2. General Settings for Full Closed Loop V.3.6

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 2.1: (nothing available yet)

When in Hybrid Closed Loop, big boli were given by the user, and the loop had no business to give big ones on top. But this is fundamentally different now.

So, first we must **enable our loop to overcome the narrow safety restrictions for SMB sizes** that were appropriate in HCL.
Evidently, this could become dangerous. Please set your iobTH (section 2.4), and observe all suggestions made in this FCL-e-book *) , and in the github pages of the developers.
See also: https://github.com/ga-zelle/autoISF/blob/A3.2.0.2_ai3.0/How-to-get-larger-SMBs.pdf
/or newer branch/

***) Caution:** This entire e-book is about Full Closed Looping. **In case you intend to work with giving boli**, many suggestions made - notably in this section 2 (and in section 4) – should **not** be followed. You would have to **do extra research**, on your own data, how your bolus changes things. Mainly use the documentations referred to in section 3, then. See also remarks in section 4.1and section 7

2.1 SMB Range Extention

(preferences/OpenAPS SMB/autoISF settings/smb delivery settings/smb_max_range_extension)

2.1.1 Standard scenario: SMB sizes based on 5 minute loop calculations

Full Closed Looping requires bigger SMB sizes. Setting **120 minutes** of basal as max. SMB size as enabled in AAPS Master is necessary, but will rarely suffice.
In AAPS Preferences/OpenAPS SMB/autoISF settings/smb delivery settings, set SMB/UAM max range extention (**smb_max_range_extension**) to **2.0**. That doubles the allowed max. size, to four hours of your profile basal, *for a start.*

Even better, you could determine an estimate for *your* initial setting as in the following description, I w*ill use the symbol, **.....** , to denote where you would use your numbers. My numbers that I use for the same situation will be in parentheses (U).*

*In full closed loop, once the bg starts rising, you want to get at least half of your required meal bolus within 10 minutes, through 2 SMBs. To do that, you need ..... U (2 U) per SMB on average, and because the bolus sizes tend not to be equal when requested by the Loop, you should have at least **.....** U (3 U) as the allowable SMB size. Your hourly basal is around **.....** U (0.6 U), i.e. AAPS Master will allow a max. 2 times that hourly basal which = **.....** U (1.2 U) per SMB. To reach the intended **....** U (3 U) therefore you should set your smb_max_range_extention to **....** (2.5 = 3 U / 1.2 U)*

The profile helper in section 4.8 might be available for doing this calculation, and for a cross-check.

If you have a *very low hourly basal* rate, extensions *bigger than* 3.0 can result (maximum you can set is 5).

Note: Elevated insulin needs *in phases of elevated insulin resistance* probably will be managed with ***>100% profile* adjustments**. Then, profile basal gets elevated accordingly, and thus **will automatically allow increased SMB size**.

Watch out (in your SMB tab, or using the emulator, section 10) whether you often run into a limitation by your set smb_max_range_extention. For instance, your **attempts to increase initial SMB** sizes via elevated smb_delivery_ratio (section 2.3) and elevated bgAccel_ISF_weight (section 4.2) **might get cut by a too low smb_max_range_extention.**

The SMBs your loop requests could get reduced in size also by other safety settings, notably by your autoISF_max setting (see section 2.2)

2.1.2 Special Libre 3 (1 minute) scenario with up to 5 SMBs per 5 minutes

When receiving bg values every minute, and adjusting insulin delivery accordingly in smaller steps, probably 120 minutes of basal per SMB suffices.

Watch whether your system actually can process 5 loop calculations (and potentially give 5 SMBs) in 5 minutes. Then judge (similar like shown in section 2.1.1), what maximum SMB size you would like to see.

Probably you can leave SMB_range_extension at default **1.0** untouched (but select higher if you have an extremely low profile basal in some of your meal hours, or if your loop "misses" many of the opportunities to make an adjustment, every minute).

Integration of 1-minute values is new from autoISF 3.0.1 (May 2024). Please watch your SMB tab (difficult in 1 minute segments => make screenshots, or employ Emulator!), and report experiences / stay in touch with other users.

2.2 Max and Min autoISF Ratio

(preferences/OpenAPS SMB/autoISF settings/autoISF_min and autoISF_max)

For a start, set **autoISF_max** = 2.0 . –

Your CGM (1 or 5 minute re-calculations) should not matter for this parameter.

This allows *up to doubling* of ISF aggressiveness if "requested by the "… _ISF_weights" (see section 4). This is just a first step.

