HYDRONIC_Analog_Values Filtering

Contents

This document describes the filtering mechanism of the Hydronic project that is applied on the values of the analog input channels. The filtering mechanism only concerns AIN5A and CIO5A (channel inputs only).

2 Slaves filtering mechanism

The filtering mechanism is implemented in the slaves' firmware, which means that they constantly apply the filtering to any active channel. The implemented filter described below is the current configuration that the master indicates to the slaves through the RS485 bus via command messages. Optionally and for future purposes there are additional filters implemented in the slaves' firmware that a new master firmware can configure.

The slaves filter the channel readings and send back to the master two values. The raw value and the filtered value. The two values are then translated to channel readings according to the IO definitions (e.g. Thermistors vs Analog voltage) from the master.

2.1 Common filter types overview

In general, the investigated and implemented common filters for the slaves are the ones described below and the selection of which one to be used is based on the specific application and needs of the system.

- *Moving average filter:*
  The filter uses past values to determine the actual one. The filtered value is calculated from averaging the past values with the current raw value that is read. Usually three or four such values are used for averaging. This filter is not considered to be suitable for the HYDRONIC system, due to spikes occurrences, which cannot be rejected by this type of filter, but on the contrary affect substantially the final filtered value.
- *Weighted moving average filter:*
  One variant of the normal moving average filter is the weighted moving average filter, which biases with weights the past values and the current raw value during averaging. Similar to the normal moving average filter, this filter is not considered to be suitable for the HYDRONIC system, due to spikes occurrences, which cannot be rejected by this type of filter, but on the contrary affect substantially the final filtered value.
- *Exponential filter:*
  Exponential filter applies exponential smoothing using the exponential window function. Whereas in the simple moving average the past observations are weighted equally, exponential functions are used to assign exponentially decreasing weights over time. Usually this filter is ideal for seasonal data. Seasonality is the presence of variations that occur at regular intervals less than a year, such as weekly, monthly, or quarterly. Seasonality may be caused by various factors, such as weather, vacation, and holidays and consists of periodic,

repetitive, and generally predictable patterns. This filter is not considered to be suitable for the HYDRONIC system, due to expected system's stable and no seasonal behavior and due to spikes occurrences, which spikes cannot be rejected by this type of filter, but on the contrary affect moderately the final filtered value, not only during the spike event, but also over future samples (at least 30 – 0.5 sec after the spike event).

- *Double exponential filter:*
  Simple exponential smoothing does not perform well when there is a trend in the data. In such situations, the double exponential filter is applied using the recursive application of an exponential filter twice. The basic idea behind double exponential smoothing is to introduce a term that considers the possibility of a series exhibiting some form of trend, which gets updated via exponential smoothing. This filter is not considered to be suitable for the HYDRONIC system, due to expected system's stable and no seasonal/trend behavior and due to spikes occurrences, which spikes cannot be completely rejected by this type of filter, but on the contrary affect slightly the final filtered value. However, the double exponential filter would be a good filter choice if it could reject more effectively the spikes.

2.2 HYDRONIC filter type selection

For the HYDRONIC system, the commonly used filters do not seem to be very effective. The system must be able to eliminate invalid readings of the channels due to electrical noise, external interferences and physical connection imperfections.

The main problem that the final applied filter has to deal with, is how to make a compromise between spike rejection and value fluctuation. The system shall be able to reject noise and spikes in the raw values, for example due to random overvoltage events, while at the same time it shall be able to discern changes in values magnitude and accept them, without falsely identifying them as spikes and thus rejecting them. A reading shall be considered as spike when it deviates unreasonably from the current system operating level of the read values and lasts for only few samples (i.e. 100 samples – approx. 1.5 - 2 sec latency).

The filter response shall follow the channel value changes immediately (after a predefined number of read samples – by default set to 100 samples – approx. 1.5 - 2 sec latency) and the latency shall be kept relatively low. A tradeoff must be made between spike robustness and latency.

Considering all the aforementioned reasons, a special custom filter was implemented in the slaves directly, the custom memory filter.

2.2.1 Custom memory filter

This system-specific filter uses at its core a four-point averaging filter combined with specific thresholds in order to determine whether a spike is desired or not, according to the spike duration (in samples, by default set to 100 samples – approx. 1.5 - 2 sec latency), magnitude and relative deviation from the current average value. These thresholds concern the current reading deviation from the previous filtered value, which is considered as the current average value. Both percentage deviation and absolute deviation values are considered to be these threshold features and by default the percentage deviation is set to 500% and the absolute deviation is set to 100 units.

For example, an analog input resistance channel reads 110 Ohms and the system is in stable state and keeps reading this value over many samples. Due to i.e. some kind of wiring instability, the reading becomes 600 Ohms momentarily and then gets back to normal, 110 Ohms. Apparently, the 110 Ohms readings are correct and shall be accepted by the filter. On the contrary, the 600 Ohms readings lasted only over few samples, the percentage deviation was 600/110 = 545% and the absolute deviation was 600-110 = 490 Ohms. The above deviation level and duration level thresholds classify these samples as not desirable and thus they are rejected. This results in stable filtered readings of 110 Ohms during the 600 Ohms actual raw readings due to the wiring instability.

But if the erroneous readings last for many samples (over 100 samples by default - 1.5 - 2 sec latency) and not just for few (less than 100 samples), then the filter considers them as the new valid stable state and accepts them.

Despite the fact that the system is in stable state, the readings are expected to still fluctuate slightly due to inevitable electrical noise interferences. In order to smooth such noise fluctuations of the readings, the system incorporates a grid mapping feature.

2.2.2 Grid mapping

Since the read values are raw decimal numbers in general, they can be assigned whatever possible values, up to a few decimal places precision. In order to achieve relatively constant filtered values behavior for small actual raw value deviations, a grid mapping feature is integrated inside the custom memory filter, which allows only predefined values steps as values resolution. The currently implemented predefined step is 0.25 Ohms for resistance readings and 0.02 mA for current readings. Essentially, this technique floors every filtered value to the nearest step-valid value. For instance, 24.73 values become 24.75, assuming a 0.25-step grid. Grid mapping technique is applied to resistance and current measurements. However, it is not applied to voltage readings, since voltage values are mapped to narrower scale (0-10 V) and two-point decimal precision is necessary.

The above grid mapping step values were selected in such a way that the loss of information due to the grid mapping transformation shall be minimized, the final values shall be representative of the initial ones and the whole system behavior shall remain intact.

The filtering mechanism is implemented inside the slaves, the master device itself is not able to filter the raw values and in general there is no need to apply any additional filters to the already filtered values.

However, specifically for the temperature sensor values, because the filtering is applied to raw resistance values, there may be slight temperature deviations from the actual temperatures (0.2 – 0.8 degrees of Celsius fluctuations) due to the temperature transformation equations and number precisions that the master device applies to calculate the corresponding temperature values, using the raw filtered values.

In order to eliminate this behavior and to obtain stable values for thermistor sensors channels, a second grid mapping is applied at temperature level by the master device, with predefined values steps of 0.25 degrees of Celsius.