13. Other Avenues to FCL

V 2.0

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

13.1 Full Closed Loop using AAPS Master and Automations

AndroidAPS 3.0 was (Sep.2023) the first DIY system to launch Full Closed Looping as an option to manage T1D, if a described set of pre-requisites apply.

> Key pre-requisites were described in https://androidaps.readthedocs.io/en/latest/Usage/FullClosedLoop.html , and are sketched also in section 1, with case studies 1.1 – 1.4 underscoring the importance.
>
> You may (not) have noticed: There was no big „marketing fuzz“ made around that FCL option. Seeing how many AAPS users struggle with even getting their basal, ISF and SMB settings right, it would be foolish to allure everybody to a supposedly very easy way of looping. True, it can be easy. But only after doing a personalized set-up project. Setting up is easier than what autoISF and the methods we get to in section 13.3. demand, but still a project. It also requires a well mastered hybrid closed loop, to begin with.

With attention to the pre-requisites, and avoiding extreme high carb diets, many (mostly: adult) users achieve satisfactory %TIR after supplementing AAPS Master with personalized Automations that attempt to strongly elevate iob upon recognition of a meal-related bg rise.

See also Case Studies, and the randomized cross-over study involving AAPS FCL: PubMed First Use of Open-Source Automated Insulin Delivery AndroidAPS in Full Closed-Loop Scenario: Pancreas4ALL Randomized Pilot Study;

This method is **highly recommended for an entry into FCL for those who do not have the interest, or lack the time, to deal with the very much more sophisticated and demanding other routes** towards FCL, like autoISF, or also like the methods briefly presented below in section 13.3 .

Note that using the autoISF dev version of AAPS 3.2 (with ”Enable ISF adaptation..” OFF) can be a good idea, to make use of features like SMB_ range_extention and SMB_delivery_ratios > 0.5 for boosted SMB sizes, also when using just Automations for FCL.

13.2 FCL using dynamicISF with AAPS or with iAPS

As opposed to

- autoISF, with it's bgAccel_ISF component , or to….
- AAPS Master, with Automations strengthening ISF at meal-related bg rises …

dynamicISF was **not** designed to help boost SMBs asap after an omitted user bolus.

Rather (as the name also suggests) it was designed to be used in hybrid closed looping to make ISF react more dynamic to suspected swings in insulin sensitivity (which shows in bg values, and in TDD trends). It does a similar job like Autosens, but can be much more amplified (by the users tuning their dynamicISF adjustment factor (%)).

When using a fast insulin (and when some other pre-requisites discussed in section 1 are in place, too), the dynamicISF method can be applied also to Full Closed Looping. (See Case Study 13.X; not available by time of publication => this is a call for a dynISF FCL user to provide a case study that contains a 1 week 24h scatter plot as well as one analyzed meal where we can see when and how dynISF helped build iob, after not having bolussed).

It will have a principal timing-disadvantage because responses are more tied to high bg values than to acceleration (in autoISF) or to delta (in the Automations route to FCL).

On the other hand, people who 1) do have strong sensitivity swings and 2) cannot pro-actively deal with those (e.g. by making profile switches) might be satisfied with the automatic (although a bit late) adjustments that dynamicISF automatically will provide.

dynamicISF therefore could be characterized, in the FCL context, as a potential solution to a rather care-free approach for those who do not seek best-possible performance (or who take other measures, like low carb diet, to still reach pretty acceptable performance in FCL mode).

Contact (caution, both not focussed on FCL:)

AAPS / search term dynamicISF in: https://discord.gg/DfvK5HnxXu

iAPS / section dynamic-isf-cr: https://discord.gg/gGKXW5uX3m

13.3 Methods involving simple Meal Announcement that might be stretched into a Full Closed Loop

13.3.1 Boost

All of the additional code outside of the standard SMB calculation requires a daily time period („Boost window“) to be specified within which it is active.

A variation of dynamicISF is used in which also predicted bg will be considered in varying degrees (40…75%) to mimic the effects of higher insulin sensitivity at lower glucose levels.

When using Boost without carb inputs (permanent cob=0) a special **boosting of SMBs** is provided when an **initial bg rise** is detected with a meal:
delta, short_avgDelta and long_avgDelta are used to trigger an early bolus (assuming IOB is below a user defined amount).

> This procedure goes in the direction of the bgAccel_ISF route discussed for autoISF (section 4.1). If used with an excellent CGM, autoISF acceleration detection should be a bit earlier, and boosting can be made much stronger in autoISF

For safety, the user sets a value of 2.5% (up to 5%) of TDD for the max. Boost Bolus (Boost Bolus Cap).

