1 Introduction

Signals and Systems is a *Mathematica* implementation of the necessary tools for working with signals and linear systems. These tools can be used in signal processing tasks; they can also be used for educational purposes. To this end, most of the operations in the packages can be made to justify their results step by step. It also extends beyond the traditional scope of a signals and linear systems course by implementing multidimensional operations.

A variety of analysis techniques for both discrete and continuous signals are available in Signals and Systems. You can employ Laplace and Fourier transforms for continuous signals and discrete Fourier, discrete-time Fourier, and Z transforms for discrete signals. You can also make use of traditional analog filter design methods, and their translations to digital filters. Other capabilities are present, as described in this manual.

Signals and Systems does not contain a complete introduction to signals and linear systems. A standard text on this topic is *Discrete-Time Signal Processing* by Alan V. Oppenheim and Ronald W. Schafer (Prentice-Hall, 1989).

This manual assumes a general familiarity with *Mathematica*. The standard reference is Stephen Wolfram's *The Mathematica Book, Fourth Edition*.

■ 1.1 Using Signals and Systems

The application provides functionality as *Mathematica* structures and functions in a series of packages. All of the functions can be accessed by loading the initialization package, which enables a series of stubs that load each of the packages on demand.

- This loads the stubs for accessing the routines in Signals and Systems.

```
In[1]:= Needs["SignalProcessing`"]
```

There may be pauses of various lengths the first time a given function is used, as the appropriate packages are loaded.

- When a function from Signals and Systems is used for the first time, several routines may be loaded.

```
In[2]:= LaplaceTransform[1/t, t, s]

Out[2]= LaplaceTransformData[1 - UnitStep[s],
         RegionOfConvergence[0, 0], TransformVariables[s]]
```

The on-line version of the documentation provided with the application is accessible via the Help Browser, in the Signals and Systems category of the Add-Ons item. Here you can browse through the notebooks and execute code. Alternatively, you can open notebooks outside the Help Browser by locating them in the `Documentation` subdirectory of the application's directory.

■ 1.2 Organization of the Packages

Signals and Systems is organized as a collection of files in the `SignalProcessing` directory. Within the directory is a collection of `.m` files that can be loaded to prepare stubs to all of the functions available to you. These stubs cause only the packages you use to be loaded when needed. In normal practice, you will explicitly load only the initialization package, as described earlier. There is also a file called `Compatability.m` that can be loaded to allow a degree of compatability with the old freely distributable release of the `Signal Processing Packages`. A special file `Bridge.m` is used by some of the packages to handle loading of certain standard *Mathematica* packages, and to perform operations like replacing the standard version of `LaplaceTransform` by the Signals and Systems version.

There are four directories of packages: `Analog`, `Digital`, `Support`, and `SystemDesign`. You will usually not load the packages directly, but it is useful to know where to find them if you are interested in examining algorithms or writing your own routines that access these functions. (You can generally locate the package containing a particular function by using the command `Context`, which returns a context corresponding to the filename of the package.)

ASPAnalyze.m	analog signal analysis routines
FilterDesign.m	filter design interfaces
Fourier.m	forward and inverse Fourier transforms
InvLaplace.m	inverse Laplace transforms
Laplace.m	forward Laplace transforms
LSolve.m	solving differential equations via Laplace transforms
LSupport.m	auxiliary routines for Laplace transforms

Packages in the `SignalProcessing`Analog`` directory.

DFT.m	forward and inverse discrete Fourier transform
DSPAnalyze.m	discrete-time signal analysis routines
DSupport.m	auxiliary routines for discrete-time transforms
DTFT.m	forward and inverse discrete-time Fourier transform
InvZTransform.m	inverse Z transforms
MDDTFT.m	multidimensional extensions to discrete-time Fourier transforms
MDInvZTransform.m	multidimensional extensions to inverse Z transforms
MDZTransform.m	multidimensional extensions to forward Z transforms
ZSolve.m	solving recurrence equations by Z transforms
ZSupport.m	auxiliary routines for Z transforms
ZTransform.m	forward Z transforms

Packages in the `SignalProcessing`Digital`` directory.

Ptolemy.m	export to Ptolemy
header05.pt	a Ptolemy header file for export to Ptolemy

Packages in the `SignalProcessing`SystemDesign`` directory.

AnalysisSupport.m	common symbols for DSPAnalyze and ASPAnalyze
AnimateConvolution.m	"flip-and-slide" method of graphically displaying convolution
Convolution.m	piecewise convolution routines
Decimation.m	two-dimensional decimation design and analysis, with visualization
DeltaSupport.m	support for operations involving delta and step functions
FilterSupport.m	auxiliary routines for filter analysis and design
LatticeTheory.m	manipulation of resampling matrices, including Smith form decompositions
MDPlotting.m	multidimensional extensions to plotting routines
MDSupport.m	general auxiliary routines for multidimensional operations
Polygons.m	operations on two-dimensional convex polygons
ProgrammingExtensions.m	useful internal functions for common operations
ROC.m	region of convergence manipulation, auxiliary routines for transforms
SigProc.m	signal processing functions defined
SupCode.m	common auxiliary functions
TransformSupport.m	additional auxiliary functions for transforms
Window.m	definitions of common windows

Packages in the SignalProcessing`Support` directory.

There is also a subdirectory named Documentation that contains the documentation notebooks accompanying Signals and Systems, and a directory named Kernel that contains initialization files.