2. General Settings for Full Closed Loop

V.4.1

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

No medical advice
research paper

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 wi*ll 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 (e.g. with a “jumpy” CGM!), 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, and some of it 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*).

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.

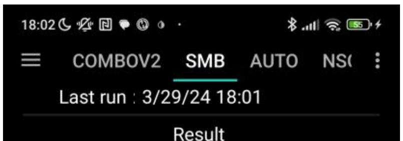
- In low carb meals, that iobTH level should not be reached => the autoISF parameters („weights") need to be tuned carefully, so SMB sizes are *not always* huge and bounce against the iobTH restriction, but show different behavior for different meals

- 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



18:02

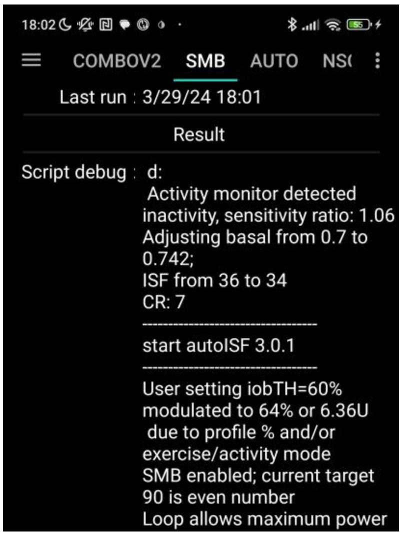
COMBOV2 SMB AUTO NS(

Last run : 3/29/24 18:01

Result

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

  The iobTH can also get temporarily modulated.
  The example on the right shows it ***elevated*** in a case of detected ***in*-activity**:
  In the exercise mode, it will automatically become radically *reduced* to limit iob during sports (see section 6, “dynamic iobTH”).



Last run : 3/29/24 18:01

Result

Script debug : d:
Activity monitor detected inactivity, sensitivity ratio: 1.06
Adjusting basal from 0.7 to 0.742;
ISF from 36 to 34
CR: 7
----
start autoISF 3.0.1

User setting iobTH=60%
modulated to 64% or 6.36U
due to profile % and/or exercise/activity mode
SMB enabled; current target 90 is even number
Loop allows maximum power

- 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 automatic-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 (if in mmol, odd decimal), you run in "AMA" mode, still 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. You could skip this section for now (and revisit later in case you miss any performance via the totally hands-off way). Or you could also just jump into the "preferred solution" presented on the next page…

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 actually set the **target glucose** values **in your profile** accordingly, *e.g. a 11-15h profile target of 76 mg/dl* if you "almost always" start a lunch between 11:45 and 14:30h. (*Should you skip eating, nothing bad would happen. Only if you do exercise instead, you likely will need a snack*).
- Note that you can opt for "turbo charging" your loop a bit with lowered ISF via making use of features as discussed in section 2.6 at points 5. and 7.
- *If you have rather irregular habits*, it might be worthwhile to either
    - **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
    - do this via an **Automation** (see next page) - or
    - just forget about it. Setting an EatingSoonTT would often have only a minor effect, as:your meals are usually spaced by a couple of hours => glucose should not be elevated as you approach the next meal. (Or, if glucose is still on its way down, it comes with insulin on bord and supports momenbtum for acceleration after the next meal start. …

Speaking about acceleration: In autoISF FCL, first SMBs driven by bgAccel_ISF are decisive. They get cut in size, in case bg sits below target (formula see in section 4.2.4). Hence, a low temporary (even-numbered) bg target secures earliest-possible ramping up of iob after meal starts. There is a *fully automatic* way to integrate this aspect:

Preferred solution with respect to Eating SoonTT

My preferred (fully automatic) solution is to ***not*** set an EatingSoonTT ahead of the meal (at my normal, even-numbered, profile target). But, as a refinement, I use an Automation:

I have my loop ***automatically set*** a low (even-numbered) TT around the time when it gives first SMBs, i.e. at certain bg delta. Two reasons:

1. "to orient the calculated insulinRequired towards a more aggressive target"
2. to get full bgAccel_ISF (rather than, @ glucose < target, half, see "cap_weight" in formula for bgAccel_ISF @ https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf , quoted also in beginning of section 4.2.4)

The **Automation** must look like: CONDITIONS: likely meal time of day + 1st sign of a beginning meal (e.g. bg delta > 10 mg/dl) + iob under (?) U (to kick in only in initial phase) + no TT running => ACTION: set TT=74 mg/dl for ~30 minutes

Discussion **): This will often **not** influence the first one or two SMBs (which may already come at acceleration detection *before* a + 10 is seen). But it 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:**
    - the next SMBs (driven still by bgAccel_ISF, or already by pp_ISF) get extra boost . . (see points 1. / 2. discussed above, under "Refinement:"), … and
    - 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)).

In conclusion, this Automation can bring a gradual improvement regarding the size of e.g. the third and the fourth SMB given after meal start.
But this Automation is not 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.

- Benefits should also be weighed against **potential draw backs** =
    - Any random bg jump will get you aggressive SMBs, whenever the Conditions you defined are given (and there is any insulin required). To prevent that, carefully "tune" jump size, or set a time window etc for this Automation!
    - No other Automations that you might have on bord, and which require "no TT set" as a condition, will be able to run. (It seems unlikely that you really would like *another* Automation kick in, in that important post-meal phase. But watch what may happen 24/7, considering all your Automations!)

