# General AutoISF

**Caveat:** *I am not a medically trained person and developed this method purely based on trial and numerical experimentation.*

This document describes the extension of my initial autoISF for use during the whole day. It explains my motivation, the method implemented, parameters to control the effect and shows improvements in glucose statistics. The ISF adaptations after meals are still under development and you are welcome to provide suggestions for further improvement. That part of the ISF adaptation may be of special interest to loopers with gastroparesis.

The initial autoISF was specific for situations of continued insulin resistance. These situations were characterized by glucose levels staying within a +/- 5% range, being above target and the absence of COB. Meanwhile a private test with presence of COB works well, too.

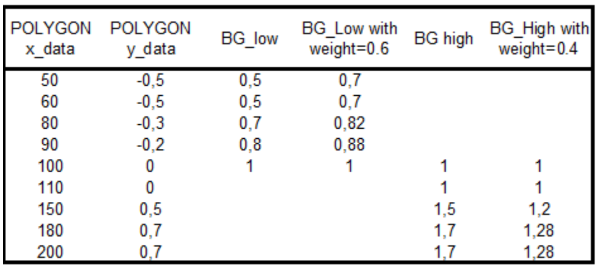
With the availability of Lyumjev more and more users loop in pure UAM mode without input of carbs and without pre-bolus. They use automation rules to change profile percentages and/or targets based on glucose thresholds or delta thresholds. This document describes an alternative method by continuously and gradually adapting ISF based on those glucose levels and deltas.

## *Background*

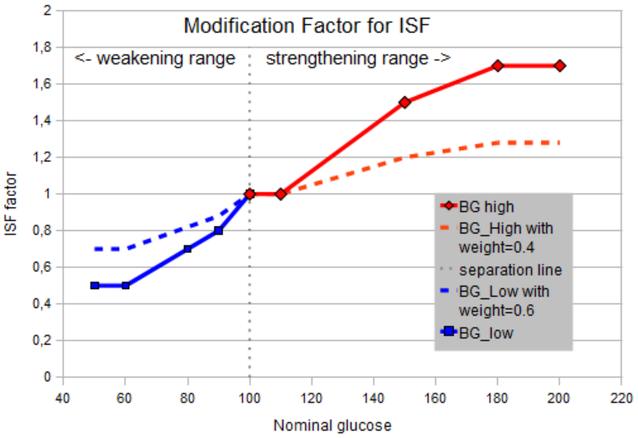
Automated changes of profile percentages have considerable limitations in their “if-then” capability. More importantly, these changes introduce sudden and considerable changes in loop parameters. In the simulation of engineering systems such jumps introduce instability and therefore have to be avoided absolutely. Instead, smooth transitions are used to gradually change the state. For example let us assume the automation rules say if glucose is above 150mg/dl then change the profile from 100% to 140%. A smoother approach could be to increase the percentage by 10% between 110 and 150mg/dl in 4 steps which is a rather tough task for the automation and most likely outright impossible.

## *Proposal*

Changing profile percentages looks like the ISF is no longer constant over the whole range of glucose values. A variable ISF means that reducing glucose by say 30mg/dl requires more insulin at higher levels than close to target. In my prototype I added a polygon consisting of pairs of (glucose / ISF correction) to map this variable behaviour:



| POLYGON x_data | POLYGON y_data | BG_low | BG_Low with weight=0.6 | BG high | BG_High with weight=0.4 |
|---|---|---|---|---|---|
| 50 | -0,5 | 0,5 | 0,7 | | |
| 60 | -0,5 | 0,5 | 0,7 | | |
| 80 | -0,3 | 0,7 | 0,82 | | |
| 90 | -0,2 | 0,8 | 0,88 | | |
| 100 | 0 | 1 | 1 | 1 | 1 |
| 110 | 0 | | | 1 | 1 |
| 150 | 0,5 | | | 1,5 | 1,2 |
| 180 | 0,7 | | | 1,7 | 1,28 |
| 200 | 0,7 | | | 1,7 | 1,28 |

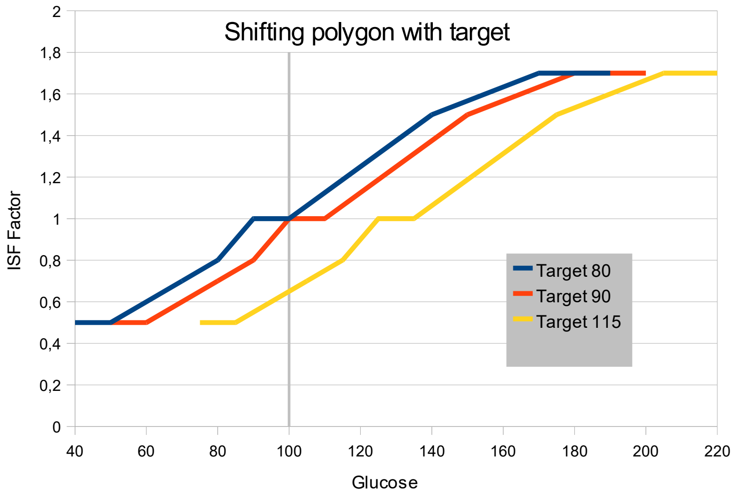


The solid lines show the master polygon data. The actual factors are interpolated between the points or extrapolated beyond the two ends of the ranges. The initial choice of correction factors was aligned to what I saw in examples of automations for profile percentages. To be able to adapt those without having to recompile the app each time when I wanted to change the polygon I introduced weighting factors to calculate how much of that correction should applied. A factor of “0” means there is no impact at all and I can gradually strengthen the effect. The figure above includes the weighted result examples as dashed lines.

There are two different, independent weighting factors, one for glucose being too high and another one for glucose being too low. With the low range factor I can use a very sensitive ISF to better keep me away from hypo.

The separation between high and low is shown above at 100mg/dl of “nominal glucose”. Nominal means the curves will be shifted horizontally such that the nominal 100 is at target_bg+10mg/dl:

| ISF Factor | Target 80 | Target 90 | Target 115 |
|---|---|---|---|
| 0,5 | 40 | 50 | 75 |
| 0,5 | 50 | 60 | 85 |
| 0,7 | 70 | 80 | 105 |
| 0,8 | 80 | 90 | 115 |
| 1 | 90 | 100 | 125 |
| 1 | 100 | 110 | 135 |
| 1,5 | 140 | 150 | 175 |
| 1,7 | 170 | 180 | 205 |
| 1,7 | 190 | 200 | 225 |



