4. Meals: Setting ISF_weights in /Preferences

v.5.2.1

Please note that with autoISF you are in an early-dev. environment, where the user interface is **not optimized for safety** of users who stray away from intended ways to use. Good safety features exist, but these are only as good as the development-oriented user understands and implements them. This is not a medical product, refer to disclaimer in section 0

Available related case studies:

Case study 4.1: Pizza

Case study 4.2: Low carb meal

Case study 4.3: Hands-off FCL on Xmas

4.1 Getting started

Caution: This entire e-book is about Full Closed Looping (FCL).

In case you intend to work with giving boli, many suggestions made - notably in this section 4, and in section 2 – should **not** be followed. Y
You should then primarily use the autoISF Quick Guide (from https://github.com/ga-zelle/autoISF), and **do extra research**, on your own data. (Look at the chart in section 4.1.2. - your bolus very much would change things there!).
If you shy away, for now, from FCL, please have a look into sections discussing methods with "**Meal Announcement**", section 07, and section 13.3-

4.1.1 Reminder of pre-requisites

This section 4. is about the core FCL aspects of autoISF. Before doing anything with this section, please make sure you have studied the preceding sections 1and 2 on the general pre-requisites for FCL and the developers "Quick guide" (see section 3) on the principal workings of autoISF. Core points are briefly summarized below.

Start with proper "safety" settings

Before you start tuning your autoISF for FCL, **make sure you have** appropriately:

- **widened the SMB size restrictions** (section 2.1),
- **elevated** the max allowed ISF amplification via your set **autoISFmax** (section 2.2)

Both of these points are extremely important: If you set (or keep in place) narrow restrictions, this will **not** allow to see effects from a more aggressively tuned ISF. Even worse, it would cover-up too aggressive settings (e.g. on the …_ISF_weights that we get to in a moment), and invariably make your loop bounce against the restriction(s).
This could even work fine, if your meal spectrum isn't broad: If, in your HCL, the **same** bolus size pretty much fitted all your meals, it could now, in FCL, be replaced by rushing, with super-aggressively modulated ISFs, into the set restrictions, to produce - with only a brief delay – the required iob that would be about equivalent to what you formerly had bolussed in your HCL.

A system that is really fit **for** the **variance** we all like to enjoy in our daily lives, though, would be characterized by "tolerating" pretty wide open safety restrictions*), while having cautiously calibrated other, notably ISF modulating, parameters (as described in sections 4.2 – 4.5).

*) Still, for safety (as also suggested in section 2.1 and 2.2), start your tuning on a middle ground, and only gradually widen SMB size and autoISFmax during your initial tuning.

Also make sure you have

- **set** your **iobTH%** (refer to section 2.4 and if available 4.8)

Furthermore, in your early test phase, it is recommended to:

- Run the system as dummy, not connected to your body (or, on own risk, connect only as long as you watch closely)
- In AAPS preferences, switch your autoISF FCL (= **autoISF/"Enable adaptation of ISF to glucose behavior"**) ON only during daytime hours of a meal, *e.g. 11-18h*, for fully automatic "full closed loop" management *of lunches*.

  You can do this switching manually *at 11 h and 18 h every day*, *or* set up an Automation that does that (see section 3.4).

Take **typical but not extreme** meals. Omit sweet drinks, or drink only slowly. You are going for a "good enough" compromise, that works with your range of usual meals.

It is wasted time to do a lot of iterations to "optimize" settings based on just 1 type of meal. See case study 8.2

Occasionally, watch the time-pattern of bg, iob (SMBs given), and insulin activity after meal start. Aside from serious "mathematical" attempts to tune settings based on data from the SMB tab (or the Emulator, section 10), just watching the curves develop on your AAPS main screen can, over time, give you "a feel" what settings, and eating behaviors, are benign or detrimental to good %TIR performance.

Importance of proper profile ISFs.

Starters on autoISF FCL who are coming from using HCL with **dynamic**ISF must be aware of the following: It is absolutely essential to build your FCL on a properly set **profile** ISFs (likely a circadian pattern over 24 hrs).

> It may not apply to you, but many dynamicISF users did never bother to determine their ISFs that would maximize their HCL performance, but employ dynamicISF so to speak for going „dynamically" through a wide range of possible ISFs, until eventually hitting a sweet spot, and the whole thing works better than before, with what they had *used as a* profile ISF (often only one, e.g. coming from Autotune).

The following is important to understand, as it leads straight into the core idea behind FCL with autoISF, too: It is a good idea to establish a well-running hybrid closed loop with set (non-dynamic) **ISF** (set in **profile** for each hour of the day). That ISF must be **aggressive enough** that it gets you down from a high around 200 mg/dl to target. That is roughly also the way you experimentally determined it (so I hope. See https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings/blob/HCL-.-settings-main-repo-(pdf)/ISF%20determination_V.3.33.pdf).

- Using *that strong* value also *at lower bg,* (on the way "up" , after meal start), is very positive: We do *not* want to have a *softer* acting loop when at *lower* bg (which is what dynamicISF tends to do!). autoISF will, in contrast, temporarily sharpen your ISF when, at low bg, acceleration is detected..
- On the way down from peak, towards glucose target, a somewhat too strong ISF should not hurt because much of the time your loop (well supplied with insulin before, „on the way up") is zero temping , or at least has only a small gap to correct, from predicted bg to target bg.
- You have no business to be much above 200 mg/dl where an *even stronger ISF* may or may not help. It sure does not help at an occlusion which is about the only reason to see super high values as an experienced looper.

Pegging ISF strength to bg level therefore **does not make sense in FCL**. You will use the autoISF toolbox to get strongest ISF **at low** but beginning-to-rise bg,

> Note: There are very much refined versions of dynamicISF that can have beneficial applications, notably in HCL …And, yes, I know, bg levels can also correlate with insulin sensitivity. But let us not get into "chicken or egg type" discussions.
> Rather, focus on doing a good tuning job, and use superior approaches to account for sensitivity changes in a more pro-active manner (before running into sky-high bg (or into hypos)):

Going to autoISF FCL, you absolutely must **anchor on the proper profile_ISF.**

The profile is not "set in stone", though. To use above terminology again: **Pegging ISF strength to** your current **insulin sensitivity** – very much like you had done all along in HCL - **does make sense in your FCL...**

> (…and the fact that autoISF afterwards "anyways" often strongly modifies ISF is not a reasonable counter-argument).

There are fully automated, as well as manual ways for sensitivity adaptations of the profile ISF:

Profile ISFs can get **fully automatically adapted**, e.g. by Autosens, or by the Activity Monitor, which in autoISF we rather use (section 6.5).

Which of your basic related settings (in AAPS/Preferences) produce exactly which adaptation can be seen right in the top lines of your SMB tab, at each loop decision. Likewise, it can be retrieved later in logfile analysis (see Emulator, section 10)

Furthermore, when using autoISF you can – as you did in the past, e.g. around exercise, or in times of illness – temporarily **manually modify** your profile ISFs

Also these effects are quantified in SMB tab and logfiles *).

*) Furthermore, the results from autoISF are explained in the SMB tab, and multiplied with (original or adjusted) profile ISF to result in the ISF (called "sens") used in the current insulinRequired calculation

All three top buttons in AAPS (%profile switch, exercise and TT) can be freely used to adapt to changes in sensitivity/resistance, turning into a yellow color to alert you to this. (More about your "FCL cockpit" see section 5.2.2.).

For a start, please spend a couple of days (if not weeks) to **get your key autoISF related settings right**, strictly **on/for days with your normal insulin sensitivity**. This is what this section 4 is about.

Importance of starting from a well-performing Hybrid Closed Loop

A **satisfying performance in Hybrid Closed Loop** mode is a pre-requisite. Expect to reproduce about the same %TIR also in your FCL, but with less daily interaction, once established.
Note that this refers to prior use of „vanilla" software, without fancy „dynamic add-ons" (such as: Autotune determined factors, dynamicISF etc). that may have introduced bias into the profile settings you bring with you into FCL now.

To reach a satisfying performance you must start from a hybrid closed loop in which you did **master your meal management well** using the oref(1) algo SMB+UAM**.**
This is a pre-requisite **to be able to forget it** … - because the initial tuning that we now turn to demands, that you analyze your **prior best practice as your blueprint** to find appropriate settings and „teach" your FCL to come up with the necessary iob.

This is the main subject of this section 4 (finding settings for automatic meal management).

Do not copy settings from other FCL loopers

When setting your parameters, don't use any given numerical example (not even as "a starting point"). Instead, anchor on **data from your *successful* Hybrid Closed Loop!**

> Most *examples given in this paper* are from an adult diabetic (Lyumjev, G6) whose insulin sensitivity can be characterized as follows: approximately 37 U TDD, thereof 13 U profile basal, at about 200g daily carbs from mainly lunch and dinner; no couch snacks or sweet drinks. The user also participates in multiple instances of daily moderate exercise such as dog walking, biking and gardening. In Hybrid Closed Loop, a typical meal bolus was 8 U that was sometimes reduced such as when activity followed the meal.
> After seeing some more inputs from a variety of users we might put together a profile helper for some rough orientation, and for plausibility cross-checking, in section 4.8

Importance of going step-by-step

Section 5 will explore avenues to manage "disturbances", i.e. time blocks or situations that might demand enhanced or reduced loop aggressiveness.
Section 6 will focus on the exercise mode, and the activity monitor.

> In case you have a strong interest in the Activity monitor (section 6.6), you can start with calibrating that, and have it run already in the weeks when you go through sections 4 and 5.
> In case you use an EatingSoonTT at meal start (the author recommends to try without), note that any active TT shuts activity monitor automatically off while that TT is active.

Resist the temptation to make use of the tools presented in sections 5 and 6 too early.
On your **first** setting-up and tuning attempt, **it is strongly recommended that you not "play around" with all ultimately available features, but stick to the sequence of steps** to take.

Yes, "playing around" with the many extra buttons often will help find an improvement. But you likely create an instable FCL that, already at fairly standard situations, uses up some of your FCL's principal capacity to correct for disturbances. This limits what will be left to manage extreme situations.

Caution: **Once you created a maze of little errors and counter-strategies/counter-errors**, **it will be nearly impossible to find your way** out of this mess, **towards better settings**, at any later point of time.

