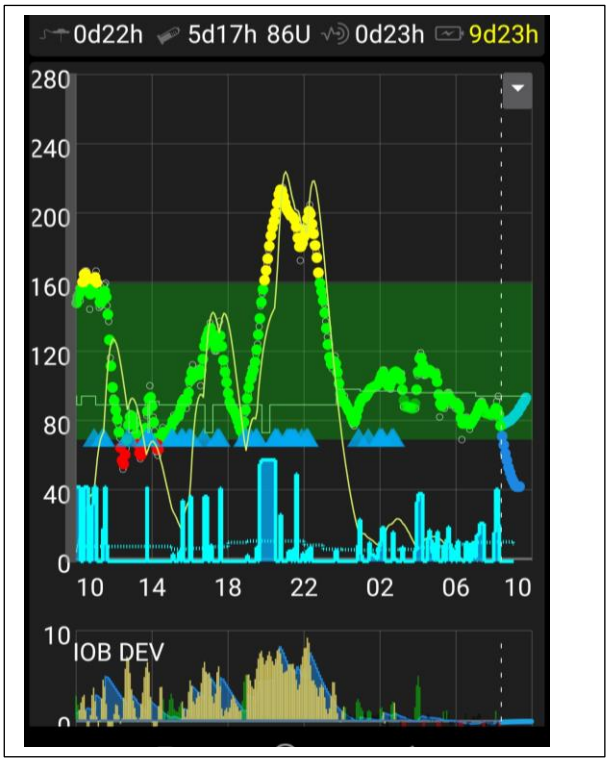
Case study 9.1: **Unintended loss of loop aggressiveness**

V.0,1

Report from a user of autoISF 3.0 who, “out of the blue” (i.e. after days with perfect %TIR) sees his loop “without bite”, and bg skyrocket.



In the Trouble Shooting section 9.3 the following potential reasons for this are laid out (shortened):

- *Meals are not recognized asap*
- *First SMB(s) seem a bit delayed*
- *SMBs are too weak*

All this seemed not at play, here….
…but the next (and last) point:-

- *… iobTH might be too low, and therefore cuts SMBs too early*

  Check whether your FCL really operates with the iobTH you think it uses.

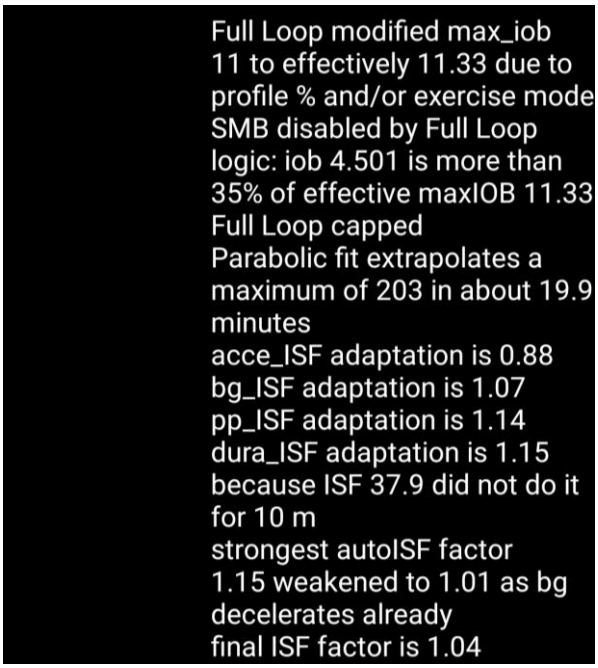
Indeed, my default iobTH_percent is 65, which should allow delivery of SMBs up to 7.1 U (=multiplied with my iobMAX of 11U). But my actual iob seemed to stagnate between 3 U and 4 U, despite bg climbing to well over 200 mg/dl (which is unusual, and was actually the reason why I recognized the problem at all).

I could rule out an occlusion because my iob was not revving up high (and also, my cannula time was well under 48 hrs still).

Eating started around 18:30 h. Well over an hour the buffet snacking went quite well regarding the glucose curve (see 1st slide in the series of 6, further below).

