1. Pre-requisites for Full Closed Loop V 2.3

Please note that with autoISF 3.0 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

1.1 Well-tuned hybrid closed loop

It is advisable to first establish a well-tuned hybrid closed loop before considering the transition to FCL. There are two important reasons for this:

- The UAM full closed loop requires a highly personalized (individual) tuning of settings, so the loop will give insulin mimicking YOUR successful hybrid closed loop mode.
- The UAM full closed loop comes with new parameters to be set and tuned. It would be problematic to set and tune several new parameters before the basics were tuned “right”. Errors could easily be balanced with counter-errors. This can work in single scenarios, but would create a highly unstable system, hard to re-calibrate better later.

1.2 Fast insulin (Lyumjev, Fiasp, Apidra?)

If the user does not bolus for meals, clearly a very fast insulin is needed so, upon realization of a starting meal-related glucose rise, the loop has any chance to eventually keep glucose in range (by common definition, under 180 mg/dl (10 mmol/l))

A modelling study (details see https://androidaps.readthedocs.io/en/latest/Usage/FullClosedLoop.html#fast-insulin-lyumjev-fiasp) can show in quantitative terms that **faster insulins**

- will result in significantly **lower** glucose **peaks** than slower insulins
- **tolerate** a couple of minutes **delayed** first meal bolus while not incurring unacceptable height of peaks
- minimize the effect on glucose peak from **different** carb load (**meal sizes**).

In conclusion, do not attempt FCL with other insulin than Lyumjev® or Fiasp®, unless, maybe, if you are on a very moderate to low carb diet. (According to case study 1.2, Apidra® might work, too, but Humalog® would not work well).

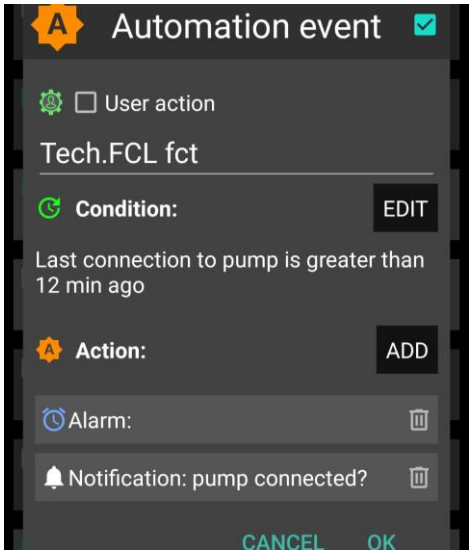
1.3 Reliable insulin delivery from the used pump/cannula/insulin system

Good Tolerance of Lyumjev (or Fiasp): Occlusions threaten the function of the full closed loop.

It is very important to have an eye on the time a **cannula (or pod)** is in use (many find **48 hrs** to be the **limit**), and whether hard-to-explain glucose rises happen at ever increasing „fake“ iob (even before a 48 hr routine replacement). (See case study 1.1: You easily lose 25% TIR that day)
It is absolutely contra-indicated to attempt FCL coming from leaking pods and associated erratic sensitivity swings that may or may not have been somewhat controlled and tolerable by dynamicISF or other measures when you were Hybrid Closed Looping,

Stable pump connection

In FCL you absolutely rely on your pump delivering, without any further delay, the much needed insulin, after any meal start.
Hence it is absolutely essential not to “miss” any problems from a lost Bluetooth connection.
An Automation similar to the one pictured here → should help recognizing eventual problems.



See also case study 1.4

1.4 Excellent CGM

You do not give a meal size-related bolus any longer. That leaves all insulination jobs to the algorithm! As glucose values are the very basis for this, please **inform yourself well about** how **your CGM** 1) principally performs 2) whether and how this may depend on data flow and intermediate apps you use 3) specifically, how and where any smoothing is done, and what this might imply for the ISF boosting method you will be using See for instance here:
https://androidaps.readthedocs.io/en/latest/Usage/Smoothing-Blood-Glucose-Data-in-xDrip.html

Around meals, a stable Bluetooth connectivity is absolutely essential, too, so CGM, loop, and pump can do their job without losing more valuable time (see case study 1.4).

Then, but even more importantly in *all other* day and night *times*, the CGM should not produce any artefacts (jumpy values; see case study 1.3) that the loop could **misinterpret** as sign of a starting meal. Note that also calibrations could produce jumps.

autoISF has also a couple of in-built checks on the quality of the recent CGM values. Hence, a CGM with more scatter will make the loop lose more time, and lead to higher peaks and lower %TIR.
So, if you are unhappy with a slow reaction of your loop it could be because the loop is unhappy with your CGM. Consult the detail info given (at the time) in your SMB tab, or look it up later in the logfiles (using the emulator eventually).