Note that it is principally not easy to conclude on suitability of tuning:

- AutoISF comes with very many (currently 18) extra parameters, and even when employing the emulator (sections 10 and 11), it is quite hard to analyze their interaction.

One principal reason why things are difficult to analyze is, that you really can only analyze one decision, and that will put you on another bg curve. So, you can never see the full effect, along more than half an hour or so, that *any* change would really result in.

- Understandably, many loopers rather “move forward” to an over-patch for identified problems, and not bother with a more “puristic” step-by-step approach to do things right from the ground up.
- Aware of above sketched conundrum, the AAPS autoISF developers offer the ultimate tool to investigate “what-if”, regarding a setting change you may contemplate: A nice lady voice on your smartphone can tell you, at each loop decision, where your contemplated change would make a difference (in SMB size). This offers an opportunity to watch closely, with or without implementing that change. (It is always your spontaneous choice, whether you want to “follow the lady’s suggestion and manually add to the SMB, as suggested). More see in Section 11.4

But, we are getting ahead too far here. You first must find a starting point for key settings, which works reasonably for not too-challenging meals in your personal spectrum.

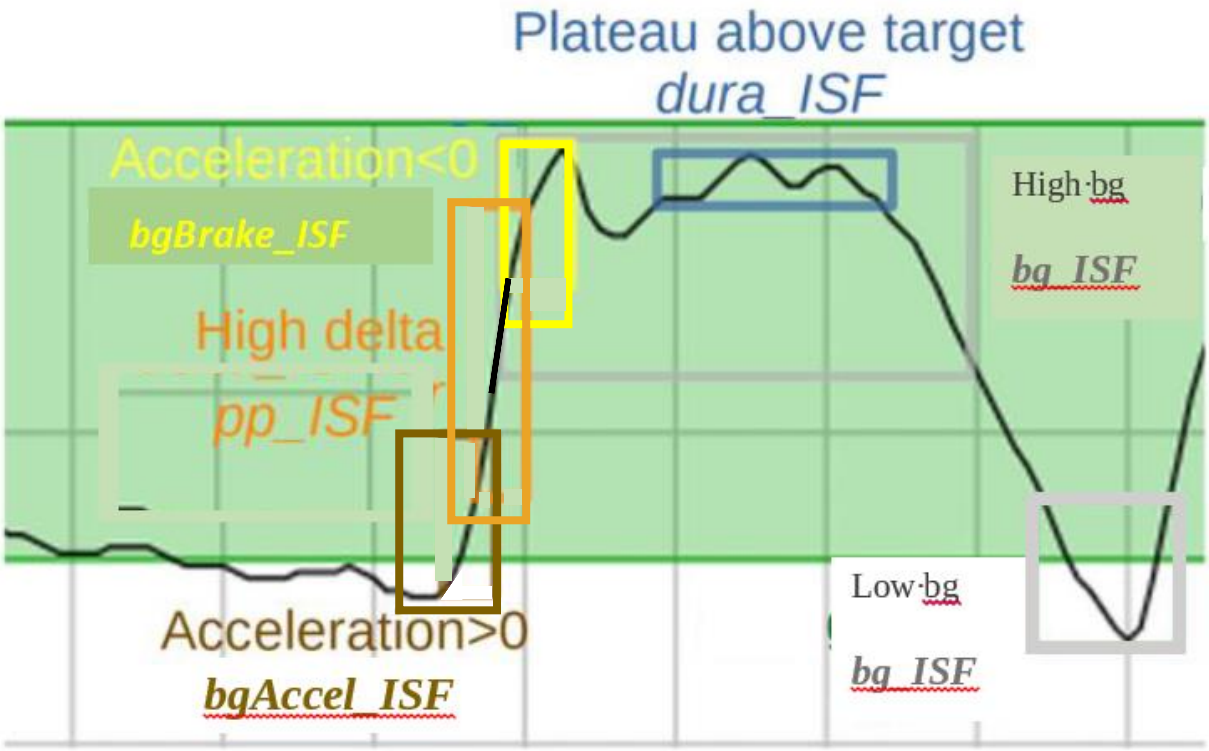
Before getting into this, let’s first have a look on how autoISF basically works. (More see in Quick guide by the developer, referenced in section 3.2; or directly at https://github.com/ga-zelle/autoISF).

4.1.2 autoISF factors overview in typical glucose chart

The core challenge of your UAM Full Closed Loop is to recognize a meal start from the glucose trend, and ramping up iob.

When setting up your autoISF Full Closed Loop, **you must set several ISF_weight parameters in AAPS Preferences**/OpenAPS SMB/autoISF settings.

They relate to different stages of the typical glucose curve after starting a meal:



Note: ***bg_ISF*** is not used much in FCL, as it is rather late to act on high (or low) bg level that developed. But, feel free to experiment, e.g. in case you have indications, in your data, that in the past dynamicISF was useful to manage bg extremes in some situations.

The core advantage of using autoISF with oref(1) SMB+UAM (in FCL as well as in hybrid closed loop) is that it manages the glucose curve it sees developing, **no matter what the underlying reason** is.

42 potential factors were identified (see: https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings/blob/FCL-w/autoISF/42%20factors%20influence%20bg.pdf), so, no wonder, that loopers who meticulously input their carbs will often *not* see the expected result.

In case you like to learn more about the theoretical background for the autoISF development, here some core aspects the key developer shared in a DIY forum:

Theory behind autoISF *gazelle, Aug. 2024*

I have recently managed to work through the large pile of older (German) diabetes journals. I've now reached issue 5/2024 and was amazed by the article "How insulin is produced and works in the body".
According to the article, there are essentially 2 phases of release:

- *Phase (1) lasts only a few minutes and releases a large dose from the beta cell's supply when glucose rises.*
- *(2) After that, insulin is newly produced and released evenly over a period of hours.*

These two phases remind me very much of initial acce_ISF and later dura_ISF in autoISF.
This may explain why it works so well for many users. I had no idea about this process in the body when I developed both effects.

1. *acce_ISF was based on* ***Newton's 2nd law of mechanics****, i.e. inertial mass * acceleration = force.*
   For me, this meant that when the FC is accelerated, a "sweet force" is acting.
   To counteract this, the acceleration must be counteracted.
   Strengthening the ISF, i.e. increasing resistance, corresponds to an increase in inertial mass, so to speak.

2. *dura_ISF was an analogy to the* ***PID controller*** *in technology:*
   P stands for proportional control, in this case the deviation of the FC from the target value;
   I stands for integral of the deviation, in this case the sum of the deviation from the target;
   D stands for differential control, in this case the delta of the BG.

The components for P and D are contained in the regular AAPS, but the I component was missing. So I added it, as a rough approximation, using dura_ISF.

*I find it very interesting that both solutions came about from analogies to technology/mechanics without direct knowledge of **hormone metabolism**. I have always been interested in using analogies to transfer solutions to other fields.*
Translated with DeepL.com (free version)

Remark: This just broadbrush sketches the general idea. Lots of dev work had unfortunately to go into details, like how to deal with everyday imperfections of our CGM systems, and still come up with meaningful mathematical (parabola fit) earliest-possible detection of "real" bg rises.

4.1.3 Getting ready to set your autoISF_weights

Before you progress, make sure you studied the flowcharts in section 3 that describe how autoISF calculates the **effective**(ly used) **ISF**.

> Warning: **Any bolus you „sneak in" will severely distort the glucose curve. That could** render your tuning of weights (see below) useless, and could **make your loop act in unpredictable ways (**potentially also dangerous, however, your set iobTH (section 2.4) should help here, too).
>
> In case you feel tempted to use boli, be ready for some own extra research, and refer to section 7.

After doing the prep work as outlined in section 2 **you now get to calibrate your FCL to your normal meal spectrum** by initially **setting and tuning the various _ISF_weights**, that dynamically change with bg curve characteristics as sketched in the chart on the previous page.

Please stay away from extremes (regarding both, meals and exercise) **when you go through this** section 4. It is about getting a first *roughly right* set of settings, as a basis.

Researching your standard meal patterns, and finding settings for the various -ISF_weights is the core job in setting up your autoISF FCL.

> Depending how varied your diet and general lifestyle are (and your expectation of %TIR you like to reach), this could be the main job at hand. However, there is much more you *could* do *later*, and that will be outlined in later sections 5 and 6.

Consult sometimes your SMB tab, to see how the applied effective ISF (named **sens** there) is calculated. (Example given in section 5.4.5).

4.2 Meal detection and managing the initial bg rise: bgAccel_ISF

4.2.1 Mimicking a HCL bolus in FCL using bgAccel_ISF

When looping without carb inputs and without giving a bolus ourselves, the first crucial setting is to set the **bgAccel_ISF_weight** so that SMBs are requested immediately when the loop detects an acceleration in your blood glucose (bg) that is starting to rise.

Ideally **within about 20 minutes after acceleration detection, which would be the first up to 4 SMBs, as much iob should automatically be supplied as we would have given with our bolus in hybrid closed loop.**

As the biggest principal challenge for the FCL is big **high/fast carb** meals (from within your personal "spectrum"), we start with a focus to get sufficiently big SMBs going for those.

> Note, though, that in a **low carb** meal scenario, the first 4 SMBs would have to automatically result much smaller (which, after careful tuning, is possible with the same parameter settings, see e.g. case studies 4.2 vs 4.3).

Rule of thumb: Two of the first three SMBs each (in this test based on a big meal) should be about ¼ to 1/3 the size of a bolus in your HCL „career" (for a similar meal).

Going over 1/3 could be problematic

- if your diet contains occasional **low carb** (or brief **snack**ing), it is not helpful if your settings make your loop invariably "bounce" over your iobTH (and then you would need extra snacks to balance the auto-generated iob, to prevent hypos),
- also if your **CGM quality** is sometimes unreliable, and might produce an artefact that could be mistaken for a meal start.

Be vigilant about this topic! And please do not choose the supposedly easy way, to just set safety restrictions (allowed max SMB size, or autoISFmax) so low, that your loop never can exceed 1/3. Try to really tune the _ISF_weights appropriately. (Only that way, your loop can "accommodate" the entire meal spectrum, and also states of adapted general insulin sensitivity).