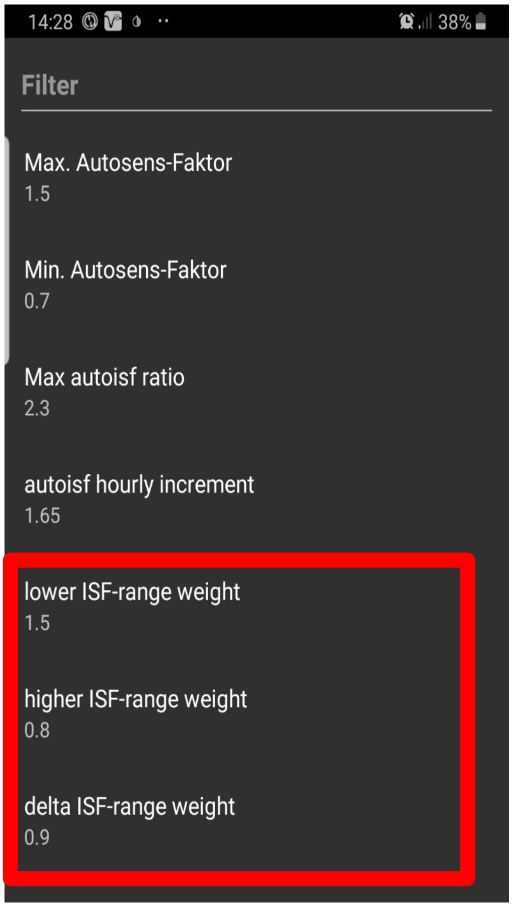
The figure above shows 3 examples of target_bg and where the correction lines with weighting of 1.0 will be located finally. The “+10mg/dl” was added to the target as an extra safety margin.

The equivalent method is used for glucose delta with its own polygon and weighting factor. However, high glucose and high delta should have a coupling effect which reflects that the delta impact depends on where glucose is at the moment. This is still in development.

The original “Max autoisf ratio” from the original autoISF is still used as the upper limit.

The strongest correction out of traditional autoISF, high glucose, high delta and autosense wins. If, however, glucose is below the nominal 100mg/dl, then the weakening of ISF wins.

Only ISF is adapted and nothing else just like in the original autoISF method.



Filter

Max. Autosens-Faktor
1.5

Min. Autosens-Faktor
0.7

Max autoisf ratio
2.3

autoisf hourly increment
1.65

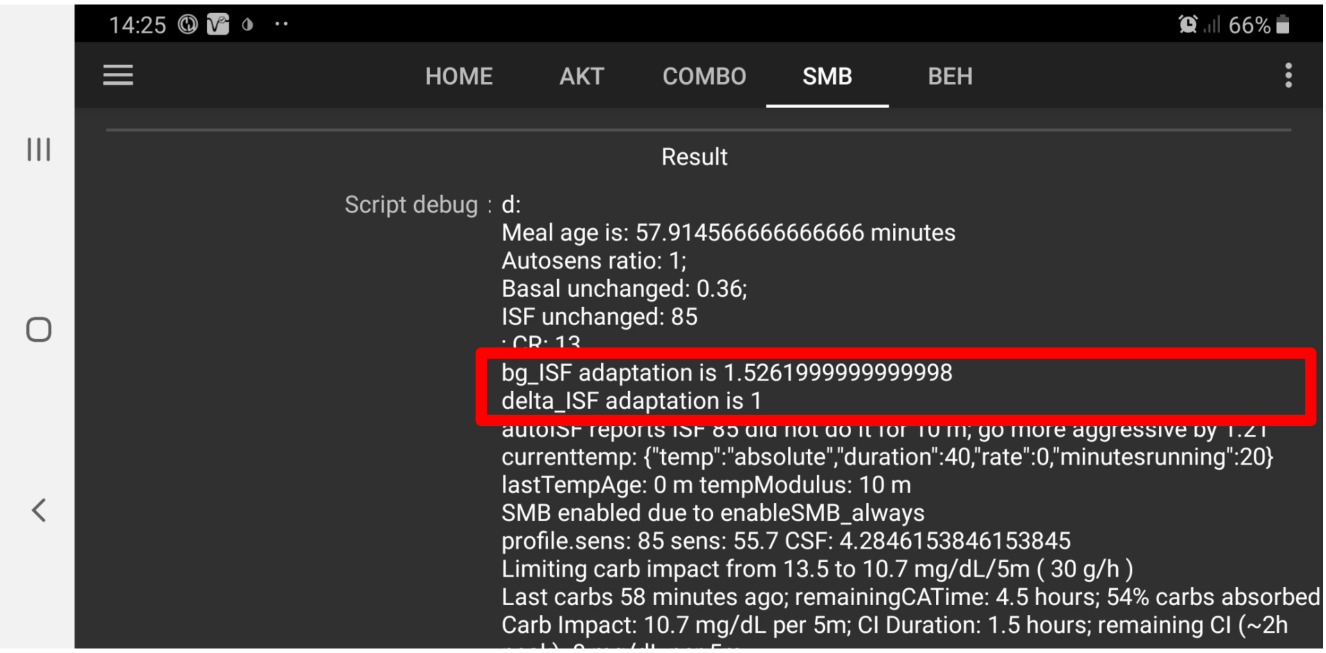
lower ISF-range weight
1.5

higher ISF-range weight
0.8

delta ISF-range weight
0.9

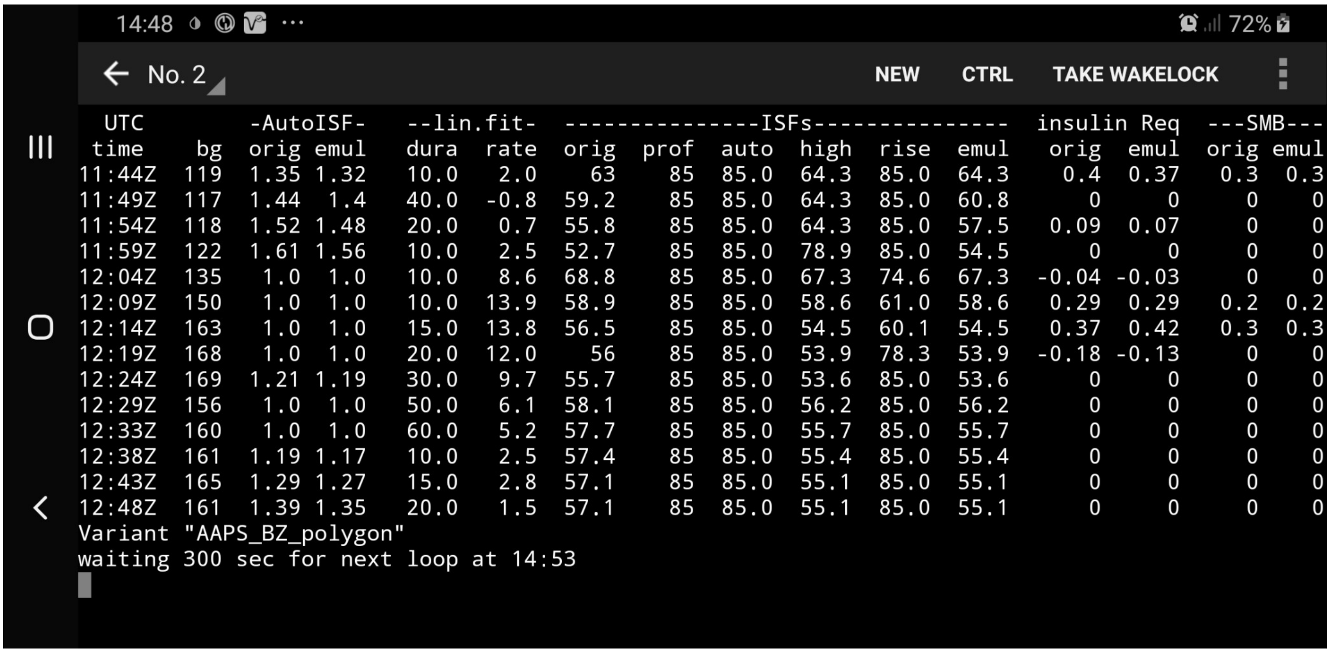
### *How can I see what the general autoISF does each time?*

