MatrixPilot
Lidar characterization

Introduction

The purpose of this document is to characterize the behaviour of the rangefinder Lidar.

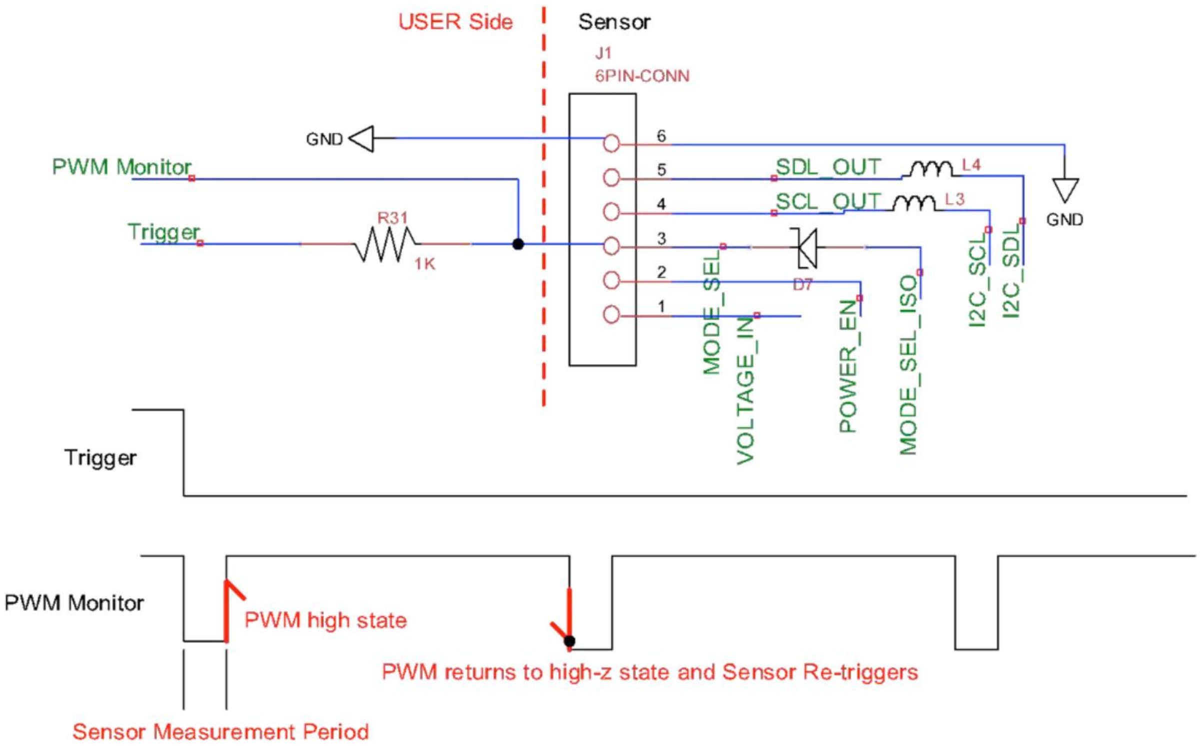
The reference of the sensor: Lidar Lite 3 from Garmin, manufactured in 2016/09.
The version tested here was not the Lite 3 HP.

Ref documents:

[1] operating-manual-llm20c132i500s011.pdf

[2] pli-06-instruction.pdf

All the tests were executed in PWM mode, in respect to this schematic (cf doc [1]), except the 1KΩ resistor replaced by a 2.2 KΩ:



Test tools

To be able to test the sensor with some agility or flexibility the following equipment was integrated:

1. Arduino UNO with Arduino-PWM-Reader Github library
2. RFD 868+ Telemetry at 40 Hz
3. Saleae Logic Analyser
4. Power bank
5. Special analysis software developed in Delphi for this study.

