata(i).resid);
27    maxF=max(Gagedata(i).resid);
28    rangeF=maxF-minF;
29    resF=Gagedata(i).resid;
30    %Next line normalizes data between 0 and 1
31
```

```
32    %Individual Line Plot Options
33    if Color==0
34    plot(Gagedata(i).date,((resF-minF)/rangeF),'DisplayName',Gage1(i,1))
35    elseif Color==1
36    %Color Option
37    plot(Gagedata(i).date,((resF-minF)/rangeF),'b','DisplayName',Gage1(i,1))
38    end
39  end
40  end
41
42
43  if Temperature==1
44  for y=1:length(Temp1)
45    %hold on
46    minT=min(Tempdata(y).resid);
47    maxT=max(Tempdata(y).resid);
48    rangeT=maxT-minT;
49    resT=Tempdata(y).resid;
50    %Individual Line Plot Options
51    if Color==0
52    plot(Tempdata(y).date,((resT-minT)/rangeT),'DisplayName',Temp1(y,1))
53    %Color Option
54    elseif Color==1
```

```
55    plot(Tempdata(y).date,((resT-minT)/rangeT),'r','DisplayName',Temp1(y,1))
56    %plot(Tempdata(y).date,Tempdata(y).resid,'DisplayName',Temp1(y,1))
57    end
58  end
59  end
60
61  if PrecipitationQuarter==1
62  %for j=1:length(Precip1)
63  for j=1:2
64    %hold on
65    minP=min(Precipdata(j).resid);
66    maxP=max(Precipdata(j).resid);
67    rangeP=maxP-minP;
68    resP=Precipdata(j).resid;
69    %Individual Line Plot Options