The result section of the SMB-tab was extended to include feedback for the new alternatives, namely glucose too high, glucose too low, delta high. Here is an example case where glucose was too high and ISF would be strengthened by a factor 1.5. Traditional autoISF would strengthen ISF by 1.2 only, so the contribution from high glucose won.



```
Script debug : d:
               Meal age is: 57.914566666666666 minutes
               Autosens ratio: 1;
               Basal unchanged: 0.36;
               ISF unchanged: 85
               : CR: 13
               bg_ISF adaptation is 1.5261999999999998
               delta_ISF adaptation is 1
               autoISF reports ISF 85 did not do it for 10 m; go more aggressive by 1.21
               currenttemp: {"temp":"absolute","duration":40,"rate":0,"minutesrunning":20}
               lastTempAge: 0 m tempModulus: 10 m
               SMB enabled due to enableSMB_always
               profile.sens: 85 sens: 55.7 CSF: 4.2846153846153845
               Limiting carb impact from 13.5 to 10.7 mg/dL/5m ( 30 g/h )
               Last carbs 58 minutes ago; remainingCATime: 4.5 hours; 54% carbs absorbed
               Carb Impact: 10.7 mg/dL per 5m; CI Duration: 1.5 hours; remaining CI (~2h
               peak): 0 mg/dL per 5m
```

To see what happened in previous loops you can use the same methods available in the original, specific autoISF method. Here is an example from running the emulator on the AAPS phone

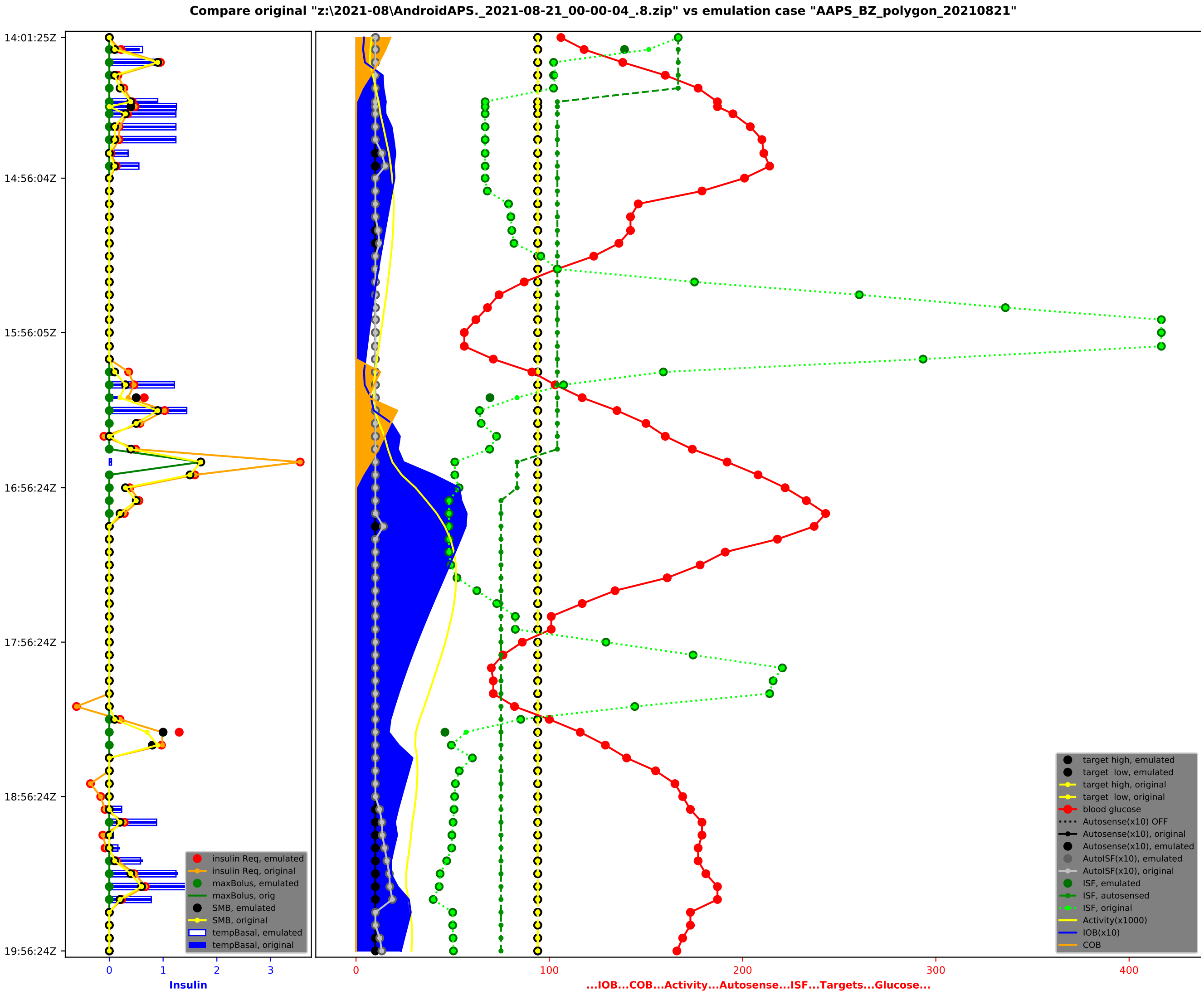


```
 UTC        -AutoISF-  --lin.fit-  ---------------ISFs--------------  insulin Req  ---SMB---
 time    bg  orig emul  dura  rate  orig  prof  auto  high  rise  emul  orig  emul  orig emul
11:44Z  119  1.35 1.32  10.0   2.0    63    85  85.0  64.3  85.0  64.3   0.4  0.37   0.3  0.3
11:49Z  117  1.44  1.4  40.0  -0.8  59.2    85  85.0  64.3  85.0  60.8     0     0     0    0
11:54Z  118  1.52 1.48  20.0   0.7  55.8    85  85.0  64.3  85.0  57.5  0.09  0.07     0    0
11:59Z  122  1.61 1.56  10.0   2.5  52.7    85  85.0  78.9  85.0  54.5     0     0     0    0
12:04Z  135   1.0  1.0  10.0   8.6  68.8    85  85.0  67.3  74.6  67.3 -0.04 -0.03     0    0
12:09Z  150   1.0  1.0  10.0  13.9  58.9    85  85.0  58.6  61.0  58.6  0.29  0.29   0.2  0.2
12:14Z  163   1.0  1.0  15.0  13.8  56.5    85  85.0  54.5  60.1  54.5  0.37  0.42   0.3  0.3
12:19Z  168   1.0  1.0  20.0  12.0    56    85  85.0  53.9  78.3  53.9 -0.18 -0.13     0    0
12:24Z  169  1.21 1.19  30.0   9.7  55.7    85  85.0  53.6  85.0  53.6     0     0     0    0
12:29Z  156   1.0  1.0  50.0   6.1  58.1    85  85.0  56.2  85.0  56.2     0     0     0    0
12:33Z  160   1.0  1.0  60.0   5.2  57.7    85  85.0  55.7  85.0  55.7     0     0     0    0
12:38Z  161  1.19 1.17  10.0   2.5  57.4    85  85.0  55.4  85.0  55.4     0     0     0    0
12:43Z  165  1.29 1.27  15.0   2.8  57.1    85  85.0  55.1  85.0  55.1     0     0     0    0
12:48Z  161  1.39 1.35  20.0   1.5  57.1    85  85.0  55.1  85.0  55.1     0     0     0    0
Variant "AAPS_BZ_polygon"
waiting 300 sec for next loop at 14:53
```