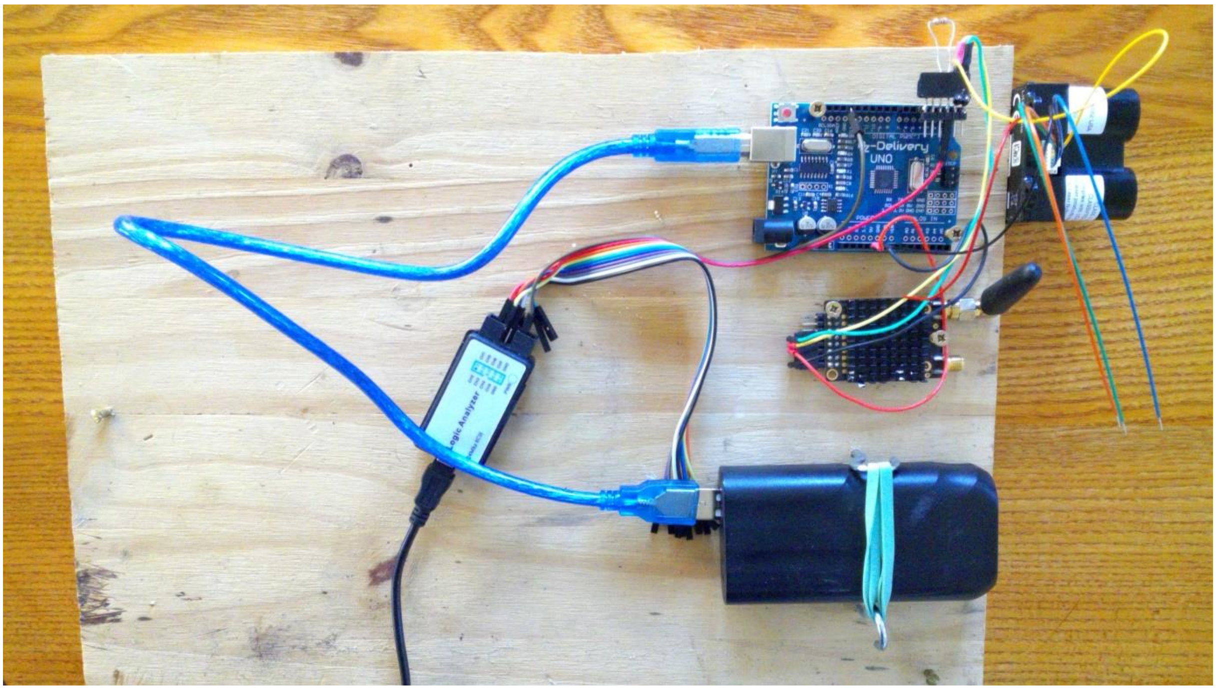


Figure 1 : Test set

View of field experiment:

Figure 2 : The target: wall concrete blocks

Figure 3: Starting point, around 14 m from the target

Figure 4 : Out of range point, around 65 m from the target.

Results

All trials were done with the Lidar PWM output grounded with 2.2 kΩ resistor permanently.
The positive pulse duration is polled each 25 ms. For long range, when this pulse is longer than 25 ms, this method can give the same value two consecutive times.

(It is not a good idea to monitor the trigger periodically because the trigger transition can cut the positive response pulse.)

On the table tests

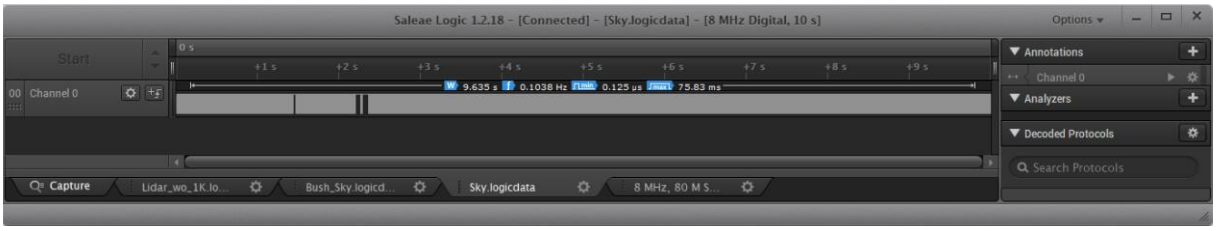
The Lidar point to the wall of the "laboratory" i.e. the living room, about 2 meters far away.