70    if Color==0
71    plot(Precipdata(j).date,((resP-minP)/rangeP),'DisplayName',Precip1(j,1))
72    %Color Option
73    elseif Color==1
74    plot(Precipdata(j).date,((resP-minP)/rangeP),'c','DisplayName',Precip1(j,1))
75    end
76  end
77  end
```

```
78
79  if PrecipitationYearly==1
80  %for k=1:length(PrecipY1)
81  for k=1:2
82   %hold on
83   minPy=min(PrecipYearlydata(k).resid);
84   maxPy=max(PrecipYearlydata(k).resid);
85   rangePy=maxPy-minPy;
86   resPy=PrecipYearlydata(k).resid;
87   %Individual Line Plot Options
88   if Color==0
89   plot(PrecipYearlydata(k).date,((resPy-minPy)/rangePy),'DisplayName',PrecipY1(k,1))
90   %Color Option
91   elseif Color==1
92   plot(PrecipYearlydata(k).date,((resPy-minPy)/rangePy),'o','DisplayName',PrecipY1(k,1))
93   end
94  end
95  end
96
97  else
98  if Flow==1
99  for i=1:length(Gage1)
100
```

```
101    minF=min(Gagedata(i).resid);
102    maxF=max(Gagedata(i).resid);
103    rangeF=maxF-minF;
104    resF=Gagedata(i).resid;
105    %Next line normalizes data between 0 and 1
106
107    %Individual Line Plot Options
108    if Color==0
109    plot(Gagedata(i).date,Gagedata(i).resid,'DisplayName',Gage1(i,1))
110    elseif Color==1
111    %Color Option
