Real-time Audio Equalization from Microphone Input

MatDeck is a very convenient tool for audio processing and analysis. There are plenty of functions which support working with audio including very common wave read(), wave play() and wave write(). Further, MatDeck has been enriched with Audio Toolkit that has functions which support generating the test signal, and analysis of audio signal in the frequency domain. Equalizer Toolkits in MatDeck contain three different graphic equalizers including 31-band equalizerfirform(), equalizerinverseform(), and 10-band equalizer10bandsform(). There are several examples which illustrate the use of these audio functions in wave file processing, and audio equipment testing. However, those examples deal with recorded audio signals.

In this example, we examine the real-time audio processing from a microphone input. The audio processing is done by using afore mentioned graphic equalizer 31-band equalizerfirform().

Microphone Input

The microphone input is performed using audio in() function. The audio in() function has three arguments: sampling rate in Hz, number of bits per sample, and the number of audio channels. By calling the audio in(), the microphone is enabled, a TCP/IP channel is created. Audio data in a format defined by the three arguments is sent through the TCP/IP channel.

```
w_sender := audio in(44100, 16, 1)
```

The audio signal is read from the created TCP/IP channel, therefore the document must be connected to the channel.

```
w_receiver := channel connect("127.0.0.1", 1805)
```

It is possible to check the properties of the received signal, these parameters will be used later for the real-time audio play function, audio out() :

```
w_props := audio in properties(w_receiver)
```

The audio is read through the created TCP/IP channel piece by piece, this piece is called chunk in audio processing. We can use variable to store each piece, and we can perform processing on every piece of the audio.

Audio Output

The audio out() function is used to create a TCP/IP channel for audio output towards speakers. It has three parameters: sampling rate in Hz, number of bits per sample, and the number of audio channels. Let us create a channel below:

```
w_output := audio out(w_props[0], w_props[1], w_props[2])
```

The TCP/IP channel is used to write data which is continuously played on speakers.

Graphic Equalizer

Graphic equalizers can be used to change the spectral properties of the audio piece in question. The graphic equalizer form is initialized for the real-time operation. The widget contains sliders which can be adjusted in order to increase or decrease sound in the bands of interest. The real-time check box should be selected.

```
w_equalizer := equalizerfirform(0, "equalizer01")
show(w_equalizer)
```

Processing

All processing is done within the function receive() which continuously reads the audio chunks from the microphone through the input TCP/IP channel. Audio chunks are processed by a equalizer using the function equalizerfirresults(). Equalized chunks are sent to speakers through the output TCP/IP channel. The function receive() is called every time the audio chunk arrives from the microphone.

```
on event(w_receiver, receive())
```

The function ‚receive' is given in a sequel.

```
receive()
{
  w_data := channel read(w_receiver, false)
  w_equalized := equalizerfirresult(w_equalizer, w_data)
  channel write(w_output, w_equalized)
}
```

The sound from the microphone is continuously, chunk by chunk, received, equalized and then heard on the speakers.