In this case I had opted to include columns for traditional autoISF, linear regression fit, the various ISF proposals, insulin req. and SMB for display on the phone. The meaning of the various ISF columns is as follows:

- orig  the actual value used by AAPS
- prof  the value defined in the pump profile settings
- auto  the value after applying autosense corrections
- high  the value proposed according to glucose being too high or too low
- rise  the value proposed based on high delta
- emul  the final value suggested by the emulation; if it differs from “orig” then the emulation used different weighting factors to test that impact

On Windows the emulator also creates a visual representation of what happened in the various phases:


Compare original "z:\2021-08\AndroidAPS._2021-08-21_00-00-04_.8.zip" vs emulation case "AAPS_BZ_polygon_20210821"

After 19:00UTC you see the traditional autoISF building up and there are two phases of low glucose managed by more sensitive ISF and three phases of high glucose managed by less sensitive ISF.

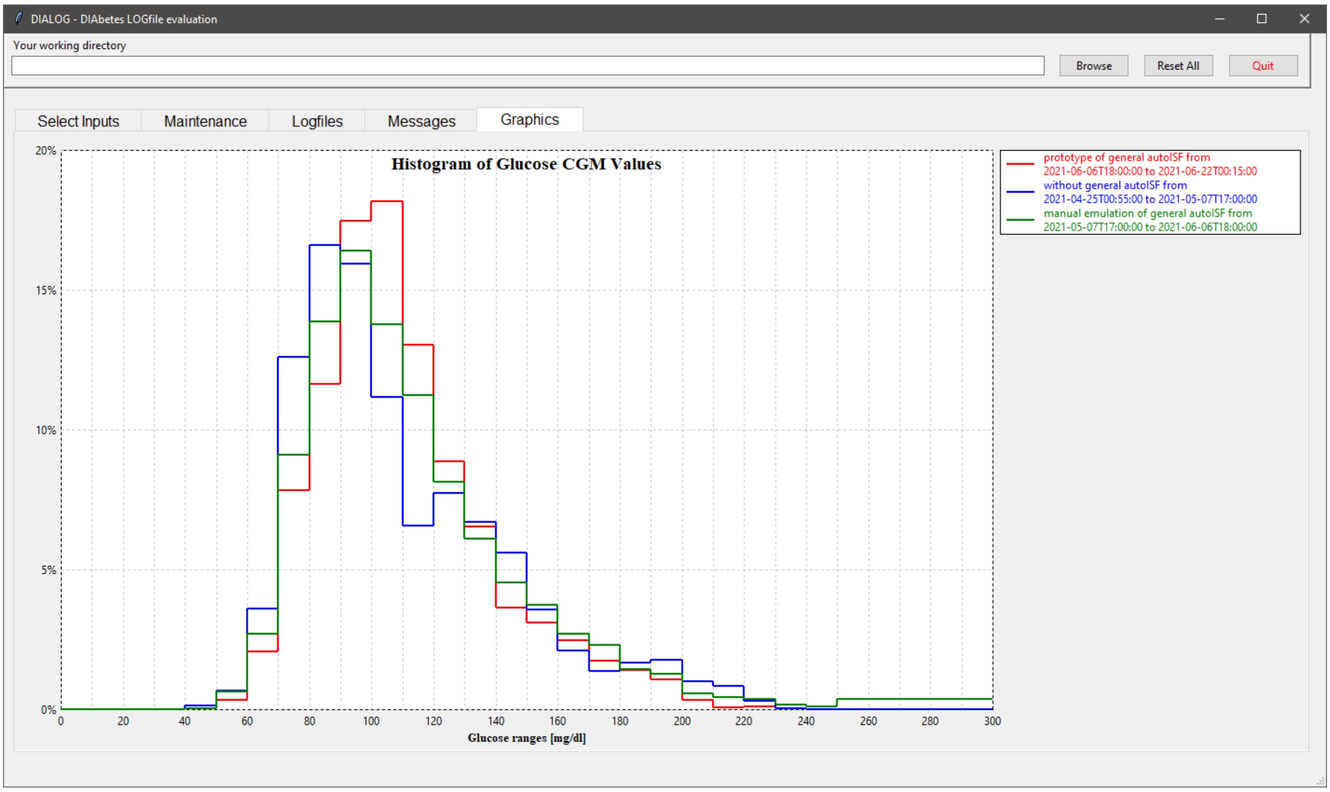
### *First Results*

All the following comparisons were done during a phase of testing Lyumjev. The overall statistical data for the test phases are summarized in this table:

| Insulin | general autoISF method | from | to | count | avg | median | std.dev | lower | TIR[%] | higher | GVI | PGS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Lyumjev | none | 2021-04-25 00:55 | 2021-05-07 17:00 | 3544 | 108,1 | 98 | 33,7 | 5 | 89,6 | 5,3 | 1,37 | 15,33 |
| Lyumjev | manual emulation | 2021-05-07 17:00 | 2021-06-06 18:00 | 8425 | 110,4 | 103 | 33,5 | 3,8 | 91,8 | 4,4 | 1,46 | 13,26 |
| Lyumjev | AAPS prototype | 2021-06-06 18:00 | 2021-06-22 00:15 | 4334 | 108,4 | 103 | 27,7 | 3 | 94,6 | 2,5 | 1,49 | 8,75 |
| FIASP | AAPS prototype | 2021-08-05 22:30 | 2021-08-24 12:00 | 5205 | 112,2 | 107 | 29,2 | 2,7 | 94,3 | 3 | 1,51 | 9,64 |

The improvement can be seen clearly in the numbers for TIR and PGS (Patient Glycemic Status as shown in Nightscout or NS-Reporter; lower is better). By the way, with relatively high TIR the numbers for PGS are dominated by the TIR contribution and therefore both tell the same story.