112    plot(Gagedata(i).date,Gagedata(i).resid,'b','DisplayName',Gage1(i,1))
113    end
114  end
115  end
116
117
118  if Temperature==1
119  %for y=1:length(Temp1)
120  for y=1:2
121    %hold on
122    minT=min(Tempdata(y).resid);
123    maxT=max(Tempdata(y).resid);
```

```
124    rangeT=maxT-minT;
125    resT=Tempdata(y).resid;
126    %Individual Line Plot Options
127    if Color==0
128    plot(Tempdata(y).date,Tempdata(y).resid,'DisplayName',Temp1(y,1))
129    %Color Option
130    elseif Color==1
131    plot(Tempdata(y).date,Tempdata(y).resid,'r','DisplayName',Temp1(y,1))
132    %plot(Tempdata(y).date,Tempdata(y).resid,'DisplayName',Temp1(y,1))
133    end
134  end
135  end
136
137  if PrecipitationQuarter==1
138  %for j=1:length(Precip1)
139  for j=1:2
140    %hold on
141    minP=min(Precipdata(j).resid);
142    maxP=max(Precipdata(j).resid);
143    rangeP=maxP-minP;
144    resP=Precipdata(j).resid;
145    %Individual Line Plot Options
146    if Color==0
```

```
147    plot(Precipdata(j).date,Precipdata(j).resid,'DisplayName',Precip1(j,1))
148    %Color Option
149    elseif Color==1
150    plot(Precipdata(j).date,Precipdata(j).resid,'c','DisplayName',Precip1(j,1))
151    end
152  end
153  end
154
155  if PrecipitationYearly==1
156  %for k=1:length(PrecipY1)
