4. Meals: Setting ISF_weights in /Preferences

v.4.7

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:

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 5.2.2.2).

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.

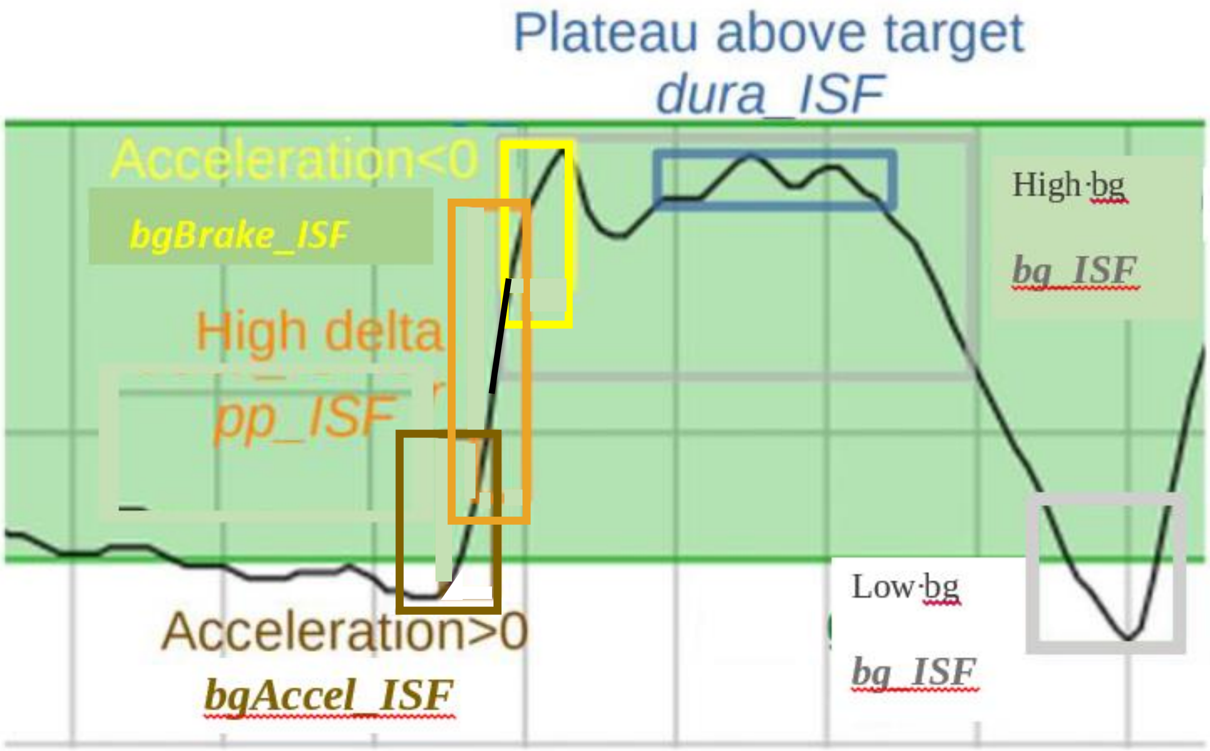
Before getting into this, let’s first have a look on how autoISF basically works. (More see in Quick guide by the developer, referenced in section 3.2; or directly at https://github.com/ga-zelle/autoISF).

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.

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

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.
(You can <u>skip reading the following</u> example calculations, unless you want to know more quantitatively how things work).

> *Example 1: Going from bgAccel_ISF_weight of 0.2 to 0.16 (20% less).*
> *If your profile_ISF is 40 mg/dl/U and with bgAccel_ISF_weight = 0.20 you saw acce_ISF factor of 1.31, this would ((if the acce influence dominates and is used as effective ISF)) lead to the effectively used ISF of 40/1.31 = 30.53 mg/dl/U. For an intended correction by – 10 mg/dl the insulinRequired would calculate to 10 / 30.53 = 0.328 U.*
> *Now, going with a 20% reduced bgAccel_ISF_weight of 0.16:*
>
> *acce_ISF = 1+ bgAccel_ISF_weight * internalFactor*
>
> *before 1,31 = 1 + 0.20 * iF => 0.31 = 0.20 * iF => iF = 1,55*
> *after ? = 1 + 0.16 * iF => ? = 1 + 0.16 * 1.55 = 1.25*
> *New effective ISF would be 40 / 1.25 = 32.05 mg/dl/U. For an intended correction by – 10 mg/dl the insulinRequired would calculate to 10 / 32.05 = 0.312 U, which is 4.9% less.*
>
> *Example 2: Going from bgAccel_ISF_weight of 0.2 to 0.10 (50% less; or doubling in the other direction).*
> *If your profile_ISF is 40 mg/dl/U and with bgAccel_ISF_weight = 0.20 you saw acce_ISF factor of 1.31, this would ((if the acce influence dominates and is used as effective ISF)) lead to the effectively used ISF of 40/1.31 = 30.53 mg/dl/U. For an intended correction by – 10 mg/dl the insulinRequired would calculate to 10 / 30.53 = 0.328 U.*
> *Now, going with a 50% reduced bgAccel_ISF_weight of 0.10:*
>
> *acce_ISF = 1+ bgAccel_ISF_weight * internalFactor*
>
> *before 1,31 = 1 + 0.20 * iF => 0.31 = 0.20 * iF => iF = 1,55*
> *after ? = 1 + 0.10 * iF => ? = 1 + 0.10 * 1.55 = 1.155*