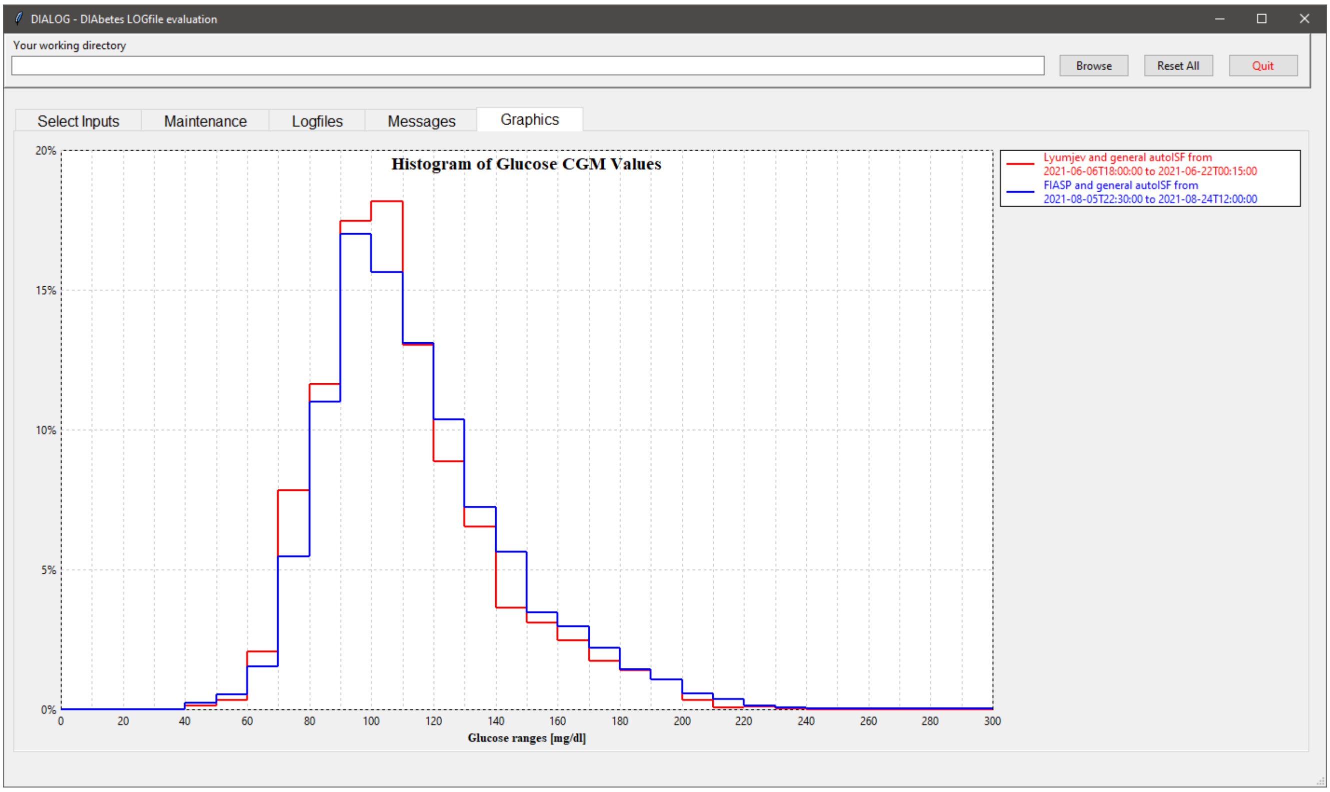
Interestingly the new algorithm results in a larger improvement in the statistical numbers than the change of insulin type when I returned to FIASP after the Lyumjev test phase. However, the histogram shows more detail and provides hints for further improvement.



- The blue curve shows the base results, i.e. before starting the development of general autoISF
- The green curve shows the initial effect of manually emulating the general autoISF method
- The red curve shows first results with the prototype implementation of the general autoISF

One clear effect is the reduction of time below range and close to that low range border. Whereas that was already visible during the emulation phase the prototype implementation improved it further, may be because it could now also be active during sleep. The other reason is that with emulation I can only add small boli after the loop executed but cannot reduce a bolus that is already injected.

Once my test supply of Lyumjev was used up I returned to FIASP and tuned the parameters further. The comparison Lyumjev versus FIASP with both using the general autoISF prototype looks like this:


DIALOG - DIAbetes LOGfile evaluation
Your working directory
Browse
Reset All
Quit
Select Inputs
Maintenance
Logfiles
Messages
Graphics
Histogram of Glucose CGM Values
Lyumjev and general autoISF from
2021-06-06T18:00:00 to 2021-06-22T00:15:00
FIASP and general autoISF from
2021-08-05T22:30:00 to 2021-08-24T12:00:00
20%
15%
10%
5%
0%
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
Glucose ranges [mg/dl]

There are less values within or close to the hypo range which is a combined effect of a small increase in the target and sharper action of the polygon lower end weight. More time spent in the range 120-180 requires further tuning or it may in part be due to the characteristics of Lyumjev.

## *How would it interact with Automation rules?*

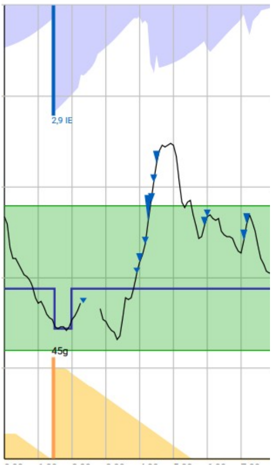
Obviously there is a conflict when automation changes profiles or their percentages. Both, the automation and this extended general autoISF would change ISF in the same direction which must be avoided under all circumstances. Currently none of the two knows about the other action and therefore this means that the user has to make sure that only one of these methods should be active.

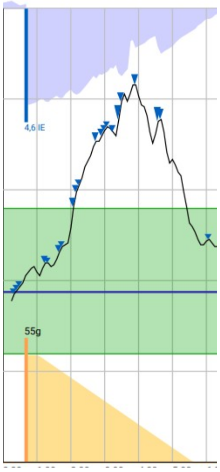
Automation rules which change the target may be less of a problem, especially when they where already activated in conjunction with profile actions. If automation was used to avoid hypos then the general autoISF would act in the same direction and its weighting and the target increase have to be adjusted together.

## *Testing special adaptations after meals*

In my daily patterns I often saw a rise in glucose levels after meals. Typically

- this rise started 45-60 minutes after the meal
- continued for 2 hours in a nearly straight line
- reached levels beyond 180 or higher
- and then came down rather slowly.





This looks like a digestion which is rather slow and can therefore not be treated by strengthening the IC ratio because that would lead to hypos. For loopers with gastroparesis theses situations may sound familiar although with even longer time scales. In order to reduce these postprandial rises I introduced an additional ISF contribution based on those rises and independent of the delta contribution mentioned above.

First, there are two new menu options:

1. postprandial ISF time window – defines for how many hours after the last carbs the contribution should be active. Please observe the related menu option just above which defines your maximum resorption time.

2. Enable autoISF postprandial all day – with this switch you can enable the feature always, e.g. for times longer than the maximum 10 hours. This can be especially useful in case of gastroparesis. In my case and after enabling it I saw it becoming active to the fight the morning rise due to the dawn phenomenon.