157  for k=1:2
158    %hold on
159    minPy=min(PrecipYearlydata(k).resid);
160    maxPy=max(PrecipYearlydata(k).resid);
161    rangePy=maxPy-minPy;
162    resPy=PrecipYearlydata(k).resid;
163    %Individual Line Plot Options
164    if Color==0
165    plot(PrecipYearlydata(k).date,PrecipYearlydata(k).resid,'DisplayName',PrecipY1(k,1))
166    %Color Option
167    elseif Color==1
168
plot(PrecipYearlydata(k).date,PrecipYearlydata(k).resid,'o','DisplayName',PrecipY1(k,1))
```

```
169    end
170  end
171  end
172  if SaveFigure==1
173    Folder='J:\Research\Writing\Figures\Custom Plots\';
174    Tar=strcat(Folder,Name);
175    print(gcf,Tar,'-dpng','-r1000')
176  end
177  end
178
179  hold off
180  legend;
181  dateFormat=11;
182  datetick('x',dateFormat);
```

APPENDIX B. PROCESSED DATASETS BY LOCATION AND DATA

B.1 Flaming Gorge Flow Gage

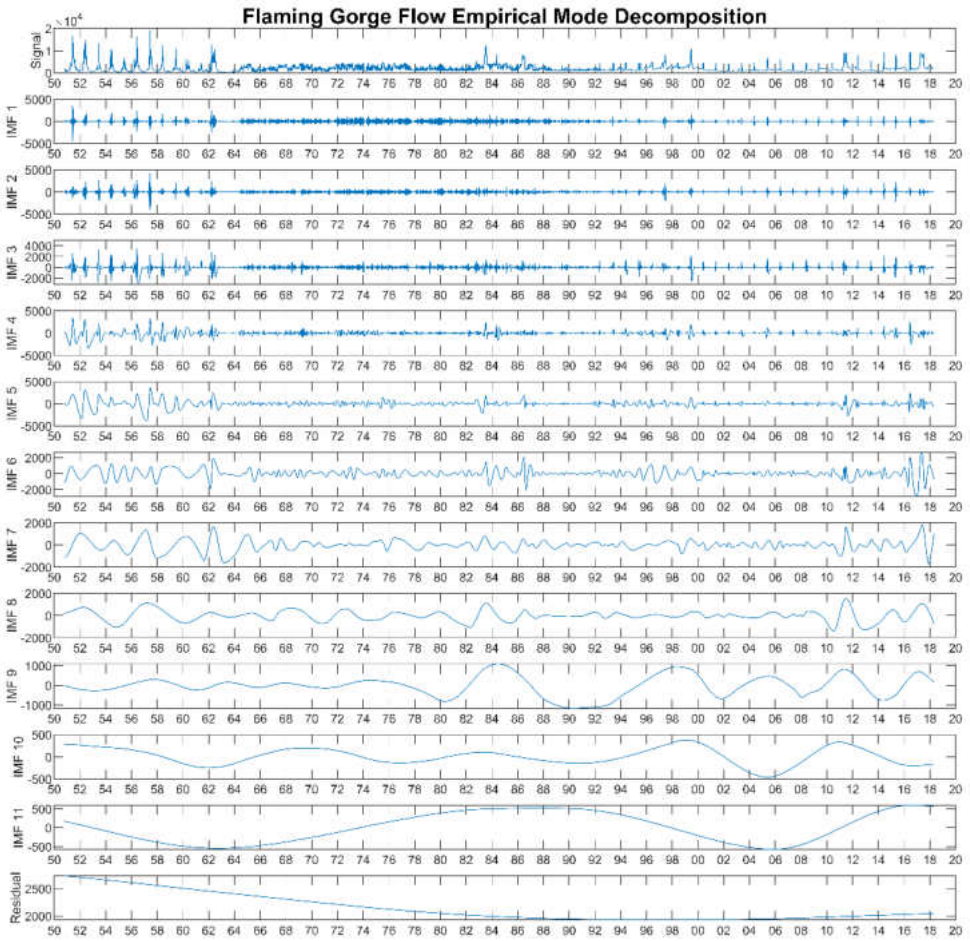

Figure B.1-1:Flaming Gorge Flow Empirical Mode Decomposition

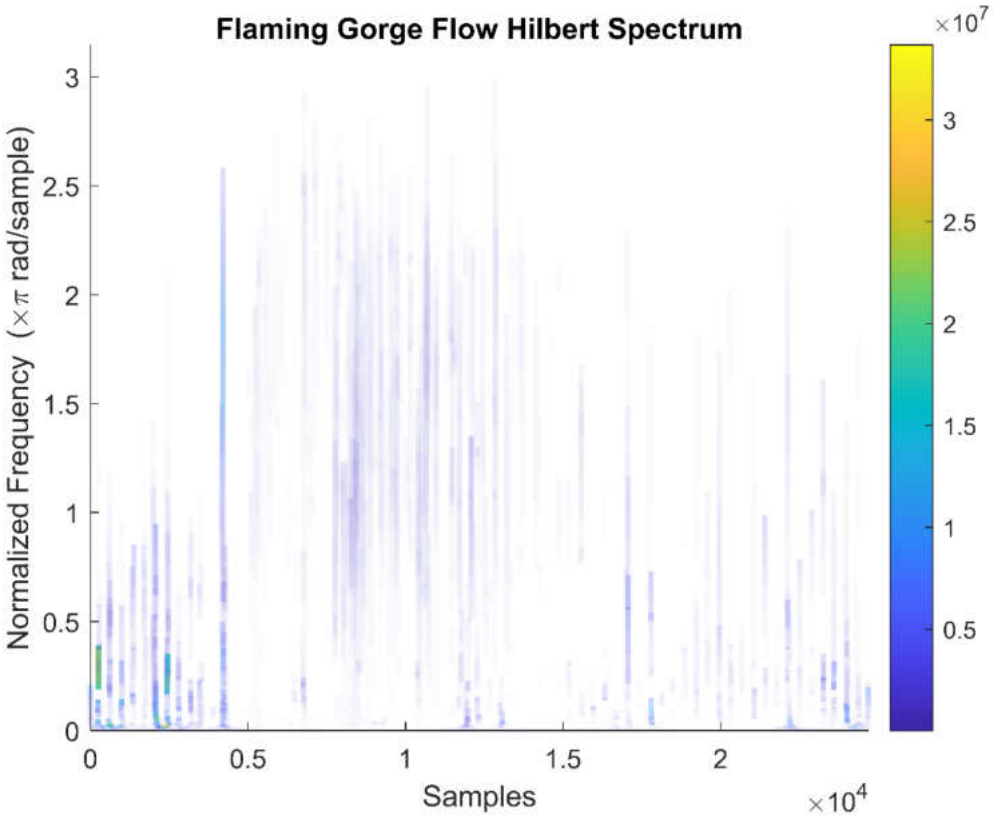

Figure B.1-2:Flaming Gorge Flow HHT

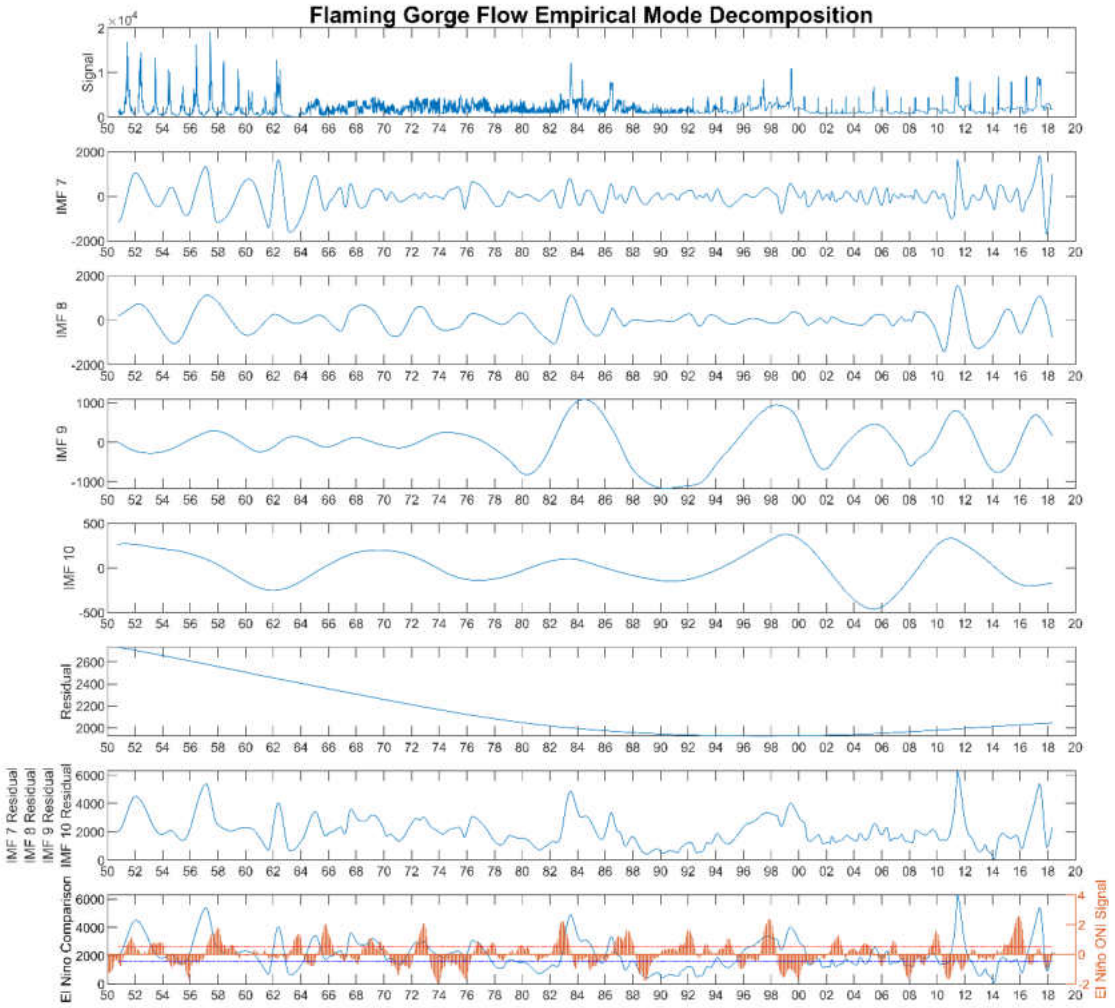

Figure B.1-3:Flaming Gorge Flow Empirical Mode Decomposition IMF 7,8,9 and 10

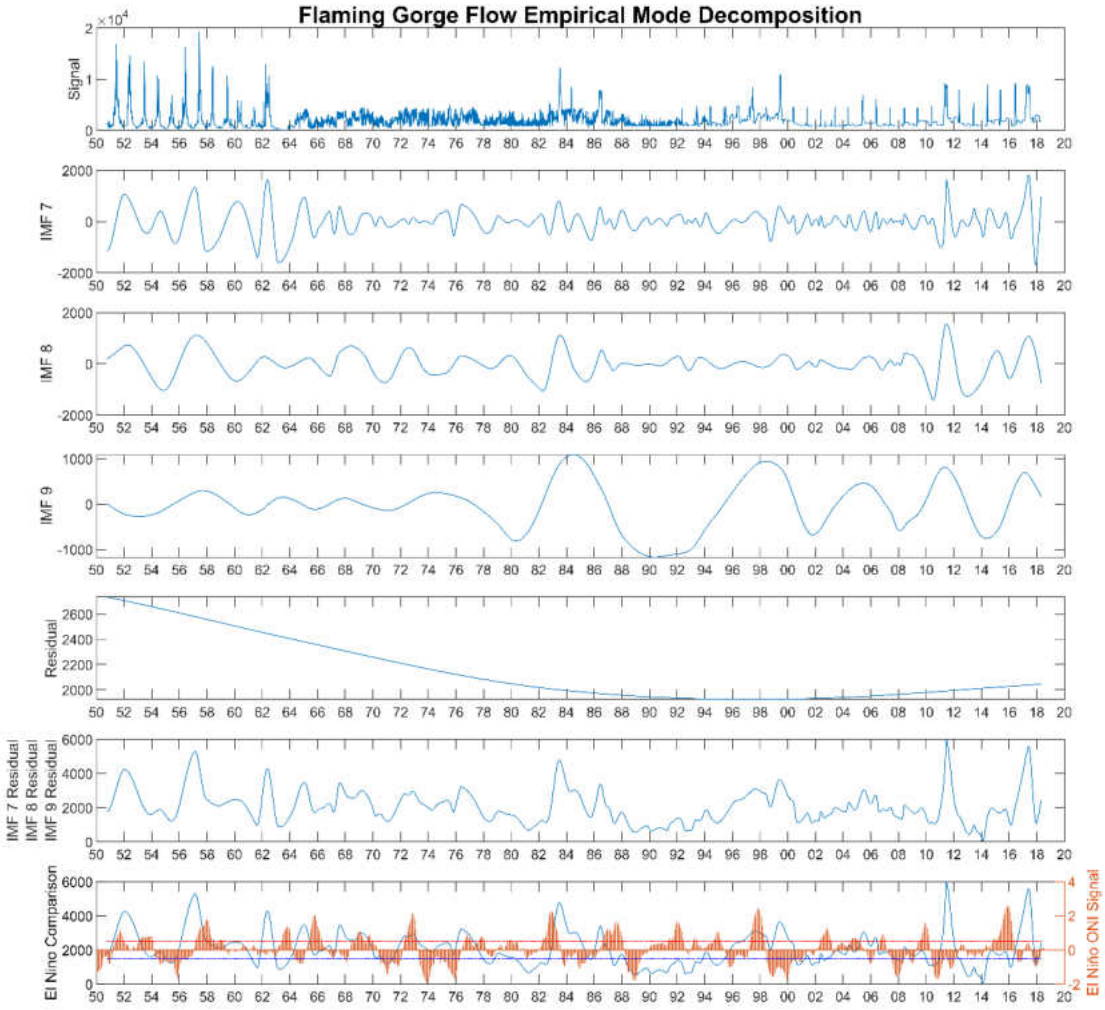