**) Remark: This is a bit pre-mature here. You may need to come back after having worked through section 4.

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.
    - Note that by generally going without any carb inputs, we "give up" the **opportunity** we may have utilized in Hybrid Closed Loop **to differentiate** allowed SMB sizes "after meals" (i.e. with, then, cob>0) vs. maybe a milder reaction in other times using a smaller selected setting for UAM minutes of basal/SMB.

      With autoISF, we just pro forma use the max setting, and tune further using the toolbox of SMB_range_extention and the various ...ISF_weights, while observing our glucose curves (section 4.1.2).

      Section 5 will elaborate on how, in FCL, you can differentiate "aggressiveness" even further, if needed in special situations.

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, enable: SMB, SMB with high TT, and SMB always. Also enable UAM, of course. Then 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**

    This option, to switch SMB off at any odd-numbered glucose target, makes the restriction, to shut out SMBs at elevated target, unnecessary. It is the reason why "Enable SMB with high TT" is set to **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. See also remark at point 5.

> If for curiosity you want Autosens ON, do so only temporarily and with very narrow Autosens min and max settings, like 0.9 – 1.1. The AAPS main screen shows you the different "opinions" about how ISF now should be modulated by Autosens or by autoISF, and the SMB tab would show you how both would interact if both active.

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**

This needs to be ON, so via a set half-basal exercise target, loop aggressiveness (profile basal, ISF used) can automatically get significantly ***reduced***. Effect is the stronger the lower this parameter is set (in AAPS/Preferences default is 160 mg/dl*), and the higher the TT.

- Details see on Exercise Mode page of the autoISF Quick guide: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf.
- *If* set *OFF*, an elevated TT has only a "lamer" effect (ISF is then *not* "softened"(raised), loop is just not aiming for the usual profile target with the corrections).

Likewise, you can opt to use this "half basal…" feature to ***increase*** loop aggressiveness (beyond the basic effect the set low bg target you are shooting for, triggers) in case you switch "low_TT_reduces_ sensitivity" ON in /Preferences:

5. Low TT lowers sensitivity: Is **default OFF**. This is to make sure your loop does not go into turbo aggressiveness before you 1) have halfway decent settings tuned in and 2) know more about how your loop operates – or *could* operate "with more bite". So:

- consult https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf.
- ..and **switch** to **ON when ready**!

6. In any case, leave on **OFF** the other two, that "work the other way around" in that they would make detected sensitivity raise or lower the bg **target**.

For instance, if you set "resistance lowers target", there is very limited experience with how it interferes with autoISF. (It might for instance result in an unintentional switch to a calculated odd bg target, and disable SMBs).
Pre-autoISF, that setting probably was useful to you when also using Autosens (point 2.).

7. Half basal exercise target: Was already discussed at point 4. Leave it at default **160 mg/dl** 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 (you set for your half-basal ex. target in AAPS/Preferences) have stronger effects

   - of reducing basal, and elevating ISF (=sharply will reduce loop aggressiveness), as desirable for exercise (and with your selected TT as a fine-tuning tool).
   - and, if desired –see point 5.- also the opposite effects, to increase loop aggressiveness with lowered ISFs at low TTs, as desireable maybe in an “EatingSoonTT” context.

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

   - After giving up Autosens, you might like this feature on board asap. So, you could actually start on section 6.6, and have this setting on **ON**, already before going into section 4.
   - Be aware, though, that any set temporary glucose target will interrupt your Activity Monitor. It is generally a good idea to operate with as few Automations as possible when you enter your autoISF tuning phase. (Tipp: Just try to avoid stuff you needed Automations for, while you get your basics right).

9. 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. G7: In contrast to G6, the G7 transmitters do not give smooth(ened) values. Hence you probably should use exponential smoothing in AAPS. (Stay connected with other G7 users in FCL to find out more).

   Note re. Libre3 (1 minute): 1 minute CGMs require sophisticated smoothing; as of Dec.2024, autoISF dev is still optimizing the math so smoothing yields results while minimizing the general “time lost” penalty (as is noticeably hurtful when smoothing a 5-minute system).

   The author has no experience at all with this sensor. Please stay connected e.g. via https://discord.gg/tamvhh57Xs

10. 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.

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

12. 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.

13. Protect settings in AAPS/Preferences with a (short) **password**, just to secure that scrolling through the many settings will never make un-intended changes.

14. Eliminate the buttons at the bottom of the AAPS home screen as soon as you can (in AAPS Preferences/Overview/Buttons: all OFF).

    This can strongly enhance personal safety. (Adult: if temp. losing control of phone; kid: if "playing around" on the phone screen).

    Note: The insulin button would allow to start a dangerous insulin shot.
    PS: Pressing on one of the top three buttons would be much less critical: It might require more inputs (like also a duration) to do anything at all. And it would only tweak insulin sensitivity in whatever the loop will do (if anything) in upcoming 5-minute segments.

15. Analyze, in your data, problems you sometimes encounter, or nearly encountered. Then **define alarms** that make your system safer and better. Sophisticated options for alarms are offered by xDrip, and notably when thinking of Automations (with an alarm as Action) in AAPS.

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

- https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf
- and get oriented, directly there in Github, or via section 3, regarding other support materials available from developers

before activating your autoISF for FCL meal management (section 4).