Maximum low pulse width (sensor measurement period): 7.75 ms
Minimum low pulse width (sensor measurement period): 2.7 ms

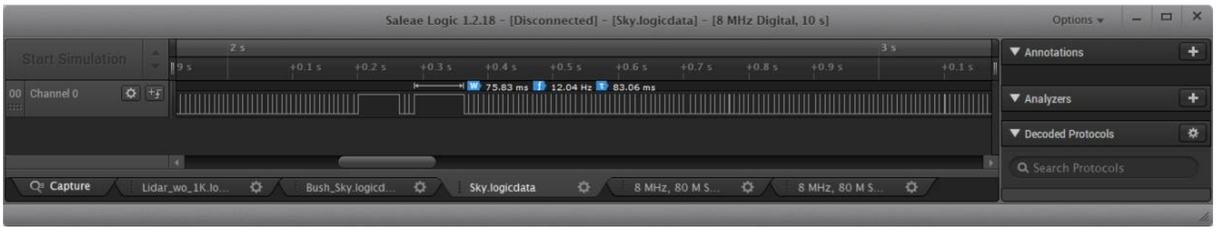
The high pulse duration depends upon the range by the simple law: duration (µs) = range (mm).
The "nominal" duration of the positive pulse is 2 ms because the wall of the living room is standing at 2m. During 10s of measurements without any movement of the Lidar head, the minimum duration was 1.925 and maximum 1.965 ms, meaning 40mm noise peak to peak or 6 mm standard deviation.

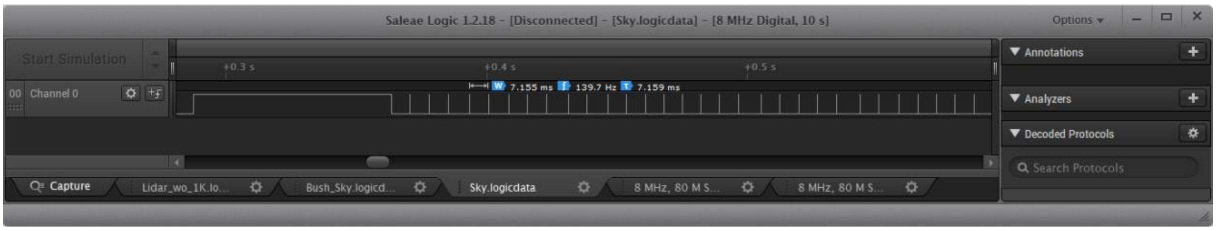
The minimum duration recorded was 42µs when something is put just in front of the Lidar glasses.

When looking to the sky, outside the window, the PWM output is reduced to a very small pulse of a fraction of µs (Minimum recorded was 0.125 µs, but note that the analyser sample frequency is 8 MHz) every 7 to 10 ms, with, sometimes, long pulses (here under 75ms but 136 ms was seen during tests).



Zooming in here on the above record between 2 and 3s, we can observe that nominal behaviour of an unlocked Lidar is to send a very short positive pulse every about 7ms. But it can be disturbed by long response, here 76 ms.





At this stage of the tests, it was interesting to study what appended at the limit of detection. Go to the trials done outside.

On the field trials

With good confidence in this Lidar Lite 3 sensor, the starting point is situated at about 14 meters from the target, a concrete cinder block vertical wall (see fig 2 and 3).

The trial consists of walking backward from the starting point until loss of reception. At the approach of this limit, the speed was reduced slowly. Arriving at about 65 m from the target, the reverse path is undertaken to return to the starting point. During this travelling, the operator tries to point as well as possible the Lidar line of sight perpendicularly to the target plane. It was not so easy!