You may have to elevate autoISF_max further, later, if your attempts to tune the …ISF_weights (section 4.) often run into a limitation by your set autoISF_max

For situations of increased insulin sensitivity (less insulin need), you must specify, in your settings (preferences) also what your *lower* limit (for weakening of ISF, compared to profile_ISF) may be.

autoISF_min should be set to 0.3. Again, see whether you ever run into that limit, that your loop e.g. "would like" to act softer, but bounces against that set limit.

Do not keep autoISF_min at 0.5 or even higher, because that would for instance preclude later, that your exercise setting can strongly "soften" your loop.

2.3 SMB Delivery Ratio

(preferences/OpenAPS SMB/autoISF settings/smb delivery settings/smb_delivery_ratio)

2.3.1 Standard scenario, using 5 minute loop calculations

Use the *fixed* **smb_delivery_ratio** and increase the setting (from AAPS default 0.5) to 0.6 or 0.7.

In AAPS Preferences, the smb_delivery_ratio *can* also be set *dynamically* (changing with glucose level). - This seems a feature geared more towards hybrid closed loop applications. (In FCL, we like strong loop aggressiveness at low (but rising) bg, not "wait for" high bg).

So, set your ratio to 0.6 or 0.7 before doing any _weights tuning. Your choice will magnify every SMB, also in phases where you actually want less, so do not exaggerate.

0.6 gets you 20% , 0.7 gets you 40% more insulin 5 minutes earlier, which is a good thing in FCL where you are late with your first meal insulin. But you do not get 20-40 % more really: You gradually will receive the full insulinReq only in increased % increments, 5 -10 minutes earlier.

> The delivery ratio is per se not changing the insulinReq, it just defines what portion gets delivered now vs 5 or 10 minutes later... if the BG trend keeps up...
>
> In that way, keeping the number closer to 0.5 protects against a jittery CGM mostly.
>
> As in FCL we have an above-avg CGM quality, we can safely go for 0.6 or 0.7 SMB delivery ratio as kind of our tuning baseline.
>
> In case you use very strong **smoothing** (e.g. exponential smoothing of incoming CGM values by AAPS) you probably can afford to go more towards 100%. This would make up for probably only some of the “time lost by smoothing” , regarding getting iob up at first signs of bg rising.
>
> But, the closer you set towards 100%, the more have an eye on how each single bg value AAPS works with triggers SMBs.

The jumpier your CGM, the closer remain near 0.5!

Note that it is generally a flawed strategy, to boost *“across the board”* with things like (*always*) a 100% delivery rate, and then limit problems from it via reducing the *dynamic range* of other highly important parameters (for example, of autoISFmax). While, taken together, these two elements of your flawed strategy would keep you safe, they make your FCL less dynamic, which essentially will translate into giving up a couple of %TIR.

In case you had tuned with a 100% SMB delivery ratio, and now go lower:
I would not expect major re- tuning required, but look into implications for the set iobTH%:
A **100% SMB delivery ratio** made you often **bounce over iobTH in a more "nervous" loop**. You now could fine tune that more sensibly, probably elevate the iobTH even (and maybe also , slightly, the accel weight…which, for safety against your wild 100% setting, had maybe to be lowered before)(Or, If you were unsafe before, leave iobTH where it was, and you are safer now, with a lower SMB delivery ratio)

2.3.2 Special Libre 2 or 3 scenario, when using 1 minute values

The recommendation there is to set the data flow Libre -> Juggluco -> AAPS -> exponential smoothing, and use a **SMB delivery ratio under 50%..**

To understand how the 1-minute values are used

- for the (still) 5 minute incremental loop calculations, now done every minute, looking back how the last 5 minutes (evtl. interpolated) went
- for the parabola fit based acceleration detection

please consult the related section (last chapter) in the developers' Quick Guide:
https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

Also, in this very new application, it is especially important to share experience with other FSL users in Discord -> https://discord.gg/tamvhh57Xs

2.4 Safety Against too Aggressive Settings: iobTH%

(preferences/OpenAPS SMB/ autoISF settings/smb delivery settings/iob_threshold_percent

…which gets multiplied with preferences/OpenAPS SMB/Maximum total IOB OpenAPS can't go over (U)

A safety net is needed because autoISF shoots big SMBs when glucose levels begin to rise; but you do not want to bounce into your ultimate maximum total iob (iobMAX) safety setting too often.

Similarly, this safety net is needed also if using autoISF in Hybrid Closed Loop (HCL), where, after a user bolus already provided some iob, autoISF could add too big SMBs to be safe.