Second, the additional weighting factor “*postmeal_ISF_weight*” was introduced and added to the absorption menu. How is this ISF adaption factor calculated?

## 1st try in the emulator using a rise rate from a linear fit to the glucose curve

In this case I used a linear regression analysis, i.e. the best fit of a straight line to the glucose curve. The rise named “*slope_70*” was multiplied by “*postmeal_ISF_weight*” from above:

ISF_factor = 1 + slope_70 * postmeal_ISF_weight

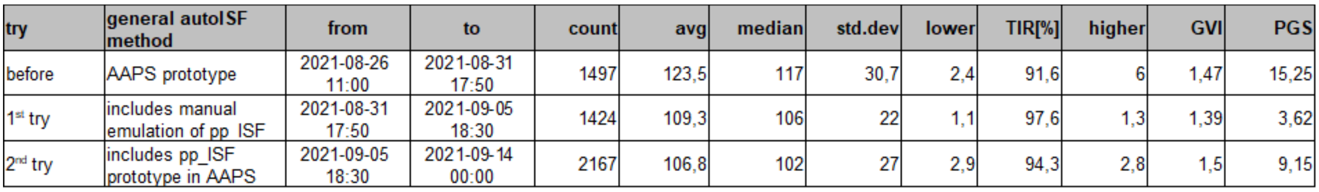
As a result I saw a reduction of the postmeal peaks to below 160mg/dl and a general reduction in the frequency of glucose values beyond 140mg/dl. TIR and GVI also improved.

## 2nd try in AAPS using the regular “*delta*”

After running “1st try” for a week I was fed up with having to manually release those incremental boli all the time. Also, the method looked very promising and I implemented it in AAPS directly. As the linear regression was not implemented I just used “*delta*” for a start:

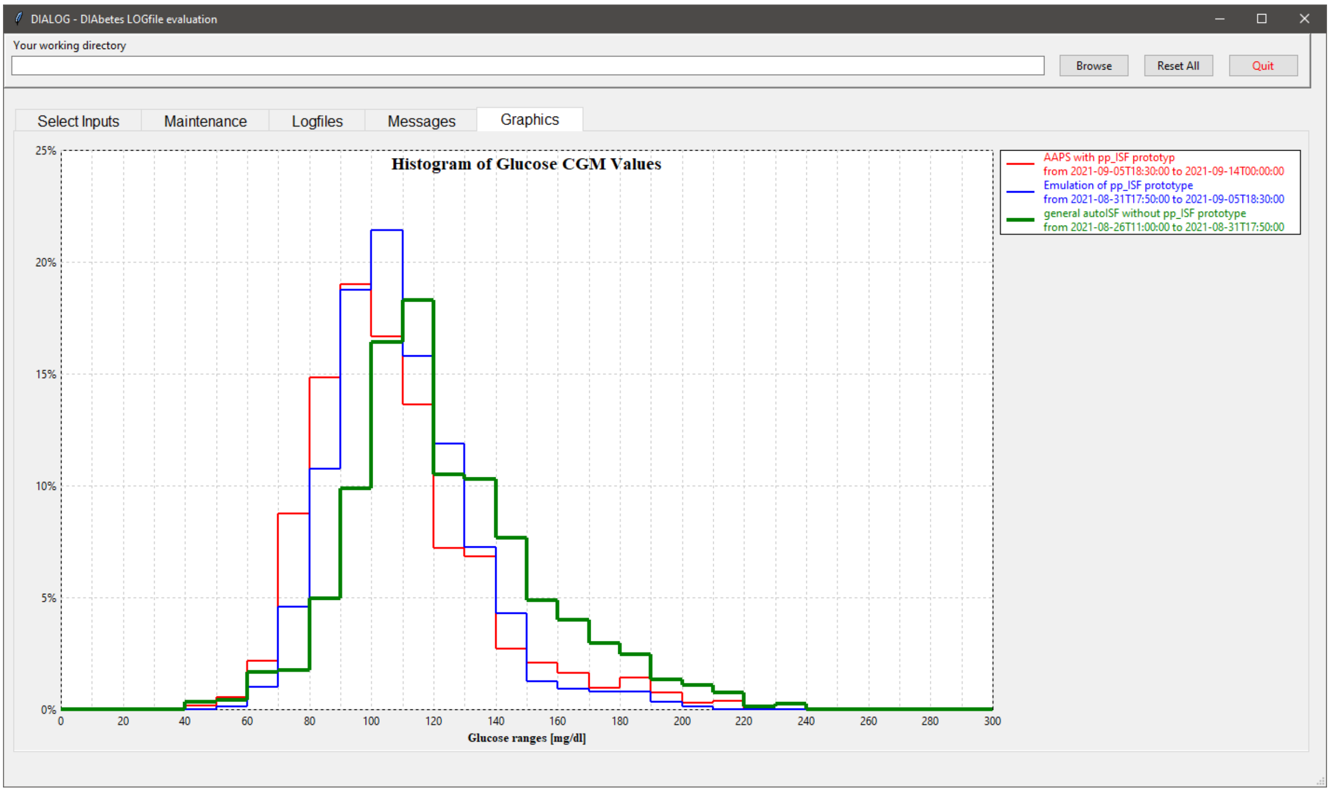
ISF_factor = 1 + delta * postmeal_ISF_weight

The results in the statistical values are very similar to “1st try”:



| try | general autoISF method | from | to | count | avg | median | std.dev | lower | TIR[%] | higher | GVI | PGS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| before | AAPS prototype | 2021-08-26 11:00 | 2021-08-31 17:50 | 1497 | 123,5 | 117 | 30,7 | 2,4 | 91,6 | 6 | 1,47 | 15,25 |
| 1st try | includes manual emulation of pp ISF | 2021-08-31 17:50 | 2021-09-05 18:30 | 1424 | 109,3 | 106 | 22 | 1,1 | 97,6 | 1,3 | 1,39 | 3,62 |
| 2nd try | includes pp_ISF prototype in AAPS | 2021-09-05 18:30 | 2021-09-14 00:00 | 2167 | 106,8 | 102 | 27 | 2,9 | 94,3 | 2,8 | 1,5 | 9,15 |

Currently I am running the “1st try” in parallel to the AAPS “2nd try” via the emulator and the bolus recommendations vary only occasionally.  If they do then mostly just by 0.1U. This is another indication that these two methods are very similar. The histogram below shows both variants are fairly equivalent but clearly an improvement compared to the situation without the postprandial ISF contribution.


DIALOG - DIAbetes LOGfile evaluation
Your working directory
Browse
Reset All
Quit
Select Inputs
Maintenance
Logfiles
Messages
Graphics
Histogram of Glucose CGM Values
25%
20%
15%
10%
5%
0%
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
Glucose ranges [mg/dl]
AAPS with pp_ISF prototyp
from 2021-09-05T18:30:00 to 2021-09-14T00:00:00
Emulation of pp_ISF prototype
from 2021-08-31T17:50:00 to 2021-09-05T18:30:00
general autoISF without pp_ISF prototype
from 2021-08-26T11:00:00 to 2021-08-31T17:50:00

The evaluation periods were rather short and consequentially numbers can change somewhat with longer periods. At least the perids are back to back without gap, so seasonal or similar trends can be excluded.