Here above the global result:

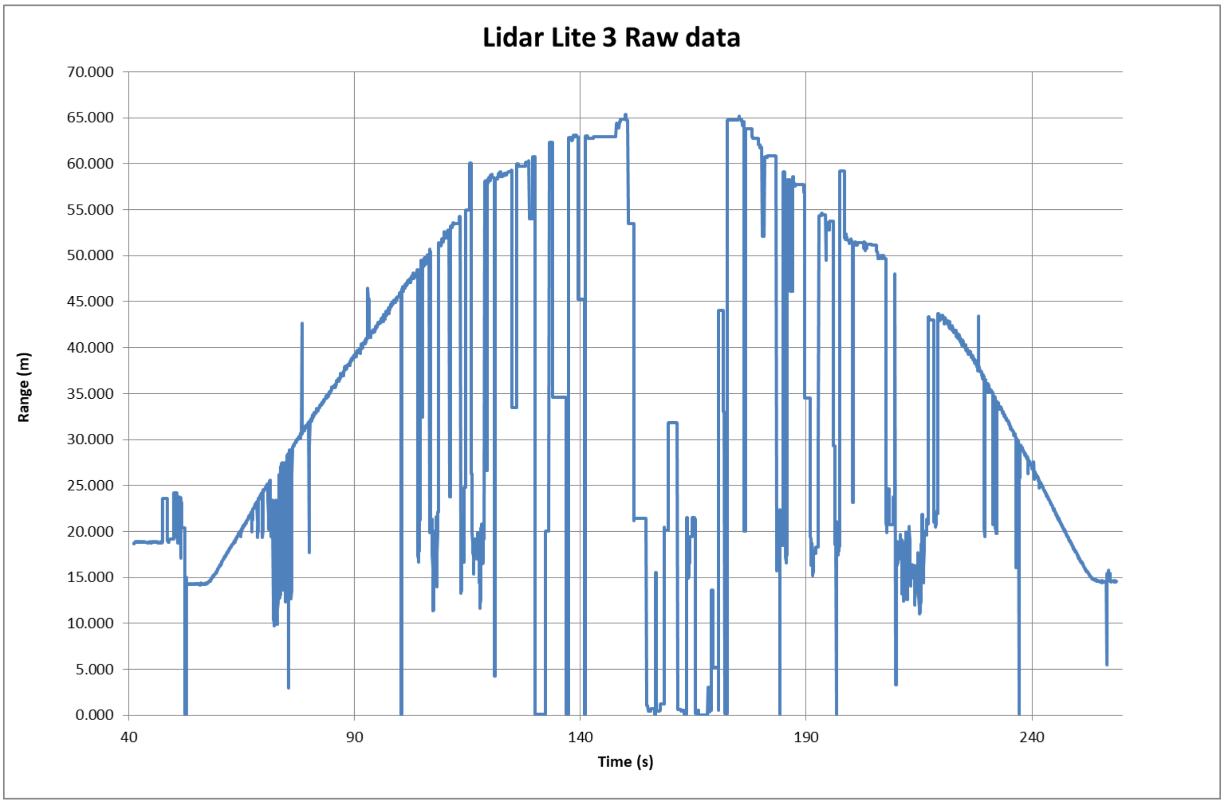


Figure 5: Lidar measurements

Analysis

Room tests

First, the Lidar "Sensor Measurement Period" (cf. doc [1]) is not constant. Even if the nominal duration were about 3ms, it could be 7ms or more from time to time. Thus, it is possible that the specified rate "up to 500 Hz" could not be obtained repeatedly.

Then, the range is quite precise with very small noise and good reliability (especially if we compare to altitudes given by barometer or GPS).

Finally, when the Lidar is looking at the sky, the PWM signal is quite unpredictable. Extreme values can be observed, from 42 mm to 136 m. There is no report in the PWM signal allowing one to know that the target is out of range. On this Lidar Lite 3, there is no particular timing denoting unlock.
It is a pity that the constructor made a positive pulse even though he reports this unlocks in the I2C interface. It would be preferable to have no positive pulse at all or a fix pulse of 50ms, for example.