Therefore, we install an **iob threshold** (iobTH) which, **when**, and as long as, **exceeded, shuts SMBs off.**

Step 1: In Preferences, set the SMB toggle for even /odd targets to "ON"

autoISF 3.0.1 demands (as we recommend for FCL also for other reasons, see e.g. later in section 5.1.2 and 5.1.3) that concurrently, in AAPS / Preferences / Open APS SMB / autoISF settings / SMB delivery settings:

Enable alternative activation of SMB depending on current target *) **ON**

*) *previous autoISF versions* allowed different settings, now it is same setting, for profile target and for TT

Step 2: In Preferences, set your default iobTH_percent

iobTH is a parameter in AAPS preferences, defined there as fraction of your set maxIOB:

/OpenAPS_SMB/autoISF_settings/Full_Loop_settings:
Percentage of maxIOB above which SMBs are disabled (iob_threshold_percent,)

Step 2.1: Solidify your maxIOB

First, check whether your **maxIOB** is set reasonably in AAPS Preferences / OpenAPS SMB / Maximum total IOB OpenAPS can't go over (U).

- Input a figure (units) slightly above the max level of iob you had ever needed in your past looping history (also considering times of elevated insulin resistance you occasionally may have had to deal with)*;* set maxIOB to that value*.*

- If you do not have "old" data at hand: **Max iob** would be the sum of hourly profile basal, plus the max meal bolus you might need (g carbs digested in first ~ 2.5 h divided by IC), plus correction bolus for elevated value (max seen BG at meal starts minus target BG divided by ISF). Then take the result times a factor, e.g. times 1.2 in case you see on some days 20% higher insulin sensitivity, and will use 120% profile (switch, or driven by Autosens max, or by other settings, see section 5 and 6, later).

Step 2.2: Identify your max iob need in big meals

Now look at your meal spectrum, and what bolus size, and iob level (including from SMBs; in HCL or FCL) was useful *) in high carb meals to control your glucose. (*For instance, the author needed up to 8U early-on in big meals in HCL; and he has TDD near 40 U, and maxIOB set to 10 U).*

*) useful level = ***iob needed*** for the meal; iob may in time have gotten even higher. However, if, in the end, to prevent a hypoglycemia, you had to consume 15 g carbs, then deduct 15 g / (your IC) from that even higher iob ***you actually did temp. have***. *Example: 15 g / (10 g/U) = 1.5 U*

Step 2.3: Set your iob_threshold_% in AAPS/Preferences

Assume you want to approach no more than about 75% of that iob level (that would be useful to have at big high carb meals) via rapid SMB „fire", after a meal related bg rise is seen (*then*, *for instance, reduce from 8 U to 6 U*).

Then calculate your setting for **iob_threshold_percent** in AAPS / Preferences:

= desired total iob given via SMBs before bg peaks / iobMAX

Enter the according percentage in /Preferences

(*In the example it would be = 6 U / 10 U = 0.6 ; which means to enter 60 as percentage in /preferences*).

In section 4.8 a profile helper might be available for more guidance.

The iobTH then is calculated as follows:

iobTH = iobTH_percent x maxIOB

Step 2.4: High-carbers may need to reduce their iob_threshold_percent a bit, to factor in that *the last "allowed"* SMB can shoot *above* iobTH:

- The last SMB given **can exceed** that threshold by up to **+30%** of the effective iobTH.

  This is desirable because it allows higher iob at big high carb meals (where SMB size, when approaching iobTH, is still big); at lower carb meals either iobTH will not be reached anyways, or SMBs are quite small when reaching iobTH and will not shoot over by much.

  A big SMB that would shoot over by more than +30% will be cut at 130% iobTH.

- Until iob falls below effective iobTH, only %TBRs supply more insulin, if the loop calculates that more iob is still required.

- In low carb meals, that iobTH level should not be reached => the autoISF parameters („weights") need to be tuned carefully, so SMB sizes are *not always* huge, and bounce against the iobTH restriction, but show different behavior for different meals

- Note that when operating with an *even elevated* bg target (>100 mg/dl), iobTH can only be exceeded by **+20%** ("loop at medium power"). This makes sense, notably in an exercise context (…in which *the iobTH per se* also gets automatically lowered, as later discussed in section 6.1.3.).

