To prebolus or not to prebolus – that is the question

In this document I compare how autoISF 2.2.8.1 reacts to the same meal taken with or without a manual prebolus at the start of the meal. Carbs were not announced in both cases so it is a comparison between traditional HCL (Hybrid Closed Loop) and FCL (Full Closed Loop). It was triggered as a safety test to see what happens if a user with FCL settings administered a manual bolus without adapting related settings.

The outcome in both cases was surprisingly similar, at least in my case and for this type of meal. Would this also be true for other users? Would this also be true for other meals? Is there some basic rule that can be extracted?

CAVEAT: Because of all these questions do not infer easily that this n=1 situation is the same for you and your meals. The attachment lists the processes and analysis steps I went through in case you want to repeat them for your own case.

Background

The meals were taken at 4 consecutive days from Jan, 8th to Jan 11th 2023. It consisted of oat flakes, plain yoghurt, apple and vanilla sugar. I do not weigh any of them but for example just pour the oat flakes from the bag to give my preferred mix of dry and liquid. My estimate of the contents is 85g carbs, 17g fat, 26g protein, 9g fibre. Even during HCL times I never declared equivalent carbs but let SMBs handle them. With FCL it has really become irrelevant.

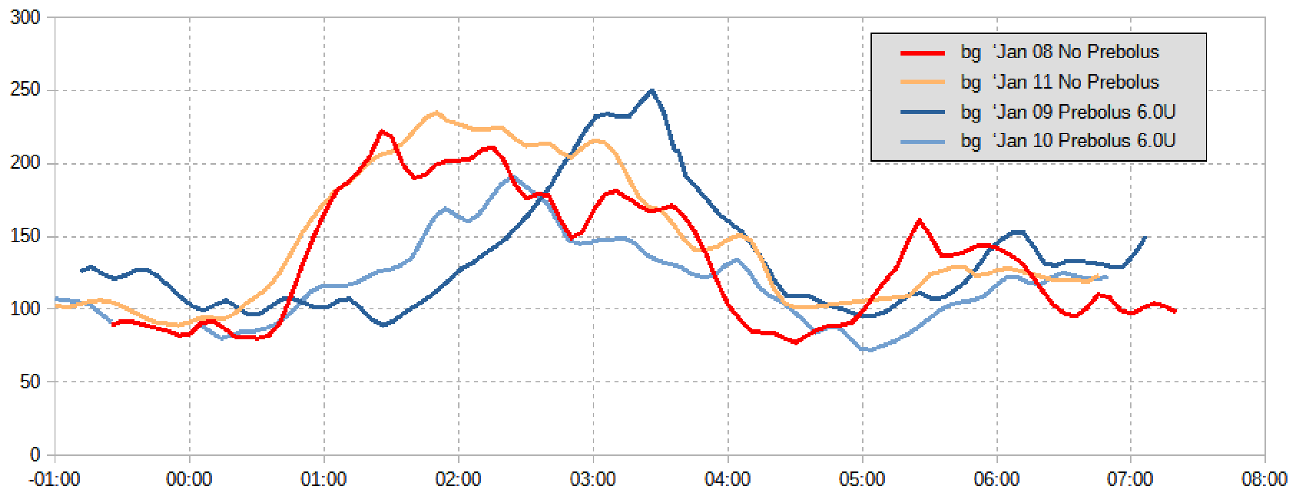
To slow down carbs arriving in the blood I used to have some very fatty starter like sausage or boiled egg. As I found out recently that sometimes backfired when the fat was digested quicker than expected and therefore blocked the insulin in the blood from doing its intended work – classic fat resistance from free fatty acids. At least that is my favourite theory and was recently proven by lower and shorter glucose peaks when skipping those starters. But that is another story.

Of those four days in the sample two were with a prebolus of 6U and two without any. All were taken at the same time of day.

