openMHA Calibration Concepts

The goal here is to present a simplified version of the system calibration with the hope that this makes the details of the existing documentation and Matlab scripts more understandable. The existing calibration documentation provided by the openMHA project provides more specific information regarding calibration methods and plugins.

Purpose of Calibration

We have two different domains: **sound levels** in the outside world and **numeric values** inside the system processor. The processor needs to know what numeric level will produce a 90dB output or what input signal amplitude corresponds to a 30dB acoustic noise level. Since microphones and receiver characteristics can vary widely, the correspondence between numeric values and sound level also varies with these components.

It would be extremely inconvenient to rewrite signal processing algorithms for every different hardware configuration because the magnitude of the internal signals changed. To solve this problem, the openMHA framework adopts the convention that signals being processed are in Pascals, the SI unit of pressure. This means a floating point value of 1.0 in openMHA corresponds to 1 Pascal, or 94dB SPL. The signals coming directly from the ADC and going to the DAC most likely won't share this same relationship. To make things work, calibrations are introduced to account for differences in transducer hardware.

How it works

The openMHA framework solves the calibration issue with a plugin, appropriately named 'transducers'. From a high level view, the transducers plugin applies gain factors to the time domain input and output signals, Figure 1. (Frequency shaping is also optionally available, to fine tune the calibration.) It usually encloses the other plugins (algorithms) so that they operate on calibrated signals.

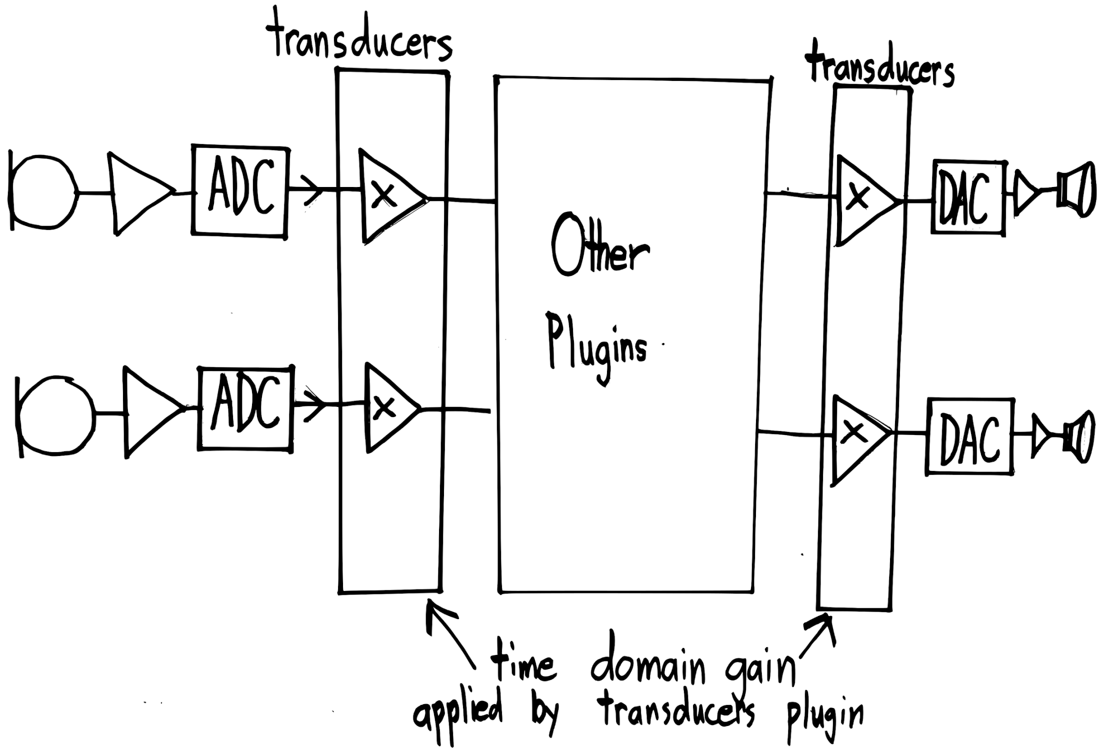

Figure 1. High level diagram showing the position of the transducers plugin in the signal path.