The best proven way to stay out of trouble currently is to use Dexcom G5 or **G6**, and to ensure via **overlapping** right and left arm sensor and transmitter utilization for always good quality values that can be used by the Full Closed Loop.
Other ways (making use also of day-1 sensor values, G7, Libre2/3…) are possible, but come with a lot of monitoring effort (best via watch) and occasional time-outs for the FCL.

One safety feature in autoISF is a **blockage of SMB delivery whenever delta bg** (within the last two 5 minute values) is **higher than 30%** *of that bg*. *So from 74 mg/dl, a jump to 97 (+23) or more would not receive SMB "response", or from 100 mg/dl to 131 mg/dl (+31) would neither.*

> Check in your (HCL or FCL) data whether at meals or sweet drinks with rapid absorbing carbs you could run into the problem that jumps are "too high" and much needed insulin will be blocked (only come via very much smaller portions: *e.g.400%TBR @ 0.6 U/h => 0.2 U in 5 minutes, instead of one ~3 U SMB. The difference of 2.8 U missed translates @ ISF~ 40 mg/dl/U into up to + 112 mg/dl higher bg peak! It will not become quite that bad, because the loop will catch up to the insulinRequired with it's next couple of decisions*).
>
> Instead searching in old data, you can also just have an eye on instances where you think a first SMB was due, but blocked. Confirm that by looking in the SMB tab, and think about a solution that would not require changing the 30% safety limit in the code. *For instance, not drinking so much juice rapidly around meal start could be a likely "behavioral" correction to get rid of the problem.*

This blockage (no SMBs) would likely last only 5 minutes (and go probably unnoticed - in a suggested improved User Interface it would be indicated for about 5 minutes via a dotted violet full closed loop logo/button on the AAPS home screen). However, not only would you lose 5 valueable minutes to get your iob substantially elevated; all following deltas are likely much smaller, and, as a consequence, you will miss some of the boost sought from bgAccel_ISF if the >30% delta was in fact (largely) due to carb absorption.

This example also underscores that the CGM in use cannot be allowed random scatter that leaves no reasonable room for safe detection of (smaller and) bigger “truly carb related” deltas

If or when (like: first half day of a new sensor) you are not sure about sufficient CGM performance you might develop for yourself an Automation with User action ticked (along the lines as used for other purposes in section 5.2.2.3). It would “ask you” before giving a SMB whether you really want it delivered. That way you can a) have a look on your glucose curve b) on the delta and acc data underneath the TT field of your AAPS home screen c) think about what sense a SMB now makes with respect to your last meal, and the carbs to be still absorbed. Ultimately, you could also d) consult some of the detailed info given (every 5 minutes) in your SMB tab.

Such User action Automations need not be active at all times, but if you have it for your first half day of a new G6 sensor for instance, you could activate that Automation from your list of Automations; after the values have settled in, you can disactivate (“shelve”) it again.

For a brief period, and if you are tech savvy, another way to deal with uncertainty about CGM would be to employ the emulator method as presented in section 11: Run a “too mildly” tuned FCL, and in parallel run a “what-if” with your more aggressive settings that you really would like to use once you are certain about your CGM.

However, I found it easiest to lay a solid groundwork by using 1 Anubis, and 2 overlapping G6 to get rid of most problems that I saw (and keep seeing, on the worse sensor of the two running for some days often in parallel) in my data.

With a sensible iobTH defined, and your standard alarms for going towards a hypo not silenced, the worst consequence from any automatically “over-treated” glucose jump should be that you need an unplanned snack for the balance of “missing” carbs.

Also FS Libre 3 is useable in the autoISF FCL context. Observe info in the general section about autoISF (https://github.com/ga-zelle/autoISF) regarding established ways to use with autoISF.

As of Dec.2023, there is still development work going around what to best make of the 1 minute values.

1.5 Meal-related limitations

Setting up a full closed loop is relatively easy for people whose diet does not consist **mainly** of components with rapid high effect on blood glucose (more see https://androidaps.readthedocs.io/en/latest/Usage/FullClosedLoop.html#meal-related-limitations)

Meals do not have to be low on carb (provided you use a fast insulin for your FCL)
Fat or protein rich diets, or slow digestion/gastroparesis, make things easier rather than harder for the full closed loop because late carbs nicely cover for inevitable “tails” of late action from SMBs needed around peak time.

Erratic consumption of snacks with fast resorbing carbs can be a problem.

> In autoISF you can reduce this problem to some extent via one or two keystrokes from your AAPS home screen. While certainly being a deviation from the FCL idea(l), this would be one of the exceptional situations where you better do a quick “nudging” step from your “FCL cockpit”. Details see in section 5.2.1 and 5.3.3.1 (4)

1.6 Lifestyle-related limitations

Technically stable system

Full closed looping requires a 24/7 technically stable system, especially regarding

- reliable CGM signals
- Bluetooth stability with the pump (see case study 1.4)
- keeping your phone in sufficient proximity at all times
- avoiding (or at least early recognition of) occlusion.