4.2.2 Widened safety restrictions

Already when tuning the bgAccel_ISF_weight it can become evident that safety restrictions (as discussed in section 2) must be **widened** further:

- Especially if your *profile basal* rate is very small, the **smb_max_range_extention** and/or the **autoISF_max** "must" often be increased further.
- Pay attention also to the **iobTH**% and, potentially, iobMAX
- Note that the smb_delivery_ratio "only" portions the insulinReq differently over the next 15 minutes (see also section 2.3), and therefore is **not** a prime tuning parameter.

In the end you should **not set these safety limits too tight,** so "nudging" aggressiveness by another 10 or 20% from your cockpit, later, will not bounce into restrictions.

On the other hand, setting **narrower** restrictions for max allowed SMB size can also become necessary:

- Poorer CGM quality demands narrower restrictions for safety reasons.
- If you use a 1-minute CGM, please observe section 1.4.2

4.2.3 Start value for your bgAccel_ISF tuning

bgAccel_ISF_weight is set default to zero in AAPS Preferences/SMB/autoISF.

To start, I would try 0.05 or **max 0.1**, and keep trying in max 0.05 steps. Soon move to 0.02 steps. From my (very limited) overview, many use around 0.2, and possibly even higher if their hourly basal rate is 0.1U or lower. (Consult section 4.8 when available). Do not be tempted to rush this setting by using large jumps in adjustments.

To monitor what is happening, and start tuning, in search of appropriate settings, you must keep (real-time) track of how autoISF uses your set bgAccel_ISF_weight:

- To do this in the **SMB tab** is possible but not very practical. You would end up making a lot of screenshots (quickly in the crucial minutes after a SMB *was* given, or when *you thought it should be* given), for later analysis.
- The superior method is to just copy **logfiles** from your phone/internal memory/AAPS/logs …
  - all zip files there
  - look up how many days of data are covered there on a rolling basis, and copy out onto your PC (see section 10.1.1) before the older ones get forever lost

  … and analyze them at your convenience later, using the **emulator** (see section 10; used e.g. in last pages of case study 4.2).
- Some emulator-based analysis is also possible within AAPS on your phone (section-11).

In any case, it is worth the effort to tune the **bgAccel_ISF_weight** in such a way that high glucose increases are already nipped in the bud, so to speak.

To summarize: In FCL, the first 3 or 4 SMBs should not be much delayed, and amount to similar iob like your "former boli in HCL".
Depending on details about the carb absorption characteristics of your meal, and the performance of your CGM, also **pp_ISF** (see 4.3) might be a fairly early contributor to getting iob up.

4.2.4 How changing the _weights influences the resulting calculated insulinRequired

(You can <u>skip reading this section</u>, unless you want to know quantitatively how things work).

The developers' documentation (Quick Guide) https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf gives the following equation:

acceISF is the factor by which autoISF wants to sharpen the profile ISF in a certain situation of bg acceleration:

acce_ISF is calculated by

acce_ISF = 1 + acce_weight * fit_share * cap_weight * acceleration (eq.1)

where	fit_share	a measure of fit quality, i.e. 0% if unacceptable up to 100% if perfect;
	cap_weight	is 0.5 below target and 1.0 otherwise;
	acce_weight	is bgAccel_ISF_weight for acceleration away from target, i.e. mostly positive or bgBrake_ISF_weight for acceleration towards target, i.e. mostly negative

fit_share should be close to 1.0 (if good CGM quality.

cap_weight is 1.0 for bg>target

Having a EatingSoonTT at first acceleration *can help avoid* the factor iF getting cut in half!

profile_ISF / acce_ISF = effectively used ISF (sens)

(…if the acce influence dominates and is used as effective ISF. Else, see flowcharts in Quick Guide)

When looking at the same acceleration moment, we can combine all three last factors of eq.1 into a term "iF".
For estimating the effect from using another ISF_weight, we then have two (eq.1) , with two unknowns, iF ("cancelling out"), and the sought acce_ISF (for new _weight).

Example: Your profile_ISF is 40 mg/dl/U. Using bgAccel_ISF_weight of 0.2
you saw effectively used ISF of 30.3 mg/dl/U (box C13 in table below) = 40 / 1.32
=> factor for acce_ISF = 1.32.
For an intended correction by – 10 mg/dl the insulinRequired would calculate to 10 / 30. 3 = 0.330 U.

	A	B	C	D	E	F	G	H
1	Profil ISF (mg/dl/U)	40		**bgAccel_ISF tuning -**				
2	InsReq (U) @ iF=0	1		see **FCL e-Book** section **4.2.4**				
3	bgAccel_ISF_weight:	0,2			0,1			InsReq effect
4	iF	acce_ISF_old	ISF_old	InsReq_old	acce_ISF_new	ISF_new	InsReq_new	
5	0	1	40	1	1	40	1	0%
6	0,2	1,04	38,5	1,04	1,02	39,2	1,02	-2%
7	0,4	1,08	37,0	1,08	1,04	38,5	1,04	-4%
8	0,6	1,12	35,7	1,12	1,06	37,7	1,06	-5%
9	0,8	1,16	34,5	1,16	1,08	37,0	1,08	-7%
10	1	1,2	33,3	1,2	1,1	36,4	1,1	-8%
11	1,2	1,24	32,3	1,24	1,12	35,7	1,12	-10%
12	1,4	1,28	31,3	1,28	1,14	35,1	1,14	-11%
13	1,6	1,32	30,3	1,32	1,16	34,5	1,16	-12%
14	1,8	1,36	29,4	1,36	1,18	33,9	1,18	-13%
15	2	1,4	28,6	1,4	1,2	33,3	1,2	-14%
16	2,2	1,44	27,8	1,44	1,22	32,8	1,22	-15%
17	2,4	1,48	27,0	1,48	1,24	32,3	1,24	-16%
18	2,6	1,52	26,3	1,52	1,26	31,7	1,26	-17%
19	2,8	1,56	25,6	1,56	1,28	31,3	1,28	-18%
20	3	1,6	25,0	1,6	1,3	30,8	1,3	-19%
21	3,2	1,64	24,4	1,64	1,32	30,3	1,32	-20%
22	3,4	1,68	23,8	1,68	1,34	29,9	1,34	-20%
23	3,6	1,72	23,3	1,72	1,36	29,4	1,36	-21%
24	3,8	1,76	22,7	1,76	1,38	29,0	1,38	-22%
25	4	1,8	22,2	1,8	1,4	28,6	1,4	-22%
26	4,2	1,84	21,7	1,84	1,42	28,2	1,42	-23%
27	4,4	1,88	21,3	1,88	1,44	27,8	1,44	-23%
28	4,6	1,92	20,8	1,92	1,46	27,4	1,46	-24%
29	4,8	1,96	20,4	1,96	1,48	27,0	1,48	-24%
30	5	2	20	2	1,5	26,7	1,5	-25%
31	formula (line 30)	=1+$A30*B$3	=$B$1/B30	=+$B$2*$B$1/C30	=1+$A30*E$3	=$B$1/E30	=+$B$2*$B$1/F30	"=G30/D30-1
32	Note: To determine not the incremental effect between two weights (E3 vs.B3), but the acce_ISF effect							
33	for the investigated _weight (E3): Enter 0 in (B3)							
34	Column (H) then shows the full effect of acce_ISF with the selected weight (E3)							

Now, going with a 50% reduced bgAccel_ISF_weight of 0.10:

acce_ISF = 1+ bgAccel_ISF_weight * internalFactor

before 1,32 = 1 + 0.20 * iF => 0.32 = 0.20 * iF => iF = 1.60 (see box A13)

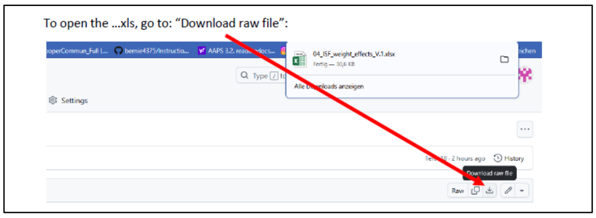
after ? = 1 + 0.10 * iF => ? = 1 + 0.10 * 1.6 = 1.16

New effective ISF would be 40 / 1.16 = 34.48 mg/dl/U (box F13).

For an intended correction by – 10 mg/dl the **insulin**Required would now calculate to 10 / 34.48 = 0.290 U, which is 12 % less (H13).

At higher acceleration points of the bg curve (column A, line 14 and higher), the effects would be more pronounced (column H, line 14 and higher).

An **Excel tool** is provided in e-book section 04 for this calculation. However, easier to handle might be the:



Simple "trial and error" approach

To get a feel for how changing the _weights influences the resulting calculated insulinRequired, it is best to start cautiously and **just do** 10 to max 20% **steps up, and watch out for the effects**. Doing similar step sizes should yield about similar effects each time.

Never forget to look into how other .._ISFs play into the effective ISF (named **sens** in the SMB tab), which overall results.

4.2.5 Characteristics of a well tuned-in bgAccel_ISF_weight

Your starting point was to set the bgAccel_ISF_weight so FCL works in a rather high carb meal.

Now you must check (and potentially fine tune) so it **will not "shoot iob too high"** with the first 3 or 4 SMBs **in other meals from your spectrum:**

- For meals that are in the **lower** (!) range of the "fast **carb load**" of your cluster, the necessary insulin supply for the first two hours or so might pretty much be provided already with the first 3 or 4 SMBs

  The glucose curve, at such meals, begins to flatten early in this SMB phase, so a de-celeration **(braking)** follows very soon (-> section 4.4). Clearly, the first 3 SMBs, in such cases, must remain below iobTH.

- **Low carb** meals are principally easiest for the FCL. However, you must secure that your bgAccel_ISF driven **first SMBs** remain small. This is principally possible also with a fairly aggressive bgAccel_ISF_weight set, because both acceleration and initial deltas are small when eating low carb. (Regarding the detected acceleration, the stakes may be high for the CGM and smoothing method you chose).

- A stage where moderate amounts of carb absorption and of insulin usage/need hold a balance could protract – at moderate bg elevation -over hours. The **dura_ISF** might play a bigger role, then, as e.g. in the low carb example in case study 4.2.

In case you run into limitations, see next sub-chapter.