For stronger boost**,** the default AAPS 50% SMB_delivery_rate can be overwritten with a higher insulin percentage determined by the user. The SMB_delivery_ratio is called „Boost insulin required percent**“** here, and suggested not to go over 75%. The % can be defined variable with bg value (like also in autoISF).

The Boost function automatically shuts off as soon as delta and the average deltas are aligned, i.e. when the accelerated rise goes over into a constant rise (compare pp_ISF in autoISF). However, the boost function is only „dormant“ if the boost window lasts longer for more meal-related accelerations.

Additional functions are a step-count modified dynamic_ISF, inactivity detection etc

A couple of safety feature are integrated. The user can define an iob limit for boosts, here called UAM Boost max IOB **.** In Preferences/Treatments There is also a user adjustable Low Glucose Suspend threshold. This allows the user to set a value higher than the system would normally use, such that when predictions drop below this level (65…100), a zero TBR is set.

More info: https://discord.gg/nYC4T9PgCR ; https://github.com/tim2000s/no-bolus-dev ; https://github.com/tim2000s/Boost-master-v3/blob/master/README.md

Contact: Tim Street @ diabettech.com

13.3.2 AIMI

AIMI has a single goal: to minimize the decisions necessary to maintain the target range, simplify the composition of the profile for the user or doctor accompanying the patient, and allow the patient to live normally without having to count carbohydrates or even without signifying physical activity (especially for brisk walking).

A key component of AIMI concept is to give a **small pre-bolus before each meal** ("Meal Announcement" that also provides some pos. iob).

• A **simplified profile** composition (neutral ISF around 100, DIA 9, target 90-90, a single value for basal, a ratio that is not used in AIMI, so not important) For a first basal estimate, you can use the TDD / weight ratio.

• Some variables in preferences that are important (AIMI_UAM which allows AIMI to make decisions, Max SMB size which is the highest value for an SMB, B30_duration (which is the duration during which the **basal will be forced after a manual bolus**), B30_upperBG and B30_Upperdelta (these last two variables represent the conditions for replacing smb with a consistent TBR depending on the delta)

• The basal profile is calculated by a polynomial equation.

• The ISF is calculated from the TDD **(dynamicISF)** and is adjusted based on the evolution of TIR throughout the day and the **detection of physical activity.**

• The detection of glycemic rise (or the opposite situation) is also calculated by a polynomial equation, which will influence the change of target but also the replacement of SMB by a TBR between 100% and 500% or by an SMB of the same equivalence.

• SMB calculation is done in several ways specific to AIMI depending on the evolution of the delta and IOB, with a distribution that can be done in three parts depending on the conditions.

Example scenario of execution, on almost all existing variants:

1. Make a "standard" manual bolus. I usually do 1.5U or 2U with luymjev
2. Just after this bolus, AIMI will force the 500% TBR for a duration defined by the user. The observation made is that the absorption of insulin such as humalog for example is accelerated and will strongly limit the first wave.
3. Depending on the options chosen, it is possible to receive an SMB of the initial manual bolus size after the duration of the 500% TBR
4. Then the rest of the calculations will depend on the result of a polynomial equation and its evolution.
5. A few hours later, if the patient decides to take a walk to go shopping, or other activities requiring movement, the phone sensor will send information on the number **of steps taken.** This will result in a reduction of the profile to about 60%. The return of the profile to normal

will be done in stages, in the first half hour following the activity, the profile will be restored to about 80%.

The AIMI developer has been working on using machine learning (using tensorflow lite).

More info https://discord.gg/7ehczAfZ

The developer hasn't kept the code public. AIMI can only be obtained as an apk via joining their WhatsApp group. Given the very high number of changes happening in this AAPS variant, it is probably deemed important to keep it in a tight sub-community. But, caution: This can be seen as violation of the Open Source principle

Contact: Mathieu Tellier @ AndroidAPS User; FB/Twitter: @MTR93600

13.3.3 EatingNow (EN)

This version of AAPS has evolved over time using elements from AIMI and Boost. It includes a modified dynamicISF which moves ISF modulation in the direction as pioneered by autoISF, and also uses Automations for FCL.

"Eating Now" (EN) allows user definable SMB's when deltas are sufficient and accelerating. The intent of this plugin is the same, to deliver insulin earlier using mostly the AAPS predictions.

