Software Requirements Specification for PolyHarmonics

Nolan Driessen

October 10, 2015

Contents

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit with the SI name in parentheses.

symbol	unit	SI
Hz	frequency	hertz
rad	angle	radians
s	time	second
V	voltage	volt

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The symbols are listed in alphabetical order.

symbol	unit	description
A	V	Amplitude
e	unitless	Euler's number
f	Hz	Frequency
i	unitless	Imaginary unit, $i = \sqrt{-1}$
k	unitless	Index of input signal
n	unitless	Index of summation
N	unitless	Number of discrete samples taken
t	s	Time
X_k	unitless	Complex number encoding phase and amplitude of x_n
x_n	unitless	Time wave input for the Discrete Fourier Transform (DFT)

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
DFT	Discrete Fourier Transform
GS	Goal Statement
LC	Likely Change
R	Requirement
SRS	Software Requirements Specification
T	Theoretical Model

2 Introduction

A system is needed to efficiently and correctly interpret a signal to relate its properties to damage in plastic parts. The following section provides an overview of the Software Requirements Specification (SRS) for PolyHarmonics. This section explains the purpose the document is designed to fulfill, the scope of the requirements and the organization of the document: what the document is based on and intended to portray.

2.1 Purpose of Document

The main purpose of this document is to describe the analysis of acoustic signals. The goals and theoretical models used in the PolyHarmonics code are provided, with an emphasis on explicitly identifying assumptions and unambiguous definitions. This document is intended to be used as a reference to provide all information necessary to understand and verify the analysis. The SRS is abstract because the contents say *what* problem is being solved, but not *how* to solve it.

This document will be used as a starting point for subsequent development phases, including writing the design specification and the software verification and validation plan. The design document will show how the requirements are to be realized, including decisions on the numerical algorithms and programming environment. The verification and validation plan will show the steps that will be used to increase confidence in the software documentation and the implementation.

2.2 Scope of Requirements

The scope of the requirements includes analyzing an input signal and transforming it to an understandable form. Given a series of TDMS files generated from LabVIEW DAQ software, PolyHarmonics reads the data, outputs waveforms, and writes text files for interpretaion of damage in plastic parts.

2.3 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by [1] and [2], with some aspects taken from Volere template 16 [3]. The presentation follows the standard pattern of presenting goals, theories, definitions, and assumptions. For readers that would like a more bottom up approach, they can start reading the data definitions in Section 6.2.3 and trace back to find any additional information they require. The data definitions provide the definition of the DFT and algebraic equations that describe the analysis of signals. PolyHarmonics solves these equations.

The goal statements are refined to the theoretical models, and theoretical models to the data definitions.

3 Stakeholders

This section describes the stakeholders: the people who have an interest in the product.

3.1 The Client

The client for PolyHarmonics is Dr. Michael Thompson. The client has the final say on acceptance of the product.

3.2 The Customer

The customers are the end users of PolyHarmonics. They will be senior undergraduate and graduate level students of McMaster University, studying in Chemical Engineering.

4 General System Description

This section provides general information about the system, identifies the interfaces between the system and its environment, and describes the user characteristics and the system constraints.

4.1 User Characteristics

- The end user of PolyHarmonics is expected to be a senior undergraduate or graduate level student in Chemical Engineering or equivalent.
- The end user is expected to have an understanding of LabVIEW and the ability to operate it.

- The end user is expected to have an understanding of ProMV and the ability to operate it.

4.2 System Constraints

- PolyHarmonics must function with the LabVIEW [4] software used to gather input, as well as produce text files in a format ProMV [5] can read.
- PolyHarmonics must be developed for and be compatible with Python version 2.7 and all associated modules.

5 Scope of the Project

This section presents the scope of the project. It describes the expected use of PolyHarmonics as well as the inputs and outputs of each action.

5.1 Product Use Case Table

Table 1: Use Case Table

Use Case No.	Use Case Name	Actor	Input and Output
1	Input	User	File Directory (input) Starting Frequency (input) Stopping Frequency (input) Frequency Step (input)
2	Filter	PolyHarmonics	TDMS File set (input) Filtered Data (output)
3	Analyze	PolyHarmonics	TDMS File set (input) Filtered Data (input) Transformed Signals (output)
4	Output Transformed Signal	PolyHarmonics	Transformed Signals (input) Plots (output) Text files (output)

5.2 Individual Product Use Cases

- Use Case 1 refers to the user providing input to PolyHarmonics. Once the user has input a directory PolyHarmonics shall search the directory for TDMS files. The starting, stopping and step frequencies are also input for use within the analysis.