Figure B.1-4:Flaming Gorge Flow Empirical Mode Decomposition Residual Superimposed on IMF 7 8 and 9

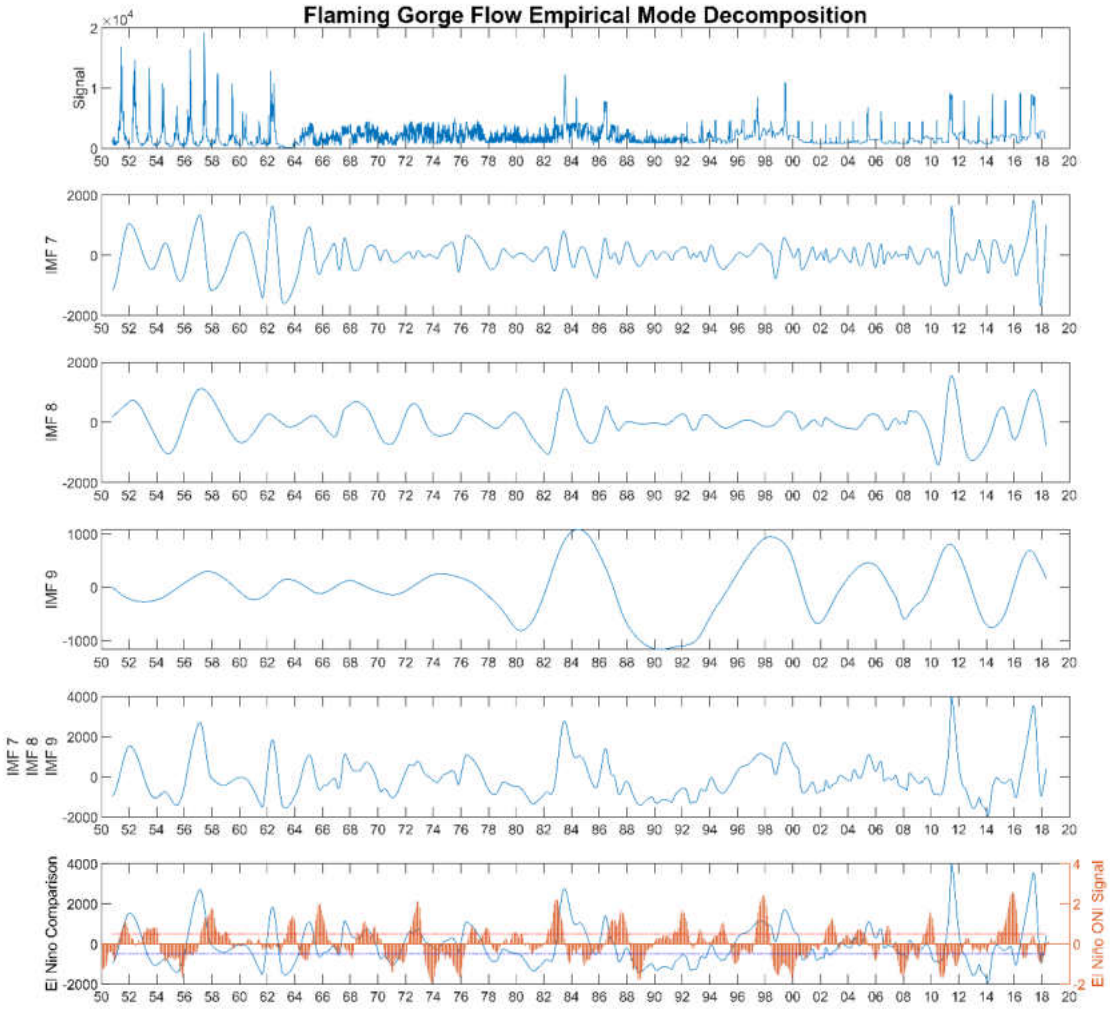

Figure B.1-5:Flaming Gorge Flow Empirical Mode Decomposition IMF 7 8 and 9

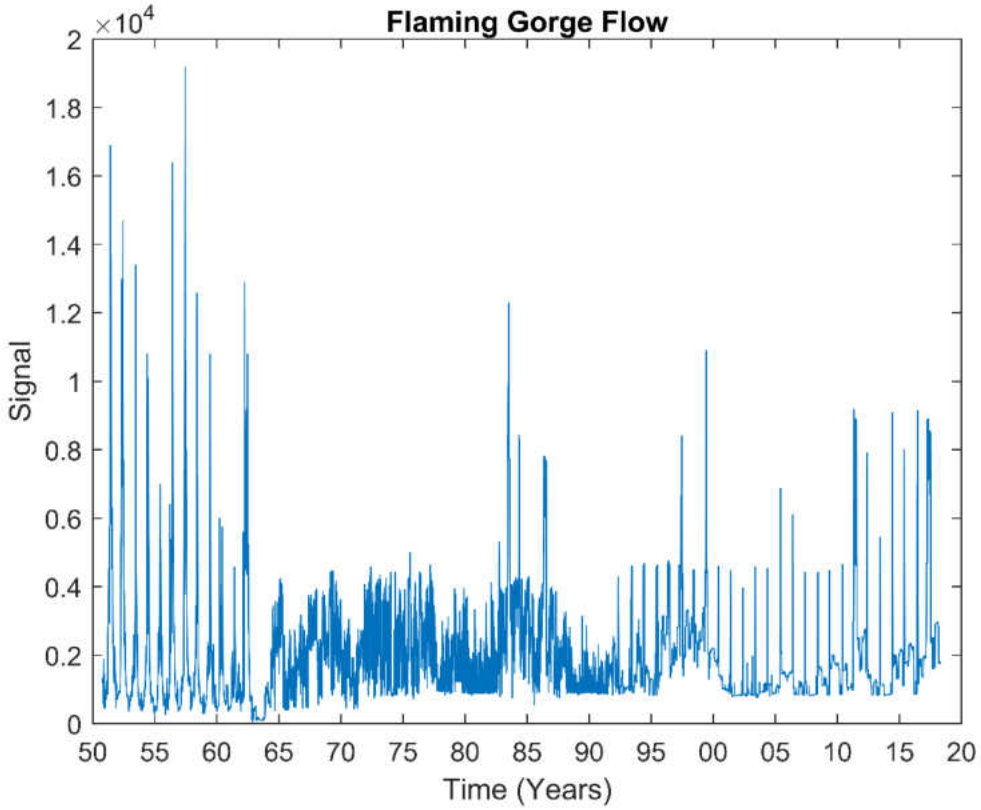

Figure B.1-6:Flaming Gorge Flow

B.2 Jensen Flow Gage

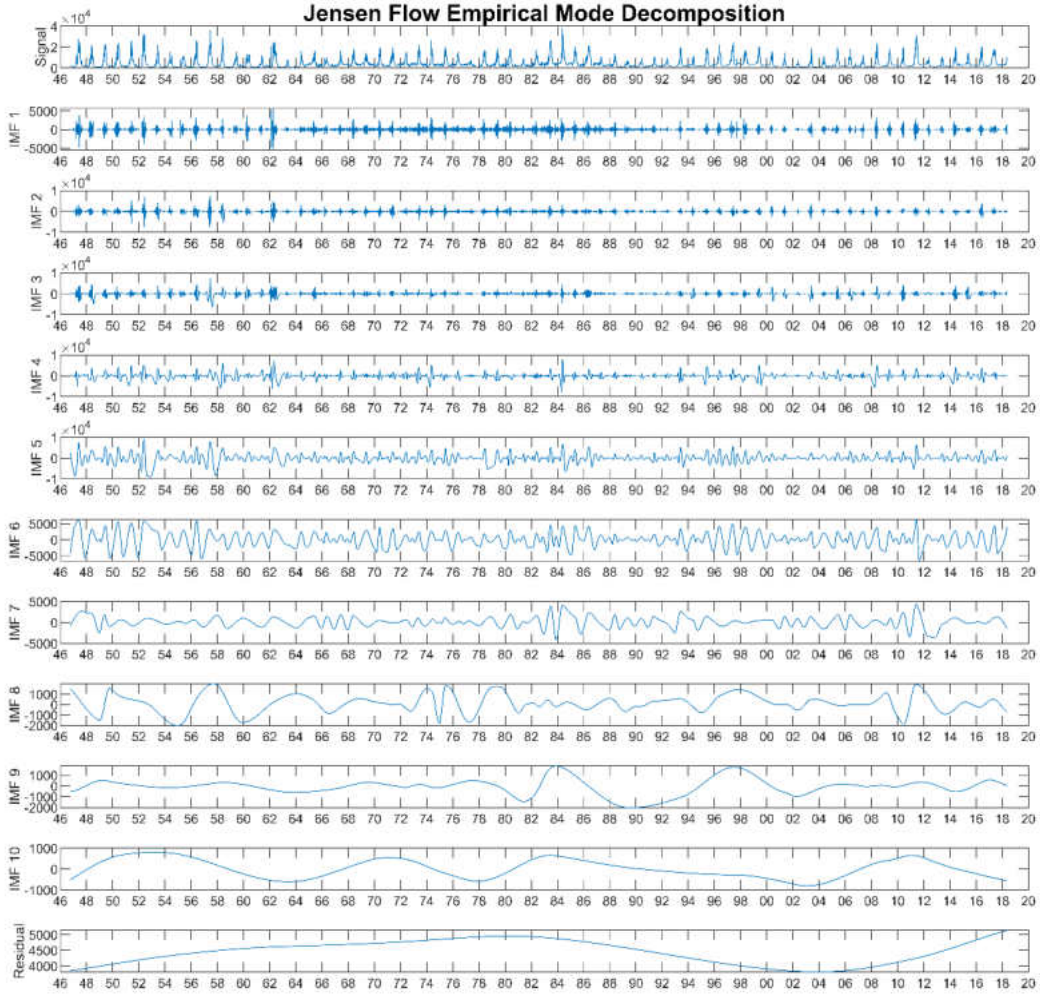

Figure B.2-1:Jensen Flow Empirical Mode Decomposition

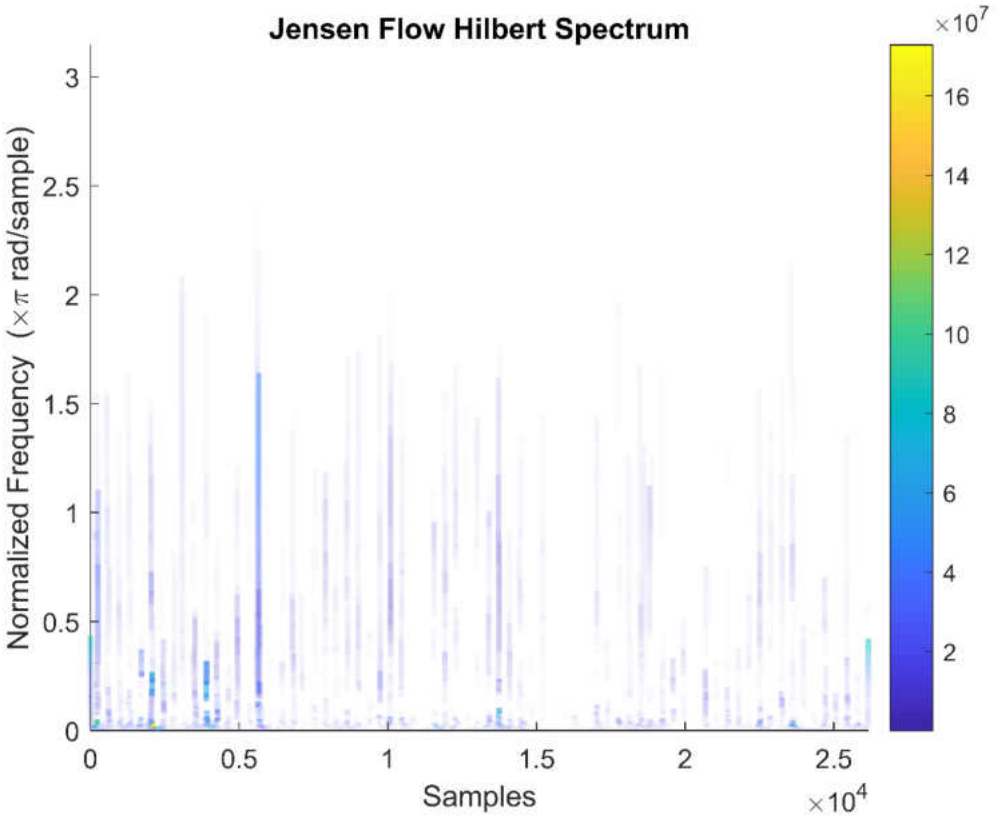

Figure B.2-2:Jensen Flow Empirical Mode Decomposition HHT

Figure B.2-3:Jensen Empirical Mode Decomposition IMF 7 and 8

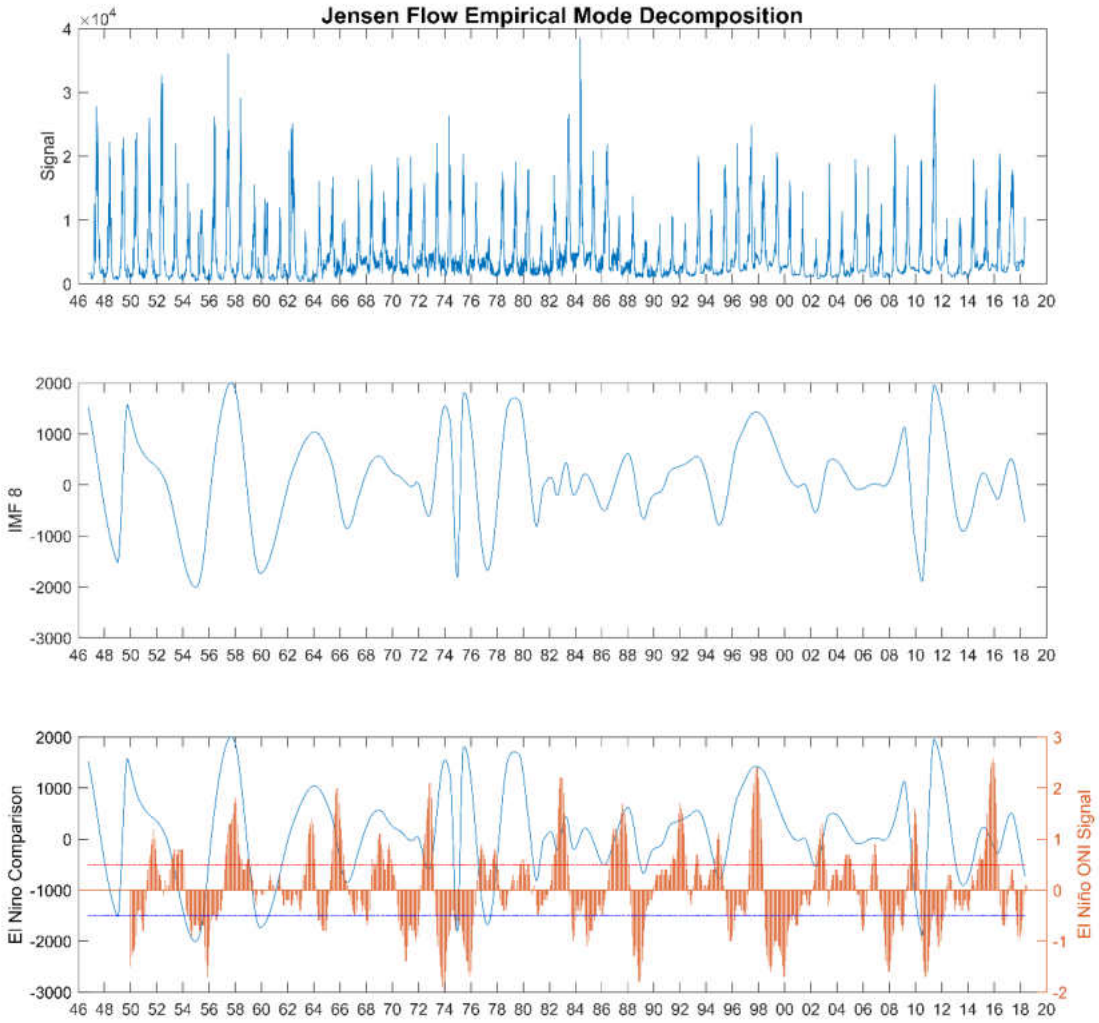

Figure B.2-4:Jensen Empirical Mode Decomposition IMF 8

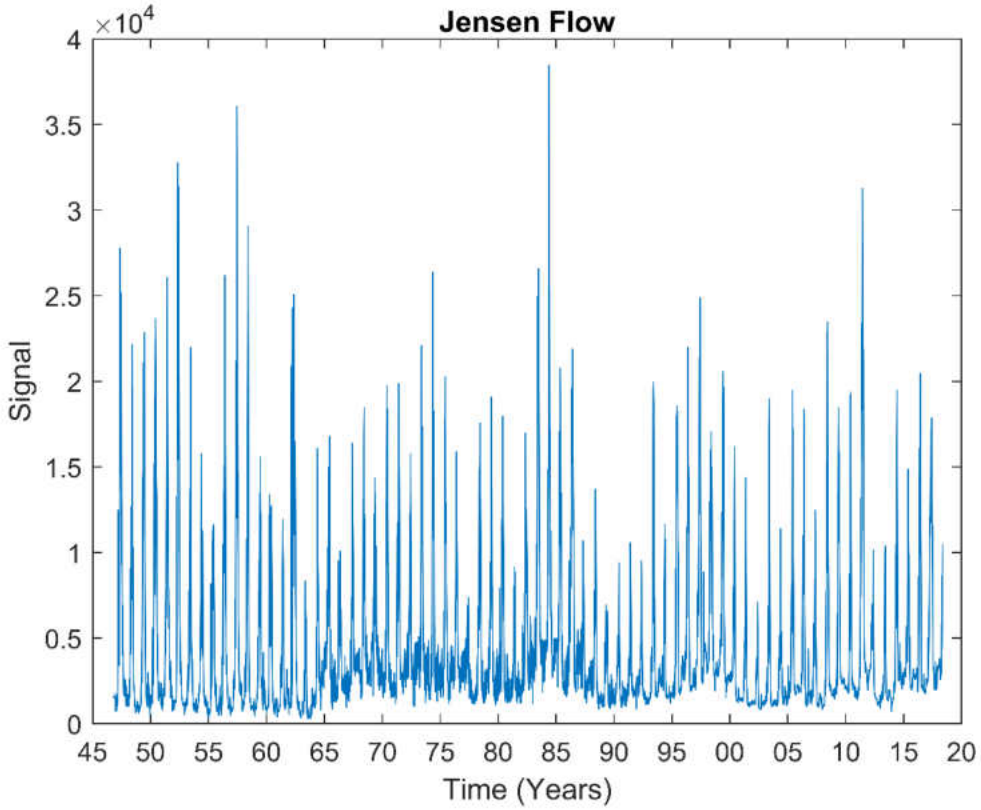

Figure B.2-5:Jensen Flow Data

B.3 Little Snake- Lily Flow

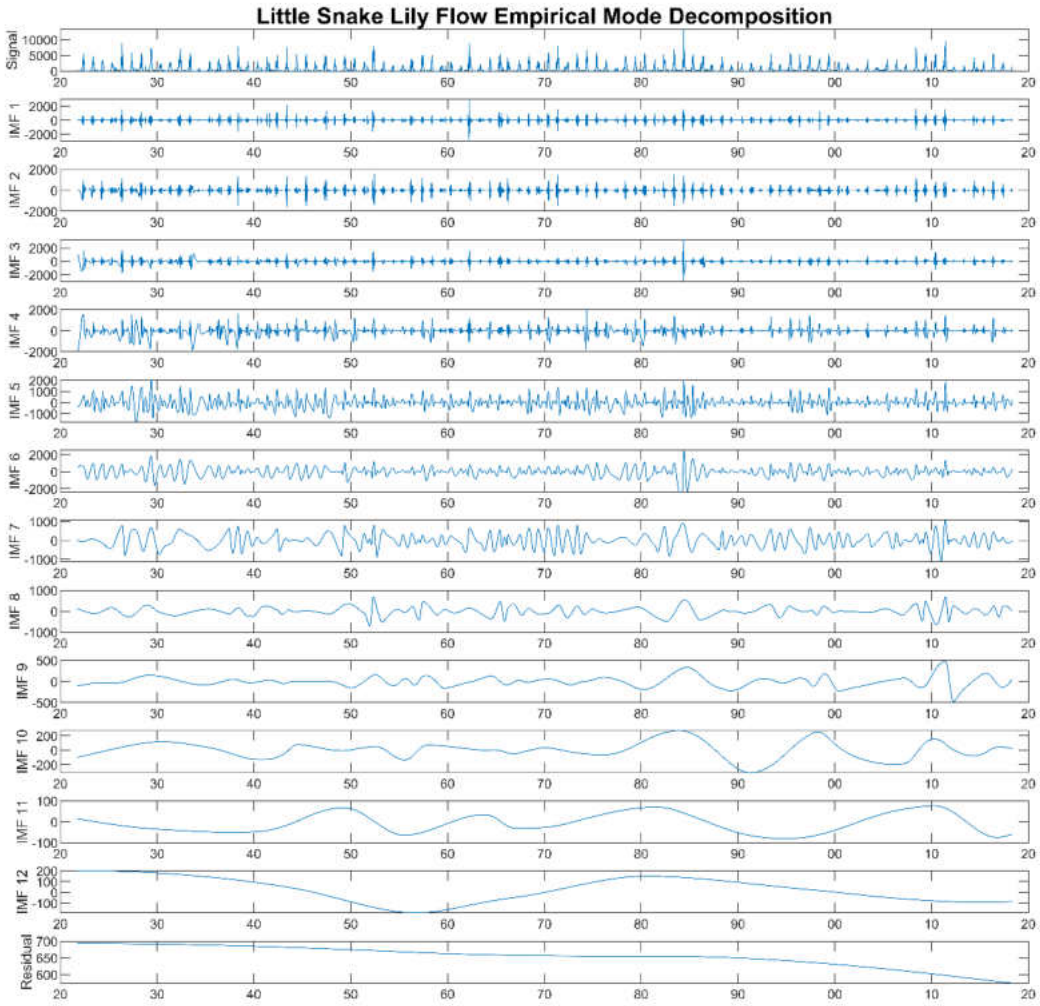

Figure B.3-1:Little Snake Lily Empirical Mode Decomposition

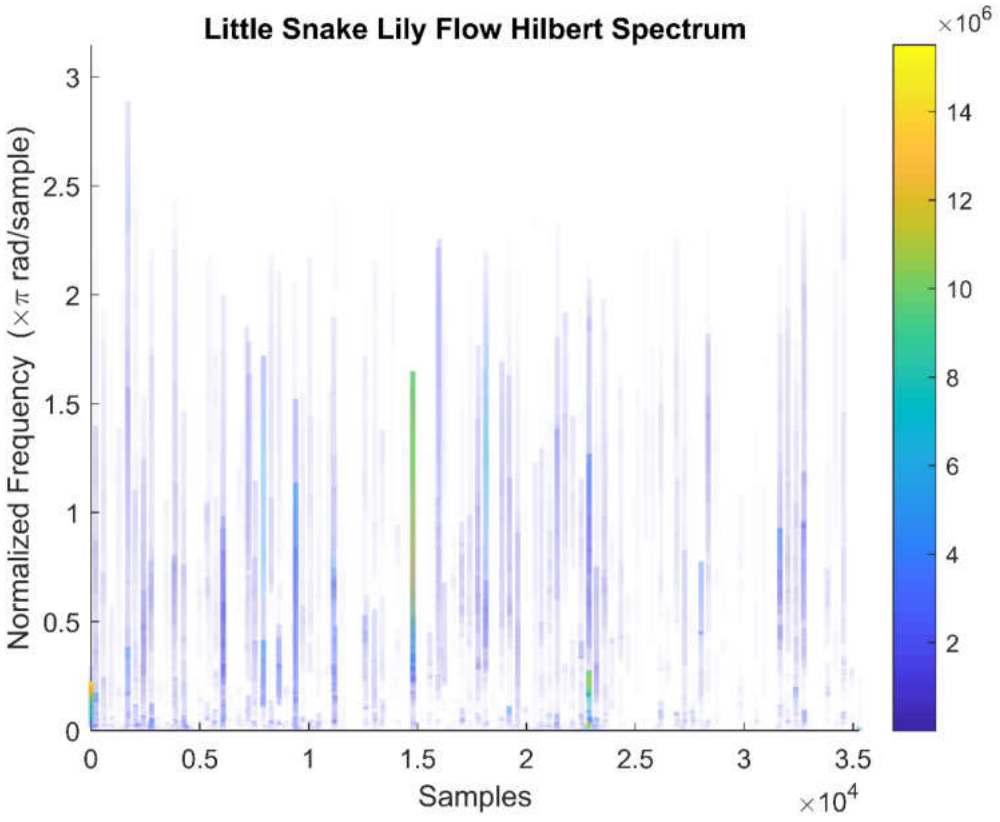

Figure B.3-2: Little Snake Lily Empirical Mode Decomposition HHT

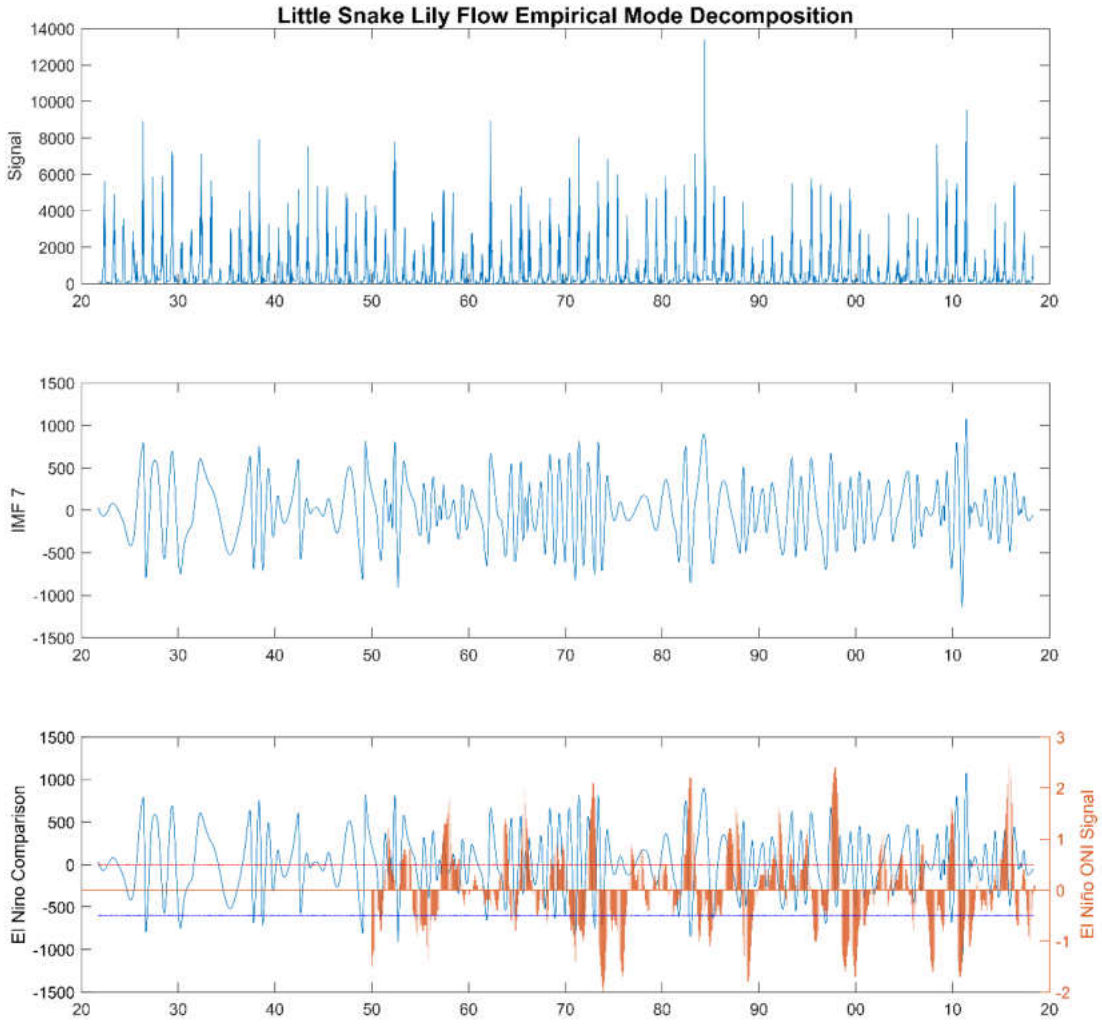

Figure B.3-3: Little Snake Lily Empirical Mode Decomposition IMF 7

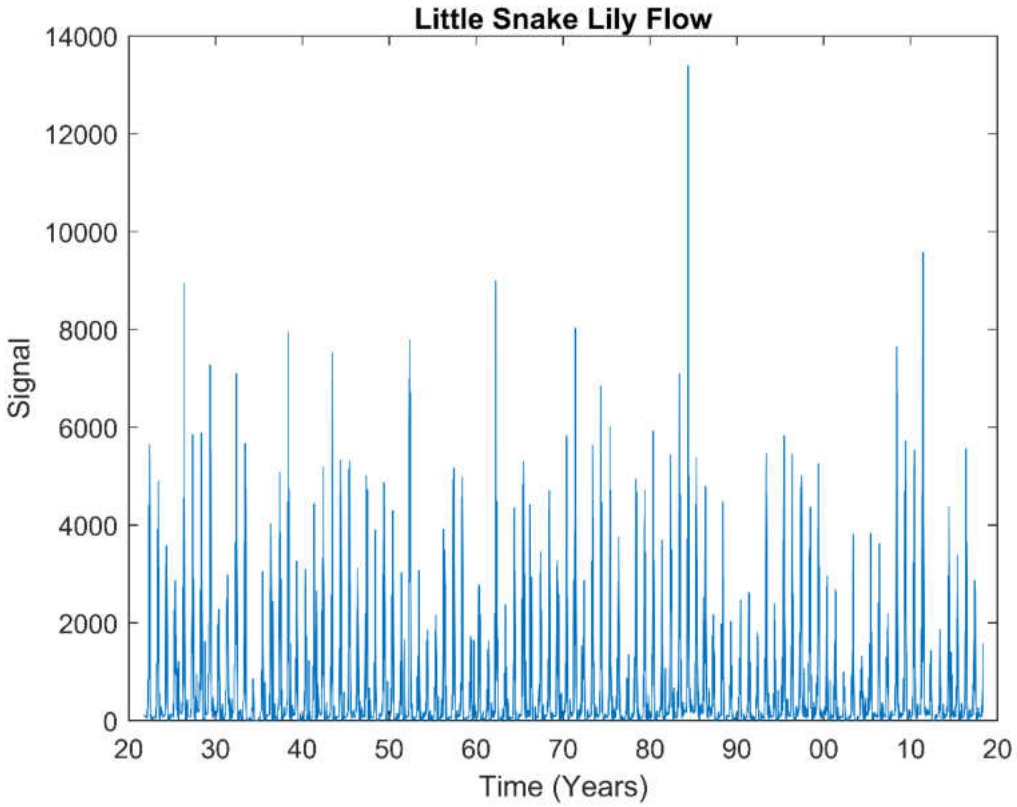

Figure B.3-4: Little Snake Lily Flow

B.4 Yampa Maybel Flow

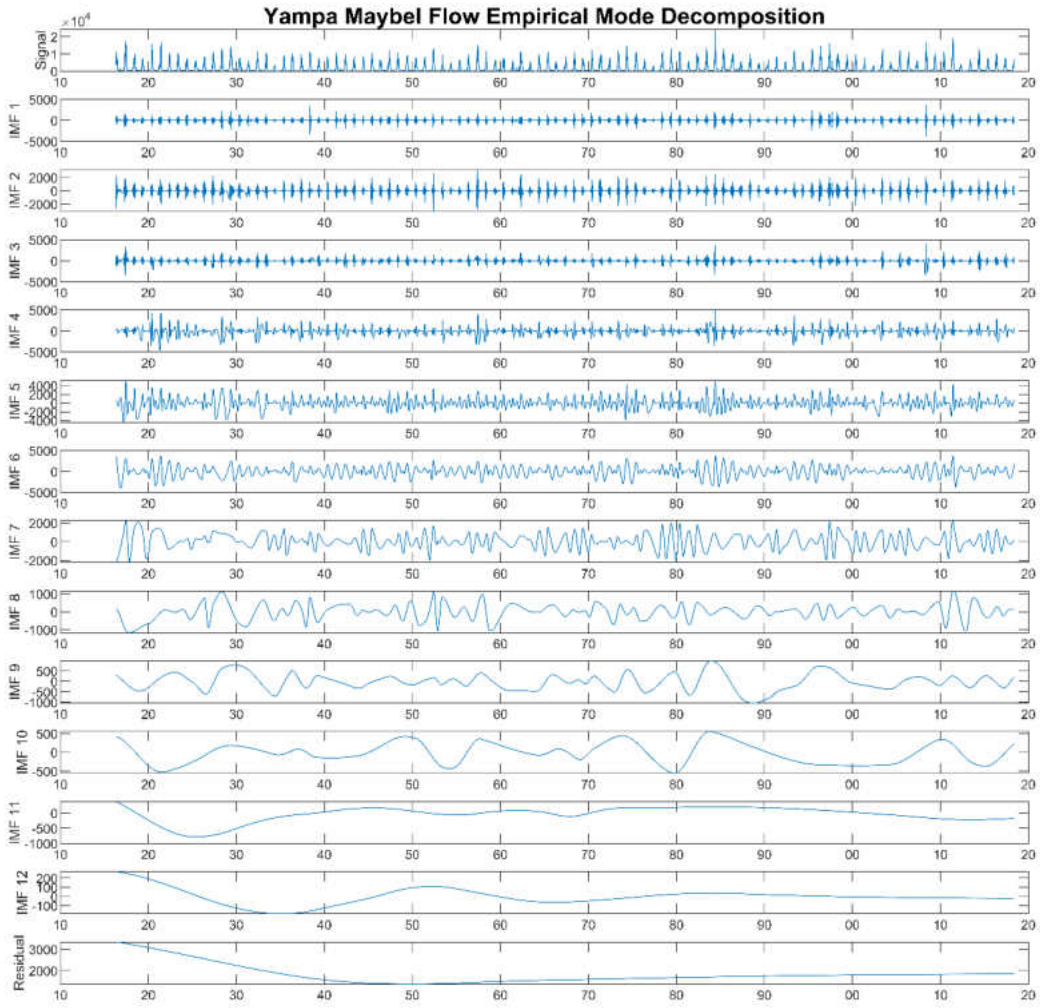

Figure B.4-1: Yampa Maybel Empirical Mode Decomposition