However, around 20.30 it became apperent that glucose was rising more than usual. Looking first through details in the AAPS home screen/SMB tab (what limited SMBs?), at 20:26 h:

x 328



iob rose only to 4.501 U, because disabled from 35% (!) of 11.33 U => disabled above 3.97 U

Looking up my settings in AAPS/Preferences/SMB/autoISF, I realized that my iobTH_percent was sitting on 35, effectively allowing only half of the urgently required insulin.

However, I had no idea *why that was*. Being at a party, I had also no further time to dig any deeper about this. I just set my default 65 percent value, and had autoISF loop (and party, with further snacking) continue.

The problem switching on SMBs again *now* could be: bg is de-celerating already (acce_adaptation 0.88) which will reduce the elevated aggressiveness that both, pp_ISF and dura_ISF call for (1.14 resp. 1.15) to only 1.01

I was aware that I was kind of in uncharted territory there because (apart from very few instances of an hour or so lost Bluetooth right after a meal start) I had no precedence, and could not be sure how exactly autoISF would deal with bringing a high bg down, based on only half *of the usually present* iob from SMBs *in the initial half hour* of rising bg. All my…ISF_weights were calibrated *without looking into the scenario that was now present*.

- It could well be that my autoISF cannot act aggressively enough now, at high bg, but de-celeration. Normally, that is when much *milder* ISF modulation is desireable, and _weights therefore were set this way.

- But also the opposite might happen. Generally, *the later* iob is devised against going high, *the bigger* the ensuing hypo danger an hour or so later.

- Even worse, a too aggressive treatment of scenarios with high bg and de-celeration might, in the past, have been camouflaged: Because at bg acceleration SMBs were triggered that quickly surpassed my default iobTH, little if any additional insulinRequired usually resulted in the stage of high bg. In my related tuning, I might have dialed in settings that were too aggressive, but never came into play really.

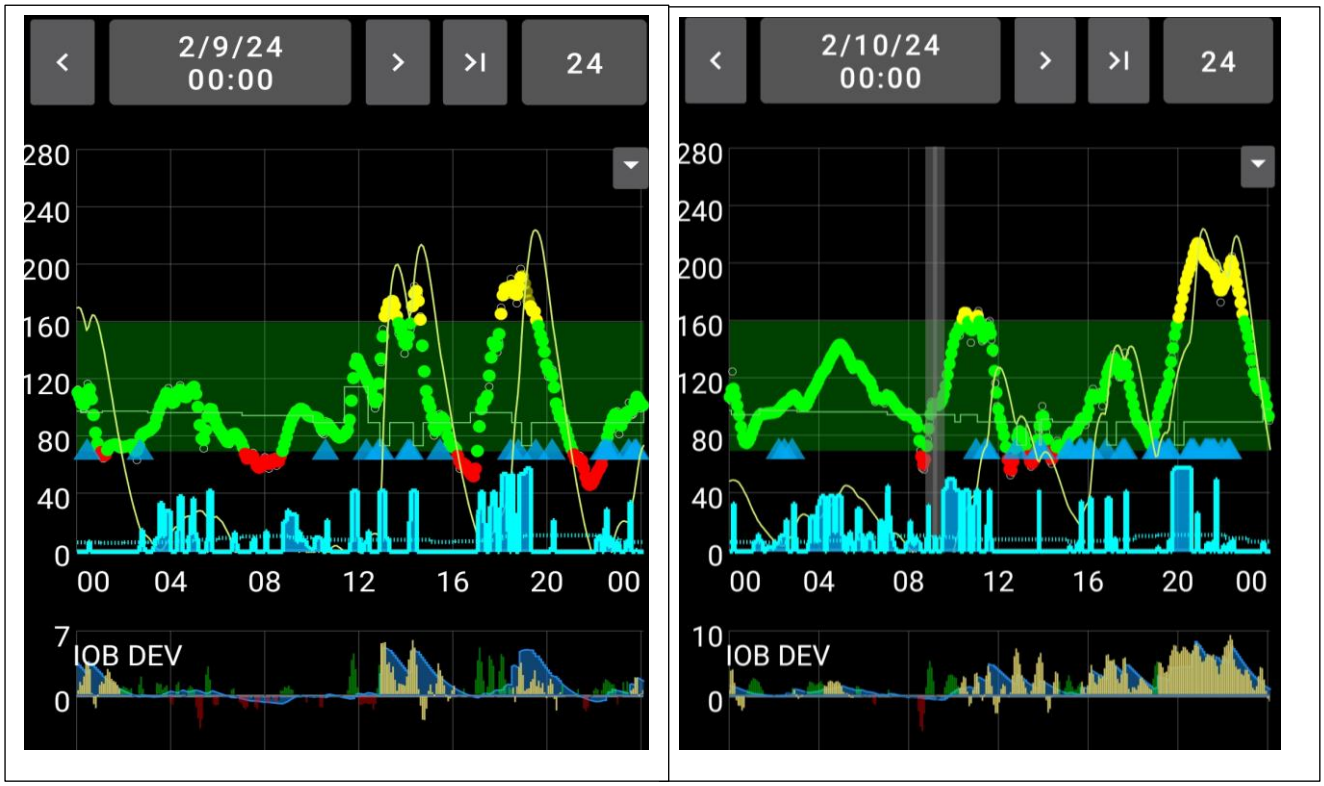
What that meant for the party evening was, that I just needed to keep looking into my smartphone every now and then, to get a feel where things are headed (even if a bit annoying to other guests).