		Insulin during 5 hours after meal				
Date	**Condition**	**Bolus**	**SMB**	**Basal**	**Sum**	**bg max**
‘Jan 08	**No Prebolus**		12,8	1,5	14,3	153
‘Jan 09	**Prebolus 6.0U**	6,0	6,7	0,3	13,0	161
‘Jan 10	**Prebolus 6.0U**	6,0	4,6	0,4	11,0	250
‘Jan 11	**No Prebolus**		11,4	0,6	12,0	191

Comparing glucose history

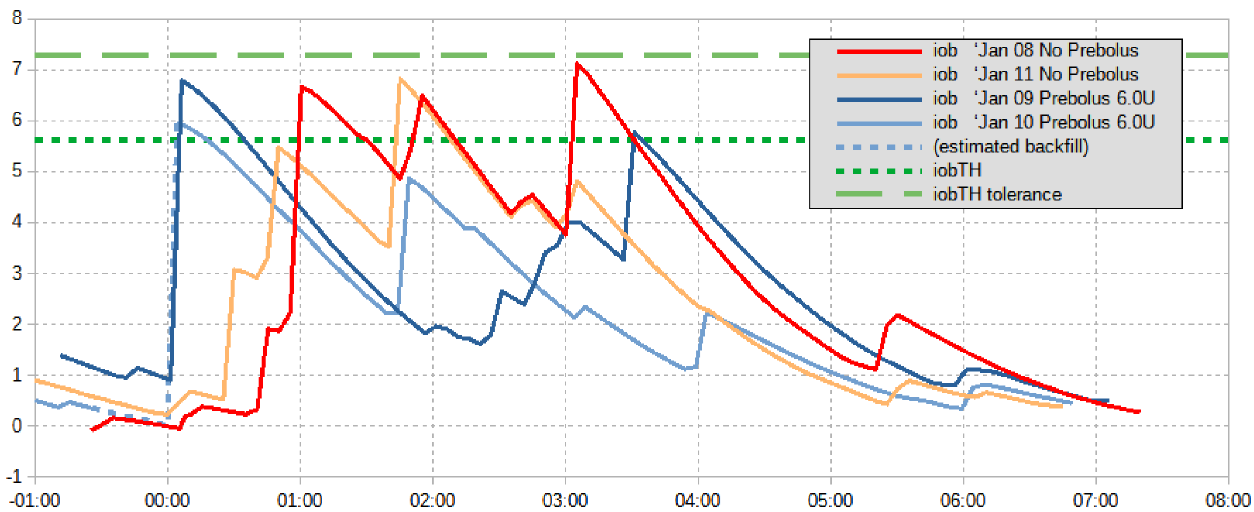
Here is the plot of glucose shifted in time such that time "00:00" means start of meal:



The red-ish curves are for FCL mode and are very similar between them. The blue-ish curves are for cases with prebolus and are similar between them. In total all four are somehow similar apart from the time shift in glucose onset which is to be expected.

Comparing the IOB history

This is getting more interesting than the glucose behaviour:



First some explanations how to read this graph. The light blue curve for 10.Jan. has a gap just before the meal because the Eversense transmitter needed to be recharged, a very unfortunate timing on my side. I cannot remember but may be that was the reason why I gave the bolus that day and did not rely on FCL. Re-attaching the transmitter means the initial 20 minutes have less stable readings and I shut off SMBs for such periods. In the glucose curve in the preceding section this gap is hidden by the other curves. The dashed line bit is an estimate of what it would look like without the CGM gap.