Assuming a maximum reasonable range of 15m, that is 15 ms, the maximum reading rate could not be greater than 15+8ms = 23 ms. Document [2] announces a 40m range, meaning 40 ms high pulse width. Taking into account such data, the maximum rate becomes 50ms or 20 Hz, if the PWM output is used.

Test range trials

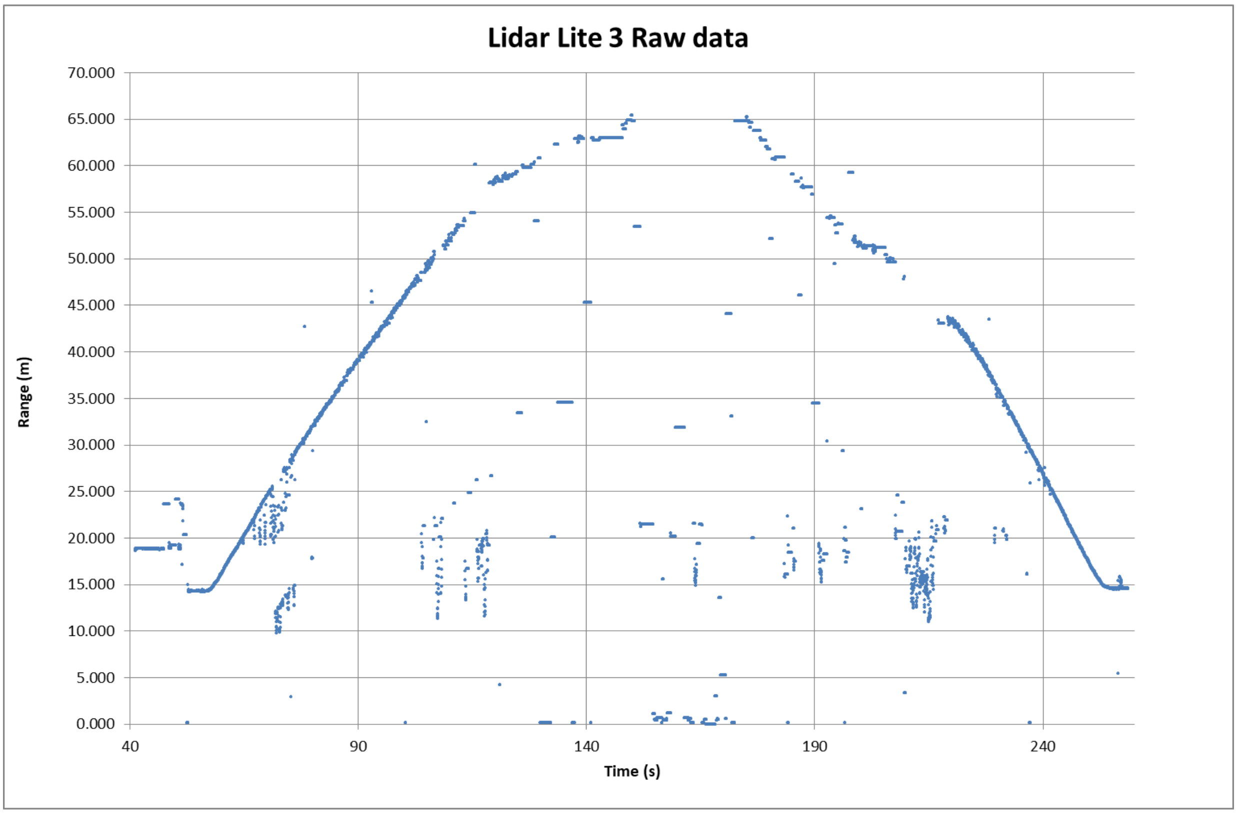
It could be interesting to analysis the result shown in figure 5.