So, next day, time to find out more

> *…. Observe the Caution notes (e.g. in section 5.1.4) about needing re-sets to default, after an Automation had lowered the effective iobTH*.

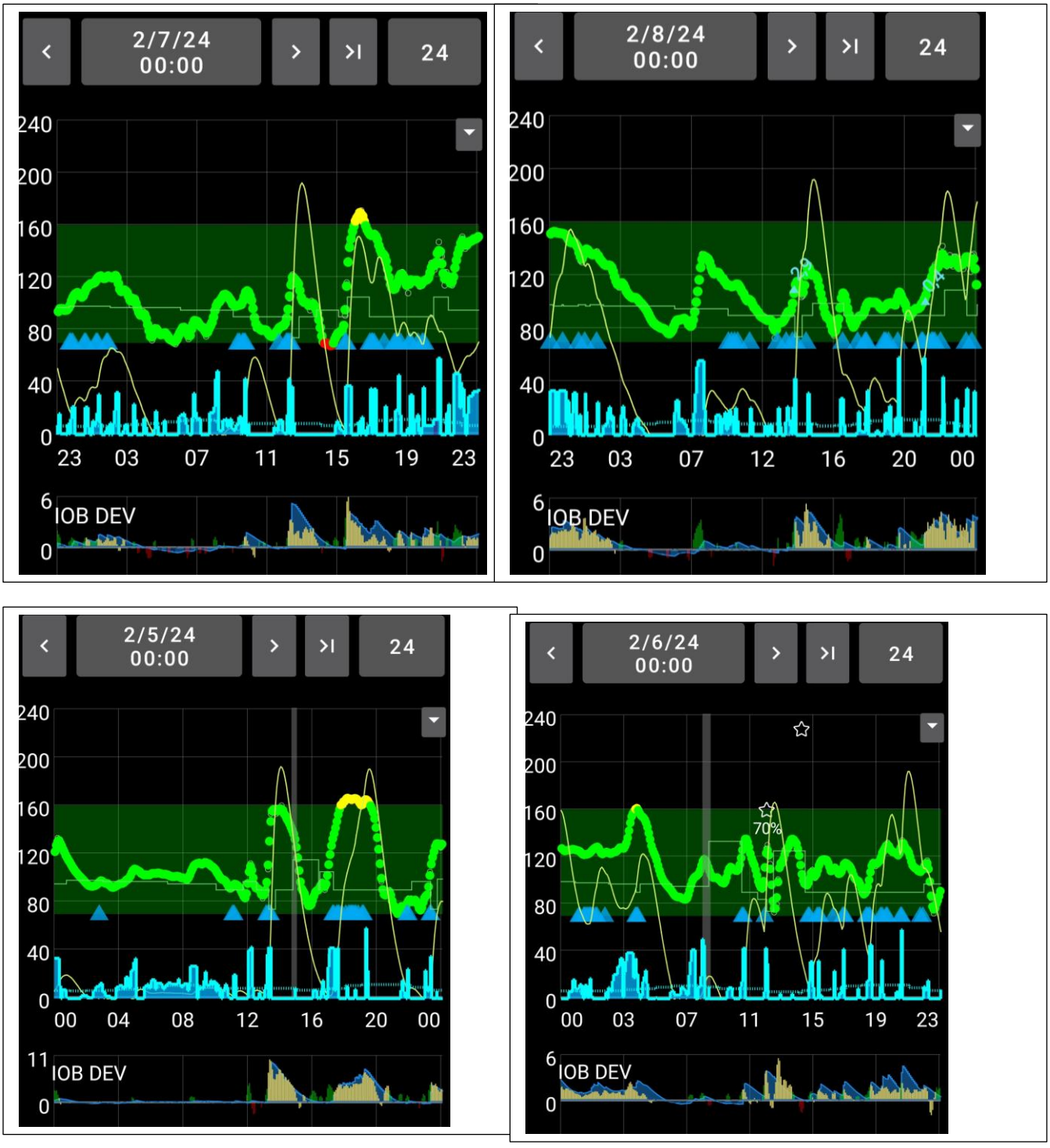
Looking at the general curves of the past days, and searching where, *maybe via an Automation*, the problem might have originated:

The following 6 screen pictures represent bg, insulin activity, and iob patterns on the day I noticed the problem (2/10), and on 5 preceding days (2/9-5): #6 – 11 / x443 …

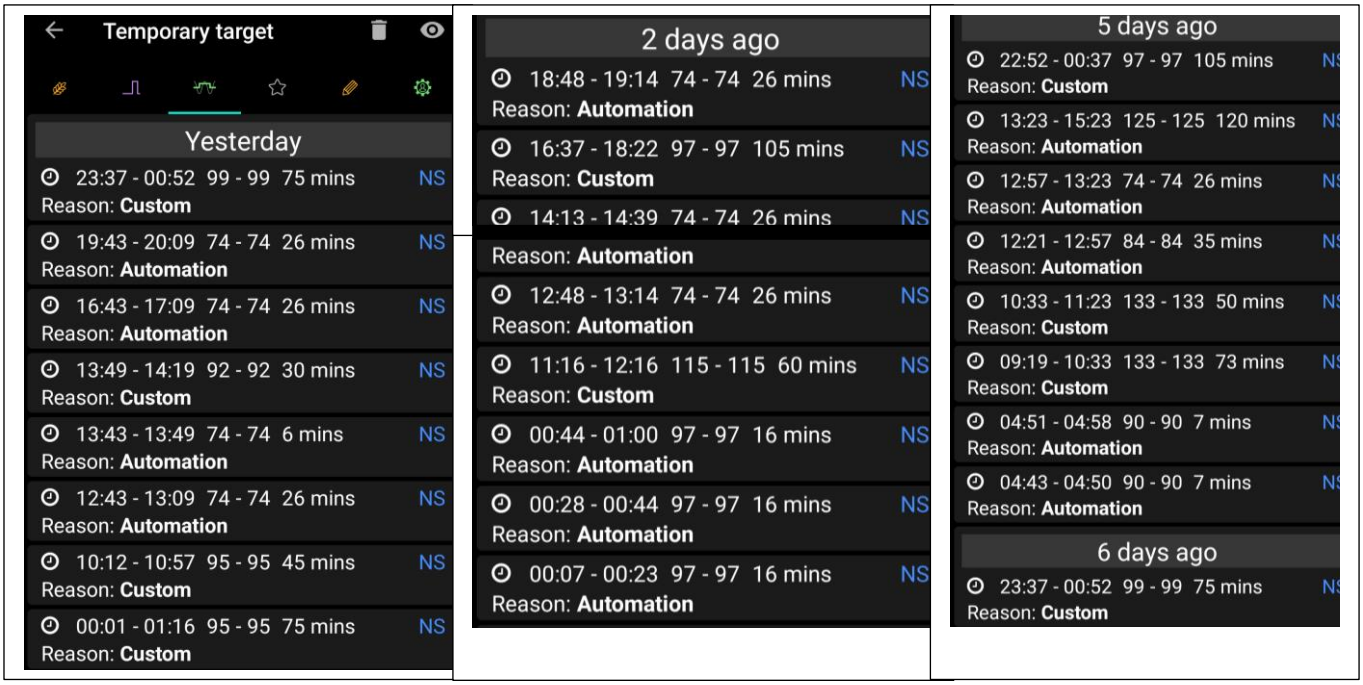


It looks like the problem originated anywhere *after* noon on 2/9, the day before.