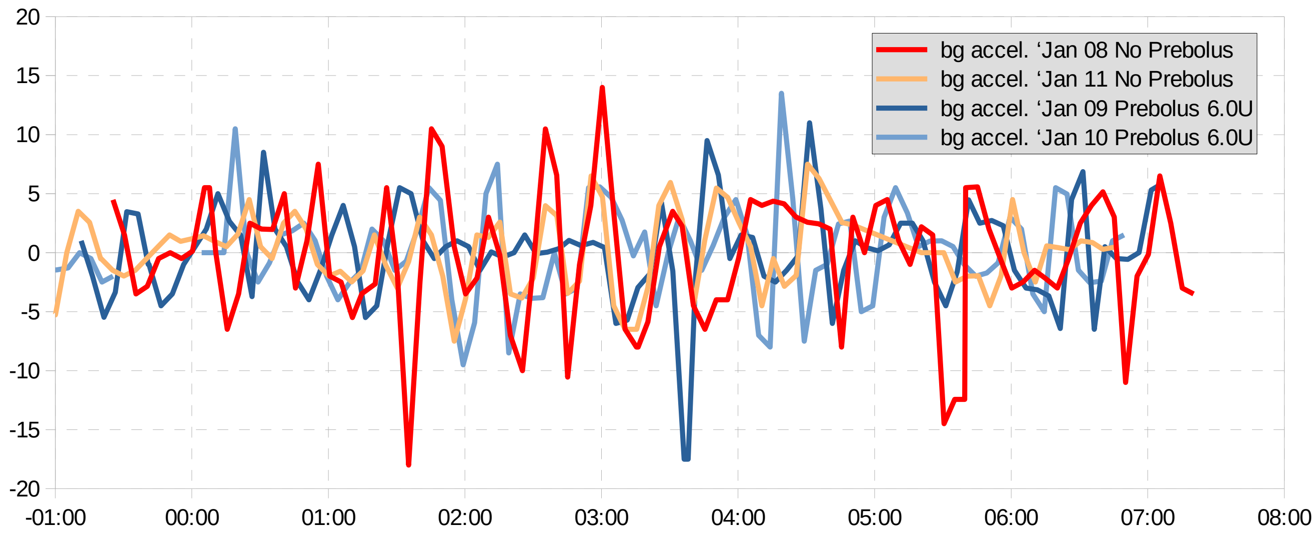
The two dashed lines in green show the levels of my active iobTH, the IOB threshold as advocated by BerNie in his FCL guide. I do not use automations to shut off SMB but have an autoISF internal method in my prototype which does it. It has the additional feature of allowing to exceed the regular iobTH threshold by 30% (iobTH tolerance) on the first occasion of wanting to pass it. The SMB is reduced in such cases accordingly. Thereafter SMB are disabled until below the regular iobTH threshold again. In all this happened four times on the two FCL days. On the prebolus days it came close once. This capping of SMB due to exceeding the iobTH tolerance level become active on 10.Jan. 3 hours after meal start. Insulin required was 4U and SMB delivery ratio was 0.95 normally resulting in 3.8U. But because of the capping only 3.5U were allowed.

The IOB levels were similar on all four days. That is a result of tuning the weights such that the same meal can be managed in FCL mode. So no real surprise here.

Looking at the decay of the prebolus I nevertheless see SMBs after two hours. From that time onwards I was in FCL territory anyway and there is no real difference between the two cases downstream.

Comparing glucose accelerations

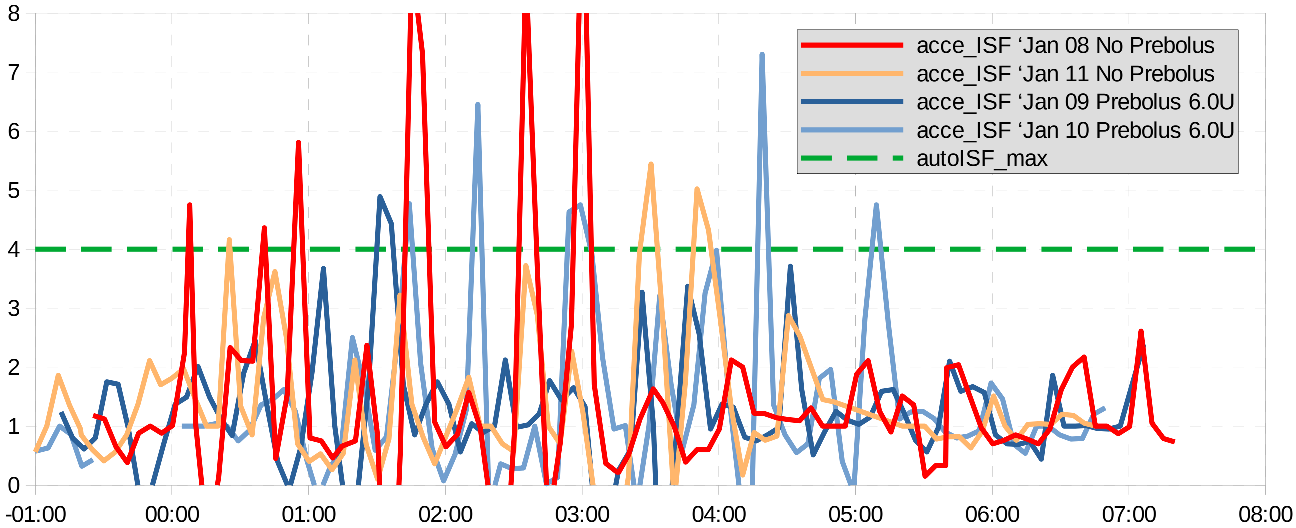
As expected the glucose curves have different basis shapes in the two cases. With prebolus there is no sharp initial rise and therefore lower accelerations can be expected. That would hopefully trigger less SMB to avoid overdosing. Here is what the glucose accelerations looked like on the four days:



Like mentioned in the preceding section the focus is on the first 2 hours. Extreme negative acceleration is of less interest as the important question is about overdosing. There are some peaks in the positive direction early in the prebolus cases. Would they lead to unwanted SMBs? That becomes clear in the next section.