The first good news is the maximum range of 65 m. The Lidar measurement at this distance is not very stable and, probably, cannot be used with confidence. Nevertheless, it proves that the announced 40 m range is quite realistic. Even if the Lidar were pointing to a bad albedo target.

Secondly, there is lot of signal losses. This must be taken into account when we have to process these data. Here the loss can be due to instability of the line of sight, but, that’s life!

That means that the real problem we have to solve is to validate the measurements.

Even at good range, around 15 m for example, the Lidar can have difficulties to lock on the target, especially if its reflectivity varies in great proportions.



Looking to the measurement plotted by plots instead of line, shows that there must be some data processing that could eliminate bad value and extrapolate between good values.
A child can show the most probable valid data!
With only one sensor giving the range, it is not so easy to find a model that can represent a good approximation of the real range.
Thanks to the filtering described in annex 2, the result seams not to bad:

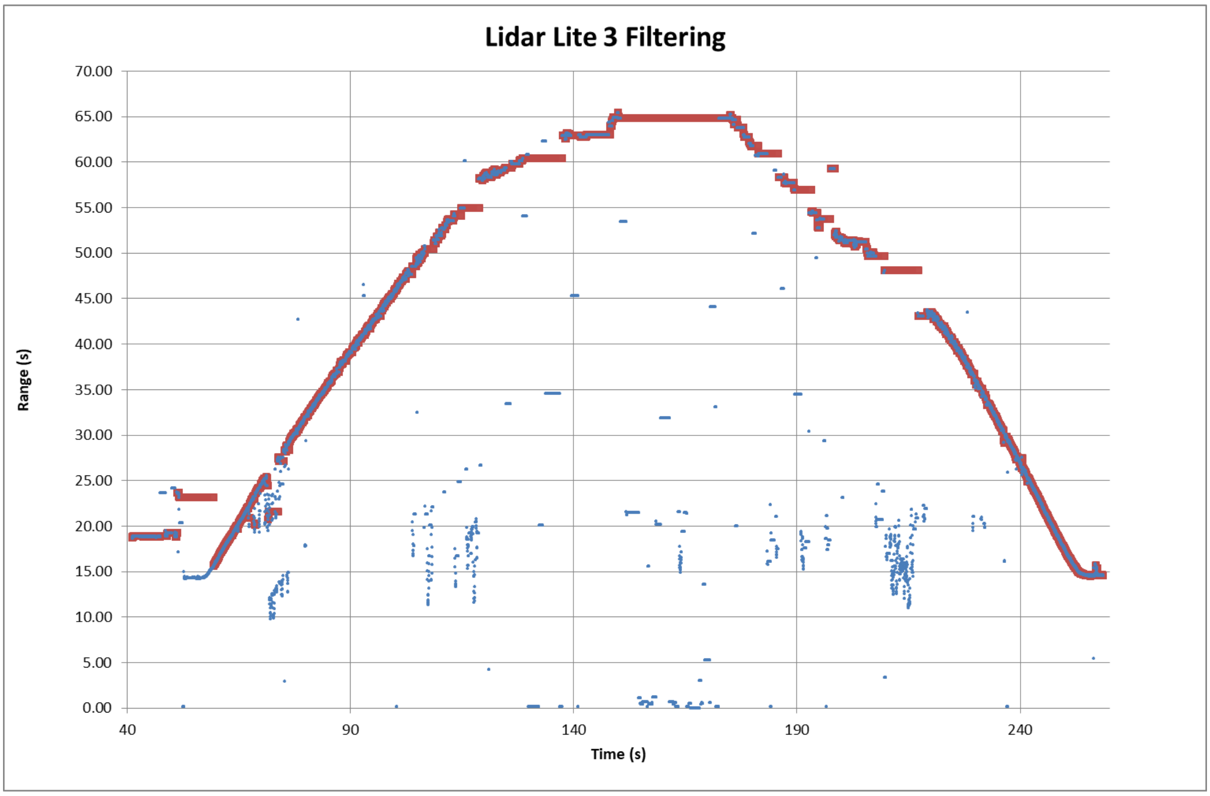


Figure 6: Filtered range

In blue, the raw data and, in red, the filter range estimator are plotted. A great part of incoherent plots are eliminated and the filter output gives (continuously) the best estimation of what could be the real distance.