As all other variants for FCL, also EatingNow requires to set glucose TT occasionally, to nudge the loop in certain direction, notably to announce and be prepared for activity.

Operating Modes provide 3 levels of „aggressiveness" in 3 time windows:

- Master AAPS w/up to 120 min basal per SMB when EN is off (usually set for night-time).
- EN (usually set for daytime) is when the modified algorithm is capable of boosting ISF and insulin delivery. At BG level rises within the EN Window, a „UAM maxBolus" is given as a first SMB. Recommended Setting: 1h current basal in units (max allowed: 2).
- ENW: A further boosted SMB will be issued in this ENW time window (e.g. for breakfast, or generally for the first meal of a day, after fasting, with higher insulin need). Upon detection of rising glucose, a SMB called Breakfast COB maxBolus is given by the loop. Recommended Setting: 25% of average breakfast total units

EN uses the dynamicISF concept, modified to making ISF stronger as and eventualBG predictions. Increase.

Specifically for the ENW (usually: breakfast window), an additional boost factor called Breakfast ISF/CR Percentage (e.g. 125 or 150%) can be applied

A setting „TIRS“ proivides a very simple version of autoISF (dura_ISF) and sharpens ISF temporarily when bg „seems stuck“ above a certain value.

Autosens sensitivityRatio will be overridden by EN sensitivity options.

SMB delivery ratio for insulinReq. Is set to 65% for when EN is disabled (overnight, usually). It is recommended to set maxSMBBasalMinutes and maxUAMSMBBasalMinutes to 30 minutes max as these will be used when EN is OFF or in SLEEP mode. Falling back on OAPS SMB settings is considered as the safe mode should you experience any issues with sensitivity or EN settings in general
It is set 85% for an active ENW, or 75% when EN is on but ENW not active

Furthermore, SMB optionally can be disabled day/night below defined bg level/s (SMB Disabled)

More info https://discord.gg/XqhnPRChEP (method description in pinned post)

https://github.com/dicko72/AAPS-EatingNow scroll down to README.md
Contact: dicko via Discord channel

13.3.4 Tsunami

The Tsunami loop algorithm analyses blood glucose and insulin activity developments to estimate bolus requirements during meals, without the necessity of carb announcements.

Users must make a **meal announcement via a button** on AAPS main screen. It switches on the main Tsunami algorithm for a finite amount of time.

In between meals (when Tsunami is inactive), users are given the choice between running a weaker version of the Tsunami algorithm (called wave), or falling back to oref1.

A “historic” merit of this method was that it pioneered a BG smoothing algorithm that later became included as a plugin in AAPS.
The insulin models dynamically readjust DIA based on bolus size so that a user-set, fixed DIA value is no longer needed.

For best results, it is recommended to issue a **bolus** at the beginning of a meal to account for the disadvantageous kinetics of subcutaneously administered insulin in a UAM setting.

More info https://discord.gg/veRKcgwVUT GitHub repository: https://github.com/piecycle/tsunami official documentation: https://cdn.discordapp.com/attachments/969948954949189633/972852790739238992/tsunami_guide_3_2.pdf
Contact: nichi#1391 on discord / piecycle on GitHub

13.4 No-Bolus Looping with Carb Entries

Some oref(1) loopers attempting to go full closed loop reported that they do best when they (do not bolus but) give their loop precise carb (and absorption time) information. This:

* announces a meal to follow (so it is not UAM, but might be called full closed looping if the insulin management is left 100% to the loop)
* provides data on cob, and with the glucose and insulin activity info the loop has, it can always calculate how much more carbs are to become absorbed (to the extent the carb-related infos the user put in is correct)
* will display realistic cob info to the user, including cob info looking forward (rather than only calculating carb deviations for the past minutes or hours, and making some coarse assumptions for the upcoming hour). It gives the user better feeling of safety if she/he can see cob info in addition to the available iob info, and insulin activity prediction.

With detailed carb (amounts + absorption times) inputs, the loop has best-possible info to provide „the best expert fit" of insulin activity and carb absorption.

It still rarely can come close to physiological values, because the time-delays inherent in our „artificial pancreas", notably the stretched out DIA, make it difficult still, compared to a real pancreas.

So, carb inputs could help. However,

- only to the extent amounts and time pattern for absorpotion („eCarbs") are correct ((which, every day, is a mission impossible))
- the oref(1) loop still largely „waits for glucose to rise", and there is no significant time advantage from inputting carb info

  Only the ***user***-bolussing *for* ***expected*** carb absorption in hybrid closed loop offers a convincing time advantage (but with associated risks).