4.2.6 Suitability for many types of meal

For a **hands-off FCL**, your settings have to fit

- ***in each*** of your **meal time**s

  What helps here is that, *between* your daily mealtime slots, your **circadian profile ISFs** (upon which the autoISF modulations build) automatically make a differentiation (as was the case in your HCL).

- for the whole **range of <u>your</u> meals.** All this is principally possible, but: …

What if you still have meals that you cannot make fit?

In extreme cases you will have to balance too high running iob with additional carbs (a late additional snack against going too low), and in the opposite case, you will have to reckon with temporarily exceeding the glucose target range, and losing some %TIR for this day.

If your meals vary very strongly, there are **avenues to ease your initial tuning job, or to optimize overall resulting loop performance:**

- Automations allow you to differentiate. For instance it is possible to apply different iobTH_percent and/or different bgAccel_ISF_weights for meals in different **time windows** or geo locations (details see sections 3.4 and 5.1)

  In case you use autoISF on the Trio or iAPS platform for i-phones, you may need to use a third party automation software, or "middleware" (! call for a case study 4.X)

- you can pre-program **custom buttons for special** meal (or snack) **types**, with different underlying FCL settings (see "cockpit", section 5.2.2.3)

- You can **modulate FCL aggressiveness manually** making use of the top 3 buttons in the AAPS home screen: These turn yellow during temporary switched %profile or glucose target (section 5.2.2.2)

Experimenting with the three above mentioned "avenues", the author found:

- the last point easiest to occasionally use, and the first one hardest.

- it worth investing some effort (also using the emulator a couple of times) to iterate through the typical meal spectrum a couple of times, for finding a "good enough" set of .._ISF_weights and other settings (like autoISFmax, iobTH% etc), *and* ***not do much extra differentiation****.* (More see in section 5).

4.2.7 Summary on tuning for the initial SMBs via bgAccel_ISF

Early strong iob also will **ease the tuning** task **for the subsequent phases** of the meal, because there is, then, largely zero-temping (as well known from HCL-times after your administered bolus). Also, the lower and shorter lasting the glucose peak, the lesser the hypo danger from the activity tail of SMBs given *when* glucose was „stuck" high.
However, it is important **not too super-aggressively** tune bgAccel_ISF_weight up, so, regardless of the type of meal, very big SMBs invariably would result.
Rather, the rough idea should be:

- SMBs driven by bgAccel_ISF: initial iob for **all meals**. SMB sizes vary, because accelerations and deltas vary.

  So, at high carb meals it depends on your settings, and on the evolving bg curve, whether the first few bgAccel_ISF driven SMBs get you already up to iobTH in high carb meals, or whether this happens in the *overlapping* next stage.

So, looking a bit ahead to the next chapters:

- SMBs driven by pp_ISF: to the extent there is strong (near-linear) bg rise (at **big meals rich in carbs**) with big or small deltas, iob is now driven towards (and potentially over) iobTH.

  In low carb meals this period can be extremely short, with iob remaining under iobTH (example see case study 4.2)

- SMBs driven by bgBrake_ISF, bg_ISF, or dura_ISF:

    Note that *all of these* can overlap with the pp_ISF stage. Consult the csv table output from the Emulator (example given at end of case study 4.2) as to which of the _ISF categories drives the effectively used ISF (and what change of the …_ISF_weights would change this. Consult decision flowcharts for effective_ISF in pages 1-6 of the Quick Guide.pdf in https://github.com/ga-zelle/autoISF).

  Depending on the shape of the bg curve after the initial strong rise, and depending on insulinReq. and on iob (> iobTH?), autoISF can provide more SMBs to bring bg to target. This case applies to **low carb** meals. The dura_ISF is also useful to manage temporary insulin resistance often observed late in **fatty** meals.

It is worth investing effort (following the sequence of steps in sections 01-04 of this FCL e-book) **in your initial project to establish a good set of ISF_weights** for your meal spectrum. This will keep interventions in daily life to a minimum.
Unless your lifestyle, or health and body weight change radically, this should be **a *one-time* effort** (in your initial weeks establishing your FCL), with *no need* to fine-tune much later (see section 8).

4.2.8 Note regarding acceleration "happening again" in late part of dropping glucose

(Skip, unless interested)

After the peak, in the late stage of *falling* bg, the glucose curve is like an accelerating parabola again. The algorithm tries to evaluate when and at which bg level complete digestion of the meal and a bg minimum will result. Insulin required to stabilize around target bg is usually very small, and the adaptation of ISF in that stage relatively unimportant. See in your SMB tab, how, at "already falling" bg, the ISF modulation is taken back.

> In version 2.2.8.2 there was a potential deficiency in situations where glucose was falling and the glucose acceleration was already positive. That meant a minimum glucose level can be extrapolated. If that happens to be less than target and expected in less than 15 minutes then there should be no strengthening of ISF as it would lower glucose even more. Therefore bgBrake_ISF_weight is used now instead of bgAccel_ISF_weight. But those situations were rare and less critical than might be expected at first sight. The reason is that in most cases the predictions ended up even below their threshold meaning SMB were disabled.

4.3 Managing strong bg rises: pp_ISF

4.3.1 Main function of pp_ISF in autoISF FCL

In the later phase of acceleration and in the earlier phase of deceleration there is a more or less linear increase of bg with **high deltas**, and corresponding extra insulin need.

- With **higher carb load** meals, or meals that come with a sweet drink, the increase will be particularly strong, and (if not already driven there by bgAccel_ISF) now reach, and with the last "allowed" SMB exceed, the valid iobTH.
- With **low carb** meals, there is only a very un-pronounced (short, with weak deltas) "pp_ISF phase". (Example see end of case study 4.2).

autoISF should now "fight" this with the help of the post-prandial ISF, set via **pp_ISF_weight,** after you have set your bgAccel_ISF_weight.

4.3.2 Tuning pp_ISF_weight

To tune-in your **pp_ISF_weight,** please do this with a really high carb meal (from within your typical meal spectrum) *after* you have set a halfway suitable (not too aggressive) bgAccel_ISF_weight.

> Note that if you rush into pp_ISF tuning while "still having a too aggressive bgAccel_ISF", the latter is covering up the requirement you now really want to calibrate for in pp_ISF!

So, at a meal in the upper spectrum of your carb load, carefully begin with a starting value for *pp_ISF_weight* of 0.005. Observe the reactions and check the SMB-tab before you increase it cautiously for the next days.

> Best practice is to analyze the emulator tables (discussed in section 10, and example given in the pizza case study 4.1)

How changing the _weight influences the resulting calculated insulinRequired

(You can continue 2 pages down at 4.3.3, unless you want to know quantitatively how things work).

The developers' documentation (Quick Guide, page 5) https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf gives the following equation:

$$\text{pp\_ISF} = 1 + \text{delta} * \text{pp\_ISF\_weight}. \qquad \text{(eq. 2)}$$

profile_ISF / pp_ISF = effectively used ISF (sens)

(….if the pp influence dominates, and is used as effective ISF. Else, see flowcharts in Quick Guide)

Note that the pp_ISF effect only comes in when there is positive short_avg_delta, AND bg is minimum 10 mg/dl **above bg_target** already.

- Before, we need bgAccel_ISF do its job!
- Evidently, by setting a lowTT around meal start, you can influence (not the size, but) *how soon pp_ISF will "compete with"* your bgAccel_ISF (… and this can play out very differently, at different kinds of meal, too.)

Tuning: When looking at the same moment (same delta), the effect from using *another* ISF_weight, can be mathematically be solved with two (eq.2):

Example: Your profile_ISF is 40 mg/dl/U. Using dura_ISF_weight of 0.02 you saw effectively used ISF of 30.3 mg/dl/U (box C13 in table below) = 40 / 1.32
=> factor for pp_ISF =1.32.
For an intended correction by – 10 mg/dl the insulinRequired would calculate to 10 / 30. 3 = 0.330 U.

	A	B	C	D	E	F	G	H
1	profile.ISF (mg/dl/U)	40		**pp_ISF tuning -**				
2	InsReq (U) @delta=0	1		see **FCL e-Book** section **4.3.2**				
3	pp_ISF_weight:	0,02			0,03			InsReq effect
4	delta (mg/dl)	pp_ISF_old	ISF_old	InsReq_old	pp_ISF_new	ISF_new	InsReq_new	(E3 vs.B3)
5	0	1	40	1	1	40	1	0%
6	2	1,04	38,5	1,04	1,06	37,7	1,06	2%
7	4	1,08	37,0	1,08	1,12	35,7	1,12	4%
8	6	1,12	35,7	1,12	1,18	33,9	1,18	5%
9	8	1,16	34,5	1,16	1,24	32,3	1,24	7%
10	10	1,2	33,3	1,2	1,3	30,8	1,3	8%
11	12	1,24	32,3	1,24	1,36	29,4	1,36	10%
12	14	1,28	31,3	1,28	1,42	28,2	1,42	11%
13	16	1,32	30,3	1,32	1,48	27,0	1,48	12%
14	18	1,36	29,4	1,36	1,54	26,0	1,54	13%
15	20	1,4	28,6	1,4	1,6	25,0	1,6	14%
16	22	1,44	27,8	1,44	1,66	24,1	1,66	15%
17	24	1,48	27,0	1,48	1,72	23,3	1,72	16%
18	26	1,52	26,3	1,52	1,78	22,5	1,78	17%
19	28	1,56	25,6	1,56	1,84	21,7	1,84	18%
20	30	1,6	25,0	1,6	1,9	21,1	1,9	19%
21	Note: To determine not the incremental effect between two weights (E3 vs.B3), but the pp_ISF effect							
22	for the investigated _weight (E3): Enter 0 in (B3)							
23	Column (H) then shows the full effect of ppISF with the selected weight (E3)							

Now, going with a 50% stronger pp_ISF_weight of 0.03 (table: E3):

pp_ISF = 1+ delta * pp_ISF_weight

before 1,32 = 1 + 0.02 * delta => 0.32 = 0.02 * delta => delta = 16 mg/dl

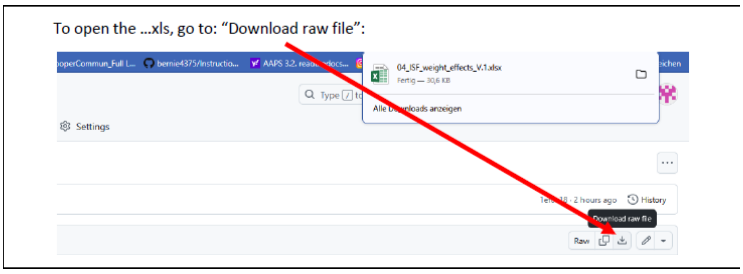
after ? = 1 + 0.03 * delta => ? = 1 + 0.03 * 16 = 1.48

New effective ISF would be 40 / 1.48 = 27.03 mg/dl/U (box F13 in table).