The main difficulty in this kind of filter is to initialize (or re-initialize) the first output values. In real life other range sensors can help to overcome this problem.

Conclusion

Lidar Lite rangefinder is a very interesting sensor: it has good performance in range finding, good precision and high data rate.
The validation of the output is a real problem, even at short distance because the reflectivity of the target is not predictable.

On the Lidar Lite 3 from Garmin, no indicator has been specified to denote that the sensor is out of range when the PWM is used, without I2C link. The solution proposed in this study consists of processing the data to eliminate wrong outputs.

Annex

Annex1 : Arduino sketch

```
#include "PWM.hpp"

PWM my_pwm(2);          // Setup pin 2 for PWM
unsigned long heure;    // Arduino current time
unsigned long duree;    // High pulse duration
void setup() {
  pinMode(3,OUTPUT); // Trigger output
  digitalWrite(3,HIGH); // Lidar not triggered, std by
  Serial.begin(115200); // Serial output rate
  my_pwm.begin(true); // PWM on pin 2 reading PWM HIGH duration
  heure = 0;            // Time marker
  delay(2000);          // Delay to wait telemetry synchronization ready
}

void loop() {
  digitalWrite(3,LOW); // Fire the Lidar output!
  delay(16);            // Delay to check the output when the Lidar is monitored
  heure = micros();    // Current Arduino time
  duree = my_pwm.getValue(); // Get the range
  Serial.print(heure);Serial.print("\t");// Output the current time to the telemetry,
  Serial.println(duree);                  // and the data
  //digitalWrite(3,HIGH);// This capacity to monitor the output is not used in this report
  delay(9);             //Complete the 25 ms data rate
}
```

Annex2: Filter used in this study

The filtering is divided in three steps:

1. Eliminate wrong unambiguous data ;
2. Evaluate the quality of the measurements ;
3. Alpha/Beta filtering.

Looking, in real time, at the data during the trial shows that when the measurements seem coherent with the real distance, the values change gently around its average. It's like the data is alive. But, when the measurement is out of range, the data is completely stable, equal to the previous or change a lot very quickly. It's like the data is a flat encephalogram with uncontrolled agitations.