- inputs require actually more attention to detail than it is good practice even in AndroidAPS hybrid closed loop, so in that respect a step back, not forward.

Entering **precise** carb information takes away a very large part of the attractivity of full closed looping.

And entering *imprecise* carb info could easy be inferior to not doing *any* carb inputs = to letting the *UAM mode* of oref(1) figure out further carbs that probably come to be absorbed in the next minutes, judging from the pattern of the calculated past *carb deviations* (see section 4.5 and https://openaps.readthedocs.io/en/latest/docs/While%20You%20Wait%20For%20Gear/Understand-determine-basal.html#understanding-the-basic-logic-written-version).

PS: Because that is so, also loopers who do carb inputs get the UAM predictions besides their other predictions, and their algo makes a judgement (every 5 minutes) as to what the best calculation might be for where glucose, underlying „real“ carb absorption, and estimated carb deviation are headed.

13.5 Machine Learning

Involving machine learning (“artificial intelligence”) could help both in the learning/tuning phase, but also in fine adjustments in daily utilization.
The study that was already referenced in section 1 discusses *on page 80* the application of machine learning in some predictions of postprandial glucose response (IEEE Control Systems Magazine, ResearchGate: The Artificial Pancreas and Meal Control. A. El Fathi et al, IEEE Control Systems Magazine Feb.2018 p.67-85.).
So there is already a body of data and evidence. To which extent it lends itself to UAM remains to be researched. For this, a body of data would have to be captured from UAM loopers, and I fear many more data would be required than what could easily be captured in Clarrity® or even in the OPEN project database.

In the DIY universe, a prototype solution was already developed for AIMI (section 13.3.3).

We might see industry come up with a 1st generation solution that will probably be geared to folks with miserable HbA1c and poor carb counting/meal handling, to offer a safe gradual improvement.

A top performing entirely self-learning system might be impossible to design:
For instance, if today you do something entirely different from yesterday (don’t we all want this freedom – even need it? Think about the fasting day following a feasting day…) there are two problems:

- Such systems rely on information from the preceding day, or an average of several preceding days
- The user does not know/learn much about how the system works, what it is calibrated for today, how she/he might intelligently change something for the specific different situation coming up. This seems like the opposite of the FCL solutions we discussed, for instance

self-defined Automations, combined with profile switches for to-be-expected temporary sensitivity shifts. (section 13.1.)

13.6 Dual Hormone Systems

Many see a dual hormone “double full loop” as the ultimate system.

The beauty of this concept would be that the second pump could influence the glucose curve via giving glucagon or an analogue, thus overcoming the strongest limitation our current systems have:
Taking basal away (zero-temping) is only a severly limited course of action against impending hypoglycemias, and therefore, to keep things safe at the back-end of each meal, fighting glucose highs is more limited than we would like to see.

In conclusion, the glucagon component not only helps stay out of hypos. It enables a more aggressive treatment for preventing, or reducing, high glucose values, as well.

While insulin and carbs have complex activity curves stretching over hours, glucagon has a window of physiological activity starting 5-10 minutes after administration, and lasting only 30-40 minutes. Compared to insulin and carbs, that makes it a better component for rapid corrections (without a lengthy “tail” of action).

As glucagon does not per se introduce more calories, but stimulates glucose release from the liver, there should at least be no concern about gaining body weight from eventual roller-coasters the dual loop might send us into. Actually there could be a nice side benefit of helping in body weight control. Also, activity/sports management could become as easy as the meal management became in the UAM step into full closed looping.

It will be interesting to see for which application(s) the dual loop will be developed and launched; as part of a full closed loop with top performance, or as part of even only a hybrid closed loop for problem patients?

It remains to be seen how well such systems work in day-to-day circumstances. And whether “real people” will be able to handle all the involved technology, and use it in ways that truly could justify the substantial extra cost.

The author currently is not really looking forward to become loaded with even more technology, and quite happy with an aggressively tuned full UAM closed loop (…and an occasional nice post-dinner or during- activity snack).

However, the dual hormone path holds enough promise to learn more about it, and to test it some time in the near future.

This is an exciting time to be part of the open source T1D community. Anyone is welcome to contribute ideas, help develop software or instructions how to use. Carefully weigh for yourself what may be your entry point for eventually surmounting the initial hurdles, and **JUST EAT happily ever after**.