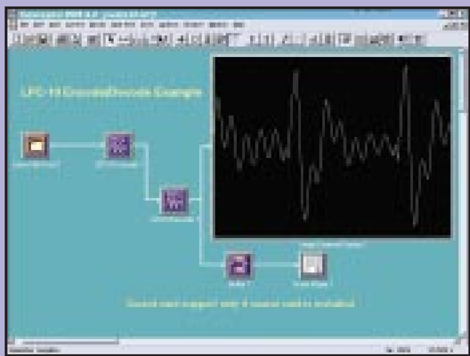
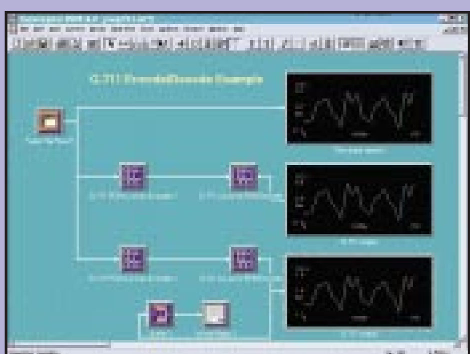
Part Number: HSWN2550

## Examples



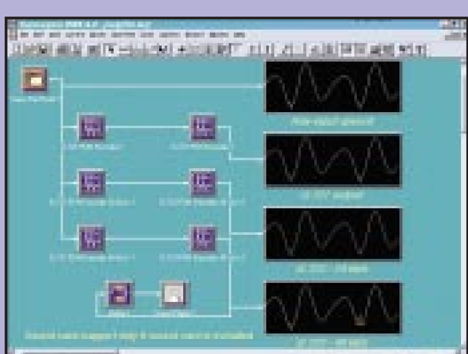
**LPC-10 Example with Sound Card**

*This worksheet provides example usage of the LPC10 Encode and LPC10 Decode block functions. A speech file is encoded at the rate of 180 linear 16-bit samples to 54 bits. The resulting decoded speech may even be heard using a standard sound card.*



**G.711 Example with Sound Card**

*This example worksheet shows a simple G.711 encoding/decoding with the resulting output directly driving a standard PC sound card such that the user can actually hear the final quality of the encoded/decoded speech.*



**G.72x Example with Sound Card**

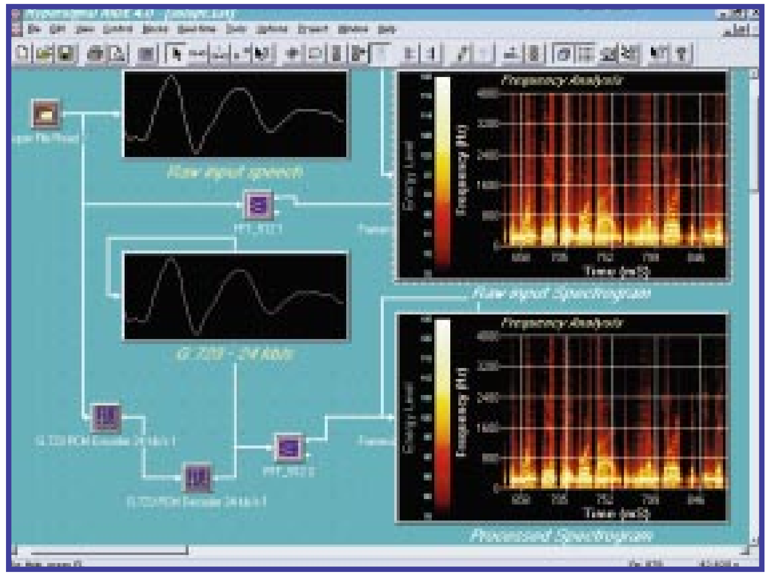
*This example worksheet demonstrates several of the speech encoding and decoding block functions in the Advanced Speech Library.*

Hyperception

The Leader in DSP

# Advanced Speech Library

## Optional Block Function Library for Hypersignal® Block Diagram/RIDE™




*Speech Processing Example using Advanced Speech Library*

## Overview

The Advanced Speech Library (ASL) available for Hypersignal Block Diagram and RIDE provides a set of speech-related simulation blocks for speech processing research and development. Block functions in the area of speech encoding/decoding, LPC filtering, and analysis are included for the speech researcher to assist in studying certain speech processing algorithms.