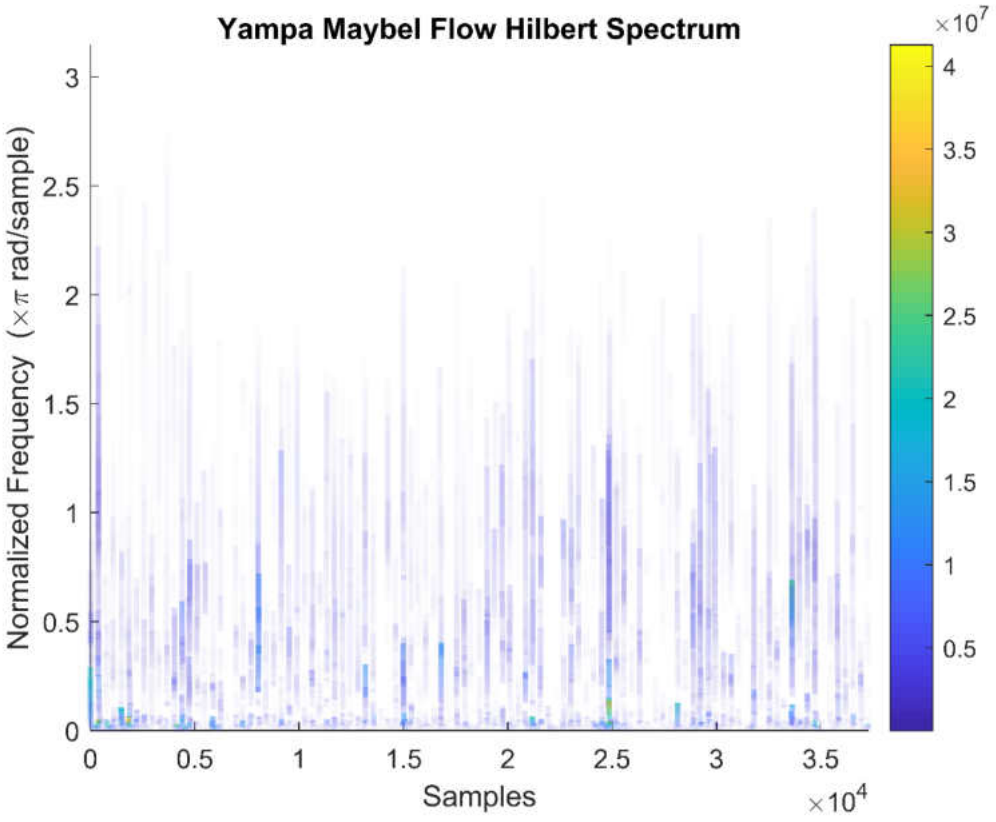

Figure B.4-2: Yampa Maybel Empirical Mode Decomposition HHT

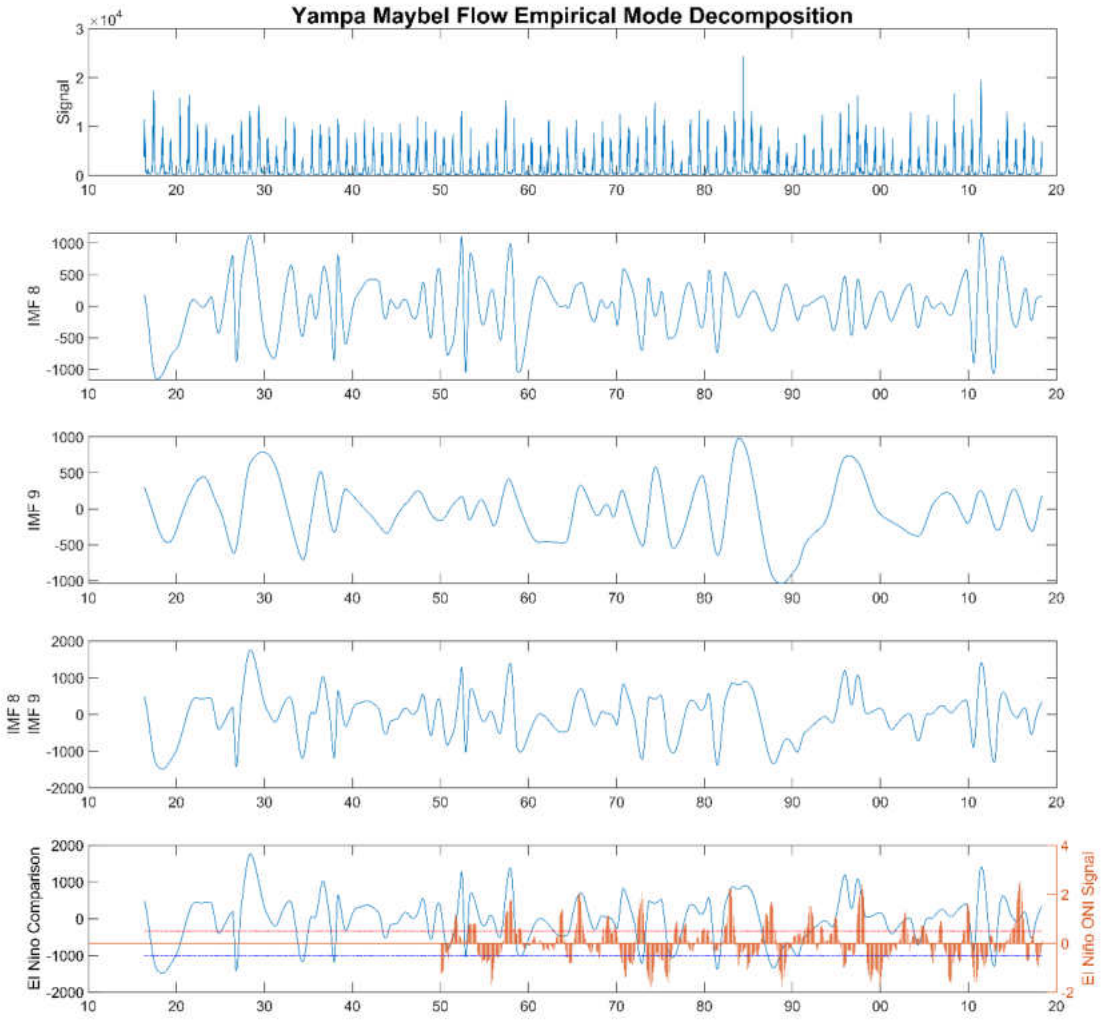

Figure B.4-3: Yampa Maybel Empirical Mode Decomposition IMF 8 and 9

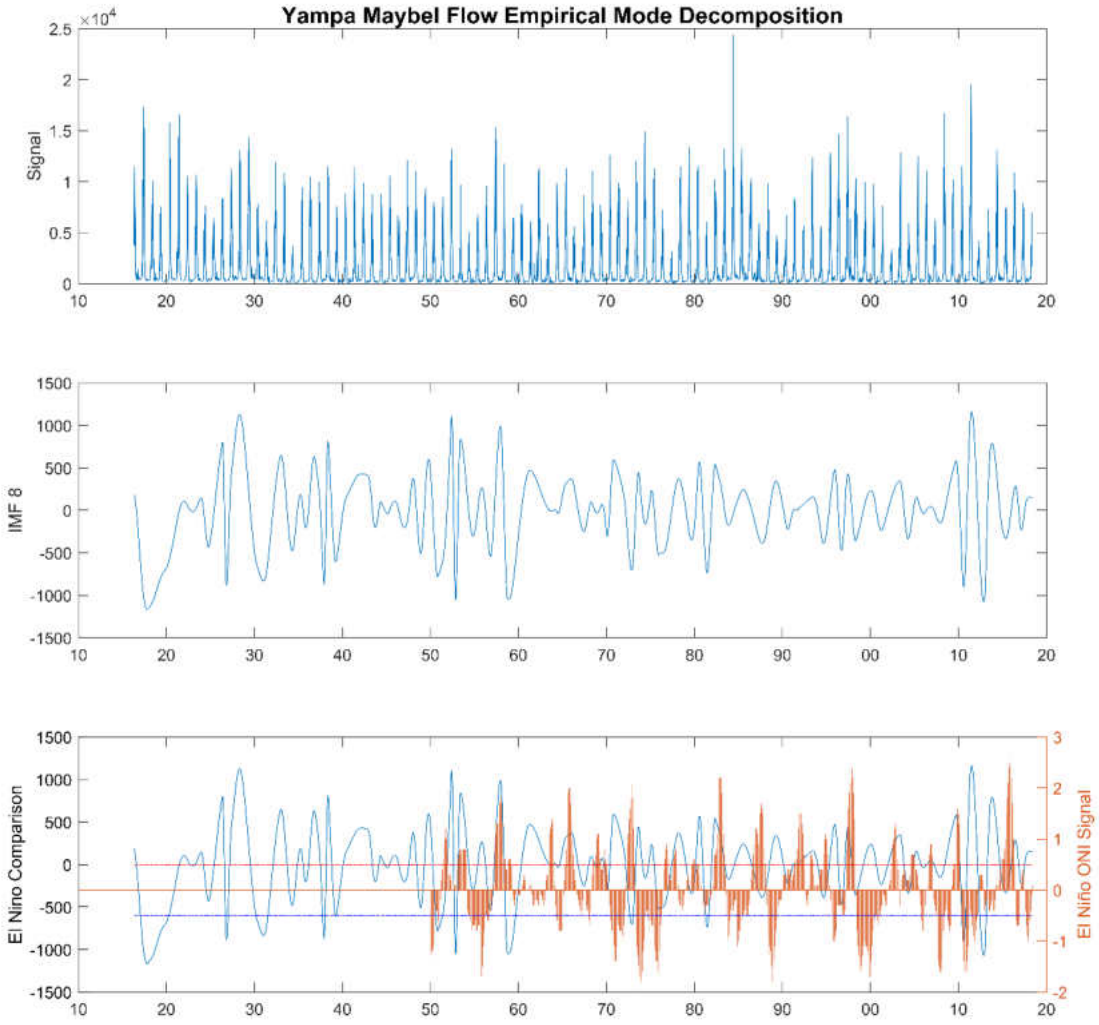

Figure B.4-4: Yampa Maybel Empirical Mode Decomposition IMF 8

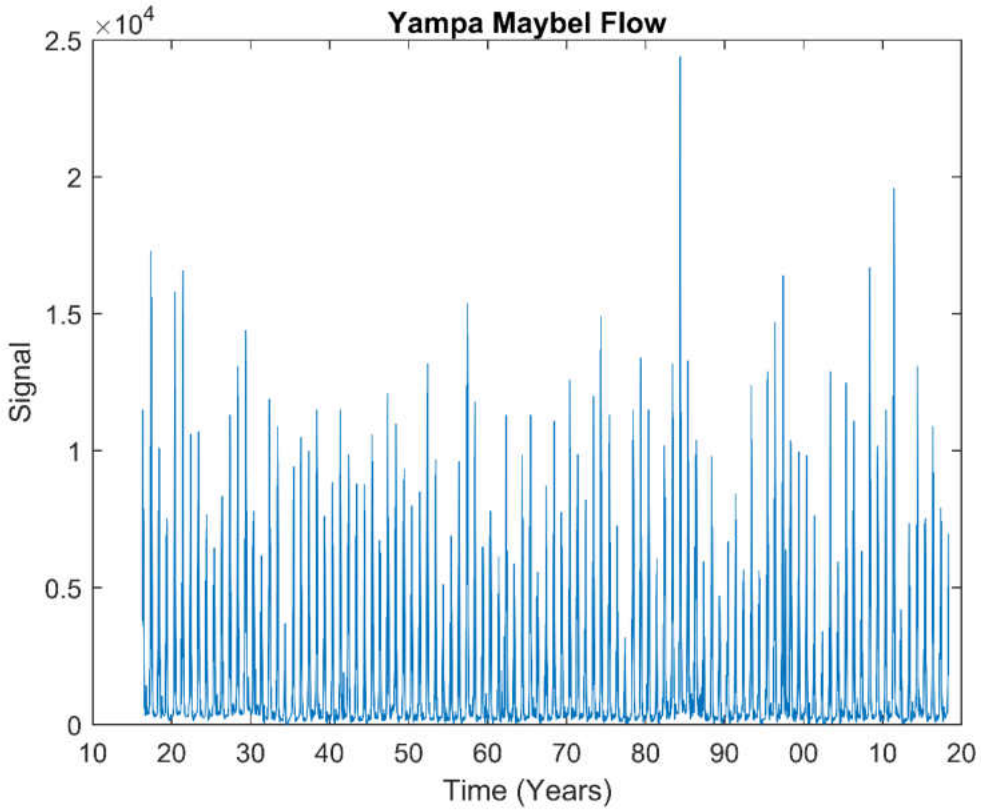

Figure B.4-5: Yampa Maybel Flow

B.5 Ourey Quarterly Precipitation

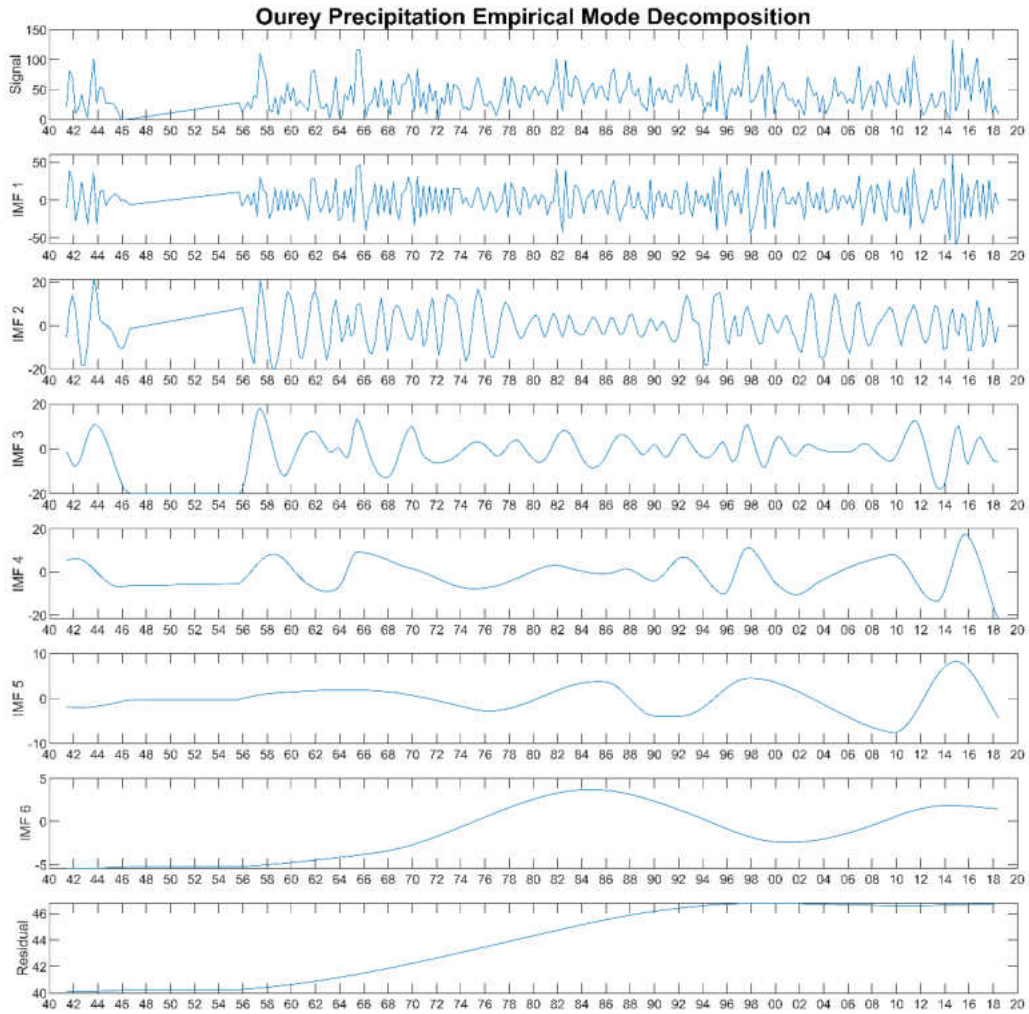

Figure B.5-1: Ourey Quarterly Precipitation Empirical Mode Decomposition

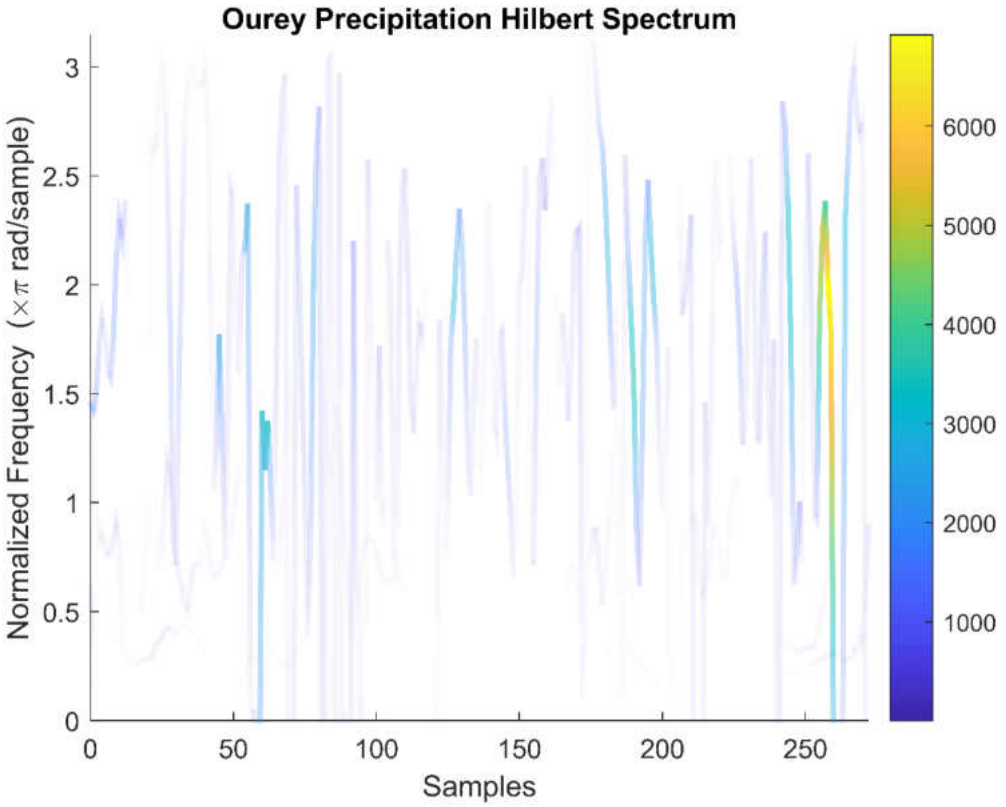

Figure B.5-2: Ourey Quarterly Precipitation Empirical Mode Decomposition HHT

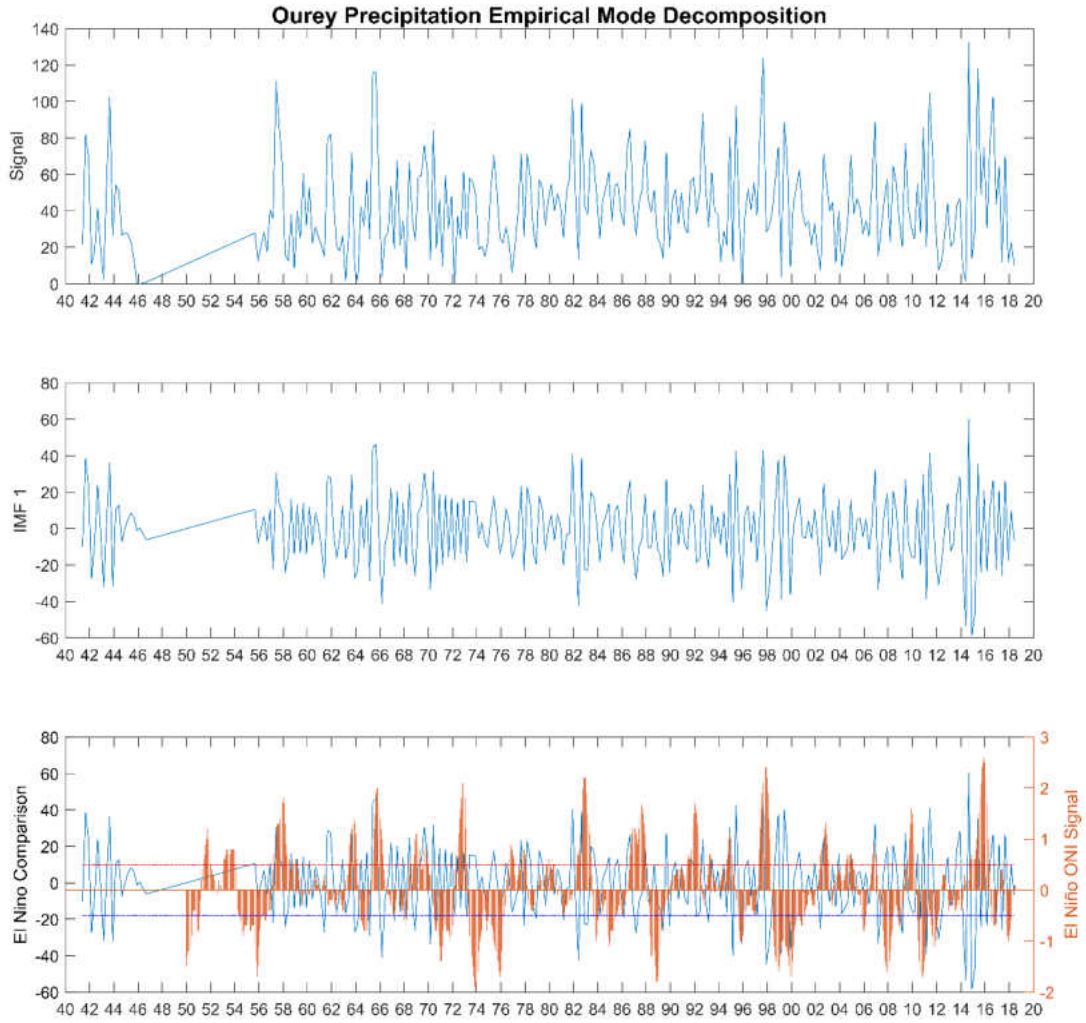

Figure B.5-3: Ourey Quarterly Precipitation Empirical Mode Decomposition IMF 1

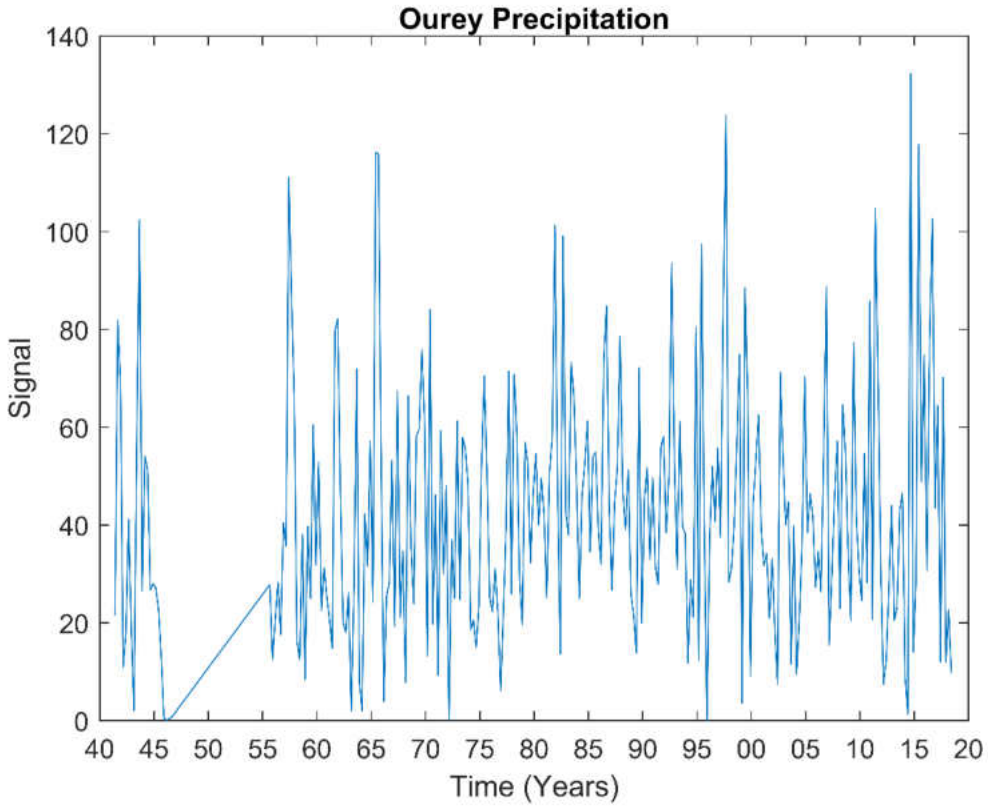

Figure B.5-4: Ourey Quarterly Precipitation

B.6 Jensen Quarterly Precipitation

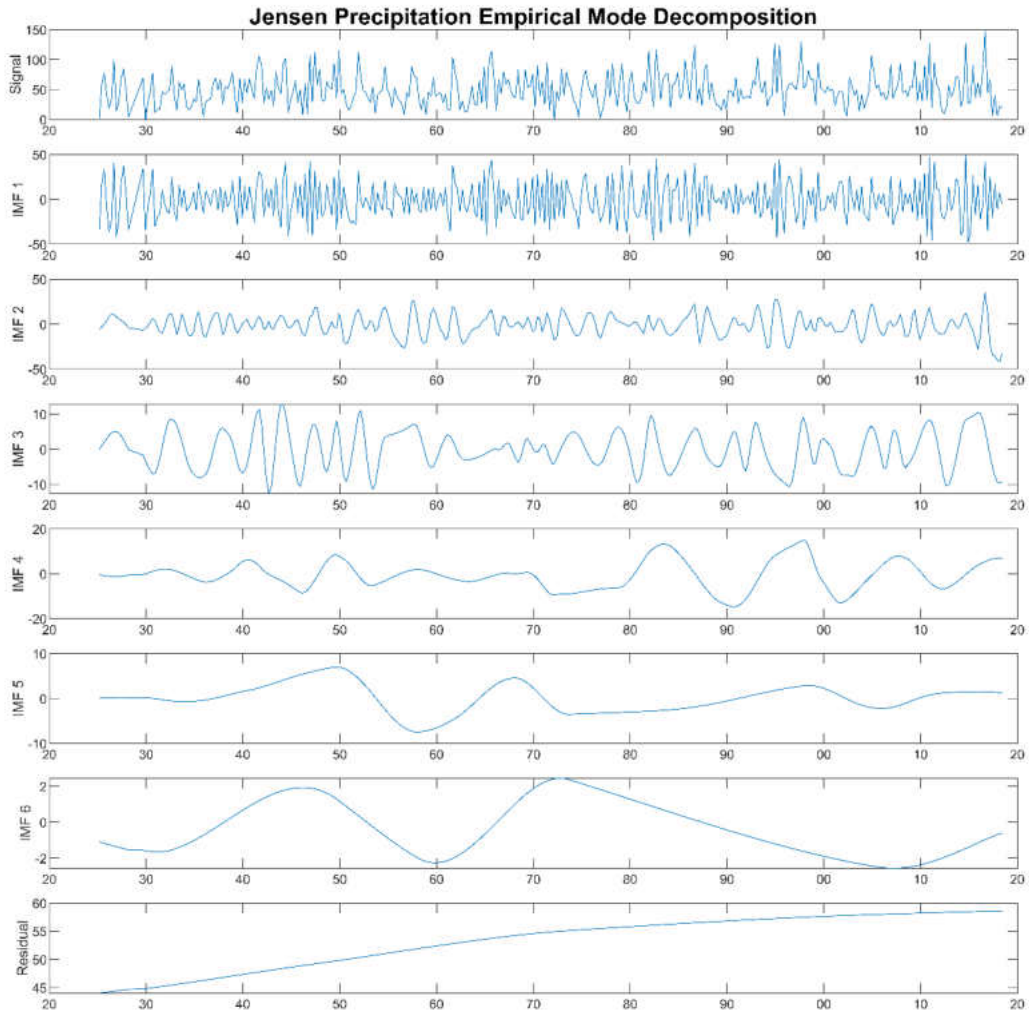

Figure B.6-1: Jensen Quarterly Precipitation Empirical Mode Decomposition

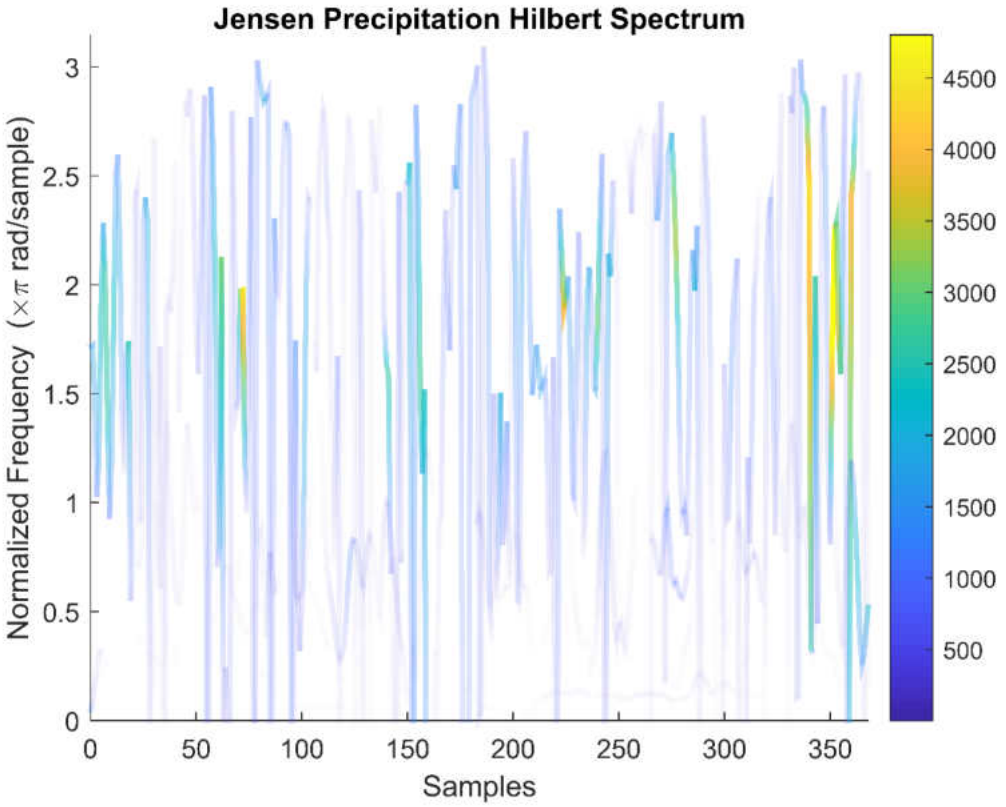

Figure B.6-2: Jensen Quarterly Precipitation Empirical Mode Decomposition HHT

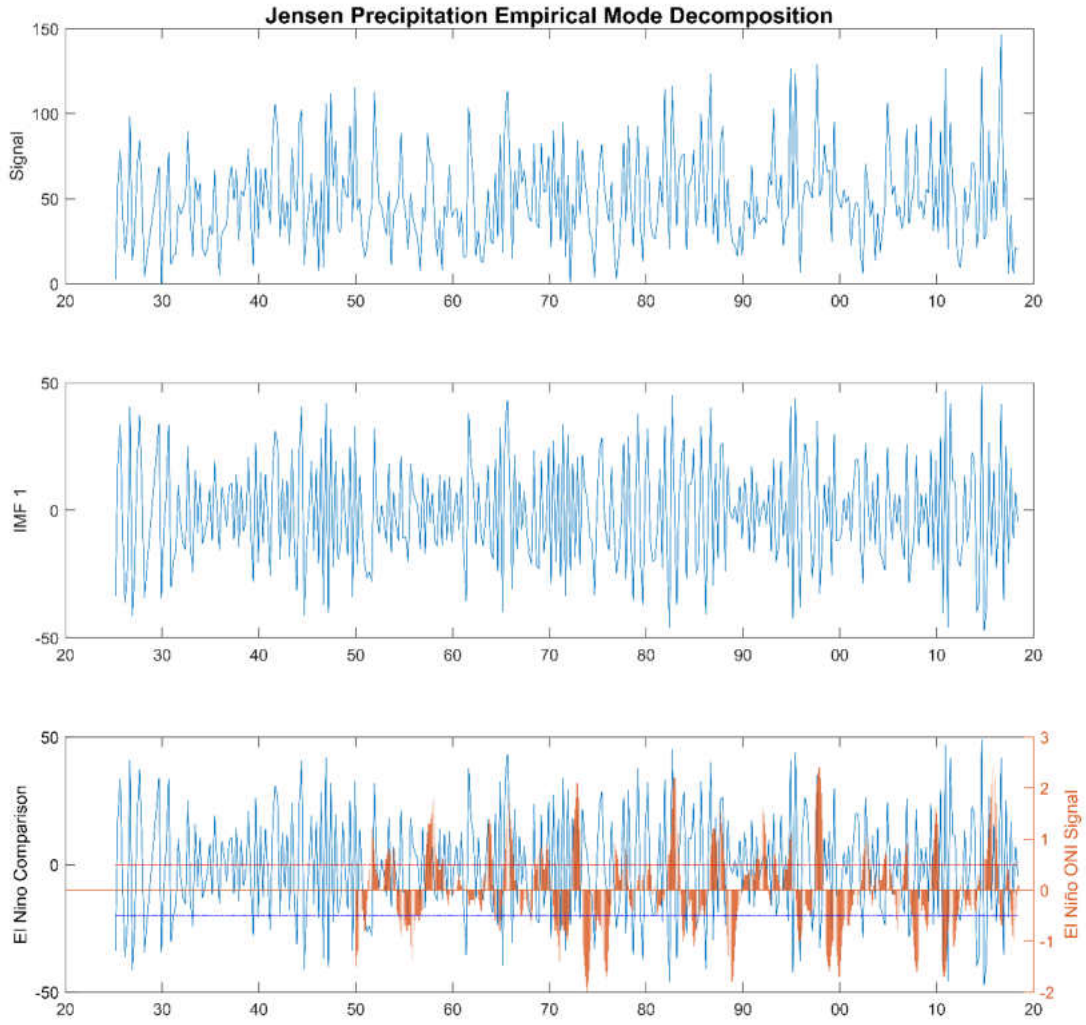

Figure B.6-3: Jensen Quarterly Precipitation Empirical Mode Decomposition IMF 1