- Use Case 2 describes the filtering of the original signals found with the TDMS Files. The input found through Use Case 1 is filtered before any analysis is done.
- Use Case 3 describes the analysis being done. PolyHarmonics will take each given input and transform the signal it contains into the frequency spectrum. This is done for both the original signals and the filtered signals.
- Use Case 4 gives output to the user in the form of a series of plots and three text files. Each TDMS file creates two plots, one amplitude vs time and another amplitude vs frequency. Additionally one contour plot is created for the directory.

6 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories and definitions that describe the analysis of the signals.

6.1 Problem Description

A system is needed to efficiently and correctly interpret a signal. PolyHarmonics is a computer program developed to interpret acoustic data and produce a waveform that can be related to damage in plastic parts.

6.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Radian: A unit of angle, equal to an angle at the center of a circle whose arc is equal in length to the radius.
- Acoustics: The science of studying mechanical waves in gases, liquids and solids. Specifically PolyHarmonics refers to the study of sound waves in solids.
- Fit Criterion: A benchmark to objectively determine whether the implemented product has properly met the requirements.
- LabVIEW: A separate software program that creates TDMS files for PolyHarmonics to analyze.
- ProMV: A separate software program that receives the text files PolyHarmonics produces as its input. Performs further analysis on the signals.

- TDMS: Labview Binary Measurement File. More information can be found here [4]

- System as is: Implementation of PolyHarmonics created before June 3rd 2015.
- System to be: The system that will be created through the use of this documentation.
- Recorded Frequency: The frequency LabVIEW received and stored.
- Emitted Frequency: The frequency sent into the plastic part

6.1.2 Physical System Description

The physical system involves testing a plastic part for damage. A signal is sent through a plastic part by an emitter and taken in by a receiver. The frequency of the received signal is then analyzed.

6.1.3 Goal Statements

GS1: Analyze an acoustic signal by receiving a directory containing one or more TDMS files as input and returning details of the transformed signal that can be related to damage in the plastic part.

6.2 Solution Characteristics Specification

6.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the data definition, or the instance model, in which the respective assumption is used.

A1: Any signal with an amplitude below MIN_AMP is noise.

A2: Any frequencies above MAX_FREQ are not relevant and will not be considered for analysis.

6.2.2 Theoretical Models

This section focuses on the general equations and laws that PolyHarmonics is based on.

Number	T1
Label	**Discrete Fourier Transform**
Input	The signal to be transformed, x_n
Output	X_k
Equation	$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-2\pi ikn}{N}}$
Description	The above equation defines the Discrete Fourier Transform. The DFT converts discrete data from a time wave into the frequency spectrum. The meaning of e^{ix} is provided in DD1. k = Index of input signal n = Index of summation N = Number of discrete samples taken x_n = A wave of Amplitude vs time to be converted to the frequency spectrum X_k = A complex number encoding both phase and amplitude of x_n
Source	http://en.wikipedia.org/wiki/Discrete_Fourier_transform
Ref. By	

6.2.3 Data Definitions

This section collects and defines all the data needed to fully define the system.

Number	DD1
Label	**Euler's Formula**
Equation	$e^{ix} = \cos x + i \sin x$
Description	Euler's formula establishes the relationship between trigonometric functions and the complex exponential function. x = Radian input
Sources	
Ref. By	T1

6.2.4 Data Constraints

Table 2 and 3 show the data constraints on the input and output variables, respectively. The column physical constraints gives the physical limitations on the range of values that can be taken by the variable. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

Table 2: Input Variables

Var	**Physical Constraints**	**Typical Value**
N	$N > 0$	
Starting Frequency	$f \leq$ MAX_FREQ	100kHz
Stopping Frequency	$f \leq$ MAX_FREQ	1000kHz
Step Frequency	$f \leq$ MAX_FREQ	100kHz
x	$0 \leq x \leq 2\pi$	
x_n	$f \leq$ MAX_FREQ (A2), $A \geq$ MIN_AMP (A1)	

(*) These quantities cannot be equal to zero, or there will be a divide by zero in the model.

Table 3: Output Variables

Var	**Physical Constraints**
A	None
f	None

7 Functional Requirements

The following section provides the functional requirements, the business tasks that the software is expected to complete.

R1: **Description:** The system shall accept a directory containing a set of LabVIEW TDMS files as input, each containing a frequency. The user shall input the starting

frequency of the set, the stopping frequency and frequency step between each input. This information will be used in the creation of the plots and text files.
Rationale: The system requires signals to analyze which are stored within files recorded by LabVIEW.

R2: **Description:** The product shall verify the input contains a signal with an amplitude greater than MIN_AMP and a frequency below MAX_FREQ. Unless both conditions are met the user shall be informed through a warning message. Otherwise the system will proceed to the analysis.

Rationale: To avoid any fatal exceptions the contents of all TDMS files must be verified.

R3: **Description:** PolyHarmonics shall use the input files to apply a Haar wavelet filter. [6] [7]
Rationale:

R4: **Description:** PolyHarmonics shall use the input files to calculate the DFT using T1.
Rationale: The signal must be transformed into the frequency spectrum to properly analyze and interpret the signal.