For an intended correction by – 10 mg/dl the **insulin**Required would now calculate to 10 / 27.03 = 0.370 U, which is 12.1 % more insulin (box H13 in table).

The % effect would be more or also less pronounced, depending how strong bg is rising (**delta**), which is of course the key idea behind pp_ISF.

An **Excel tool** is provided in e-book section 04 (Github/bernie4375) for this calculation. However, a simple “trial and error” approach as outlined initially in this 4.3.2 section (2 pages before) might be easier to handle.



To summarize, tuning-in the pp_ISF_**weight** allows the user to define her/his personal sweet spot for ISF aggressiveness in the phase of strongly rising bg.

- As different kinds of meals will have different **delta** patterns, it can well be that one pp_ISF_**weight** (as set by you in AAPS preferences) is good for all meals.

- Note that a circadian profileISF provides an avenue to still differentiate between e.g. breakfast and lunch response of your autoISF loop.

4.3.3 Loop states with very little insulin need (iob > iobTH, or 0 %TBR)

Normally (except for very low carb meals) the SMBs triggered by bgAccel_ISF_weight and pp_ISF_weight should be sufficient to reach and slightly exceed the **iobTH** (see section 2.4) so all *the other* autoISF parameters are relatively unimportant for now.

> A reason why this can work at all, also for quite a variety of meals, lies in the fact that there is an hourly carb absorption limit of about 30g/h
>
> > (Reference: Dana Lewis:https://github.com/danamlewis/artificialpancreasbook/blob/master/8.-tips-and-tricks-for-real-life-with-an-aps.md#heres-the-detailed-explanation-of-what-we-learned. (That limit can be lower, e.g. with gastroparesis or certain medications, but that would make things even easier)

So while meals might wildly vary in composition and size: What is digested, and needs insulin in the first ~90 minutes (when FCL tries to catch up with insulin need and differs strongly from HCL, with bgAccel_ISF and pp_ISF in the leading role), will be relatively close…for meals with similar *initial* glucose acceleration and rises, anyways…

The others, **low carb** with much slower initial acceleration and rise, are easy recognized as different by the loop, see section 4.4 that follows.

Depending on the type of meal and "aggressiveness" of your bgAccel_ISF_weight and pp_ISF_weight tuning, the iob will already be so high that, in the phase of decelerated glucose rise towards the peak (the "last part of the rise"), no more insulinRequired is seen by the loop.

Therefore the **bgBrake_ISF_weight** is often unimportant in meals with a relevant carb content.

> For potential relevance in low carb meals, see section 4.4.

4.3.4 “Quality control” on how well your tuning can replace your former HCL bolussing

Warning: **Occasionally consult the SMB tab to see how your settings really work.**
A setting (...ISF_weight) that is actually set too aggressive might be masked.
Tuning only works if the effects of the settings being tuned are **not** unintentionally **limited by other** (e.g.„safety“) **settings**.

Also, **always look at two or three *different* meals** before deciding whether a tuning "fits" („good enough“ for each of them). You probably will have to iterate back and forth doing this for two or three different kinds of meals …

- Case Study 4.1 (Pizza Meal) contains, towards the end, an example how you can go about tuning the _weights for various _ISF factors of autoISF.

- Case Study 8.2 shows that it is **not** worth it to seek “optimized” settings based on just one (more extreme) meal.

… until you find *one* good enough set of settings *for all* of them. Do not rush this, establishing a solid foundation will be well worth your time.

The following sections will deal with similar issues like you were facing in HCL after your given bolus lost much of its power, and SMBs were needed for the “eCarbs”.

4.4 Sluggish rise towards a bg peak: bgBrake_ISF

At a **low carb** meal, or an attempt at doing a **weight reduction diet**, (and probably also with gastroparesis, or if you take one of these novel GLP-1 drugs that slow meal absorption - Somebody, please supply a case study!)- the glucose goes up only sluggishly, and iobTH should not be reached at all.

In case you *exclusively* do very slow absorbing meals, you could of course also adjust your iobTH setting low enough to suit your *uniform* situation.

Acceleration, and the phase of strong glucose rise, are quickly over at slow-absorbing meals, and there can be:

- a decelerating bulge of insulin action that projects over an hour or longer. This is where the importance of the **bgBrake_ISF** can come in.

- a bg curve that hovers for an hour or longer around an elevated bg level, because additionally absorbed carbs, and consumption of the moderate SMBs delivered, tend to keep a balance for a while. **Dura_ISF** can deal with this (see next chapter). An example for this is given in Case study 4.2.

Note that in some data outputs (e.g. the csv/xls tables coming from the Emulator, e.g. in Case study 4.2, big table at the end there), you will see only “**acce_ISF**” results.

- In case of positive acceleration, these are driven by the bgAccel_ISF_weight setting, and results are >1.

- **In case of negative acceleration** (decelerating rise), **bgBrake_ISF_weight is applied**, , and results are < 1. (Example see in graph in section 10.3.3.3).

In full closed loop, the bgBrake_ISF_weight is often only about half as large as the bgAccel_ISF_weight (but that would also depend on your personal diet pattern and eating/digestion speed). Also here, one should approach the tuning gradually, increasing the weight coming from small values.

Please observe that **tuning bgBrake_ISF_weight must strictly be done with types of meals for which there is insulin need at de-celerating but still rising bg**.

bgBrake_ISF is totally irrelevant for hi carb meals where your loop shot over iobTH already by the time your rising towards the bg peak slows down!

Likewise, if your initial bgAccel_weight is set so strong that your first SMBs catapult you over the iobTH, no matter what type of meal: Then you must **first** find a reasonable setting for this parameter, one that works "good enough" to control your carb loaded meals, and then see whether there is "room" (and need) for milder loop response at low carb meals.

In case you cannot quite get all the ISF_weights "right" so the occasional low carb meal will not get over-treated: Avenues to adapt your loop aggressiveness are discussed in section 5.
For instance you will be able to (if needed):

- use a temp. reduced %profile
- temp. lower iobTH or bgAccel_ISF_weight
- construct for yourself a "DIY cockpit" with an extra "snack" or "low carb" button with an underlying suitable Automation

In the **late stage of still rising (!) glucose**, the Full Closed Loop typically sharply reduces SMBs already because it is "painfully aware" of the following principal conflict:

- iob (like formerly given in HCL via your bolus) must go high quickly, in order to limit the high
- However, if there is too much insulin in the system, a **hypoglycemia can happen later** within the DIA time window, because the loop can, later, only correct to a very limited extent (namely, only to the extent that it can set basal to zero).

Therefore, the core problem is that the Full Closed Loop must build up iob very quickly, **but not too much**, in the initial phase of a meal, and high bg values (out of range, >180 mg/dl) can not always be avoided.

4.5 Sluggish rise into a bg plateau, or late plateauing at high bg: dura_ISF and bg_ISF

Depending how your personal diet spectrum looks, you need to tune-in your dura_ISF primarily with large hi-FPU meals, and/or for meals at the low carb end of your diet

4.5.1 dura_ISF for sluggish rise into a bg plateau

A (in that case, often not very high) plateau can form in **low carb meals**, when, basically, carb and insulin "burn rates" might keep a balance over an hour or longer, requiring occasional moderate size SMBs.(See an example in case study 4.2)

4.5.2 dura_ISF for late/high bg plateaus

With **large or high fat/protein meals,** often a long high bg plateau is encountered (sometimes associated with 2nd "late, long stretched hill" forming for this, in the bg curve).
For such situations, autoISF features the modulation of ISF depending on bg level and on duration of **plateau** formation.

4.5.3 "One size fits all" -dura_ISF

Absolute "pros" could primarily calibrate their dura_ISF for low carb.

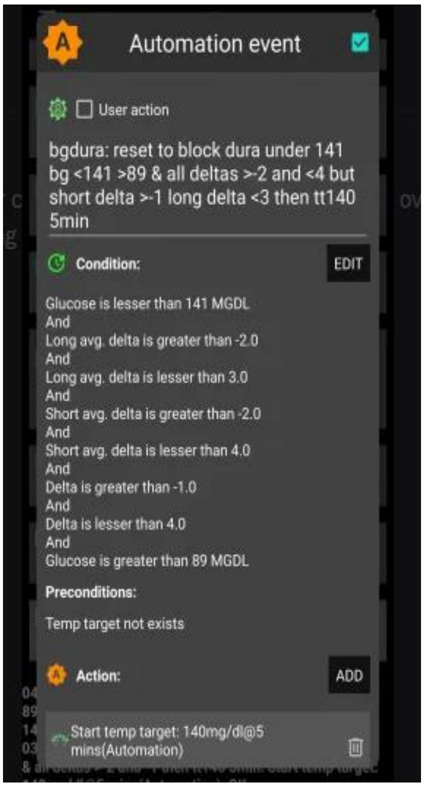
Dura_ISF has in-built amplification at higher bg levels. So, effects will automatically be boosted in case much higher plateaus develop after greasy feasts.
Should that not per se be sufficient, there is more the DIY "pro" can do:

- by adding an Automation that gives an extra boost "against" the temporary insulin resistance associated with fats (via increasing the baseline, in terms of a temp.130% profile switch, for instance. Compare at: https://androidaps.readthedocs.io/en/latest/Usage/FullClosedLoop.html#stagnation-at-high-bg-values),
- *or by* making additional use of the bg_ISF (or dynamicISF) (-> Tune it in parallel.)

The author's preference would be to go via Automation, but only in case the in-built differentiation via bg level make it necessary.