Because of the inherent efficiency of Block Diagram/RIDE, many of the block functions actually run fast enough on the host PC to drive a standard PC sound card in real-time for subjective listening and comparison. Adding these specialized speech functions is a great way to augment Hypersignal Block Diagram or RIDE for the speech processing engineer.

## Capabilities

With the Advanced Speech Library and Hypersignal Block Diagram/RIDE the speech professional can create quick proof-of-concept visual designs. This modular set of speech functions provides the user with an excellent starting point for a number of speech processing related projects. Using the powerful display capabilities within Hypersignal Block Diagram/RIDE, detailed analysis and comparison of speech processing waveforms may be performed easily.

Using the open software architecture, it is easy to add user-created block functions to support additional custom or proprietary speech algorithms; the user-created functions may be used together with the ASL block functions to achieve an overall specialized design with a minimum amount of design time. Get a quick start on your speech processing, or add to your professional speech processing development tools with this Advanced Speech Library!



©Hyperception, Inc. 1997
Printed in USA

# Advanced Speech Library

## List of Functions

### Analysis Filter (Direct Form)
*This block performs an LPC Analysis Filter using Direct Form structure. The LPC coefficients are input on the first input channel, and the signal to be filtered is on the second input channel. The output will be the filtered signal. If any integer arithmetic modes are selected, Q15 fixed point arithmetic will be utilized.*

### Analysis Filter (Lattice)
*This block performs an LPC Analysis Filter using a Lattice Form structure. The LPC coefficients are input on the first input channel, and the signal to be filtered is on the second input channel. The output will be the filtered signal. If any integer arithmetic modes are selected, Q15 fixed point arithmetic will be utilized.*

### G.711 A-Law to PCM Decoder
*This block performs decoding on the input channel from G.711 A-law to PCM Linear. For each 8-bit A-Law value, a 16-bit linear sample will be output.*

### G.711 A-Law to u-Law Converter
*This block performs conversion on the input channel from G.711 A-Law to u-Law. The input channel is assumed to be in BYTE precision, and the output is always in BYTE format. For each 8-bit A-Law value input, an 8-bit u-Law sample will be output.*

### G.711 PCM to A-Law Encoder
*This block performs encoding on the input channel from G.711PCM Linear to A-Law. For each16-bit linear sample, an 8-bit A-Law value will be output.*

### G.711 PCM to u-Law Encoder
*This block performs encoding on the input channel from G.711PCM Linear to u-Law. The input channel is assumed to be in integer precision, and the output is always in BYTE format. For each16-bit linear sample, an 8-bit u-Law value will be output.*

### G.711 u-Law to A-Law Converter
*This block performs conversion on the input channel from G.711 u-Law to A-Law. The input channel is assumed to be in BYTE precision, and the output is always in BYTE format. For each 8-bit u-Law value input, an 8-bit A-Law sample will be output.*

### G.711 u-Law to PCM Decoder
*This block performs decoding on the input channel from G.711 u-law to PCM Linear. The input channel is assumed to be in BYTE precision, and the output is always in 16-bit short integer format. For each 8-bit u-Law value input, a 16-bit linear sample will be output.*

### G.721 PCM Decoder
*This block takes 4-bit unpacked data and converts to 16-bit PCM linear data using G.721 conversion.*

### G.721 PCM Encoder
*This block takes 16-bit Linear PCM data and converts it to 4-bit encoded data using G.721 conversion. Although the input data may be of any precision, it is converted to 16-bit short integer internally prior to conversion, and the output is always unpacked BYTE format.*

### G.723 PCM Decoder 24 kb/s
*This block takes 3-bit unpacked data and converts it to 16-bit PCM linear data using G.723 conversion with an assumed sample rate of 24 kb/s. Although the input data may be of any precision, it is converted to BYTE internally prior to conversion.*

### G.723 PCM Decoder 40 kb/s
*This block takes 5-bit unpacked data and converts it to 16-bit PCM linear data using G.723 conversion with an assumed sample rate of 40 kb/s. Although the input data may be of any precision, it is converted to BYTE internally prior to conversion.*

### G.723 PCM Encoder 24 kb/s
*This block takes 16-bit Linear PCM data and converts it to 3-bit encoded data using G.723 conversion with an assumed sample rate of 24 kb/s. Although the input data may be of any precision, it is converted to 16-bit short integer internally prior to conversion, and the output is always unpacked BYTE format.*