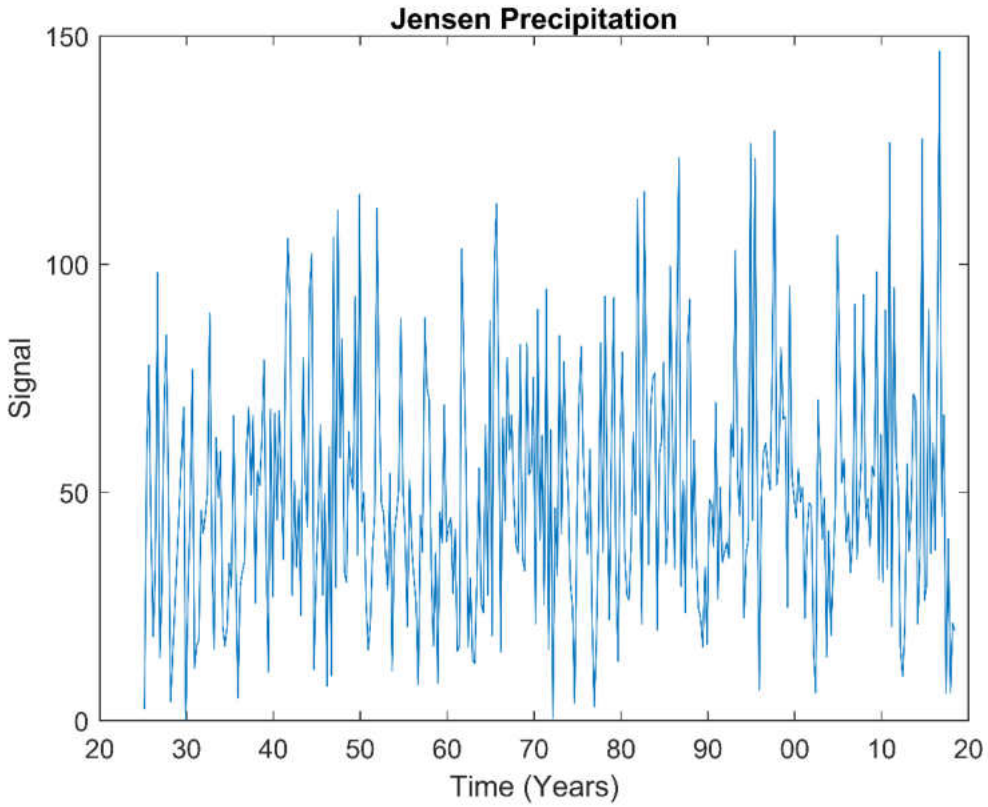

Figure B.6-4: Jensen Quarterly Precipitation

B.7 Ouray Yearly Precipitation

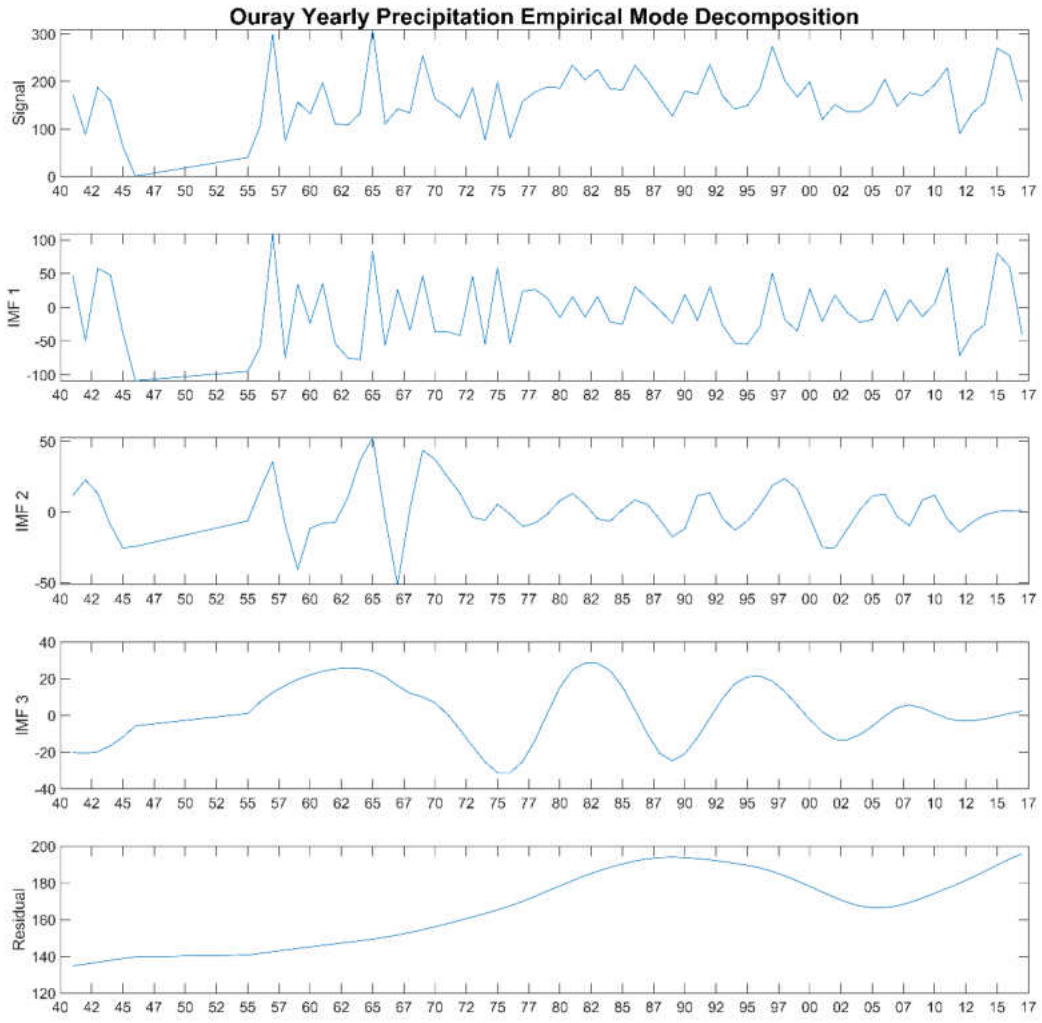

Figure B.7-1: Ouray Yearly Precipitation Empirical Mode Decomposition

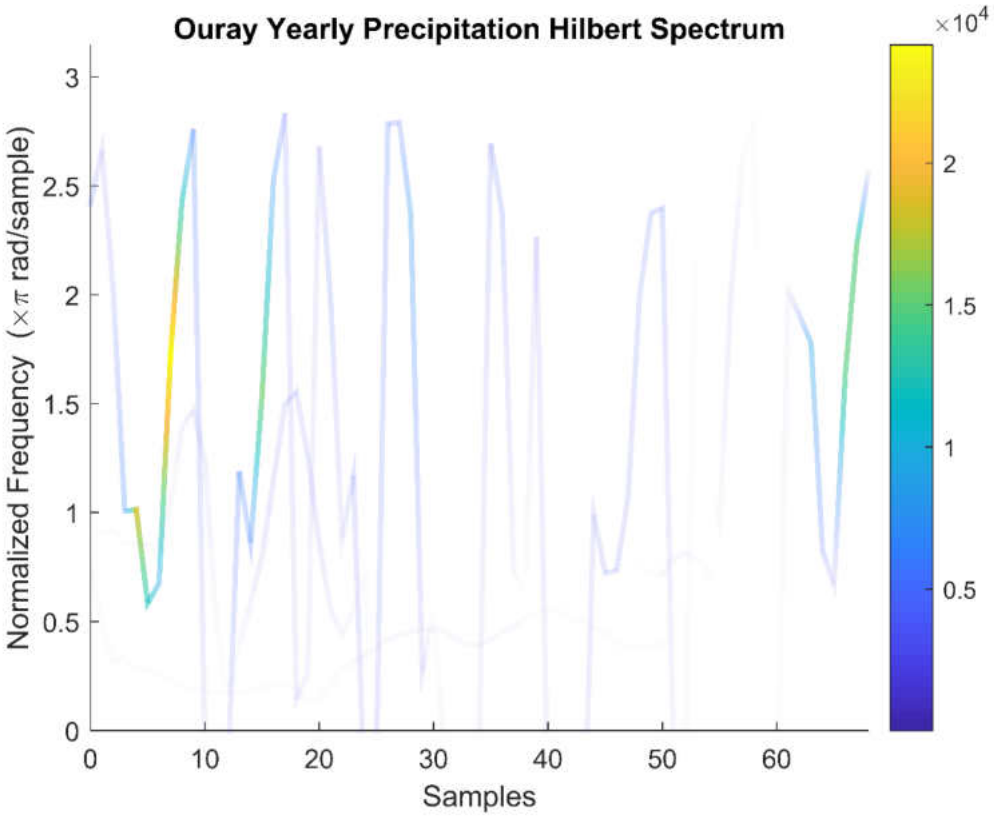

Figure B.7-2: Ouray Yearly Precipitation Empirical Mode Decomposition HHT

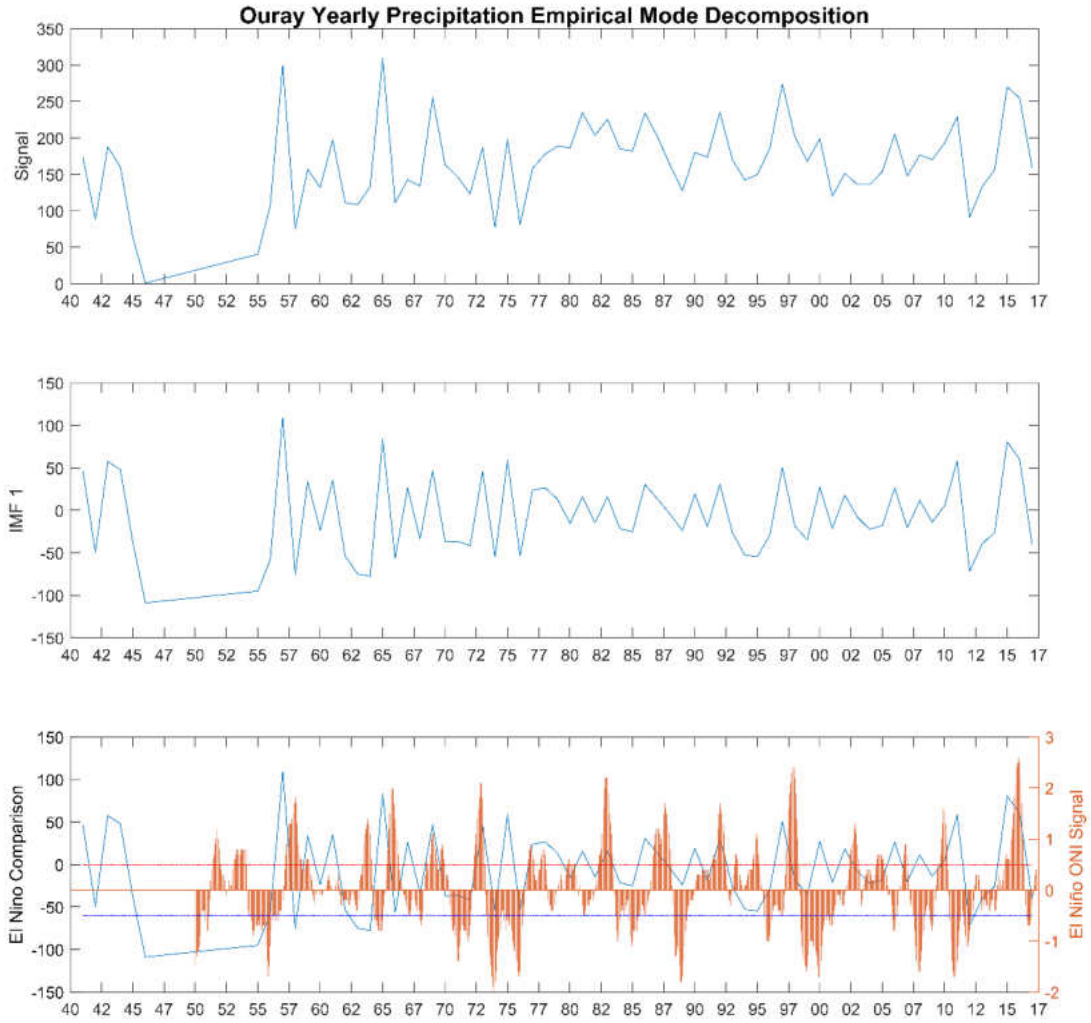

Figure B.7-3: Ouray Yearly Precipitation Empirical Mode Decomposition IMF 1

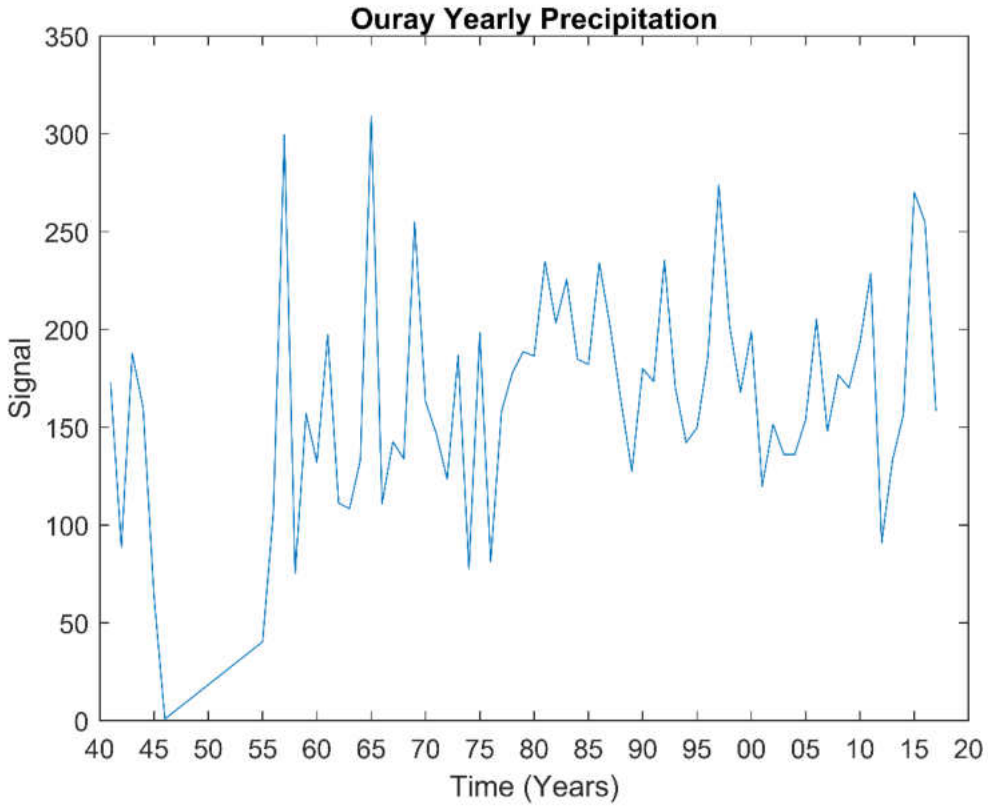

Figure B.7-4: Ouray Yearly Precipitation

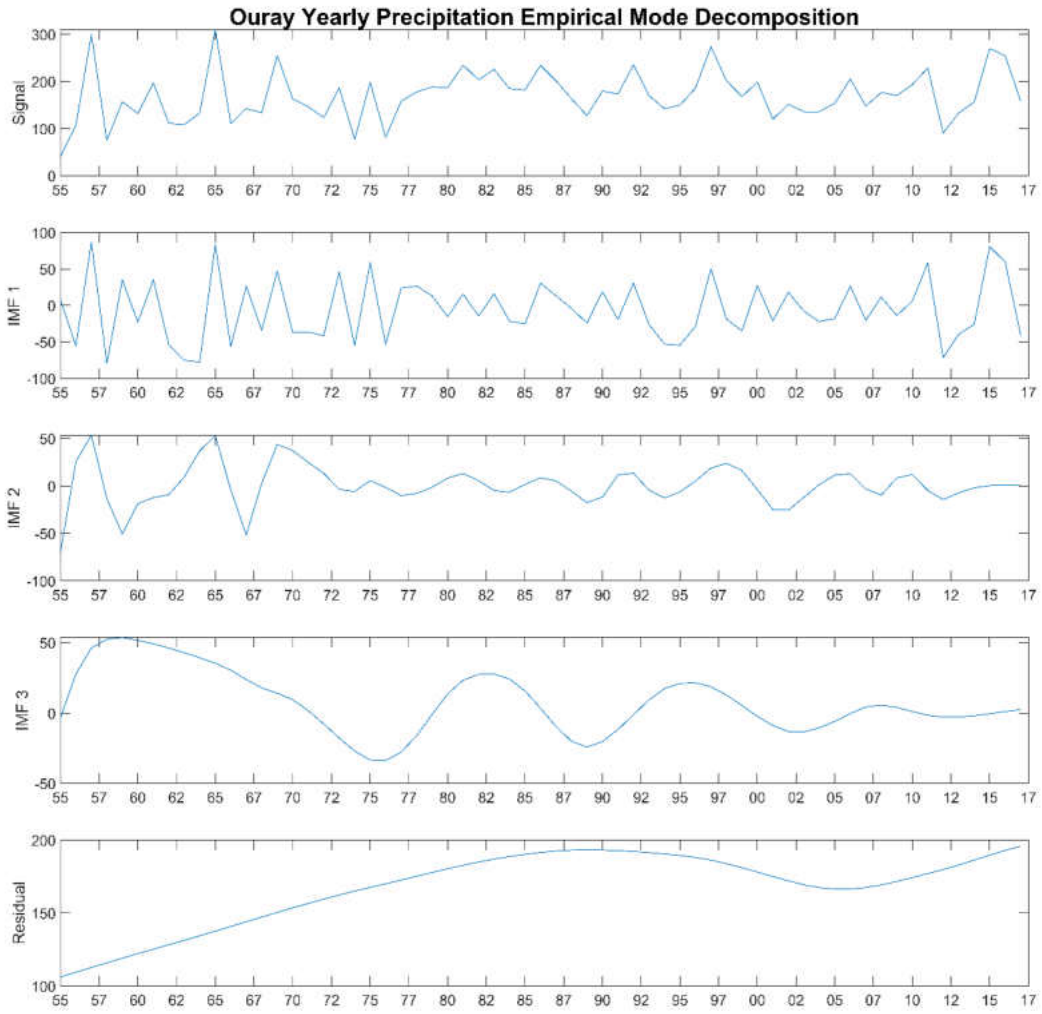

Figure B.7-5: Ouray Yearly Precipitation Empirical Mode Decomposition Shortened to Account for Missing Data

B.8 Jensen Yearly Precipitation

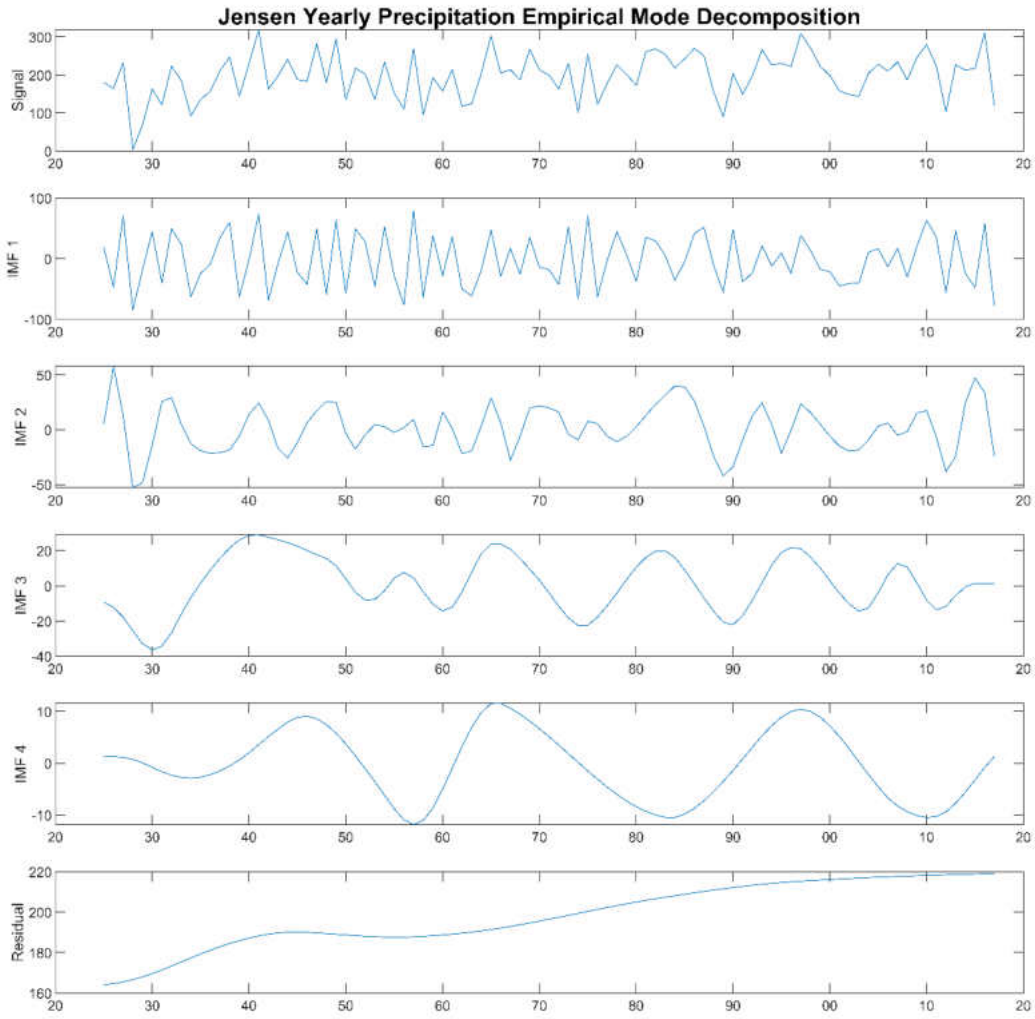

Figure B.8-1: Jensen Yearly Precipitation Empirical Mode Decomposition

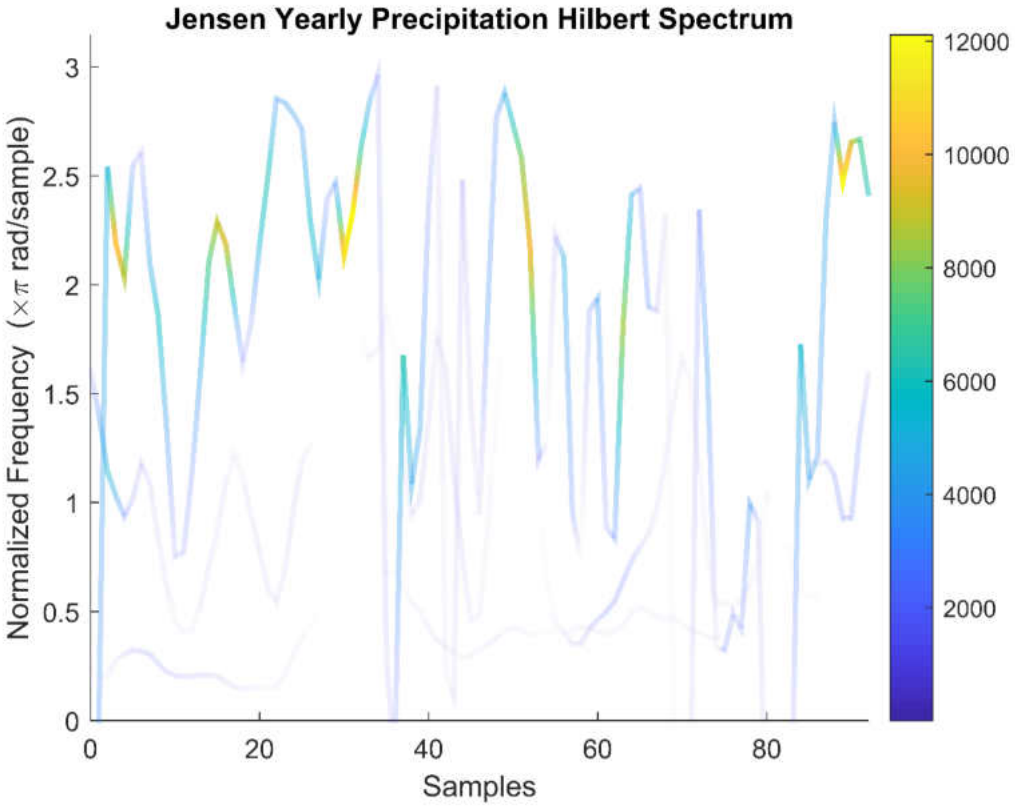

Figure B.8-2: Jensen Yearly Precipitation Empirical Mode Decomposition HHT

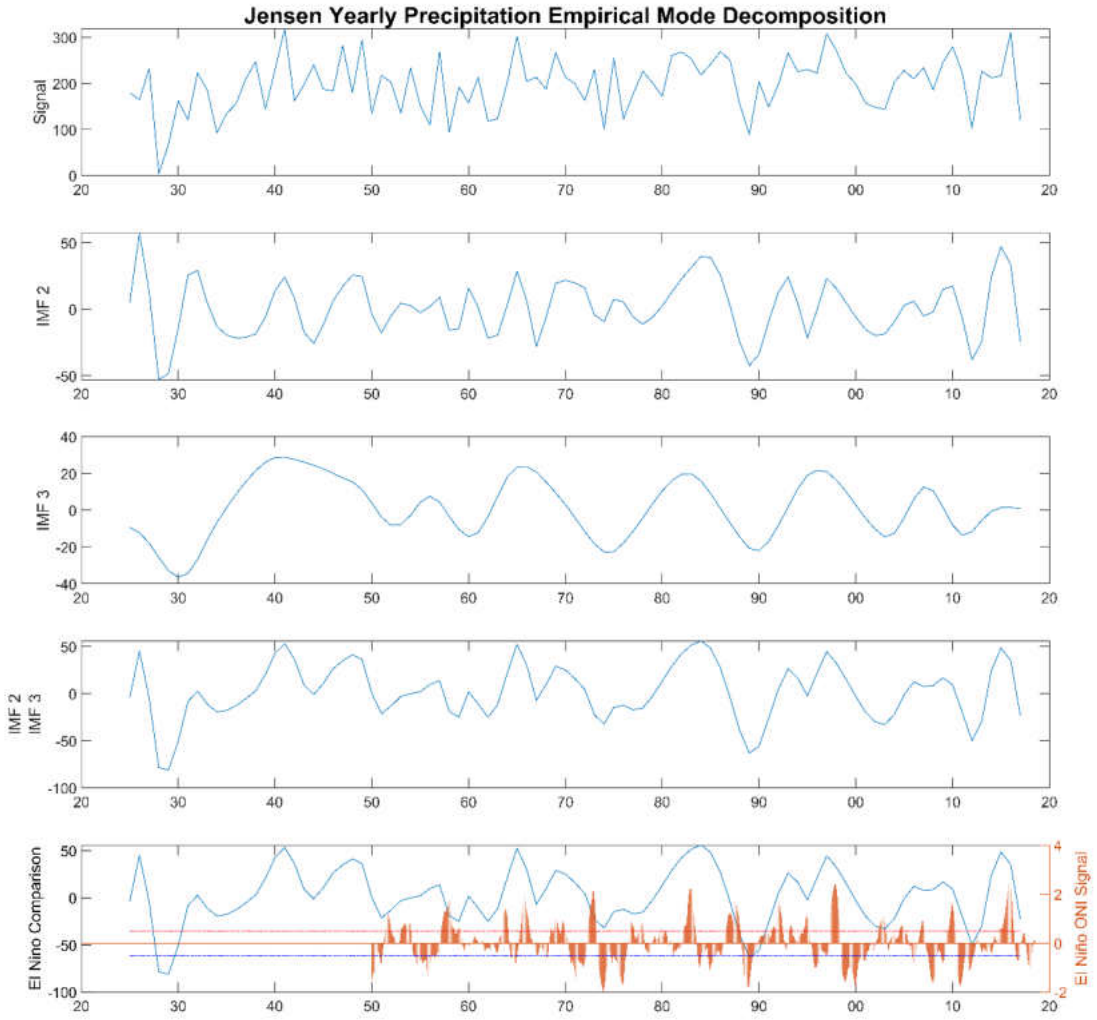

Figure B.8-3: Jensen Yearly Precipitation Empirical Mode Decomposition IMF 2 and 3

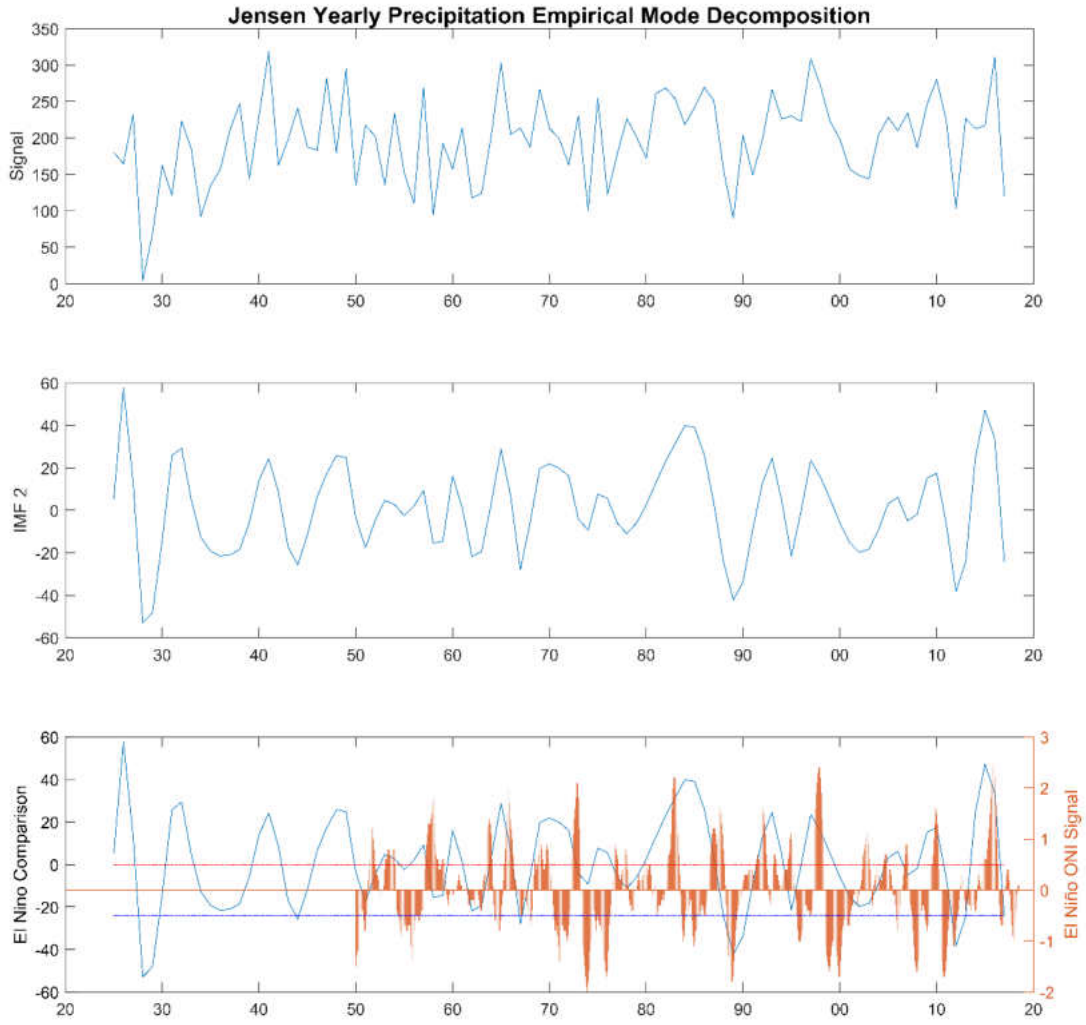

Figure B.8-4: Jensen Yearly Precipitation Empirical Mode Decomposition IMF 2