4.5.4 Options to limit iob delivered from dura_ISF

Rather than relying on your initial tuning to keep you safe from hypos also in the future, there are some extra precautions you could take. Some were discussed in Discord or in dev circles, regarding what could be done:



1) To limit the danger of going low, it can make sense to design an **Automation** which pauses the delivery of more insulin.

   This one was suggested by Alex999

   If a glucose plateau built under 140 mg/dl, do not treat via dura_ISF (because the defined Action is to set an elevated TT to a level that will not require more correction insulin.

   An alternative Action would be to set, near the actual glucose target, an odd-numbered TT (which blocks any SMB be given, while valid).

2) In an autoISF update, the **duration** in which iob is added up could be **capped** after max. 1.5 hours of any “stubborn high”.

3) Instead of 2), or additionally, the total **iob accrued in that “dura phase” could be capped** by a new related safety setting. It would probably be anchored on iobTH, and could also become a tuneable setting, maybe even a new parameter useable in Automations, too.

4.5.5 How dura_ISF works

Conditions for dura_ISF to become active:

1) Glucose plateaus, i.e. it is varying within a +/- 5% interval only, and this situation lasted at least for the last 10 minutes => Duration **dura_05** = 10 or more minutes.
2) The average glucose value in this “duration” time window is **avg05** mg/dl. Its elevation relative to bg target determines one of the factors in the equation (eq.3) for dura_ISF, see below.

Effect:

- The strengthening of ISF is stronger **the longer** the situation lasts, and **the higher** the average glucose is above target:

3) This can be individually tuned by the **duraISF_weight** to automatically manage high plateaus in bg values.

How changing the _weight influences the resulting calculated insulinRequired

(You can continue 3 pages down at 4.5.6, unless you want to know quantitatively how things work).

The developers' documentation (Quick Guide, page 6) https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf gives the following equation:

$$dura\_ISF = 1 + \frac{avg05\text{-}target\_bg}{target\_bg} * \frac{dura05}{60} * dura\_ISF\_weight \qquad \text{(eq. 3)}$$

profile_ISF / dura_ISF = effectively used ISF (sens)

(…if the dura influence dominates, and is used as effective ISF. Else, see flowcharts in Quick Guide)

When looking at the same moment, we can combine the first two factors factors of eq.3 (which vary with bg elevation above bg target, and with how long the plateau already shows) into a term "iF".

For estimating the effect from using another ISF_weight, we then have two (eq.3) , with two unknowns, iF ("cancelling out"), and the sought dura_ISF (that results for the new _weight).

Example: Your profile_ISF is 40 mg/dl/U.
Using dura_ISF_weight of 0.6 you saw effectively used ISF of 30.3 mg/dl/U (box C21 in table on next page) = 40 / 1.32 => factor for dura_ISF =1.32.
For an intended correction by – 10 mg/dl the insulinRequired would calculate to 10 / 30. 3 = 0.330 U.

Now, going with a 33% stronger dura_ISF_weight of 0.80 (box E3):

dura_ISF = 1+ internalFactor * dura_ISF_weight *

before 1,32 = 1 + 0.60 * iF => 0.32 = 0.60 * iF => iF = 0.533

after ? = 1 + 0.80 * iF => ? = 1 + 0.80 * 0533 = 1.426

New effective ISF would be 40 / 1.426 = 28.04 mg/dl/U (F21 in table).
For an intended correction by – 10 mg/dl the **insulin**Required would now calculate to 10 / 28.04 = 0.357 U, which is 8.1 % more insulin (H21 in table next page).

	A	B	C	D	E	F	G	H
1	profile.ISF (mg/dl/U)	40		**dura_ISF tuning -**				
2	InsReq (U) @ iF=0	1		see **FCL e-Book** section **4.5.5**				
3	dura_ISF_weight:	0,6			0,8			InsReq e
4	iF	dura_ISF_old	ISF_old	InsReq_old	ura_ISF_new	ISF_new	InsReq_new	E3 vs.B3
5	0	1	40	1	1	40	1	0,0%
6	0,033	1,020	39,2	1,020	1,027	39,0	1,027	0,7%
7	0,067	1,040	38,5	1,040	1,053	38,0	1,053	1,3%
8	0,100	1,060	37,7	1,060	1,080	37,0	1,080	1,9%
9	0,133	1,080	37,0	1,080	1,107	36,1	1,107	2,5%
10	0,167	1,100	36,4	1,100	1,133	35,3	1,133	3,0%
11	0,200	1,120	35,7	1,120	1,160	34,5	1,160	3,6%
12	0,233	1,140	35,1	1,140	1,187	33,7	1,187	4,1%
13	0,266	1,160	34,5	1,160	1,213	33,0	1,213	4,6%
14	0,300	1,180	33,9	1,180	1,240	32,3	1,240	5,1%
15	0,333	1,200	33,3	1,200	1,266	31,6	1,266	5,6%
16	0,366	1,220	32,8	1,220	1,293	30,9	1,293	6,0%
17	0,400	1,240	32,3	1,240	1,320	30,3	1,320	6,4%
18	0,433	1,260	31,8	1,260	1,346	29,7	1,346	6,9%
19	0,466	1,280	31,3	1,280	1,373	29,1	1,373	7,3%
20	0,500	1,300	30,8	1,300	1,400	28,6	1,400	7,7%
21	0,533	1,320	30,3	1,320	1,426	28,0	1,426	8,1%
22	0,566	1,340	29,9	1,340	1,453	27,5	1,453	8,5%
23	0,599	1,360	29,4	1,360	1,480	27,0	1,480	8,8%
24	0,633	1,380	29,0	1,380	1,506	26,6	1,506	9,2%
25	0,666	1,400	28,6	1,400	1,533	26,1	1,533	9,5%
26	0,699	1,420	28,2	1,420	1,559	25,6	1,559	9,9%
27	0,733	1,440	27,8	1,440	1,586	25,2	1,586	10,2%
28	0,766	1,460	27,4	1,460	1,613	24,8	1,613	10,5%
29	0,799	1,480	27,0	1,480	1,639	24,4	1,639	10,8%
30	0,833	1,500	26,7	1,500	1,666	24,0	1,666	11,1%
31	0,866	1,519	26,3	1,519	1,693	23,6	1,693	11,4%
32	0,899	1,539	26,0	1,539	1,719	23,3	1,719	11,7%
33	0,932	1,559	25,6	1,559	1,746	22,9	1,746	12,0%

	A	B	C	D	E	F	G	H
34	0,966	1,579	25,3	1,579	1,773	22,6	1,773	12,2%
35	0,999	1,599	25,0	1,599	1,799	22,2	1,799	12,5%
36	1,032	1,619	24,7	1,619	1,826	21,9	1,826	12,7%
37	1,066	1,639	24,4	1,639	1,853	21,6	1,853	13,0%
38	1,099	1,659	24,1	1,659	1,879	21,3	1,879	13,2%
39	1,132	1,679	23,8	1,679	1,906	21,0	1,906	13,5%
40	1,166	1,699	23,5	1,699	1,932	20,7	1,932	13,7%
41	1,199	1,719	23,3	1,719	1,959	20,4	1,959	13,9%
42	1,232	1,739	23,0	1,739	1,986	20,1	1,986	14,2%
43	1,265	1,759	22,7	1,759	2,012	19,9	2,012	14,4%
44	1,30	1,780	22,5	1,780	2,040	19,6	2,040	14,6%
45	1,40	1,840	21,7	1,840	2,120	18,9	2,120	15,2%
46	1,50	1,900	21,1	1,900	2,200	18,2	2,200	15,8%
47	1,60	1,960	20,4	1,960	2,280	17,5	2,280	16,3%
48	1,70	2,020	19,8	2,020	2,360	16,9	2,360	16,8%
49	1,80	2,080	19,2	2,080	2,440	16,4	2,440	17,3%
50	1,90	2,140	18,7	2,140	2,520	15,9	2,520	17,8%
51	2,00	2,200	18,2	2,200	2,600	15,4	2,600	18,2%
52	2,10	2,260	17,7	2,260	2,680	14,9	2,680	18,6%
53	2,20	2,320	17,2	2,320	2,760	14,5	2,760	19,0%
54	2,30	2,380	16,8	2,380	2,840	14,1	2,840	19,3%
55	2,40	2,440	16,4	2,440	2,920	13,7	2,920	19,7%
56	2,50	2,500	16,0	2,500	3,000	13,3	3,000	20,0%
57	3,0	2,800	14,3	2,800	3,400	11,8	3,400	21,4%
58	3,5	3,100	12,9	3,100	3,800	10,5	3,800	22,6%
59	4,0	3,400	11,8	3,400	4,200	9,5	4,200	23,5%
60	4,5	3,700	10,8	3,700	4,600	8,7	4,600	24,3%
61	5,0	4,000	10,0	4,000	5,000	8,0	5,000	25,0%
62	Note: To determine not the incremental effect between two weights (E3 vs.B3), but the dura_ISF effect							
63	for the investigated _weight (E3): Enter 0 in (B3)							
64	Column (H) then shows the full effect of duraISF with the selected weight (E3)							
65	Use table (I21 x S39 or higher), input target_bg at (J21), then look up iF for the observed dura							
66	and avg05 => which "iF line" applies in table (A3 x H61)							

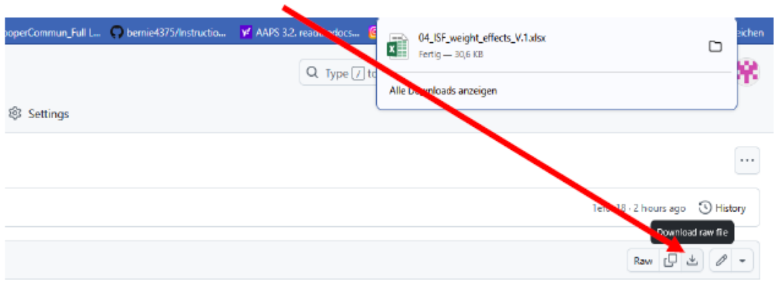
The effects could be more or also less pronounced because the iF factor strongly depends (as in eq.3, see also table on next page) on:

- how high bg is above target (**avg05**)..
- …and for how many minutes (**dura05**).
- Also, do not under-estimate the effect of a low **target_bg.** Not sure you should, but certainly *you could* add an extra boost from that via an Automation that kicks in a low TT (at the "dura situation" that you describe as a triggering condition in your related Automation):

  Example: You plateau at 180 mg/dl and your bg target is 100. The first factor in your iF term then is (180-100)/100 = 0.80.
  Now, with a TT = 74 mg/dl, that term becomes (180 – 74)/74 = 1.43
  (The resulting + 79% in your boost factor does not translate into 79% more insulin - just as we have seen "only" 8% more insulin with a 33% boost in the complete example before).