### G.723 PCM Encoder 40 kb/s
*This block takes 16-bit Linear PCM data and converts it to 5-bit encoded data using G.723 conversion with an assumed sample rate of 40 kb/s. Although the input data may be of any precision, it is converted to 16-bit short integer internally prior to conversion, and the output is always unpacked BYTE format.*

### Levinson-Durbin
*This block performs the Levinson-Durbin LPC algorithm for creating LPC coefficients. Although multiple precisions are supported, for most accuracy, either floating point, or double precision should be used. If any of the integer precision modes are selected, fixed point arithmetic will be used with Q15 scaling applied.*

### LPC Analyzer
*The LPC analyzer block accepts 180 samples and produces four outputs; the LPC coefficients, the estimated pitch period (at an 8kHz rate), the energy estimate, and the voicing decision. The LPC coefficients are bounded by +1 and -1 and are output on the first channel. The pitch estimate is output on the second channel. The energy estimate is output on the third channel, and the half frame voicing decision is output on the fourth channel; there are two voicing values for each frame indicating the voicing decision on a half frame basis.*

### LPC Synthesizer
*The LPC synthesize block accepts four input channels and produces 180 samples of output. The input channels include: the LPC coefficients, the estimated pitch period (at an 8kHz rate), the energy estimate, and the voicing decision. The LPC coefficients should be bounded by +1 and -1 and are input on the first channel. The pitch estimate is input on the second channel. The energy estimate is input on the third channel, and the half frame voicing decision is input on the fourth channel; there should be two voicing values for each frame indicating the voicing on a half frame basis.*

### LPC10 Decoder
*This block accepts incoming standard LPC-10 (54 bits) and decodes it to linear PCM data, 180 points/frame. The incoming encoded data is assumed to be organized as 54 long integer (32-bit, samples), with 1 bit/sample. The sync value is located as the last sample in the frame.*

### LPC10 Encoder
*This block accepts incoming linear PCM data, 180 points/frame and encodes it to standard LPC-10 (54 bits). The encoded data is organized as unpacked 54 long integer (32-bit, samples), with 1 bit/sample. The sync value is the last sample.*

### Schur
*This block performs the Schur LPC algorithm for generating LPC reflection coefficients from auto-correlation sequence. Although multiple precisions are supported, for most accuracy, either floating point, or double precision should be used. If any of the integer precision modes are selected, fixed point arithmetic will be used with Q15 scaling.*

### Synthesis Filter (Direct Form)
*This block performs a LPC Synthesis Filter using Direct Form structure. The LPC coefficients are input on the first input channel, and the signal to be filtered is on the second input channel. The output will be the filtered signal. If any integer arithmetic modes are selected, Q15 fixed point arithmetic will be utilized.*

### Synthesis Filter (Lattice)
*This block performs an LPC Synthesis Filter using Lattice Form structure. The LPC coefficients are input on the first input channel, and the signal to be filtered is on the second input channel. The output will be the filtered signal. If any integer arithmetic modes are selected, Q15 fixed point arithmetic will be utilized.*

### LSP to Predictor Coef
*This block function accepts floating point LPC Line spectrum pairs (LSP's) and produces the LPC predictor parameters given the order of the system.*

### Predictor Coef to LSP's
*This block function accepts floating point LPC predictor parameters and produces LPC Line spectrum pairs (LSP's) given the order of the system.*

### Predictor Coef to Reflection Coef
*This block function accepts floating point LPC predictor parameters and produces LPC reflection coefficients (rc's) given the order of the system.*

### Reflection Coef to Predictor Coef
*This block function accepts floating point LPC reflection coefficients (rc's) and produces the LPC predictor parameters given the order of the system.*

Hyperception

The Leader in DSP

## Ordering Information

**Part Number:**
HSWN2550 - Advanced Speech Library

**Hyperception, Inc.**
**9550 Skillman LB 125 * Dallas, Texas 75243**
**(214) 343-8525 * FAX (214) 343-2457**
**Internet: info@hyperception.com**
**WWW: http://www.hyperception.com**

***Optional Block Function Library for Hypersignal Block Diagram/RIDE***

*Hyperception is continually improving and modifying its product line, and reserves the right to change the specifications in this product information sheet at any time, without notice. While the utmost care and precaution have been taken in the preparation of this product information sheet, Hyperception assumes neither responsibility for errors or omissions, nor any liability for damages resulting from the use of the information contained herein. Hypersignal is a registered trademark of Hyperception, Inc., and Microsoft is a registered trademark and Windows is a trademark of Microsoft Corporation.*