1.4 ADPCM speech coding

Consider the same speech file you used in the last assignment,

`http://ptolemy.eecs.berkeley.edu/~eal/eecs20/sounds/voice.wav`

You can read this URL with the **AudioReader** actor, found in the audio sublibrary of the signal processing library, although, again, you will probably find it more convenient to copy the file onto your local disk.

You are to construct an adaptive differential pulse code modulation (ADPCM) coder using the "feedback around quantizer" structure and an LMS filter to form the approximate linear prediction. Be sure to connect your LMS filter so that at the receiver, if there are no transmission errors, an LMS filter can also be used in a feedback path, and the LMS filter will exactly track the one in the transmitter. You will use various amounts of quantization.

To assess the ADPCM system, reconstruct the speech signal from the quantized residual, subtract this from the original signal, and measure the noise power. You can also listen to the result using the **AudioPlayer** actor.

1. In your first experiment, do not quantize the signal. Find a good step size, verify that the feedback around quantizer structure works, measure the reconstruction error power and prediction gain. Does your reconstruction error make sense? Compare your prediction gain result against that obtained in the previous lab. It should be identical, since all you have changed is to use the feedback-around-quantizer structure, but you are not yet using a quantizer.

Assume you have a communication channel where you can transmit N bits per sample. You will now measure the signal quality you can achieve with ADPCM compared to simple PCM (pulse code modulation) over the same channel. In PCM, you directly quantize the speech signal to 2^N levels, whereas in ADPCM, you quantize the prediction error to 2^N levels. For a given N, you should choose the quantization levels carefully. In particular, the quantization levels for the ADPCM case should not be the same as those for the PCM case. Given a particular prediction gain G, what should the relationship be? You can use the **`Quantizer`** actor in the math palette to accomplish the quantization in both cases. A useful way to set the parameters of the **`Quantizer`** actor is as follows (shown for $N = 2$ bits, meaning 4 quantization levels):

levels: {-1.5*s, -0.5*s, 0.5*s, 1.5*s}

where "s" is a parameter of the containing model. You can create the parameter *s* by right clicking on the background of the model, selecting Edit Parameters, and the clicking on "add". This way, you can easily experiment with various quantization spacings without having to continually retype long sequences of numbers.

For each N, you should compare (a) the ADPCM encoded speech signal and (b) the PCM encoded speech signal to the original speech signal. You should make this comparison by measuring the power in the differences between the reconstructed signals and the original. How does this difference compare to the prediction gain?

2. Use $N = 3$ bits.
3. Use $N = 2$ bits.