Further back, days had excellent bg curves staying nearly always in range:



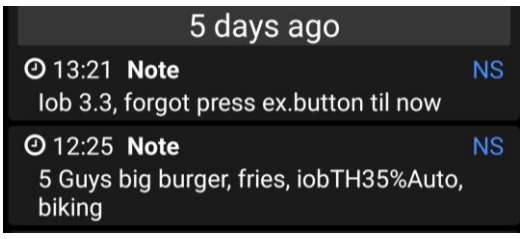
Looking at Automations that ran on the day of the problem (2/10, “yesterday”) and the day before (2/9, “2 days ago”) only reveals Automations setting the low 74 mg/dl TT at meal recognition, and some custom even or odd temp. bg target settings – none of which would do anything to the iobTH_percent set in /preferences.



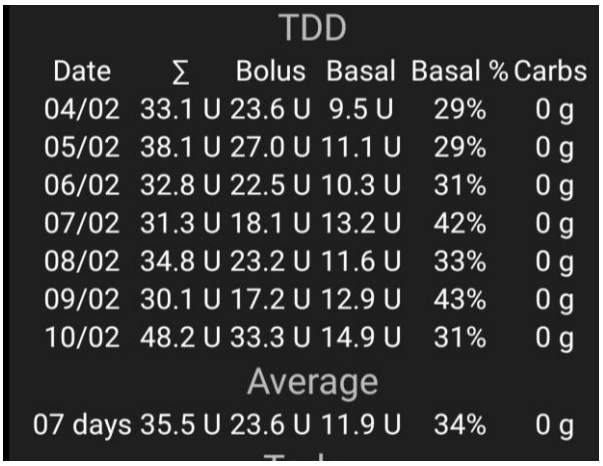
My “shelved” (in-active) User Action Automation “Meal before Exercise” is the only one *that would set* 35% iobTH, and does so in combination with setting 84 mg/dl TT. This was last used “5 days ago” i.e. 2/6 (5th screen in the series of 6 given earlier).

Looking next through all my Careportal *text* entries, nothing of potential relevance came up except, “5days ago” (2/6; 5th screen in the series of 6 given earlier), there was a similar occurrence after lunch, where manually activating the exercise button in effect stopped SMBs at iob 3.3 U: #49 / x692

In searching further for potential reasons of a lowered iobTH_percent might require to look into the logfiles.



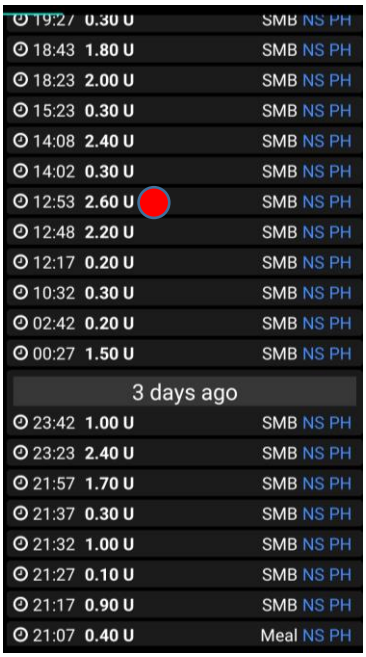
Meanwhile, reviewing what else AAPS itself has to offer, the screenshot at the right shows that the problem day stood out regarding much elevated TDD. Still, no occlusion is suspected; eating from party buffet (incl. 2 pc of pie) can explain what we see # 54 oben x207



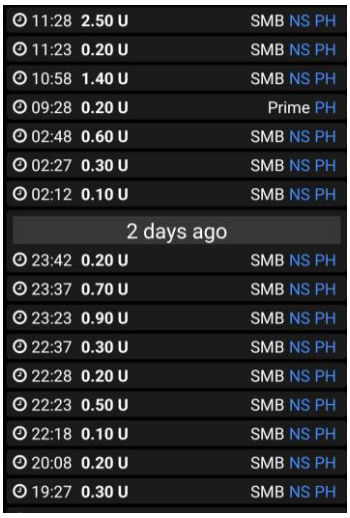
The open questions now are:

1. Was the iobTH *already since 2/5* at the low 35% level (at 3.97 U)? It sounds unlikely that I could stay *in range* for 4 days. However, these 4 days were well below average regarding TDD. Maybe, with “the last SMB” shooting a bit over 4 U, the resulting iob happened to be enough to keep bg in range?