R5: **Description:** Create a plot for each input frequency of the original data: A vs t.
Rationale: The original signal is useful for subsequent analysis.

R6: **Description:** Create a plot for each input frequency of the transformed data: A vs f.
Rationale: With the transformed data, PolyHarmonics can analyze the new signal and return the useful information.

R7: **Description:** Create a contour plot for the analyzed directory showing Recorded Frequency vs Emitted Frequency with the value of the amplitude indicated by the colour of the graph.
Rationale: The results of each recorded requency are compiled into one graph for quick interpretation.

R8: **Description:** Create a text file for the analyzed directory, of the original data. This text file contains the emitted frequency, the received frequency and the amplitude of the original signal after the DFT (T1) is taken.
Rationale: The results of the test must be documented in order to proceed with further analysis through ProMV.

R9: **Description:** Create a text file for the analyzed directory, of the filtered data. This text file contains the emitted frequency, received frequency and the amplitude of a signal filtered by a Haar wavelet which has then had T1 applied to it.
Rationale: The results of the test must be documented in order to proceed with further analysis through ProMV.

R10: **Description:** Create a text file for the analyzed directory, of the scattering data. This text file contains the emitted frequency and a ratio between the amplitude of a range near the emitted frequency and the amplitude of the transformed original signal. This ratio is used to confirm how well the received signal kept the emitted frequency.
Rationale: The results of the test must be documented in order to proceed with further analysis through ProMV.

8 Non-functional Requirements

This section provides the Non-Functional requirements, the requirements that specify criteria that can be used to judge the operation of a system, as opposed to the specific behaviours.

8.1 Look and Feel Requirements

8.1.1 Appearance Requirements

LF1: PolyHarmonics shall appear visually intuitive.

Fit Criterion: At least MIN_PERCENT_USERS shall intuitively understand how to use the interface to input a directory on their first attempt.

8.1.2 Style Requirements

N/A

8.2 Usability and Humanity Requirements

8.2.1 Ease of Use Requirements

UH1: PolyHarmonics shall be usable for collecting and analyzing data by any senior undergraduate or graduate student studying Chemical Engineering or equivalent with no training.

Fit Criterion: At least MIN_PERCENT_USERS shall be able to perform an analysis as described by Use Case 2, Use Case 3 and Use Case 4 with no training.

8.2.2 Personalization and Internationalization Requirements

N/A

8.2.3 Learning Requirements

UH2: The product shall be usable by any senior undergraduate or graduate student studying Chemical Engineering or equivalent on their first attempt.

Fit Criterion: MIN_PERCENT_USERS shall be able to complete an analysis in accordance with all Use Cases on their first attempt.

8.2.4 Understandability and Politeness Requirements

UH3: The product shall use symbols and words that are naturally understandable by the user community.

Fit Criterion: 100% of suitable users shall understand all symbols and terminology used within PolyHarmonics.

8.2.5 Accessibility Requirements

N/A

8.3 Performance

8.3.1 Speed and Latency Requirements

PR1: Each analysis shall be sufficiently fast to not interrupt work flow.

Fit Criterion: At least MIN_PERCENT_USERS shall be satisfied with the analysis speed of PolyHarmonics when running the standard benchmark test cases.

PR2: Any interaction between the user and the product shall respond fast enough to not interrupt work flow.

Fit Criterion: At least MIN_PERCENT_USERS shall be satisfied with the response speed of PolyHarmonics when running the standard benchmark test cases.

8.3.2 Safety-Critical Requirements

N/A

8.3.3 Precision or Accuracy Requirements

PR3: PolyHarmonics shall have sufficiently small relative error.

Fit Criterion: The implementation of PolyHarmonics shall have at most relative error equal to the system as is when compared to the standard benchmark test cases.

8.3.4 Reliability and Availability Requirements

N/A

8.3.5 Robustness or Fault-Tolerance Requirements

PR4: The product shall identify false input and reject it.

Fit Criterion: Input that does not fit within the Assumptions shall be rejected and the user shall be warned.

8.3.6 Capacity Requirements

N/A

8.3.7 Scalability or Extensibility Requirements

N/A

8.3.8 Longevity Requirements

PR5: The product shall function for at least MIN_LIFESPAN.

Fit Criterion: PolyHarmonics shall fulfill all Requirements for at least MIN_LIFESPAN.

8.4 Operational and Environmental Requirements

8.4.1 Expected Physical Environment

OE1: The product is expected to operate within a lab environment.

Fit Criterion: PolyHarmonics shall fulfill all Requirements within a lab environment.

8.4.2 Requirements for Interfacing with Adjacent Systems

OE2: The product shall receive input from a series of files created within LabVIEW.