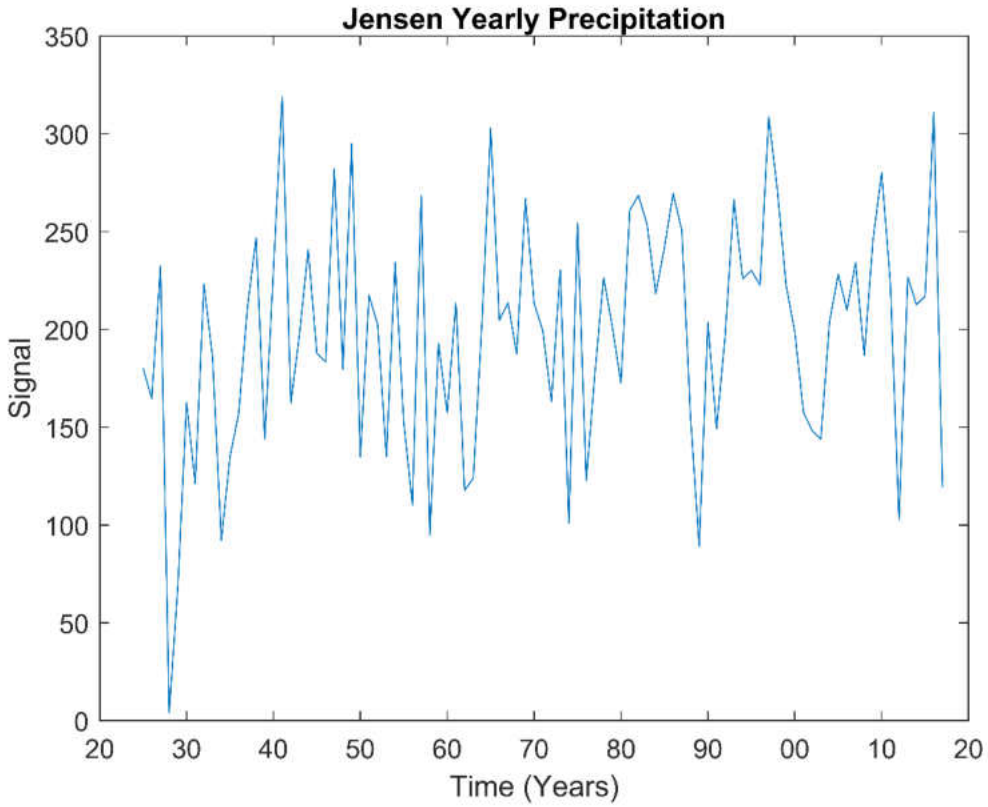

Figure B.8-5: Jensen Yearly Precipitation

B.9 Jensen Temperature

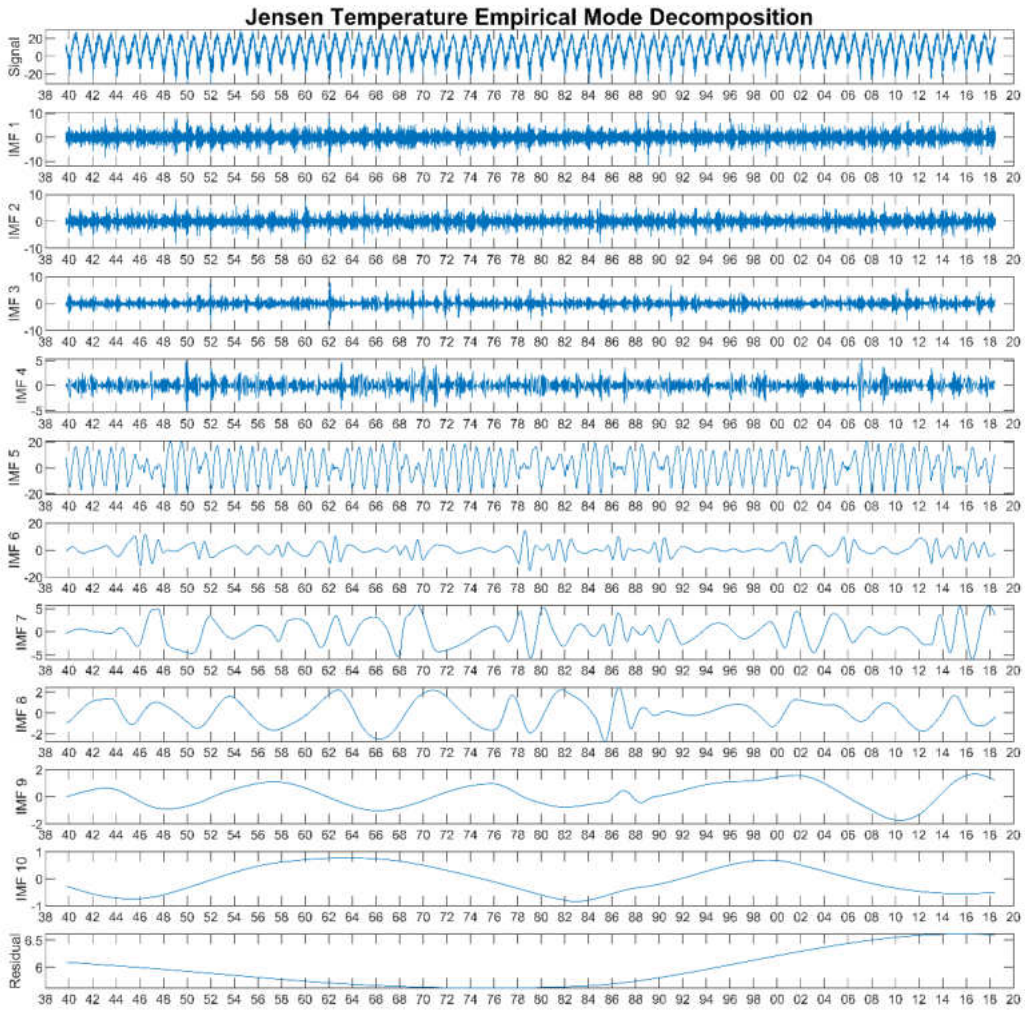

Figure B.9-1: Jensen Average Temperature Empirical Mode Decomposition

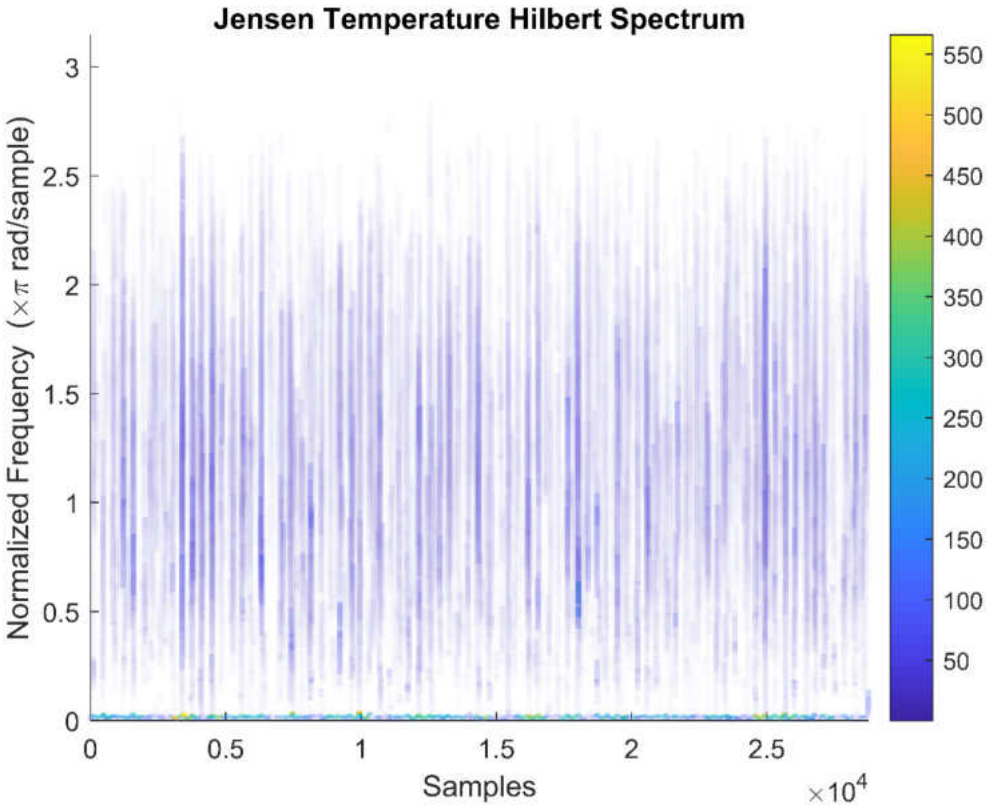

Figure B.9-2: Jensen Average Temperature Empirical Mode Decomposition HHT

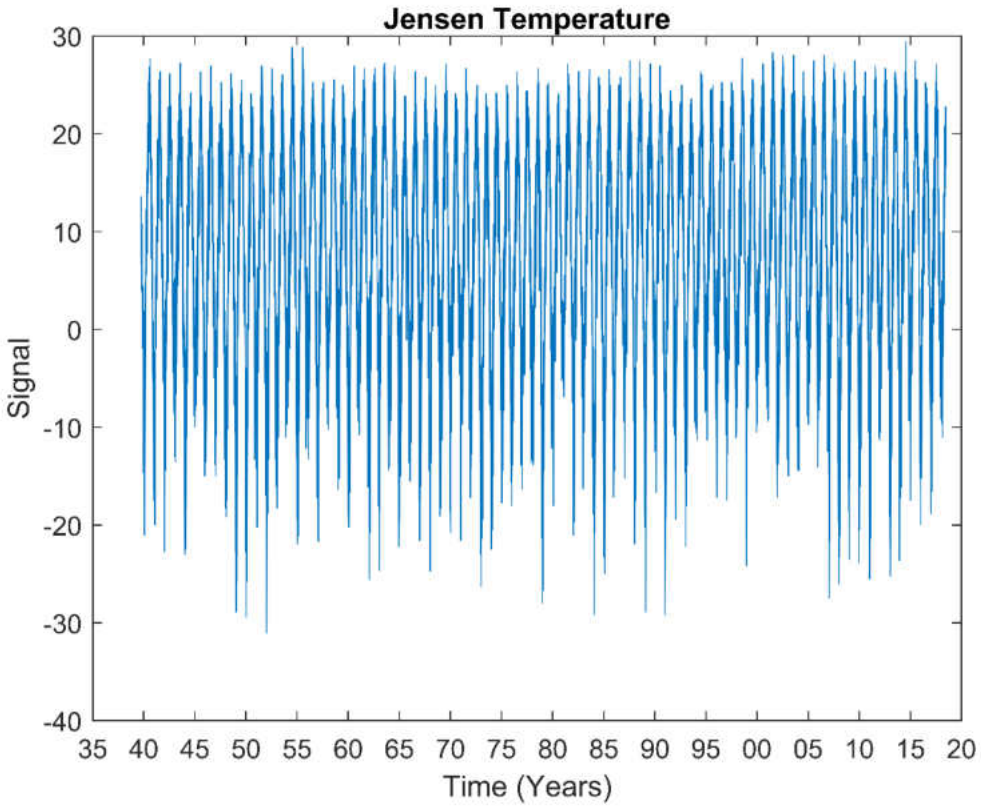

Figure B.9-3: Jensen Average Temperature

B.10 Vernal Temperature

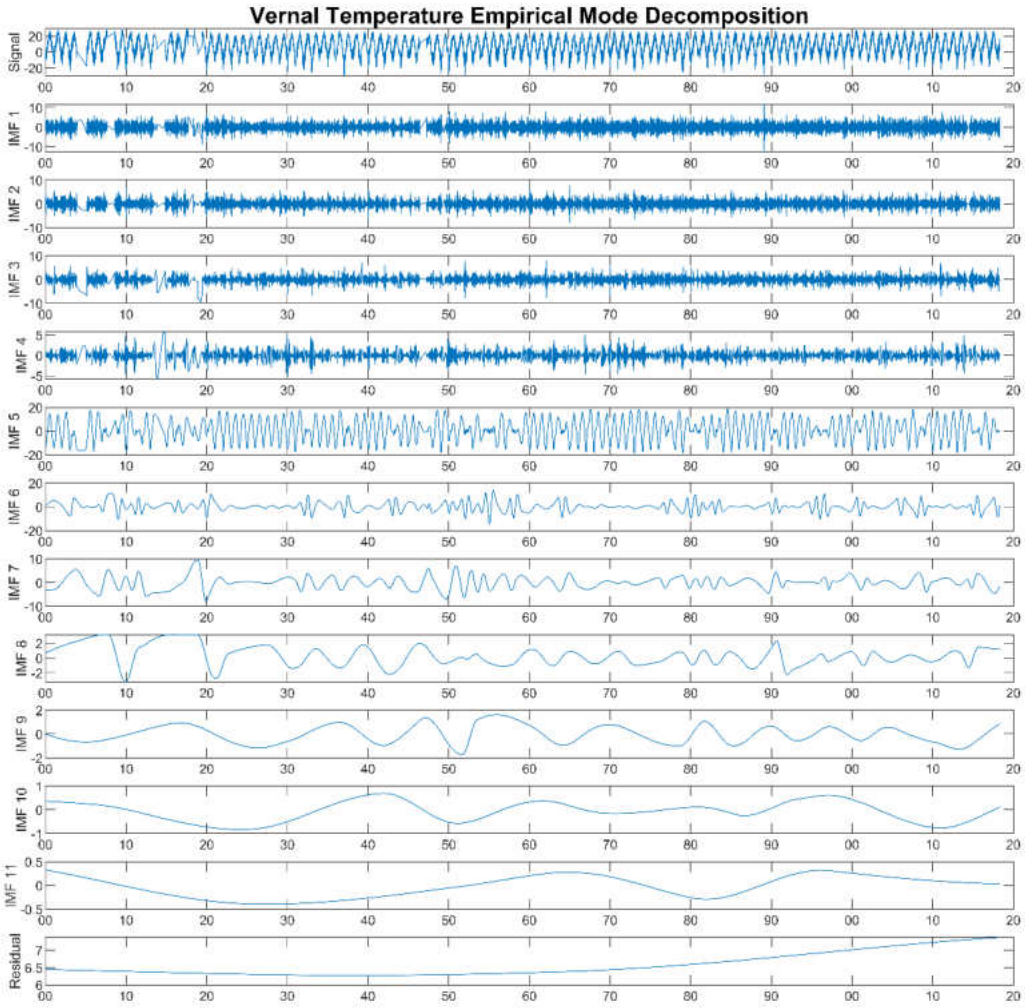

Figure B.10-1: Vernal Average Temperature Empirical Mode Decomposition

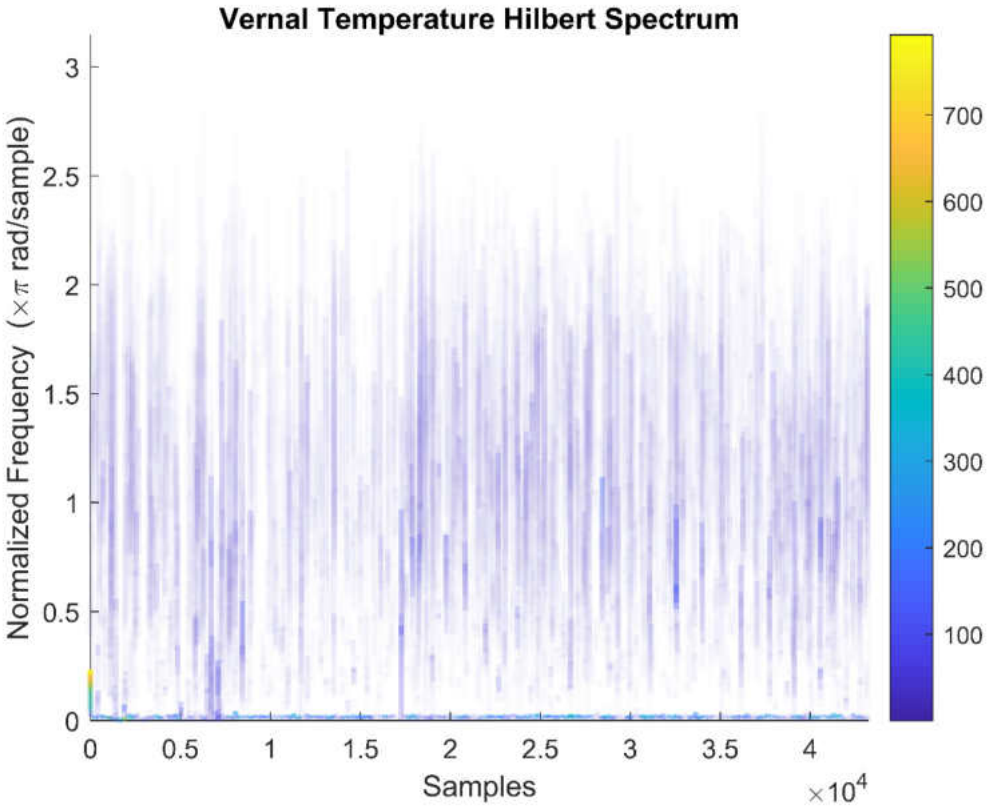

Figure B.10-2: Vernal Average Temperature Empirical Mode Decomposition HHT

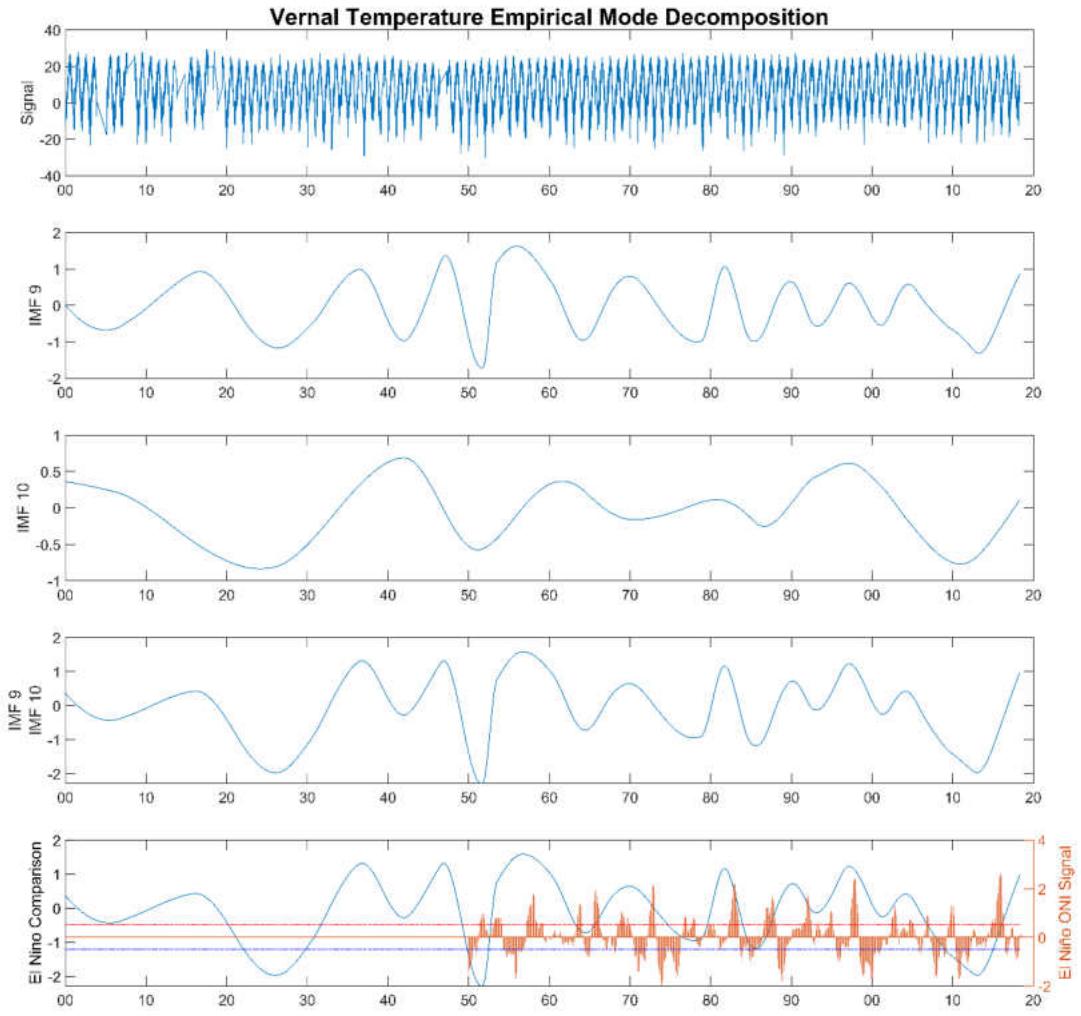

Figure B.10-3: Vernal Average Temperature Empirical Mode Decomposition IMF 9 and 10

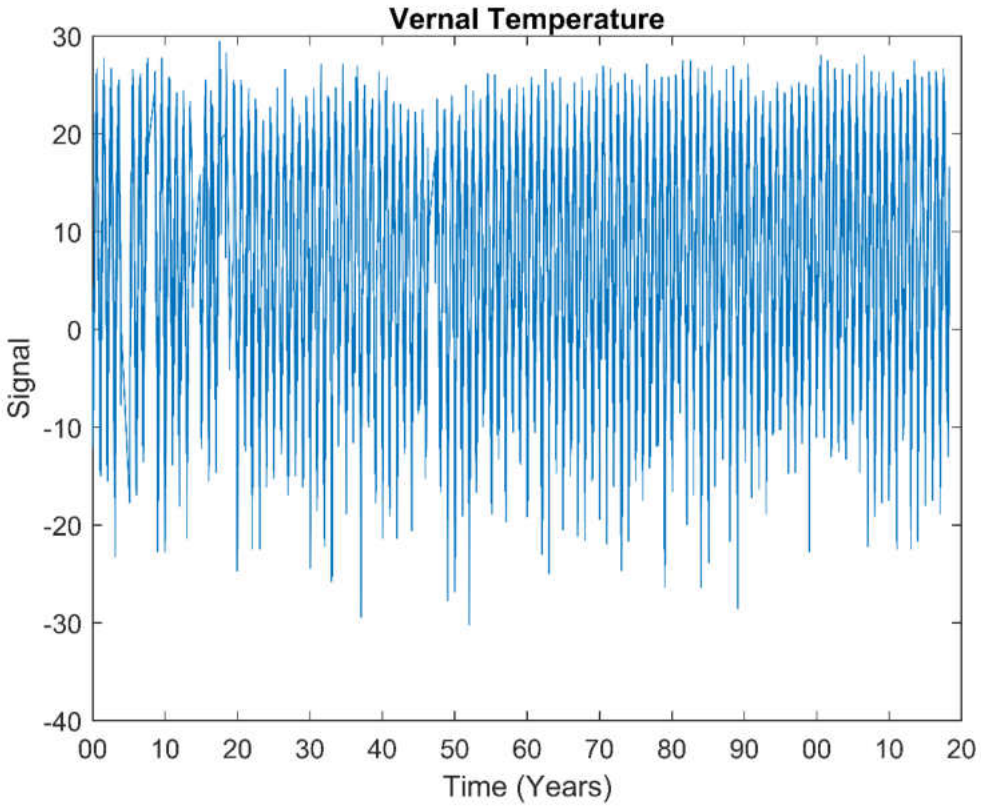

Figure B.10-4: Vernal Average Temperature

B.11 Combination

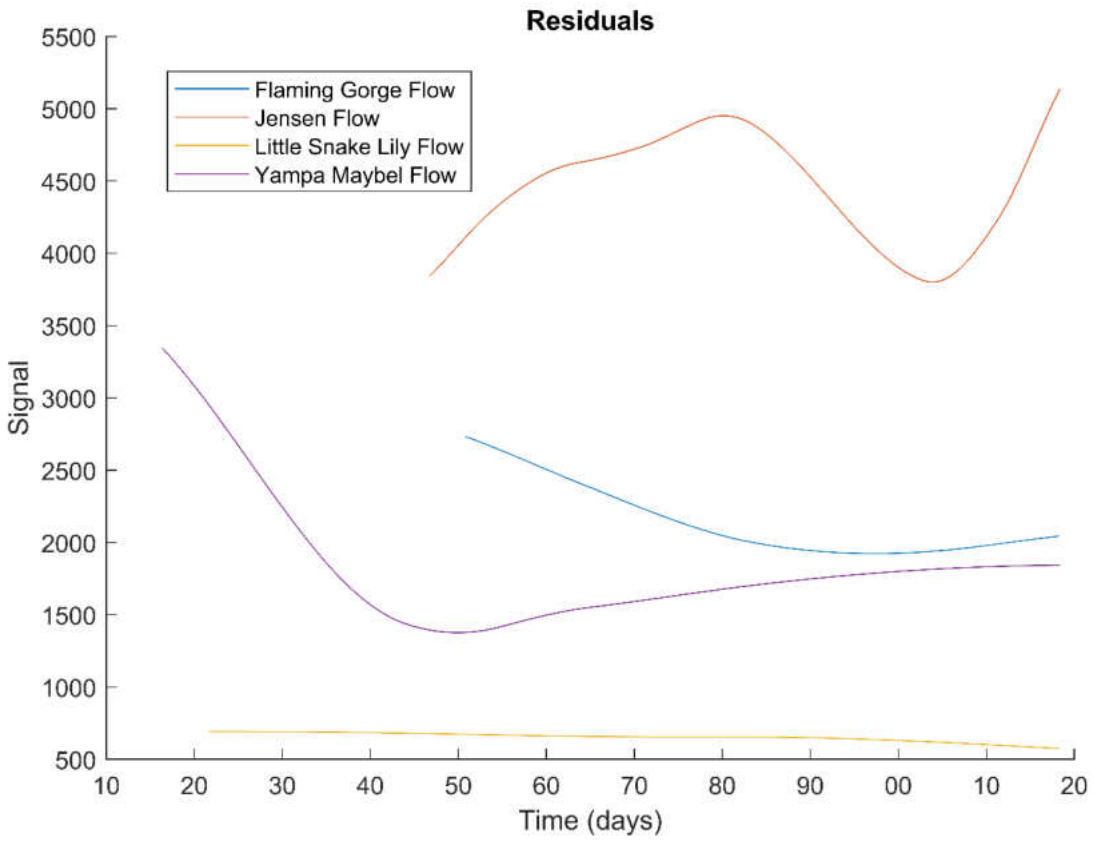

Figure B.11-1: Flow Residuals

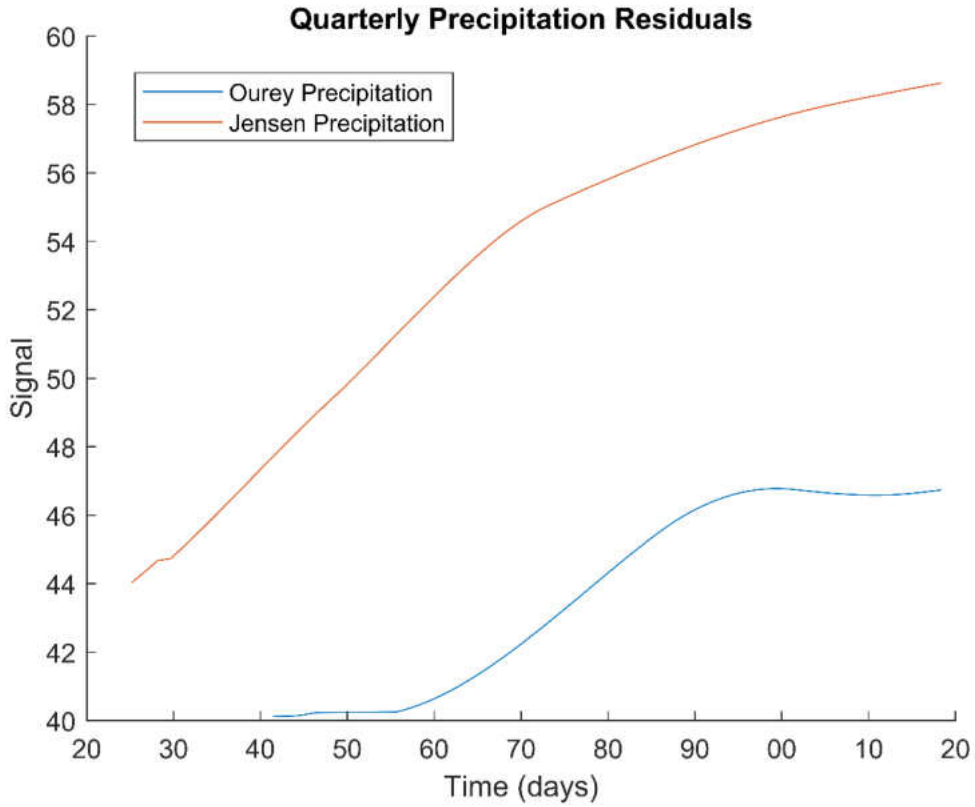

Figure B.11-2: Quarterly Precipitation Residuals

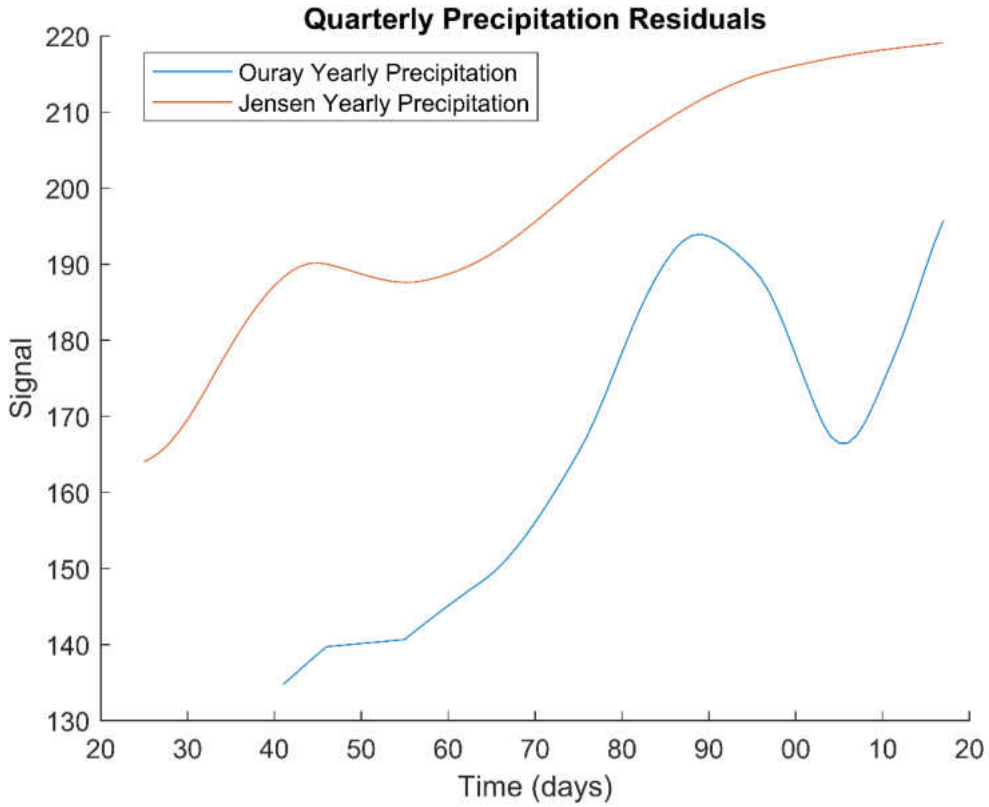

Figure B.11-3: Yearly Precipitation Residuals

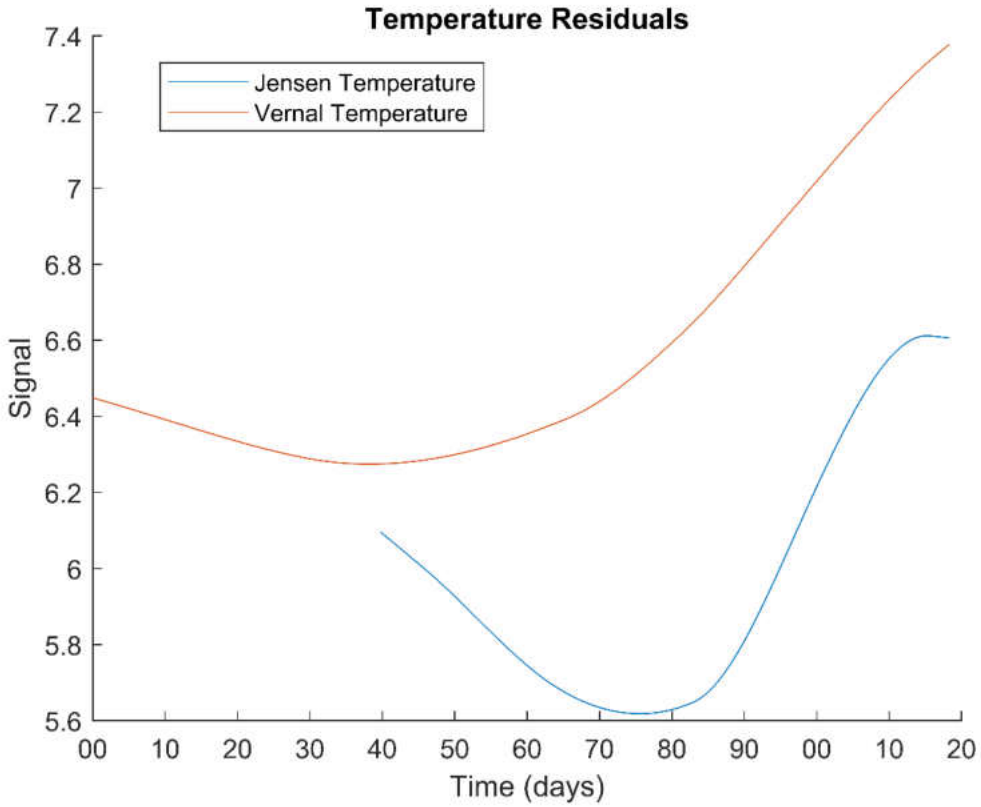

Figure B.11-4: Temperature Residuals