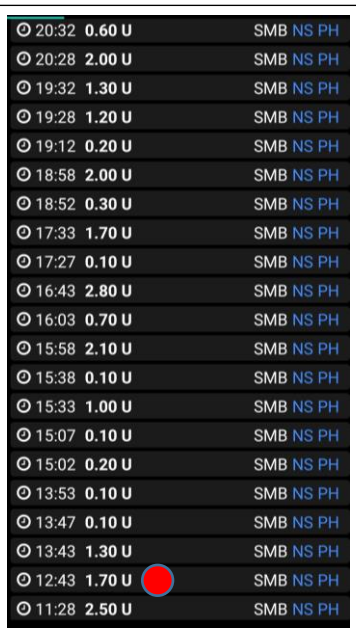
   So what were the SMB sizes on 2/6-2/10? Candidates that could shoot over 4 U, with a pause to the next SMB, are marked “🔴 “:



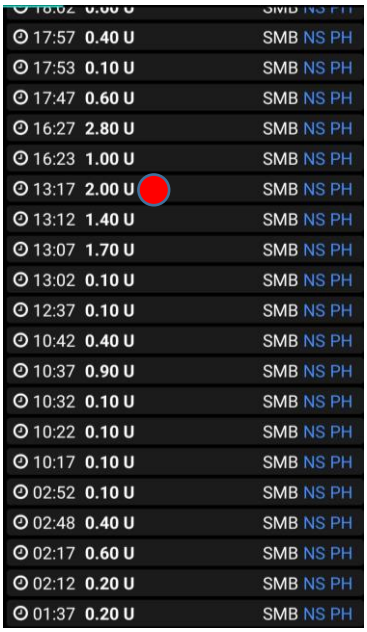
#19



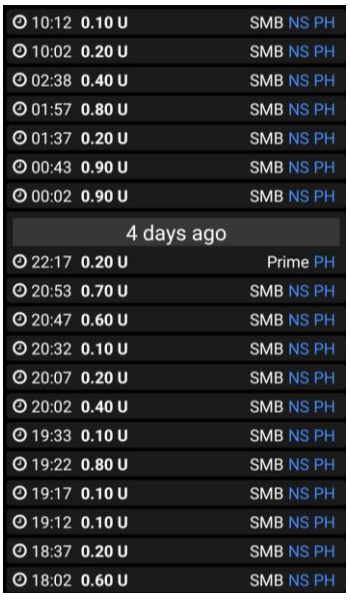
#18



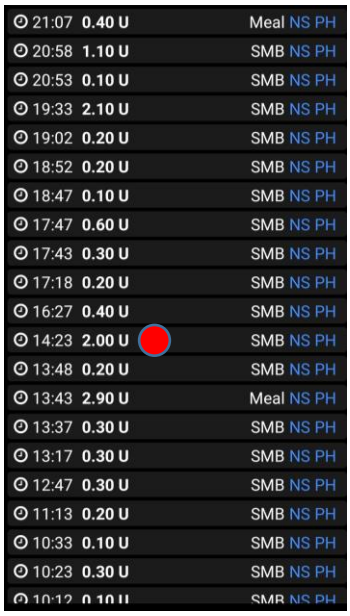
#17



#22



#21



#20

2. How do my Automations re-set the iobTH_percent to default?

   Once iob is over 3.8 U, the Automation that set 84 mg/dl TT AND the iobTH% of 35 is ended with ACTION: stop TT, and start 70% profile. Immediately following, at CONDITION 70% profile, a TT of 125 mg/dl sets in for 120 minutes, completing my typical exercise setting (TT 125 @ 70% profile) after what had been a brief interruption with higher aggressiveness until 3.8 U iob was exceeded after the meal-

   2A) Usually, the exercise button is activated in addition. This will activate the dynamic iobTH (see section and case study 6.1). After disactivation of the exercise mode, iobTH reverts to former default (?? Is that so ??)