Fit Criterion: PolyHarmonics shall receive one or more TDMS files as input in accordance with Use Case 1, if none are found the user shall be warned.

OE3: The product shall produce text files for ProMV to receive as input.

Fit Criterion: The text files produced shall be of the same format as the system as is.

8.4.3 Productization Requirements

OE4: PolyHarmonics will require the installation of multiple Python modules.

Fit Criterion: PolyHarmonics shall be able to complete all Use Cases with the use of these additional modules.

OE5: PolyHarmonics shall be able to be installed by an untrained user accompanied by nothing other than the products user manual.

Fit Criterion: MIN_PERCENT_USERS shall be able to get PolyHarmonics fully functioning within one hour of beginning setup.

8.4.4 Release Requirements

OE6: Each release will not cause previous features to fail.

Fit Criterion: After each release PolyHarmonics shall continue to fulfill all Requirements.

8.5 Maintainability and Support Requirements

8.5.1 Maintenance Requirements

MS1: PolyHarmonics must be maintainable by people other than the original developers.

Fit Criterion: The end users of the product must be able to improve, fix or add new functionality to PolyHarmonics.

8.5.2 Supportability Requirements

MS2: The product shall have a user manual accompanying its release.

Fit Criterion: Upon release of PolyHarmonics a user manual shall also be released containing details on installation and setup.

8.5.3 Adaptability Requirements

MS3: The product is expected to run under Windows, OSX and Linux based operating systems.

Fit Criterion: All Use Cases will successfully run for all platforms using a standard benchmark test case.

8.6 Security Requirements

8.6.1 Access Requirements

N/A

8.6.2 Integrity Requirements

SR1: No information shall be stored unnecessarily or distributed under any circumstances.

Fit Criterion: PolyHarmonics shall allocate no space for storage of any personal information.

8.6.3 Privacy Requirements

N/A

8.6.4 Audit Requirements

N/A

8.6.5 Immunity Requirements

N/A

8.7 Cultural Requirements

CP1: The product shall use Canadian spelling where applicable.

Fit Criterion: All non-project specific words within PolyHarmonics shall pass a Canadian spell check program.

CP2: The product shall not be offensive to religious or ethnic groups

Fit Criterion: 0% of users shall feel antagonized from use of the product.

8.8 Legal Requirements

8.8.1 Compliance Requirements

N/A

8.8.2 Standards Requirements

LR1: The product shall comply with all of McMaster University's rules and regulations regarding research development.

Fit Criterion: Each principle outlined here [8] will be followed.

9 Likely Changes

LC1: Currently only frequency is being analyzed, however it is likely that in the future different components of the system, such as the number and time between pulses, will also be taken into account.

LC2: As new information becomes relevant different methods might be used to analyze and interpret the signal.

LC3: The number and contents of the plots that are created from the transformed signal are likely to be represented in different ways in the future. In the event this changes, the UseCases and Functional Requirements will require updating.

LC4: The number and contents of the produced text files. In the event of a change, the UseCases and Functional Requirements will require updating.

LC5: The method of filtering may change as different information becomes relevant.

References

[1] N. Koothoor, "A document drive approach to certifying scientific computing software," Master's thesis, McMaster University, Hamilton, Ontario, Canada, 2013.

[2] W. S. Smith and L. Lai, "A new requirements template for scientific computing," in *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05* (J. Ralyté, P. Ågerfalk, and N. Kraiem, eds.), (Paris, France), pp. 107–121, In conjunction with 13th IEEE International Requirements Engineering Conference, 2005.

[3] J. Robertson and S. Robertson, "Volere requirements specification template edition 16." "www.cs.uic.edu/ i442/VolereMaterials/templateArchive16/c Volere template16.pdf", 2012.

[4] "The NI TDMS file format." "http://www.fermentas.com/techinfo/nucleicacids/maplambda.htm", June 2015.

[5] "Desktop multivariable analysis software." "http://www.prosensus.ca/solutions/multivariate-analysis-software/promv-desktop-multivariate-analysis".

[6] B. Vidakovic and P. Mueller, "Wavelets for kids a tutorial introduction." http://gtwavelet.bme.gatech.edu/wp/kidsA.pdf.

[7] C. Torrence and G. P. Compo, "A practical guide to wavelet analysis." http://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf.

[8] "Research integrity policy." "http://www.mcmaster.ca/policy/faculty/Research/Research%20Integrity June 2013.

[Bibliography links dont work, unsure how to handle this —ND]

A Symbolic Constants

constant name	**description**	**Referenced By**
MAX_FREQ	800kHz	A2, Table 2, R2
MAX_SECONDS	3s	PR1, PR2
MIN_AMP	0.06V	A1, Table 2, R2
MIN_PERCENT_USERS	80%	LF1, UH1, UH2, PR1, PR2, OE5
MIN_LIFESPAN	3 years	PR5