Comparing acce_ISF factor curves

The acce_ISF factors include the influence of the weights and other effects. Negative factors were cut off in the graph to get a clearer view. Here is the graph:



The original acceleration peaks on the prebolus days are no longer prominent. What was the reason?

Digging deeper in the CSV-file for 10.Jan. reveals that the first acceleration peak had a fit correlation of only 90.15%. At 100% the acce_ISF factor would be at maximum and at 90% it is zero. Therefore it is obvious why the factor was only 1.04, i.e. negligible in its impact.

Digging deeper in the TXT-file for 9.Jan. I see an acce_ISF factor of 1.9 but there is also a hint in the reason field of the loop:

```
56 add'l carbs req w/in 35m
```

That of course shuts off any SMB. Half an hour into the meal that carb requirement is obviously covered by the carbs unknown to the loop.

On both prebolus days the iobTH criterion prevents SMBs anyway that early in the cycle.

Conclusion

This analysis confirmed my subjective observation that in my situation I can use a prebolus or not without changing any setting. Especially in situations which are difficult to judge or predict I was happy to leave it to FCL to manage the meal. Examples are unknown meal content from the take away or when there was significant IOB still present.

It appears to be important to tune the FCL to administer the amount of insulin otherwise provided by the carb calculator in HCL mode. Then the iobTH will take care to not overdose. But this may also depend on my special 30% tolerance method which is not available in the public autoISF.

It is absolutely impossible to infer from this autoISF specific study what it means for the other AAPS variants.

I thought about whether the emulator could be used to investigate robustness further. However, that would require to extend its scope to deliver modified glucose and modified IOB.

Recommendation

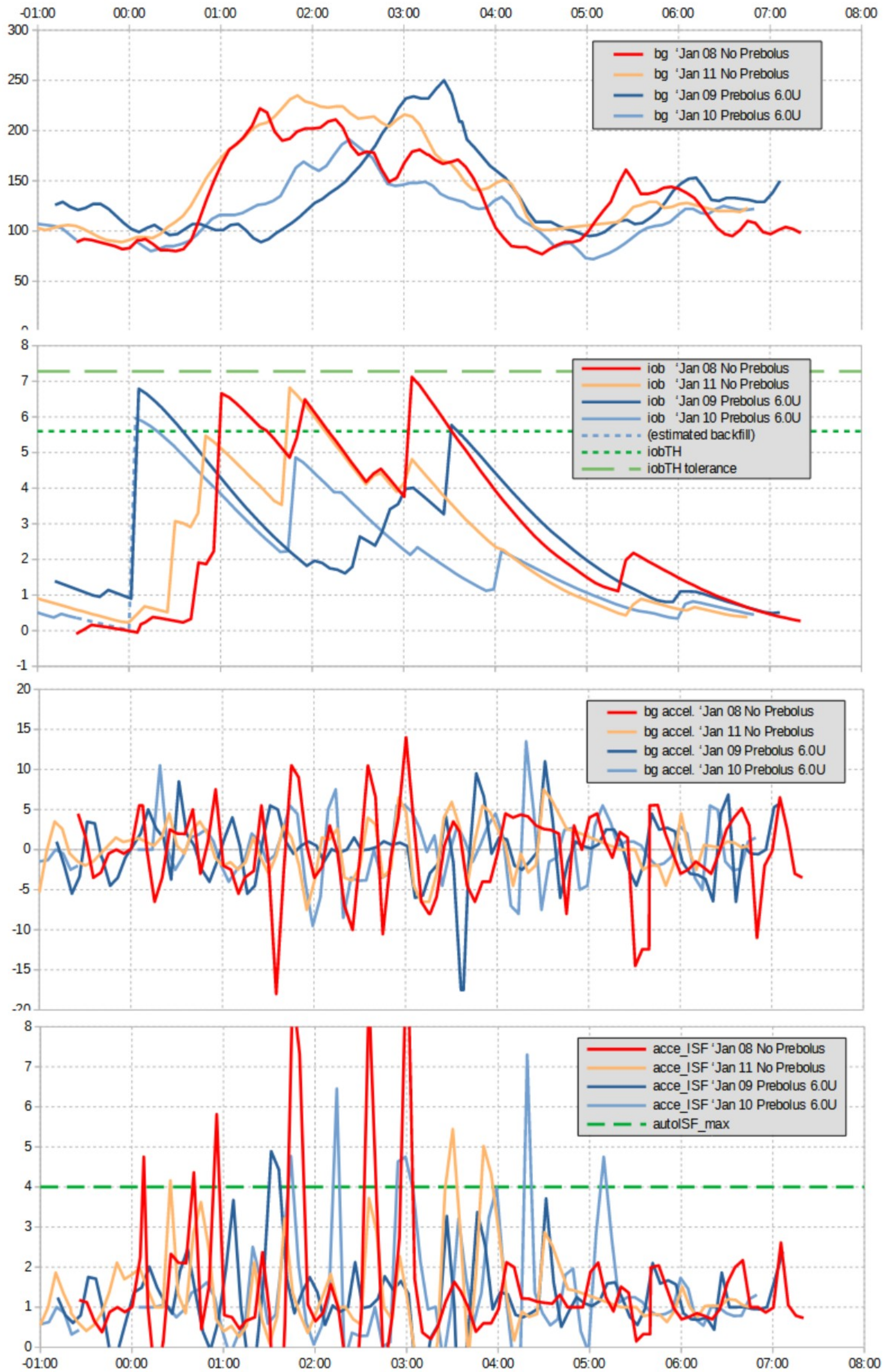
Please do not just assume the same is true in your case and for any other types of meals. More examples need to be analysed to come up with more general recommendations. Of special interest would be giving a prebolus that is significantly less than iobTH. It might just be enough to tame the glucose curve but not enough to attack the rise.