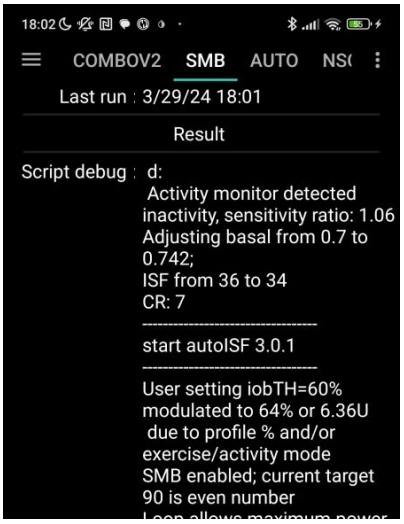
Step 2.5: *At some later stage*, come back to fine-adjust your percent setting in /Preferences,, factoring in that it can be auto-modulated.

- autoISF 3.0 and higher contains a function to **auto-adjust** iobTH **with TT** set: Dynamic iobTH (section 6.1.3). In your initial tuning, just set a iobTH_percent that is good-enough on your average day. A ***low*** (e.g. EatingSoon) TT can automatically ***elevate*** iobTH.

- An ***elevated*** (e.g. exercise) TT can automatically ***lower*** iobTH, which is highly desirable ***for exercise.*** The formula for the resulting effective iobTH is complicated, especially when the exercise mode is also activated. See sections 3.3 and 6.1.3, and example in case study 6.2

- The resulting **effective iobTH** can be seen in the SMB tab.

  The example on the right shows that iobTH can also get temp. ***elevated*** **.** for instance in the case of detected ***in*-activity**:



- From autoISF 3.0.1 onwards, the **SMB tab** starts with the Result section, and right below the / start autoISF headline, the resulting modulated iobTH is given:

Green texts describe currently not available features that were suggested for further development. In later software updates, it is desirable to see the modulated number (6.36 U in the example) also next to iob (below the glucose value in the AAPS main screen).

- Likewise, a **%profile** setting < 100% (or > 100%) will proportionally lower (or elevate) the iobTH that will be used.

effective iobTH = % temp.profile x iobTH

After the temporary % profile expired, it will automatically revert to your originally set iob_threshiold_percent.

Off-topic note, regarding the **effective ISF ("sens")**:
In the SMB tab, *above the "start autoISF.." line*, the profile ISF is given ("ISF unchanged"), eventually with adaptation by activity monitor ("adjusting …ISF from … to .. " ?) or by a TT ("adjusting …ISF from … to ..") or by a %temp. profile set ("unfortunately" still called "ISF unchanged" then).
Then follows the autoISF section explaining in detail how the recently encountered bg curve characteristics suggest adaptations, and what overall the conclusion is ("final ISF factor", calculated following the flowcharts as explained in detail in section 03.).
Below the autoISF section, the effective ISF (sens) results from dividing the (unchanged or adapted) ISF *prior to* "start autoISF", with the determined "final ISF factor" at the end of the autoISF section of the SMB tab.
Example given in section 5.4.5

If both, a % profile and a TT are set, both effects multiply.

We shall see later, how this opens nice avenues for exercise management, where we like to strongly limit how high iob shall be allowed to go. Example given in case study 6.3.

All above discussed settings must be made in AAPS/preferences

- except for the **temporary modulations** of sensitivity, which can be done from the AAPS home screen via %profile or via TT inputs

(This will in detail be presented in section 5 or, regarding exercise button, in section 6.1.3). Regarding activity monitor see section 6.6).

Setting different iobTH via an Automation.

If you do have situations where you rather use a different iobTH_percent than set in your profile (or than resulting in an eventual auto-modulation), you can change it also via an Automation. Caution: This will overwrite your iob_threshold_percent until you restore it, manually (in /Preferences), or via another Automation (see e.g. section 5.1.4).

For this reason, preferably work with the temporary adaptions via %profile, TT and exercise mode as discussed above.

Next steps:

Before going first time into into FCL mode, you must **first**

- check proper AAPS settings according to section 2.6
- **make additional settings in** /preferences/Open APS SMB/**autoISF,** when you get to section 4.

Only after you also went through section 4, you can **enter/exit FCL** (for initial tuning, or for everyday utilization) via

- in AAPS/Preferences/OpenAPS SMB/autoISF settings/"**Enable ISF adaptation by glucose behavior**" **ON** / OFF

In your multi-week FCL set up phase you will quickly notice that changing this setting back and forth "all the time", in your initial project weeks, is not convenient.