## Ideas for further and alternative tries

There are several ways for determining a suitable alternative rise rate. Meanwhile I included a linear and a quadratic best fit the glucose curve.  This offers various options for determining a representative delta which can be tested with the emulator:

1. use the traditional "*short_avgdelta*" already available in standard AAPS. It should be more stable than delta – or the option "always use short_avgdelta" was activated anyway.

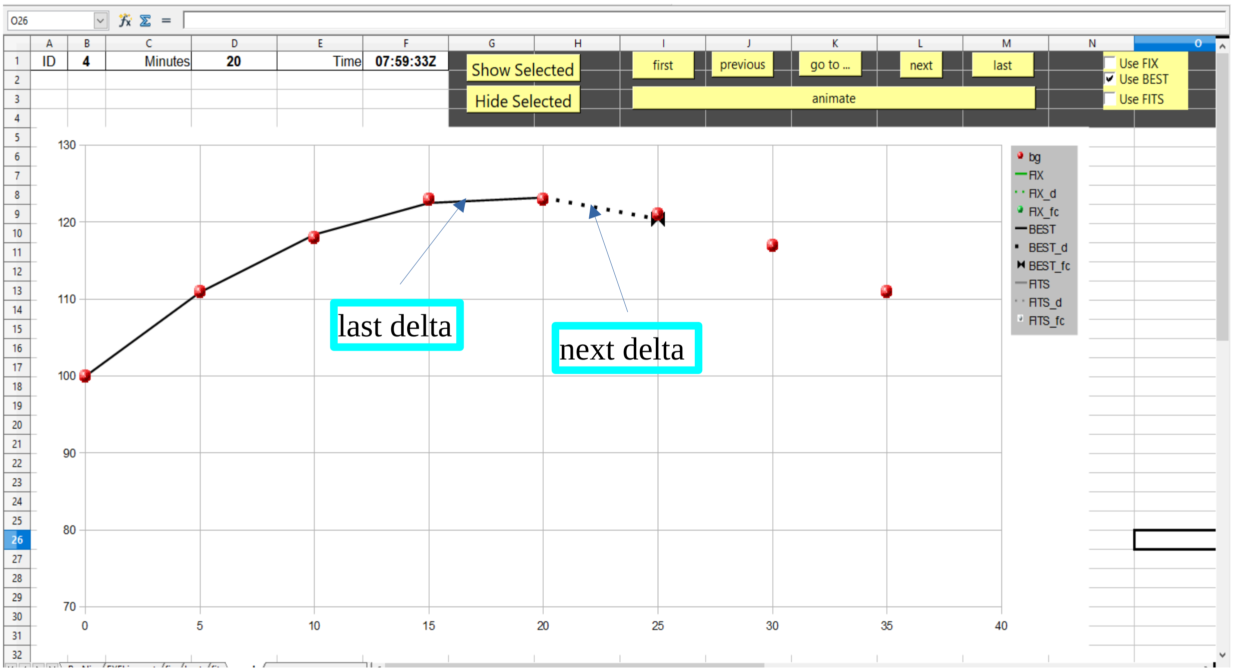
2. use the traditional "*long_avgdelta*" already available in standard AAPS.

3. From the new linear fit use "slope05", the best linear fit fro the recent 5 minutes. This corresponds to the delta time frame.

4. From the new linear fit use "slope15", the best linear fit for the recent 15 minutes. This corresponds to the short_avgdelta time frame.

5. From the new linear fit use "slope40", the best linear fit for the recent 40 minutes. This corresponds to the long_avgdelta time frame.

6. From the new quadratic fit use "*parabola_fit_last_delta*" or "*parabola_fit_next_delta*".



ID 4 Minutes 20 Time 07:59:33Z

□ Use FIX
☑ Use BEST
□ Use FITS

This graph is an extract covering 40 minutes of glucose values and shows some of the terms used in the text. The solid black line segments represent the best fit of a parabola at time 20 minutes. This fit matches the glucose history for 20 minutes (*parabola_fit_minutes*). The dotted black line is the forecast where the parabola would be 5 minutes later. Sadly it is not always as close as shown here.

The parabolic fit also delivers the "*parabola_fit_correlation*" which is a meassure of how well the fit matches the glucose values. A value of 1 means perfect match and I suspect any value below 0.9 (or even below 0.95) is just not good enough to be used as a promising forecast.

7. A weighted average or the combination of the deltas could als be tested, especially if the parabola fit has good correlation and fit length of 20 minutes or more.

For experimenting in the emulator these are typical values that may be used in your own formulas:

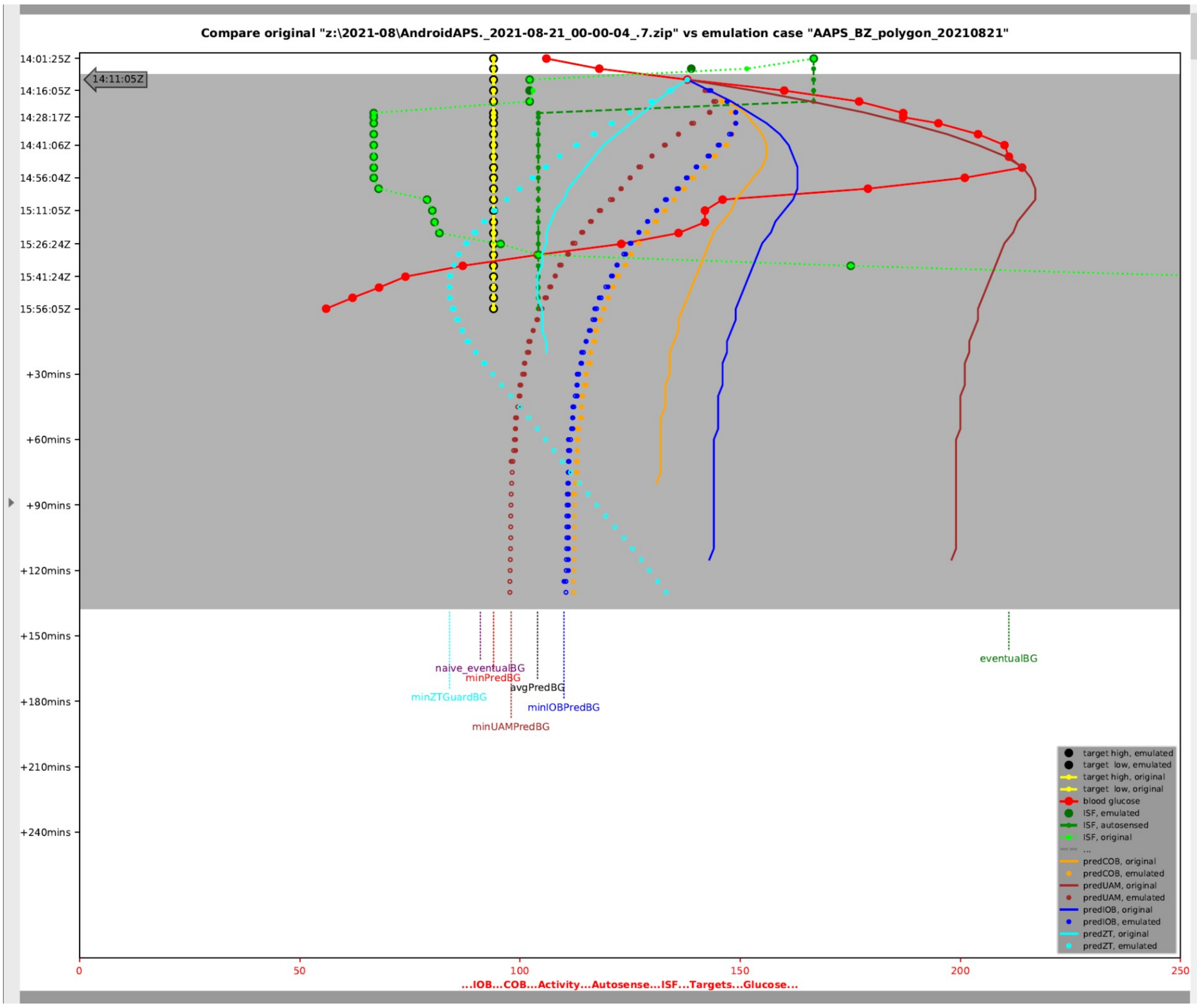
| | | |
|---|---|---|
| • | glucose_status['parabola_fit_minutes'] | how long the parabolic fit delivers useful data |
| • | glucose_status['parabola_fit_last_delta'] | last delta based on parabolic fit |
| • | glucose_status['parabola_fit_next_delta'] | forecast of next delta based on parabolic fit |
| • | glucose_status['parabola_fit_correlation'] | measure of parabola fit quality<br>from excellent(1.0) to unusable(0.0) |
| • | glucose_status['slope05'] | delta based on linear fit for last 5 minutes |
| • | glucose_status['slope15'] | delta based on linear fit for last 15 minutes |
| • | glucose_status['slope40'] | delta based on linear fit for last 40 minutes |
| • | meal_data['mealCOB'] | carbs remaining frm last meal entered |
| • | meal_data['lastCarbTime'] | when the last carbs were entered |
| • | profile['postmeal_ISF_weight'] | adjust the weight of the postprandial effect |