If you have such scenario or want to find out whether the same answers apply in your case rerun the analysis I went through. In attachment 2 you find hints what data is required and how to get it.

Status: 5.Jun.2023 @ 21:26

Attachment 1:

All results again with aligned time axes for better direct comparison



Attachment 2: Hints on how to execute the analysis yourself

The meals I enter in AAPS as notes so I have a record.

Most other information is provided by running the emulator with a time window of about 1 hour before the meal until 7 hours after. Import the CSV-file into a spreadsheet workbook as it contains the basic information. For obtaining the total SMB and total basal insulin I reran the emulator with a narrower window, namely 1 minute before meal start until 5:01 after the meal.

For the relative time axis I copied the UTC time column and in the copied column subtracted the meal start time as an offset for the graphs. I kept the original time column and because together the two columns served as a translation between real world UTC time and the relative meal time.

The glucose acceleration is not available directly but can be appended as a new column in the CSV-file with the value being simply the difference between “next-Δ” and “last-Δ” in the parabola fit columns. Consulting the glucose status in the SMB tab of the phone may help with verifying the maths.

IOB is available in the CSV-file. The iobTH values are more tricky. Easy for me because I extract them form the logfiles and store them in my own database anyway. I assume you use automations and in that case you can check the logfiles for which automation was triggered at the the time and enter those yourself into the spreadsheet. You may want to use my “find_string_batch.py” script from my github repo “...ga-zelle\Scan-APS-logfiles” to extract the relevant lines from the logfiles automatically like in my example here:

```
find_string_batch.py  Z:\foldername\AndroidAPS._2023-01-08_*.zip  dummy  automation
```

for the whole day.

The last bit of information not contained in the CSV-file already is the autoISF_max. Easy if it was constant during the relevant time window and you remember it. If not you can scan the logfiles and search for the lines containing the Profile definition with all its enable/disable, min/max and other settings as listed in the SMB tab. Alternatively, if autoISF_max actively limited acce_ISF then you can see it in the acce_ISF graph. Plot the final ISF factor as well in the acce_ISF graph. Then you can recognise where some peaks are capped at autoISF_max.

Finally you can look into the emulators TXT-file to find specific messages like why SMB was off. The case before with “additional carbs required” is such an example.

If you did not save the logfiles you can extract most of the information from Nightscout but it is significantly more and tedious work.