New effective ISF would be 40 / 1.155 = 34.63 mg/dl/U. For an intended correction by – 10 mg/dl the insulinRequired would calculate to 10 / 34.63 = 0.289 U, which is 12 % less (going the other way, 0.328 is 13.5 % more).

Example 2 (-50%) reduces _weight 2.5 times lower than example 1 (-20%), and the resulting effect (-12% vs. -4.9% insulin Required) is also factor 2.5 different.

Note: "Your" internal factor "iF" might differ; for sure it is very different between the various …_ISF components.

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

Between acceleration and deceleration there is a more or less linear further increase of bg and of 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 meals 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)

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 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 consumation 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.
- I**n 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:

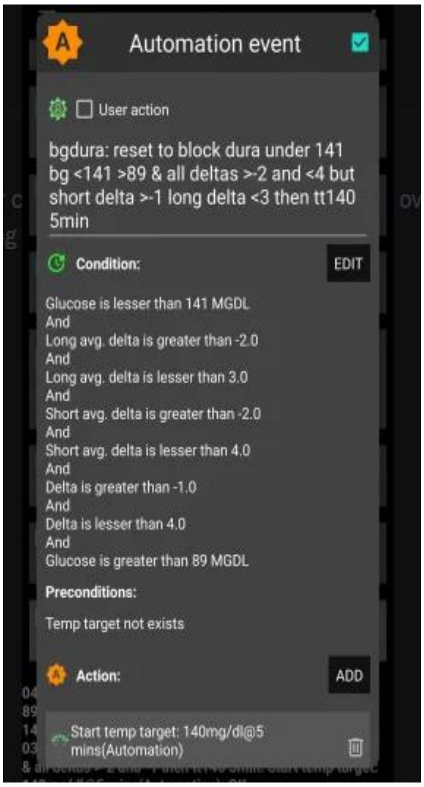
https://androidaps.readthedocs.io/en/latest/Usage/FullClosedLoop.html#stagnation-at-high-bg-values),

- *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 is varying within a **+/- 5% interval** only;

2) the average glucose (*dura_ISF_average)* within that interval is **above target**;
3) this situation lasted **at least for** the last **10 minutes** (*dura_ISF_minutes)*

Effect:

4) The strengthening of ISF is stronger the longer the situation lasts, and **the higher** the average glucose is above target:
5) This can be individually tuned by the **duraISF_weight to automatically manage** high plateaus in bg values.

The formula looks like (this, and more, see page 6 of the Quick guide at Github/ga-zelle/autoISF):

$$\text{dura\_ISF} = 1 + \frac{\text{avg05-target\_bg}}{\text{target\_bg}} * \frac{\text{dura05}}{60} * \text{dura\_ISF\_weight}$$

where

$\text{avg05} = \text{dura\_ISF\_average}$
$\text{dura05} = \text{dura\_ISF\_minutes}$

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, section „Late stage of meals“ of “Meal Management Basics”, 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.

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 aggressive settings might 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” concludes insulin need for your un-declared carbs former 4.5.3 quoted elsewhere!

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 “IC (carb ratio)…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 potentially induced hypo danger later (considering the DIA of all insulin in use).

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 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 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 (meal spectrum)..
- In that case you will occasionally run out of range (bg =70…180 mg/dl), and 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

xls based tool has been under development

- needs more user data
- we have second thoughts, because of *“Do not copy settings from other FCL loopers“* (see section 4.1.1)
- chapter with link to xls tool *might* follow later