An **Excel tool** is provided to download as raw file …
…from the e-book (Github /bernie4375/section 04)
for these complex calculations.

To open the ...xls, go to: "Download raw file":



The following table (from this tool) shows examples for combinations of bg target (yellow input field at box J1 resp. box J22) , bg plateau level (mg/dl, blue figures), and duration (minutes >10, in violet), the iF of 0.533 in our example above could originate (fields marked green):

	I	J	K	L	M	N	O	P	Q	R	S
1	target_bg	100	=>	**iF factors** resulting at different avg05 (mg/dl.bg), and duration (minutes)							
2	dura05;avg05	120	140	160	180	200	220	240	260	280	300
3	10	0,033	0,067	0,100	0,133	0,167	0,200	0,233	0,267	0,300	0,333
4	15	0,050	0,100	0,150	0,200	0,250	0,300	0,350	0,400	0,450	0,500
5	20	0,067	0,133	0,200	0,267	0,333	0,400	0,467	0,533	0,600	0,667
6	25	0,083	0,167	0,250	0,333	0,417	0,500	0,583	0,667	0,750	0,833
7	30	0,100	0,200	0,300	0,400	0,500	0,600	0,700	0,800	0,900	1,000
8	35	0,117	0,233	0,350	0,467	0,583	0,700	0,817	0,933	1,050	1,167
9	40	0,133	0,267	0,400	0,533	0,667	0,800	0,933	1,067	1,200	1,333
10	45	0,150	0,300	0,450	0,600	0,750	0,900	1,050	1,200	1,350	1,500
11	50	0,167	0,333	0,500	0,667	0,833	1,000	1,167	1,333	1,500	1,667
12	55	0,183	0,367	0,550	0,733	0,917	1,100	1,283	1,467	1,650	1,833
13	60	0,200	0,400	0,600	0,800	1,000	1,200	1,400	1,600	1,800	2,000
14	65	0,217	0,433	0,650	0,867	1,083	1,300	1,517	1,733	1,950	2,167
15	70	0,233	0,467	0,700	0,933	1,167	1,400	1,633	1,867	2,100	2,333
16	75	0,250	0,500	0,750	1,000	1,250	1,500	1,750	2,000	2,250	2,500
17	80	0,267	0,533	0,800	1,067	1,333	1,600	1,867	2,133	2,400	2,667
18	85	0,283	0,567	0,850	1,133	1,417	1,700	1,983	2,267	2,550	2,833
19	90	0,300	0,600	0,900	1,200	1,500	1,800	2,100	2,400	2,700	3,000

22	target_bg	74	=>	**iF factors** resulting at different avg05 (mg/dl.bg), and duration (minutes)							
23	dura05;avg05	120	140	160	180	200	220	240	260	280	300
24	10	0,104	0,149	0,194	0,239	0,284	0,329	0,374	0,419	0,464	0,509
25	15	0,155	0,223	0,291	0,358	0,426	0,493	0,561	0,628	0,696	0,764
26	20	0,207	0,297	0,387	0,477	0,568	0,658	0,748	0,838	0,928	1,018
27	25	0,259	0,372	0,484	0,597	0,709	0,822	0,935	1,047	1,160	1,273
28	30	0,311	0,446	0,581	0,716	0,851	0,986	1,122	1,257	1,392	1,527
29	35	0,363	0,520	0,678	0,836	0,993	1,151	1,309	1,466	1,624	1,782
30	40	0,414	0,595	0,775	0,955	1,135	1,315	1,495	1,676	1,856	2,036
31	45	0,466	0,669	0,872	1,074	1,277	1,480	1,682	1,885	2,088	2,291
32	50	0,518	0,743	0,968	1,194	1,419	1,644	1,869	2,095	2,320	2,545
33	55	0,570	0,818	1,065	1,313	1,561	1,809	2,056	2,304	2,552	2,800
34	60	0,622	0,892	1,162	1,432	1,703	1,973	2,243	2,514	2,784	3,054
35	65	0,673	0,966	1,259	1,552	1,845	2,137	2,430	2,723	3,016	3,309
36	70	0,725	1,041	1,356	1,671	1,986	2,302	2,617	2,932	3,248	3,563
37	75	0,777	1,115	1,453	1,791	2,128	2,466	2,804	3,142	3,480	3,818
38	80	0,829	1,189	1,550	1,910	2,270	2,631	2,991	3,351	3,712	4,072
39	85	0,881	1,264	1,646	2,029	2,412	2,795	3,178	3,561	3,944	4,327
39	90	0,932	1,338	1,743	2,149	2,554	2,959	3,365	3,770	4,176	4,581

Still. a “trial and error” approach might be easier to handle (see the following section, 4.5.6):

> *Off topic:* dura_ISF is also very useful in Hybrid Closed Loop. It can be used to elegantly manage, fully automatically, a temporary insulin resistance from fatty acids. Please refer to other papers for details (for instance, sections on FPU and persistent high bg in “Meal Mgt.3…,pdf”, available here: https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings).

4.5.6 Set your dura_ISF

Set a **start value of 0.2** for your dura_ISF_weight, and increase only cautiously with an eye on hypo prevention 2-3 hours later.

Caution: Fine tuning this parameter only makes sense ***after*** you tuned your bgAccel_ISF and pp_ISF well (so your thin yellow insulin activity curve shifts *as far to the left*, towards meal start, *as possible*, which will lower bg peaks and ease the job for dura_ISF).

4.5.7 Set your bg_ISF

Since in Full Closed Loop we make our loop give us the maximum SMB size it can, at the beginning of a rise, it is crucial to **resist the temptation to continue** with a particularly **strong ISF** in the meal phase with the **highest glucose** values .
This is a reason why in Full Closed Loop we do not make much use of the ***bg_ISF*** component of autoISF.

- Wanting to get most of our insulin from SMBs delivered at fairly low (but beginning-to-rise) bg implies that we do **not** make ISF weaker at low bg. Under preferences/OpenAPS SMB/autoISF/bg_ISF settings you could set **lower ISF_range_weight** = 0.0.

  If you want to analyze in your data, whether you might benefit from a milder ISF at low bg values (e.g. if you often go below target after correction of only mildly elevated bg in the preceding hours), you may want to try lower ISF_range_weight = 0.1 or 0.2. Study the effects from bgISF, and increase, or decrease, the bgISF_weight to fine tune the sought-after affect.

- The **higher_ISF_range_weight** is used when bg is above target, It then strengthens ISF the more the higher the set weight is. 0 disables this contribution, i.e. ISF is constant in the whole range above target.

  In FCL, this factor should be fairly irrelevant: Near glucose peak, zero-temping usually prevails anyway, so the settings we try might often not be used really by the loop. Very likely, you can live with setting the weight to = 0.0 here, too.

  If you want to analyze in your data, whether you might benefit from a stronger ISF at high bg values (e.g. if you often remain above target after correction of elevated bg in the preceding hours), you may want to try higher ISF_range_weight = 0.1 or 0.2. Study the effects from bg_ISF, and increase, or decrease, the higher_ISF_range_weight to fine tune the sought-after affect.

  In case bg_ISF shall play a bigger role in your loop, please consult the related developers' documentation (Quick Guide, page 4) at: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf .

4.5.8 "Quality control" on your tuning for the later half of your meal time

The later stages of meal management (both, in HCL and in FCL) struggle with the problem that there is a **hypo danger** from the "tail" of insulin activity from earlier SMBs that were needed to fight high bg or plateaus associated with temporary insulin resistance.

Once your bg sits high, neither you, nor a hybrid closed loop with all the carb info, nor your FCL can work wonders.

Very important:

- Iterate between **2 or 3 kinds of meals** (from your typical spectrum) to find **one** set of settings that works *good-enough for all.* That should be possible.
- If you can't make it work for certain meal types, see sections 4.7 and 5. what you can do then.

Observe hypo trends after meals, and

- resist the temptation to elevate the **dura_ISF**_weight very high.
- try to stay away from **bg_ISF** or dynamicISF in Full Closed Loop:
    - In FCL you probably can afford to shut bg_ISF entirely off via setting both related _weights to 0.0.
    - At least be careful, use small ISF_range_weights and check whether you are happy with the contributions to the effectively used ISFs
    - *Off topic:* If, coming from dynamicISF usage, you stay in **Hybrid** Closed Loop, but now with autoISF, you probably can use the bg_ISF parameter with higher _weights to emulate what you like to replicate from your dynamicISF experience.

bg highs will take time to resolve.
Interestingly, an after-dinner walk can work wonders sometimes (take glucose tablets along).

Zero-temping and too tightly set safety limits can be **stumbling blocks in your tuning** project:

- Investigating effects of set ISF_weights is not really possible in periods of zero-temping.
- Too tightly set safety limits "allow" tuning that really is way too aggressive, but gets "cut" by your too-cautious safety settings (e.g. for SMB_range_extention, or for autoISFmax). Aggressive settings then could not come into play most of the time.
  However, some *other* time they might come into play, and *then* produce a hypo 1-2 hours later.

Therefore, **carefully study the SMB tab** (**or** better yet, do an **emulator** based analysis, see sections 10-11) to see

- what the selected weights **would** do, **if** there was **no zero-temping** at the time, and

- whether you bump into a **set limit** already (if your bgAccel_ISF_weight makes you exceed allowed max. SMB size, then further tuning your settings only makes sense with either allowing bigger SMBs, or limiting bgAccel_ISF_weight to a lower number at which you will not frequently bounce into the SMB limit)
- at which **other** times (rather than the one you currently look at and try to improve) that selected setting might backfire

4.5.9 How your “UAM” loop concludes insulin need for your un-declared carbs

The UAM Full Closed Loop doesn't get any information from you as to how many grams of carbs will be there, for absorption.