   On that day, I activated the exercise button on 13:21 h (with just a bit of a delay, when iob was at 3.3 U ; see Note).

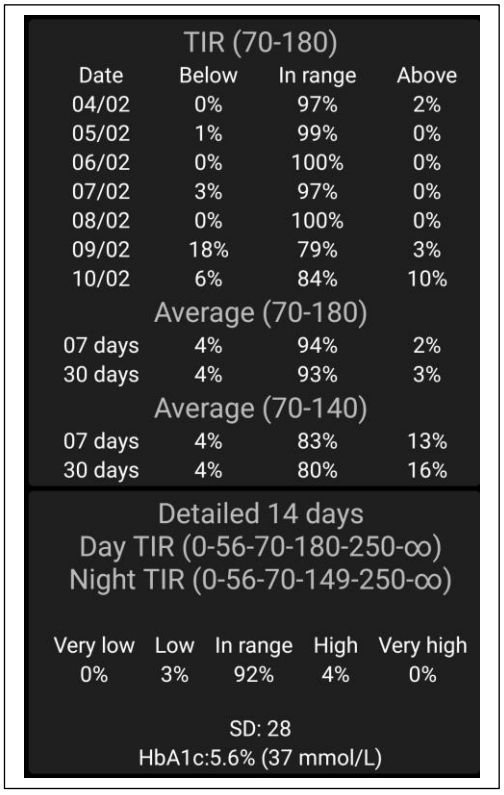
   2B) If I went without ever activating the exercise mode with dynamic iobTH, and 125 mg/dl TT, and later also 70%profile, had expired, (when?) would iobTH_percent return to default ??

3. What needs to be changed so users are better aware at all times, what the important settings are their FCL is currently working with?

Unfortunately, the problem day (10/02), plus the preceding one that probably suffered from the same problem already, thoroughly ruined a nice phase when my autoISF FCL was stable running at ~98%TIR (see screenshot)

Actually, this is overall the experience of the author who is testing autoISF 3.0 since nearly 6 months now. The nice extra features allow lots of extra fine tuning and Automation to, in principle, reach a few more %points of TIR. But time and again little stretches of flawless workings at over 95%TIR are interrupted by problems like the one reported here (or see also case studies 1.5 and 1.3).

So, these experiences MUST translate into actions, how to improve the workings of features, or the user interface to deal with them, safely and effectively.



To iron out some of the problems might be relatively easy in a software update:

The problem of un-recognized, too soft settings (un-intended low iobTH-percent; and same could happen with bgAccel_ISF_weight that was not re-set to default) must be eliminated as far as possible by:

1) Having a very obvious indicator on the AAPS main screen as to which iobTH (and bgAccel_ISF_weight?) is actually valid. Staying with the pattern of yellow %profile, TT, and exercise fields when loop aggressivenmess is temp. modulated, developers should provide a grey/yellow field with the iobTH next to the actual iob (both in U please, *not* iobTH in %,).

   This measure # 1) is the minimum I would see necessary to move autoISF into official AAPS_dev, or even into Master.

2) What might be helpful is to *always* keep the set default iobTH (and bgAccel_ISF_weight) in /Preferences, and name the temporary altered ones differently, e.g. effective iobTH or eff.iobTH%.

3) Thoughts need to go into how to *automatically* return from effective to default.

3A) If User-Automations are the reason for an effective iobTH%, maybe, like for TT, a time corridor must be set. If an effective iobTH results from modulation in exercise mode, a definition is needed when it (latest) automatically expires / tbd.

3B) Clarification is needed asap as to how, after exercise mode used dynamic iobTH, the system returns eventually to default settings as made in /Preferences.

(The latter point I will just check in the next days.

I hope I can conclude what users can do to limit and recognize the problem better. Based on that, I will complete this Case Study)

(Maybe logfiles research would also help to see which iobTH was valid all these days, and why – Except for 2/7 I have the logfiles, but I never had the latest Emulator version running yet)