

## 2. General Settings for Full Closed Loop

V.16

When in HCL, 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.

### 2.1 SMB Range Extension

(preferences/OpenAPS SMB/autoISF settings/smb delivery settings/smb\_max\_range\_extension)

Full Looping requires bigger SMB sizes. Setting 120 minutes of basal as max. SMB size as enabled in AAPS Master will rarely suffice.

In AAPS Preferences/OpenAPS SMB/autoISF settings/smb delivery settings, set SMB/UAM max range extension (**smb\_max\_range\_extension**) to 2.0. That doubles the allowed max. size, for a start.

Even better, you could determine an estimate for your initial setting using the profile helper in [section 4.6](#) (*if already launched.... Prior, you can proceed as in the following description, I will 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 loop, you want to get at least half of our required meal bolus in 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, because of the max min of basal setting's max 120 min. size limitation.)*

To avoid the requested SMBs reduced in size by safety settings, you need to make further setting adjustments in AAPS/ Preferences:

### 2.2 Max and Min autoISF Ratio

(preferences/OpenAPS SMB/autoISF settings/autoISF\_min and autoISF\_max)

Set **autoISF\_max** = 2.0 (This allows up to doubling of ISF aggressiveness if "requested by the weights" (see [section 4](#)). You can sharpen further, later.

For situations of reduced insulin sensitivity, 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 0.5, or even lower.

## 2.3 SMB Delivery Ratio

(preferences/OpenAPS SMB/autoISF settings/smb delivery settings/smb\_delivery\_ratio)

Increase fixed **smb\_delivery\_ratio** from 0.5 to 0.6. This results in the algorithm generally demanding 20% more insulin (in the calculation it does every 5 minutes). This can be sharpened significantly more (going toward 1.0), later.

*The **smb\_delivery\_ratio** can also be set dynamically (changing with glucose level). This seems a feature geared more towards hybrid closed loop applications.*

If you have an extremely low hourly basal rate, you may have to use a higher range extension along with adjusting settings such as **bgAccel\_ISF\_weight**, as discussed in [section 4.2](#)

## 2.4 Safety Against too Aggressive Settings: iobTH

A safety net is needed because autoISF shoots big SMBs at rising glucose levels, and you do not want to bounce into the ultimate iobMAX safety setting too often.

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

autoISF 3.0 totally changes how iobTH is accessed and modulated. (Before it had been done via an Automation).

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

/OpenAPS\_SMB/autoISF\_settings/Full\_Loop\_settings: iob\_threshold\_percent,

First, check whether your maxIOB is set reasonably, near the max level of iob you had ever needed in your past looping history; *for instance 10 U*.

Then look at your meal spectrum, and what bolus size, and iob level (including SMBs) was useful in high carb meals to control your glucose (*for instance 8 U*).

Then assume you want to approach no more than about 75% of that level by rapid SMB „fire“, after a meal related bg rise is seen (*6 U*).

That gives you the iob\_threshold\_percent to enter = desired SMB amount / iobMAX (*In the example it would be 0.6* ).

In [section 4.6](#) you find (*if already launched* a profile helper to guide you through your calculation.

78

79 Note that

- 80 • the last SMB given can exceed that level by max +30% of the active iobTH.
- 81 • thereafter (until iob falls below iobTH) only %TBRs supply more insulin, if calculated as still  
82 required.
- 83 • in low carb meals, that iobTH level should not be reached => the autoISF parameters  
84 („weights“) need to be tuned carefully, so SMB sizes are *not always* huge and bounce  
85 against the iobTH restriction, but show different behaviour for different meals
- 86 • autoISF 3.0 contains a function to **auto-adjust** iobTH with TT set: Dynamic iobTH, see  
87 [sections 3.3](#) and [6.1](#).

88

89 Note: Shutting off SMBs due to iobTH does not rely on an odd TT or target running - but the  
90 even/odd logic remains useful in different contexts than iobTH.

91 *With older autoISF variants, the odd TT SMB shut-off above iobTH in an Automation*  
92 *had been a bridging solution*

93

94 All above discussed settings must be made in AAPS/preferences.

95

96 To enable going into FCL mode, additional settings must be made in /preferences/SMB/autoISF, as  
97 discussed in detail in [sections 3-4](#).

98

99 Once all this set-up is in place, you can enter/exit FCL (for initial tuning or for everyday utilization)  
100 via just tapping on the green/violet closed loop icon (*if already launched*) of your AAPS main  
101 screen (*else must do it in AAPS/Preferences/OpenAPS SMB/autoISF*).

102

## 103 2.5 EatingSoon TT ?

104

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

108

109 Setting an EatingSoonTT well ahead of meal start *can* therefore be an excellent idea.

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

112

113 The good news therefore is: Setting an **EatingSoonTT is not required** for autoISF FCL.

114

115 Note that the goal of having a low glucose target to aim at can also be achieved without setting an  
116 EatingSoonTT, by automatically setting a low TT just around the time when your loop gives first  
117 SMBs. For this you can define an **Automation** like: CONDITIONS: likely meal time of day + 1st  
118 sign of a beginning meal + iob under (?) U (to kick in only in initial phase) + no TT running  
119 ACTION: set TT=~74 for ~30 minutes

120

121 *Note for users of previous autoISF versions:* You do not need the other Automation any longer,  
122 that previously was needed to shut off EatingSoonTT before iobTH is reached. autoISF 3.0 ff  
123 integrates iobTH and the associated temp. SMB shut-off into the algorithm, rather than  
124 requiring yet another user Automation for that.

125

126 You might prefer working with a pre-set low profile target (especially in case you often experience  
127 elevated glucose levels before the respective meal time slot):

- 128 • If you have relatively fixed meal time slots in the 24 hours of the day, you could set the  
129 **target glucose values in your profile** accordingly. So e.g. 11-15h target 76 if you almost  
130 always start a lunch between 11:45 and 14:30h.
- 131 • If you have rather irregular habits, it is more worthwhile to **manually** set an **EatingSoonTT**  
132 (which is quite time-uncritical) well before the start of a meal, or even (latest) when the first  
133 SMB is about to be triggered by your loop.
- 134 • As eluded to above, if your glucose is not elevated as you approach the meal (this should  
135 be the norm), you do not have to bother with setting an EatingSoonTT at all. Just have the  
136 afore-mentioned Automation help optimize initial SMB sizes.

137 FCL works in principle also **without** setting an EatingSoonTT. Try for yourself whether you loose  
138 any performance via the totally hands-off way.

139