Looking back

For the recent 5-minute segments, the UAM oref(1) loops can precisely calculate how many grams of carb “must have been digested” based on the CGM values seen, and your IC and ISF profile parameters
For more detail see chapter 1.2 on how dynamic carb absorption is calculated, in “Carb ratio (IC, CIR)…pdf” at: https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings.

Looking into the next minutes, hour

However, *here* we worry about the *late meal stage*, and our FCL has gotten no information from us about how many grams **in total** were eaten, and certainly we do not bother to give eCarbs with estimated **absorption times** (that are so essential in iOS Loop).

So, in FCL you leave your loop without knowledge when your steady-state max carb absorption phase…

- the earlier mentioned 30g/h, or
- with gastroparesis, or if on GLP-1 drug treatment, probably on a lower g/h level
- sometimes prolonged (“faked”) by a brief episode of insulin resistance to fats

…might end. Nor, whether extra carbs were added, later, or “FPUs are lurking”.

The FCL now needs to provide desired amounts of insulin, while facing a potentially induced hypo danger later (considering the DIA of all the insulin still active).

Fortunately, the UAM Full Closed Loop is *not completely clueless* regarding how carb absorption *will continue*:

It will work with a **prediction** of *further* carb absorption, building on the **carb deviation**s (=calculation of how much got absorbed in the *past* 5 minute segments), and phase out further *expected* carb decay in the course of the next 1 to max 3 hours.

For more detail see

- https://openaps.readthedocs.io/en/latest/docs/While%20You%20Wait%20For%20Gear/Understand-determine-basal.html#understanding-the-basic-logic-written-version

- *or do a real-time study* with (screenshots from) your SMB tab info.

Discussion

This UAM prediction about further carb absorption can be worse, but **can** also **be better than a prediction based on the user's „e-Carb" input** as done in Hybrid Closed Loop.

In any case, and even when having perfect knowledge about how exactly the carbs fade out in the next hours, there would still be a principal problem for the loop: Heavy insulin „fire" against highs will not work immediately (depending on the insulin's time-to-peak), and notably it comes with a significant hypo danger from the „tail" of insulin activity.
A big bolus, or also a series of boli, will rarely work exactly for several hours matching the absorption of carbs (from what, how much and and how fast the user ate).

> *Off topic closing remark*: With meticulous attention to all carb-related profile parameters, and daily inputs on amounts and absorption times, plus some "mindfulness" as to which diet habits disturb the possible balance, there are "pro" loopers (notably on iOS Loop) who achieve impressive %TIR (and % in tight range) performance. – The author consciously chose the other way, to put substantial effort into a personalized upfront system calibration, and work with a oref(1) algo system that allows (nearly) every day hands-off FCL.

4.6 Tuning your initial settings

Be pro-active: **The earlier large SMBs come** (driven by bgAccel_ISF and pp_ISF) …
> Note: Also your CGM smoothing may play a role here, that you may want to look into !

…the **less high** the overall increase in BG will be, and (provided you set a proper iobTH) the **lesser** the **risk** will be **for a hypo** after the meal.

Therefore, **put most of your FCL tuning effort into determining suitable weights for bgAccel_ and for pp_ISF, and for finding a suitable iobTH_percent**.

Low carbers probably should pay more attention on **dura_ISF**, besides seeing to it that bgAccel_ISF is not too aggressive (see above, section 4.2.5 and case study 4.2).

Later, your “FCL cockpit” will give you access to **temporarily modulate** these essential parameters (see section 5.2.), providing you an opportunity

- in your tuning phase, for more research on the fly, so to speak
- everyday, for temp. adaptations to altered insulin sensitivity, or to special disturbances (should you, occasionally, see a need).

After you tuned your **initial settings** well, there should rarely arise a need for “fine tuning” later, see section 8 and case study 8.2!

The experience of the author is that it is possible to tune the above mentioned weights for very different meals in such a way that the glucose almost always remains acceptably in range.

However, if you come to the conclusion that **differentiated settings** (for different meals or meal time clusters) would be easier to establish, and/or work better for you, the following sections suggest many options you could try and use.

4.7 Maneuvering through more complex scenarios

You now can move on, to accommodate more complex scenarios.

4.7.1 Complex meal spectrum

- Especially if you are a bit shy of using the Emulator (section 10 and 11) for really detailed analysis, it can well be that you will not hit *one* real good system calibration (section 4) for your *entire range* of diets (e.g. your *meal spectrum* at all your lunches).
  - Note that *between meal time slots* (e.g. breakfasts vs lunches) you should differentiate via different “circadian” *profile_ISF*s (with which the autoISF-effects always multiply).
- So, in case you occasionally run out of range (bg =70…180 mg/dl), your options to prevent, react, or improve are:
  - accepting a few % higher time outside of range for that day (and, if feasible, in the future avoiding what seemed to have caused it)
  - taking a snack (whenever you tend to go low from the “tails” of insulin activity that was required to fight a peak)

- doing a manual “tweak” (if you can think of one in time), to manage the problem manually. For example, briefly going into an odd TT (=temp. blocking more SMBs) can be a very easy-to-handle remedy, sometimes
- define a User Action Automation, and provide an extra “cockpit button” to announce a meal *outside of your usual spectrum*, so it will automatically be treated differently by your FCL (as you defined in your Automation; example: Case study 5.2).
- temporarily resorting to “your old” hybrid closed loop.

Instead of accepting such instances, you could launch “improvement projects”, that refine your initial tuning (section 4. and sections 8 and 9)

Note, though, that it could be near-impossible to fine-tune *if your basics never were “right”* and you got lost in a maze of errors and counter-errors.

In that case, **only a fresh start might convincingly help.**

4.7.2 Complex tasks aside from managing meals

To deal with *different* disturbances than presented by your meal spectrum (that you were calibrating for in this section 4), there will be **other** instances where **temporary modulations** of your FCL will be needed.

You have a variety of options to deal with that, and this will be the topic in section 5.

It is suggested to do **major exercise** still *in your hybrid closed loop* setting, *until* you have your FCL up and running for meals on normal days with no or only moderate exercise.
Later, implement extras as discussed in section 6 to fully implement your FCL.

4.8 Profile helper

DRAFT by John Kitching/edited

For general loop settings for FCL with autoISF, please consult [sections 2.1 – 2.6]

The following table gives a comprehensive approach how to tune your autoISF FCL better, when you see any of the problems as in *column A*.

Make sure to work through the problems in the order given in this table.

	A	B	C	D
1	Notes:1. if not entering carbs then "several hours after eating" can be applied instead of "with zero COB			
2	**The problem** (follow in order)	**Y/N**	**Possible solution**	**Based on what logic**
3	Are you having perods with zero COB and negative IOB?	Yes	Decrease basal a bit over these periods, wait a couple of days to evaluate and return here	Oref is backing off your basal to try and hold your BG steady so basal is probably too high
4		No	Continue with this sheet	
5	Are you having perods with zero COB and positive IOB?	Yes	Increase basal a bit over these periods, wait a couple of days to evaluate and return here	Oref is increasing your basal to try and hold your BG steady so basal is probably too low
6		No	Continue with this sheet	
7	Do you get roller coaster BGs?	Yes	Increase profile ISF a bit (make weaker). Also, check your set iobTH%. Reevaluate over a couple of days, return here	BG goes over target -> with too-strong ISF, too much insulin is dosed resulting in BG below target. Long zero-temping required, but resulting in high BG next ...
8		No	Continue with this sheet	
9	After your BG went high, do you get a hypo later (from remaining insulin "tail" at zero COB)?	Yes	Reduce autoISF "weights" a bit, reevaluate over a couple of days, return here	ISF is reduced (strengthened) too much by autoISF as you move away from target. Reducing the proper ISF_weight will reduce that impact. Use the **autoISF history table** for more information about **which one** is causing the issue: Preferably, "shave of late", at DURA or BG; PP or ACCEL weights might need reducing, too.
10		No	Continue with this sheet	
13	When you start eating with BG around target and iob near zero, do you get a hypo (in the first 2 hours)?	Yes	Check whether you can reduce your set iobTH%, or SMB range extention, or autoISFmax. Decrease PP (and maybe also ACCEL) "weight" , and increase profile carb ratio a bit (make it weaker, so you see the insulin needed relate well to the g of carbs digested by the time you see a hypo tendency). Reevaluate over a couple of days, return here	You're getting too much insulin to cope with the bg rises from the carbs you've eaten. Taming the loop can be done in several ways: Weakening profile carb ratio and ISF would work in all situations. Lowering ACCEL and/or PP autoISF "weights" would exactly weaken loop over-response inearly stage of BG rise. And the 3 first-mentioned safety settings just limit excesses.
14		No	Continue with this sheet	
17	When you eat, is iob building up too slow, with a lot of it added at bg peak or even still when BG reduces?	Yes	Check whether you must increase your set iobTH%, or SMB range extention, or autoISFmax. Increase PP "weight" , and decrease profile carb ratio a bit (make it stronger, so you see the insulin needed relate well to the g of carbs digested). Reevaluate over a couple of days, return here	You're not getting enough insulin to cope with the bg rises from the carbs you've eaten. If your loop bounces into any of the 3 first-mentioned limits, "tuning more aggressive" might simply not work. Making the loop more aggressive can be done in several ways: Strengthening (lowering) profile carb ratio and ISF would work in all situations. Elevating autoISF "weights" (notably PP) could exactly strengthen loop response in instances where needed (where the respective "weight" is playing a role).
18		No	Continue with this sheet	
19	Does BG stay high for long periods?	Yes	Increase DURA "weight" a bit. Reevaluate over a couple of days, return here	DURA reduces ISF (makes it stronger) if BG stays high for longer periods than it should. Reminder: Follow this sheet in order! This "last job", dealing with persistent highs via DURA, is much easier after you sorted out how to build-up iob faster, so not to go super high in the first place.
20		No	Have a short holiday with lots of carbs	
22	**Further reading:**			
23	Ga-zelle's quick start guide		https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf	
	FCL e-Book , Trouble Shooting		https://github.com/bernie4375/FCL-potential-autoISF/blob/FCL-e-book/09_Trouble.Shooting_FCL-	