This requires a habit (or, unlikely, permanent attention to details) like keeping all components well charged and in close proximity; making cannula (or pod) changes always early enough to lower the risk of occlusion (see case study 1.1); having always potentially needed parts with you.

Depending on your system, your experience with it, but also on your acceptance and general lifestyle, these aspects may or may not limit you.

Preparing for exercise

To prepare for exercise (sports, heavy work), the normal protocol with a pump or hybrid closed loop is to take actions that reduce insulin on board prior to exercise
With your full closed loop, the algorithm is tuned to detect meals and to give you insulin to counter glucose rises automatically. Setting a high temp. target and lower %profile right away (effective already around meal start) could be a problem.

Unusual activity levels therefore likely require **disciplined preparation** (especially **if you want to keep the need to snack during sports low**)

In autoISF you can reduce this problem to some extent via two or three keystrokes on your AAPS home screen. While certainly being a deviation from the FCL idea(l), this would be one of the exceptional situations where you better “flick a lever” from your “FCL cockpit” to

Extra hurdles to establish FCL for kids

To establish and maintain a FCL for kids brings about some extra challenges if:

- Lyumjev is not available or well tolerated
- Hourly basal rate is very low, providing a poor basis for big SMBs
- Diet is rich in sweet components. With the typical low blood volume of a small body, strong tendency towards very high bg spikes!
- Going through marked changes of insulin sensitivity or of circadian pattern makes it difficult to keep the FCL appropriately tuned.

  This problem is about the same in Hybrid Closed Looping. However, now you might expect miracles from the FCL. This is not going to happen. You still should try to set appropriate temp. changed profiles, that serve also as a basis for your autoISF FCL.
- Discipline is poor regarding keeping Bluetooth connectivity and infusion sites perfectly running
- Between kid and supervising parent it must be guaranteed, especially in the initial weeks, that an eye is kept on whether the FCL is working about as to be expected.

More see section 7.

1.7 Time required for setting-up

Lastly, before enjoying a functioning full closed loop you need to have a period of a some weeks with some free time and „free head“ for set-up –**.** Can you get, in the time you are willing to invest, to a result that you consider good-enough is really the question. Depending on your „habits“, and which – if any - compromises (like doing cannula/pod changes more often, never starting meals when bg sits high …) are you willing to make (and everyday able to stick to), for the ease of not having to deal with assessing meals and bolussing for them?

While setting up your personal FCL using autoISF is a substantial project, there is no need to implement it fully in one step. There is nothing wrong to go in your well running Hybrid Closed Loop mostly, while switching to FCL only for dinners, for instance, or only for

weekend lunches, as a start. Once you found feasible settings, you can expand to other meal times, and lastly towards figuring out your best strategies for challenges outside of meal windows, as we shall discuss in sections 5. and 6.

There are alternatives to using autoISF for FCL, as well. See section 13. for more info.

Notably 13.1 FCL using AAPS Master and Automations could be a much easier and more error-tolerant way of stepping into FCL. In a clinical study with 16 participants about 80% TIR was achieved without much tuning effort (source: see in section 13.1).

To close the circle to where we had started (section 1.1): A very time consuming pre-requisite might actually be to first sort out your Hybrid Closed Loop, so your profile parameters are set „right“, and your “old” data really can serve as a blueprint for what, now, you would like *your loop* to do in FCL mode

Note that if you had used dynamic parameters or special Automations („loops inside the loop“) this might have balanced some principal errors, but leaves you now without a good starting point as you must get rid of these over-patches (see also warnings at start of section 4)..

You will see also success stories of loopers who just jump into using more powerful tools, in kind of a trial and error mode, and frequently add the latest add-on, or self-constructed patch (often in form of an Automation) to counter-balance problems.
So, yes, you can also continue in that spirit. Resulting solutions may be good-enough. But they tend to be unstable and not well-understood. That is a poor basis for managing arising problems (-> fine tuning), and for adjusting to special situations (-> which setting to temporarily change). But it certainly is an alternative avenue for the impatient, less analytically, and more adventurous inclined.
In any case, PLEASE always observe the safety settings/instructions coming with the DIY dev- variant of software you select.

One key safety measure every AAPS user going towards FCL should have in place is to set an **iob threshold** (iobTH; size a bit below what you used as a bolus for bigger meals in HCL) above which no more SMBs can be given by your FCL.
This is an integrated feature of autoISF, from 3.0 version onwards (see section 2.4).