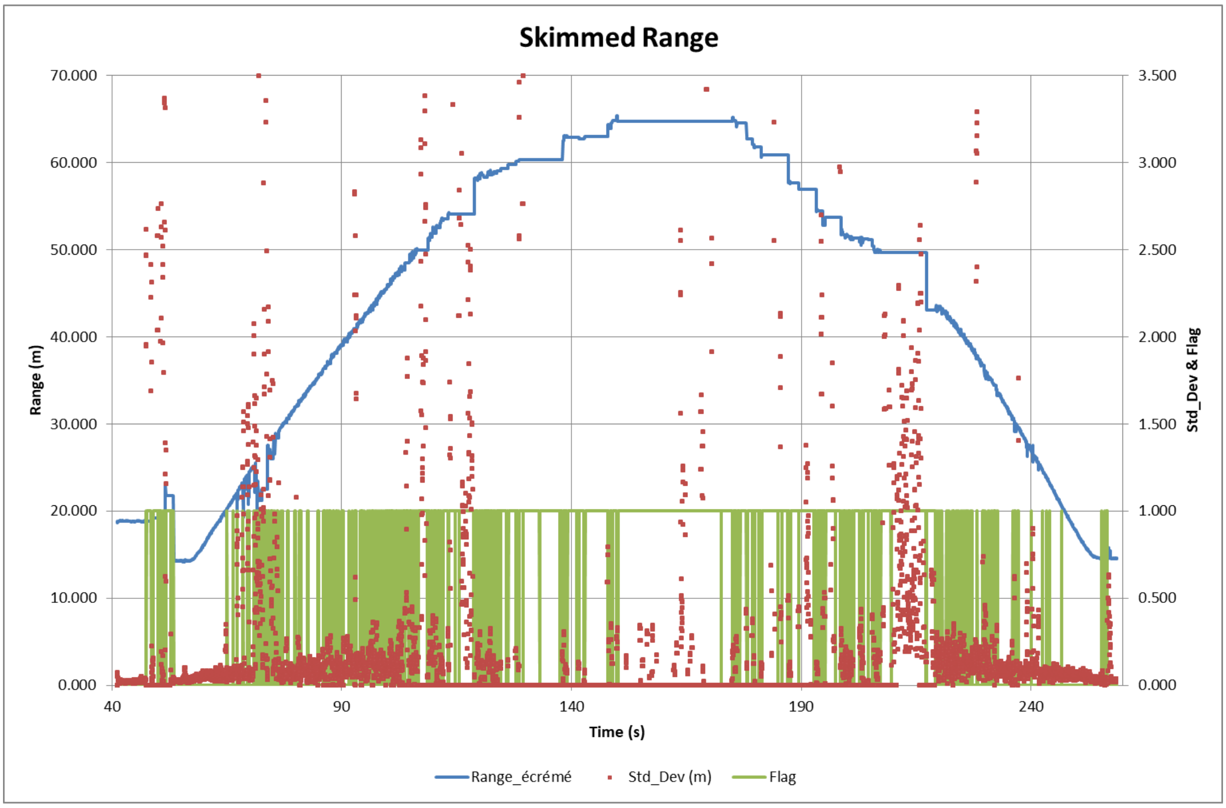
To convert this feeling in Math for data processing, the idea is to evaluate the current standard deviation σ of the few last measurements. If σ is equal to zero, that means the data is out of range and if this value is too high, something wrong probably happened.

The second idea to skim wrong data is to eliminate those which are too far from the model. Of course, this hypothesis supposes that the Lidar flies above flat ground!

Thus the first step consists of:

1 – 1 Evaluate the standard deviation σ, here on the last 5 measurements (125 ms) and invalidate the current measurement if σ is greater than 1 m/s or equal to 0 ;

1 – 2 Reset a flag to 0 if the difference between the current measurement and the last estimated range is below a threshold. The threshold (15m/Q) is not constant but depends on the confidence Q in the measurements. If not, the flag is set to 1 and discard the current measurement.



The second important step is to evaluate the quality of the measurements to be able to adjust the "depth" of the Alpha/Beta filter used in the third step.

This quality number can vary between two limits: Qmin=2 (low quality) and Qmax=10 (high quality).

If the measurement is near (<0.5m) the estimator, the quality increases. Or else, the quality decreases.

This method is already used in the altitude estimator of my MatrixPilot Jeffem's fork:

```
#if USE_LIDAR_ALTITUDE>0
    if (abs(lidar_altitude - fusion)<500)
        if (qual_lidar <qual_lidar_max ) qual_lidar=qual_lidar+1;
    else
        if (qual_lidar>qual_lidar_min) qual_lidar=qual_lidar-1;
#endif
```

Alpha/Beta filter is interesting by its simple implementation and it's no need to "recycle" the variables. It is based on least square Gaussian mathematics.

It gives an estimation of the speed, which could be useful later, to stabilize an altitude control loop, for example.

The velocity estimator, est_V, is first calculated by:

est_V = est_V + Beta/dt*(Range - est_Range) with dt the sample time (here 0.025s) Range the current measurement and est_Range the range estimator output.

Then the range estimator, est_Range, by:

Est_Range = Est_Range + Alpha*(Range - est_Range) + est_V*dt.

Alpha and Beta are calculated with the quality Q by:

Alpha = 2 * (2*Q - 1) / Q / (Q + 1) and Beta = 6 / Q / (Q + 1)