A much easier way to "switch off" FCL aggressiveness is to set an odd-numbered bg target, and an even target again when you want normal FCL aggressiveness again. With odd bg targets, you run in "AMA" mode, with autoISF boosted %TBRs but much milder acting because you will not receive any SMBs. See discussion in section 5.1

- An improved solution might become in the future available via a User Interface upgrade (described in section 5.3.1): Switching between **FCL** *and HCL* by just tapping on the **violet**/ *green* **closed loop icon** of your AAPS home screen - after this feature becomes integrated in a future autoISF version update.

2.5 EatingSoon TT ?

FCL works in principle also without setting an EatingSoonTT. Try for yourself whether you miss any performance via the totally hands-off way (*and maybe skip this section for now*).

Your FCL works best if you start meals at below-target glucose values, and ideally have a bit of positive iob at meal start. Also, a low temp. glucose target helps making SMBs (that „aim at it“) a bit bigger.

Setting an EatingSoonTT well ahead of meal start therefore is *in principle* a good idea

- If you have relatively fixed meal time slots in the 24 hours of the day, you could set the **target glucose** values **in your profile** accordingly. So *e.g. 11-15h target 76 mg/dl* if you almost always start a lunch between 11:45 and 14:30h. (*If you do exercise or physical work in that time, this would be too aggressive, and probably also un-necessary*).

- If you have rather irregular habits, it might be worthwhile to **manually** set an **EatingSoonTT** (which is quite time-uncritical) well before the start of a meal, or even (latest) when the first SMB is about to be triggered by your loop. – **Or, just forget about it:**

However, if (as to be expected) your loop anyways always regulates you down to near-target, the effect from setting an EatingSoonTT will be limited:

- If your meals are spaced by a couple of hours your glucose should not be elevated as you approach the next meal …

- If you eat more at a still elevated glucose, your loop should provide you with “a balancing” iob (and hence a prediction to get to target soon) …

…In both cases (which are the prevalent norm) setting an EatingSoonTT would only have a very minor effect.

The good news therefore is: **Setting an EatingSoonTT has only minor effects, if any, and is not required for autoISF FCL.**

Preferred solution with respect to Eating SoonTT

My preferred solution is as follows

Principal solution: I am ***not*** setting an EatingSoonTT ahead of the meal. For getting full loop aggressiveness, an even profile target (or TT) under 100 mg/dl is the only requirement.

> Note: To *prevent* full loop aggressiveness *when not needed*, e.g. at night time, I set an odd profile target (or TT); details see section 5.1.2.

Setting odd (profile or temp.) **target** is generally a recommended easy way for you **to block out SMBs**, as an **"emergency brake" for your FCL**. This should be particularly useful in your initial tuning work.

Refinement: Just "to orient the calculated insulinRequired towards a more aggressive target", I have my loop automatically set a low TT just around the time when it gives first SMBs

For this you can define an **Automation** like: CONDITIONS: likely meal time of day + 1st sign of a beginning meal + iob under (?) U (to kick in only in initial phase) + no TT running => ACTION: set TT=74 mg/dl for ~30 minutes

> This will **not** immediately influence the first one or two SMBs, which come at acceleration detection *before* a + 10 is seen. Actually this can help "de-couple" treatment of low vs high carb meals:
>
> - All meals have an acceleration stage in the very beginning, when we already do want SMBs. We should tune bgAccel_ISF_weight (in my suggested mode, without setting a TT) so that all, also low carb meals, get a proper iob boost asap.
>
> - Only hi carb meals will quickly progress into a +10 mg/dl per 5 minutes rise. The TT kicking automatically in, then, produces two highly relevant benefits:
>
> 1) the next SMBs (driven still by bgAccel_ISF, or already by pp_ISF) get extra boost, because insulinReq will be oriented towards a much lower target , … and
>
> 2) the "dynamic iobTH" is automatically elevated, exactly in the time window where it counts. This means, SMBs are shut off later, and on average you get a bit more iob for high carb meals. ((Sorry, this is a bit pre-mature to discuss here. More see section 6.1.3)).
>
> This Automation can bring a gradual improvement, but will not be essential to have. You could just not worry about EatingSoonTTs and related Automations, and leave eventual implementation of this "refinement" to a later round of fine-tuning things.
>
> Also, note that this automatic setting of an even TT will mean
>
> - No other Automations that you might have on bord, and which require "no TT set" as a condition, will be able to run ("is shut out")

- Any random bg jump will, for the duration of this Automation, get you aggressive SMBs. To prevent that you could “tune” the jump size, or set a time window (condition = likely meal time of day) for the Automation. See also the compression low example in Case study 5.3

2.6 Other settings in AAPS/Preferences for autoISF FCL

Make sure you start your migration to FCL with a **solid profile** that worked fairly OK also without a bunch of tricky Automations, and without dynamicISF (which, both, unfortunately, too often are employed to counter-act principle problems with profiles) (or even with technical loop functions, like leaking pods, see section 1).