The calibration problem consists of adjusting the signal level coming from the ADC and going to the DAC so they match the openMHA convention of 1.0 ⇒ 94dB SPL.

Before the input signal is calibrated, the 1.0 ⇒ 94dB SPL convention doesn't apply, and the number 1.0 will correspond to some arbitrary acoustic level, the peak level, depending on the microphone and preamp gain. The same situation applies to the output, where 1.0 will correspond to different SPL levels depending on the amplifier and receiver. When the transducers plugin knows the correct peak levels, it can calculate the gains needed to calibrate the input and output signals, Figure 2. The peak levels will be different for input and output and they'll also differ slightly across individual components.

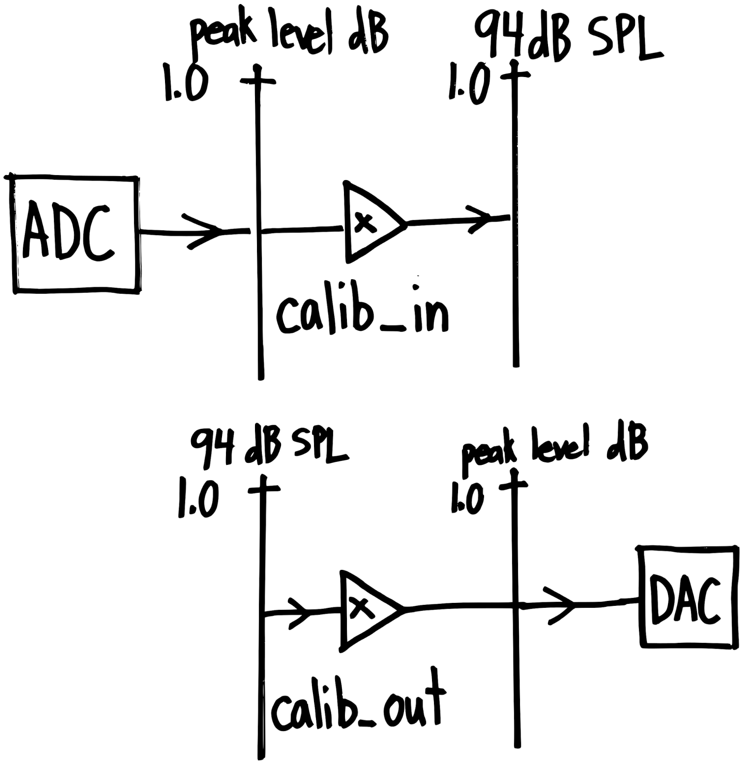

Figure 2. Relationship of ‘peak level’ to 94dB SPL in the transducers plugin.

The scalar gains are calculated using the peak level as follows:

$$input\,gain = \frac{10^{\frac{peaklevel}{20}}}{10^{\frac{94}{20}}}$$

$$output\,gain = \frac{10^{\frac{94}{20}}}{10^{\frac{peaklevel}{20}}}$$

The key configuration variables of the transducers plugin are ‘calib_in’ and ‘calib_out’, each of which has its own variable, ‘peaklevel’ that functions as a calibration term set by the user. Also, the variable ‘rmslevel’ is read by the user and represents a measurement of the rms level of the signal after it’s been adjusted with the ‘peaklevel’ calibration term on the input, and before the signal is adjusted by calibration at the output.

Calibration Methods

The user doesn’t need to calculate input and output gains or directly measure peak levels when adjusting the system calibration. Instead, the general method is to make measurements with a given ‘peaklevel’ setting, then adjust the ‘peaklevel’ based on these measurements.

The basic method to set the input calibration, involves a version of:

1. Presenting an acoustic stimulus to the system microphone at a known level.
2. Reading the value of calib_in.rmslevel.
3. Adjusting calib_in.peaklevel so that calib_in.rmslevel correctly matches the known level.

There are different procedures that will result in a calibrated output. The following outline can be used once the input is calibrated:

1. Configure the system to apply a fixed gain (10dB, for example), no other processing.
2. Present a known acoustic input level to the calibrated input.
3. Measure the acoustic output of the receiver and adjust calib_out.peaklevel so that it correctly matches the expected output level.

There are a collection of scripts and .cfg files in openMHA that detail various calibration methods.