## *Open Issue*

The 4 hour predictions are determined based on the assumption that the basic pump profile does not change within that time frame. This is not always true, e.g. when a temp profile switch ends or the circadian profile triggers a change. With this even more variable ISF that assumption is definitely no longer true and the question is whether that should be accounted for in the prediction algorithm.

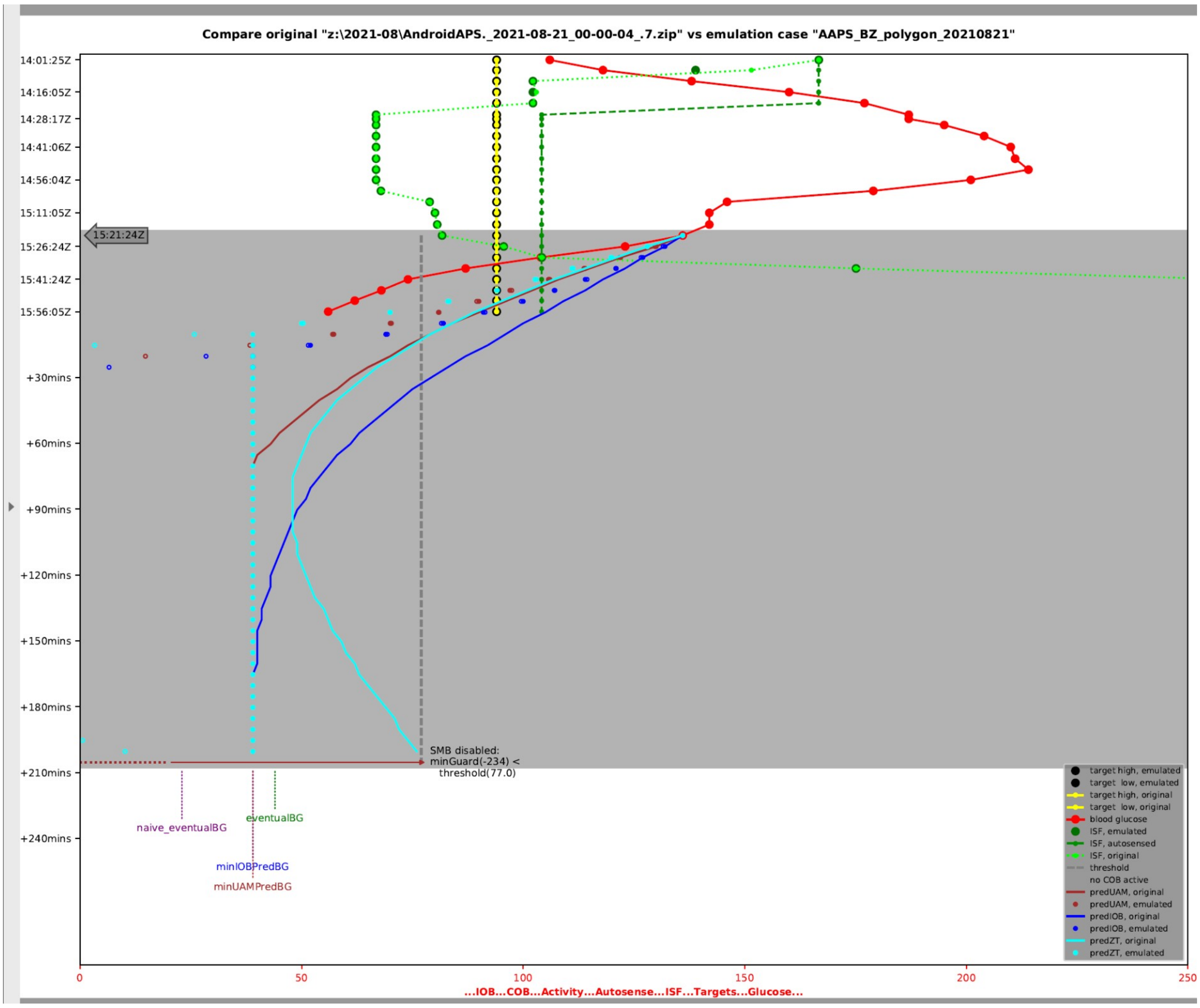
Before going live with adaptable predictions I added this in my emulator first to investigate the effect. I do not understand the resulting impact on the predictions but may be I do not really understand the various predictions in the first place. I look at two different situations:

- original predictions are rising: the IOB should have less power in the near future compared to a constant ISF, i.e. I expect the modified predictions to be at higher glucose in that phase.



However, the I see these predictions (dotted curves) are below those with ISF fixed (solid curves) during the prediction phase.

- original predictions are falling the IOB should have more power in the near future compared to a constant ISF, i.e. I expect the modified predictions to be at lower glucose in that phase.



Here the change is according to my expectation.