Indicators for a solid profile in your HCL:

- Not too short DIA for your fast insulin
- Basal keeps your bg steady in open loop testing
- Meal Management in Hybrid Closed Loop is satisfyingly established, and can serve as a blueprint for your FCL set-up.
- Most important: **ISFs** experimentally proven in relevant times of day, and “working fine” also in Hybrid Closed Loop, and allowed **SMB sizes** opened **up to 120 min** basal.

  One frequently observed “burden” loopers bring with them is that they could *not* operate with 120 minute SMB settings in AAPS Master HCL because they did their ISF tuning wrong. Their lower settings on allowed SMB sizes covered up their principal problem, which now is bound to come up and hurt.

See also beginning of section 4.

The following is *not* a list of *all* settings in AAPS / Preferences. We just like to bring up some settings that may not be fully understood, or might interfere “behind your back” with what you try to do.

1. In AAPS Preferences, go all the way down in all sub-pages of: Open APS SMB / autoISF settings/SMB delivery settings: Enable alternative activation of SMB depending on active target: **ON**

   We point to this first, because setting an **odd glucose target** in the top right TT field of your AAPS home screen will be a super convenient **“emergency brake”** for you, in your tuning process, to shield yourself against a FCL “going wild” with more SMBs.

   If you implemented your iobTH (section 2.4) well, that iobTH feature serves as a principle, automated, first line of defense against hyper-aggressive SMB fire from your FCL.

The usefulness of the additional, odd target “emergency brake” will, in the long run, lie more in preventing FCL over-reaction to bg bumps that are unrelated to a major meal (section 5.1).

2. Use Autosens: Should be **OFF**, see the pop-up warning that the feature does not make good sense when entering no carbs. (If for curiosity you want it ON, do so only temporarily and with very narrow Autosens min and max settings, like 0.9 – 1.1).

   Also, do not use Autotune. Enable SMB **always.** (I think for iAPS users we need add: .. and switch off dynamic ISF, dynamic CR, and sigmoid). You may need a look into your CGM whether or how it allows to do SMB always.

3. How frequently SMBs: **3 min**, or for Libre 3 (1 minute): 1 min

4. High TT raises sens: **ON**

5. Low TT lowers sensitivity: **ON**

6. Half basal exercise target: Put in 180 as a placeholder; you will set this later in your initial tuning when you get to tune for your favorite kinds of exercise (section 6.1.3). Lower numbers have stronger effects of reducing basal and weakening ISF for exercise.

7. Activity modifies sensitivity: **OFF** until you get to section 6.6

8. Advanced settings/Always short avg delta: **OFF** (- unless you need it ON because of jittery CGM. Yes, smoothing can reduce problems, but at the cost of losing time for recognizing true bg movements, as well). Same related to your smoothing selection in AAPS **Configuration builder** / Smoothing: **No smoothing** is the preferred solution there if/when/as long as you have an excellent CGM. Next preferred would be Average smoothing. If single values tend to hop around and cause too big SMBs in your case, you may need to try Exponential smoothing, which gives the nicest bg curves but “iron out” the early indication of a rise, which is so important in a no-user-bolus FCL.

   Note re. Libre3 (1 minute): The author has no experience at all with this sensor. Please stay connected with other users to find which recommendations apply. (1 minute CGMs might actually require smoothing; however, there it probably comes with much less of a “time lost” penalty…).

9. Back to /Preferences / Open APS SMB / Advanced setting: For both safety multipliers go **higher**, probably double, the setting as in the dialogue box recommended (for AAPS HCL), so your FCL loop will be able to do up to 500% TBR in the future.

10. autoISF settings: **Do not activate or change settings until you start** section 4.

11. Percentage of maxIOB above which SMBs disabled: Put in the number determined in section 2.4 for your iobTH%. or **50** as a placeholder before you get to that section.

Next steps

To define a reasonable figure for iobTH% and to make a couple of settings for the SMB delivery settings were your preparatory tuning tasks in this section 2.

We suggest to review the basic description of autoISF by ga-zelle referred to in section 3, before activating your autoISF for FCL meal management (section 4).