How-to …

find out whether autoISF would help if you are still on plain AAPS

This document is meant for users of regular AAPS who want to know whether migrating to autoISF can help with the performance of their loop. The method relies on running the emulator which will compare results from *orig* (i.e. your current AAPS) versus *emul* (i.e. the emulated AAPS plus autoISF). This method works because the emulator not only emulates the regular AAPS but also the autoISF add-ons. Instructions for implementing and running the emulator are available on Github at https://github.com/ga-zelle/APS-what-if.

The other prerequisite is the availability of your original logfiles covering the time window for which you want to look at the comparisons. Please be aware that as of AAPS V3 the logfiles are overwritten after 2 days latest and you should therefore download them from the phone in time.

The VDF-file for the emulator contains the instructions which enable the features you want to investigate and modify the weights from their neutral defaults to become active. These instructions mimic what you would normally select and adapt in preferences via the autoISF sub-menu.

1. Pick example and ensure the original case can be repeated

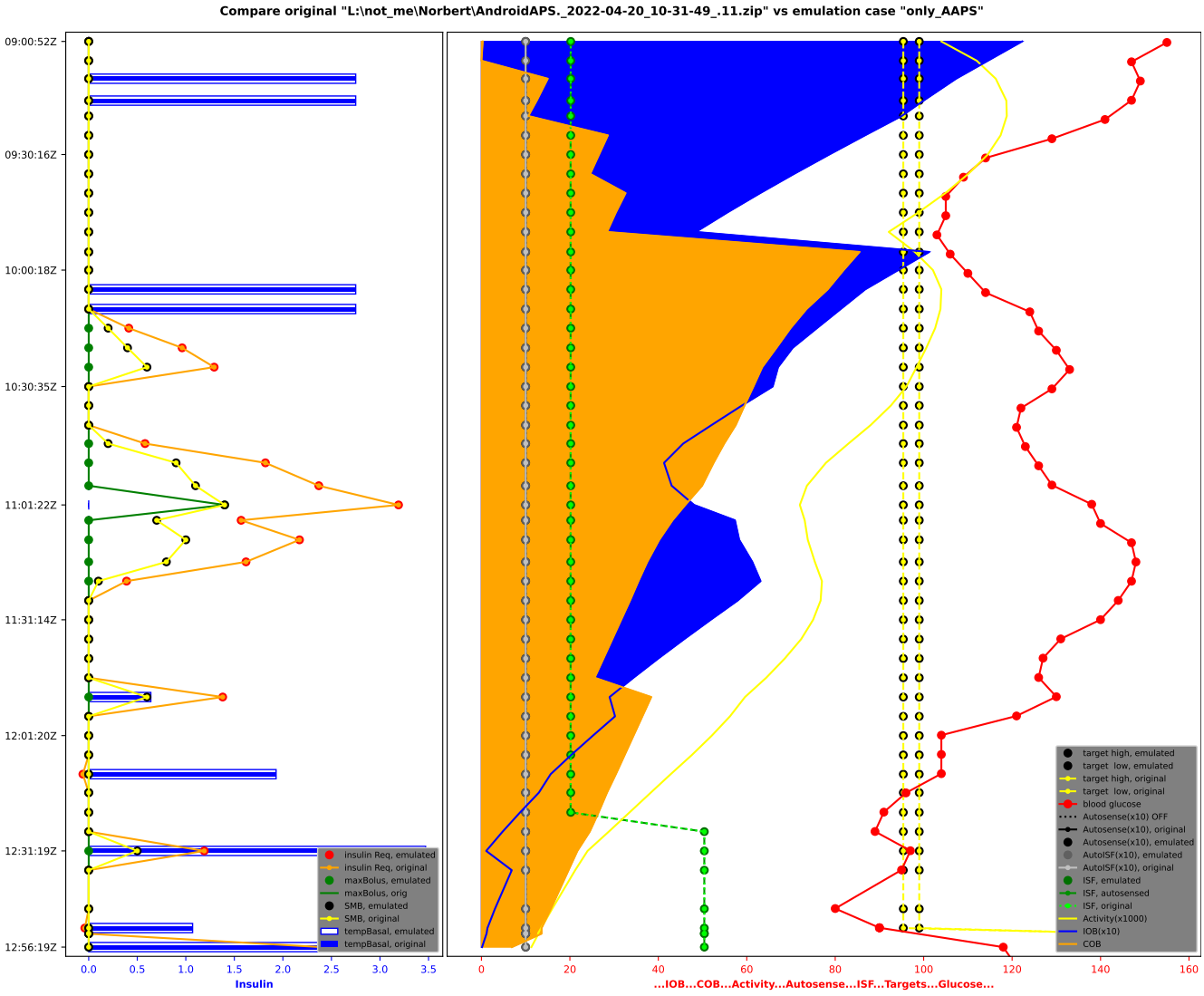
Because I run autoISF for years already I cannot use my own logfiles as a serious demo case. Therefore I want to thank n0rb33r7 from Discord for providing a sample set of his logfiles. To re-run the original case the simplest VDF-file is an empty file. Empty file means all settings remain as they were in the original run.

But because later on the autoISF features will be activated one by one a slightly different VDF-file can be used. The starting line means that the emulator ignores any autoISF algorithm. Also, it contains dummy assignments for the various features which then can just be replaced by the real stuff. This is what it looks like:

```
profile         enable_autoISF              False       ### keep it disabled

new_parameter   acce_ISF                    1.0         ### overwrite autoISF algorithm
new_parameter   bg_ISF                      1.0         ### overwrite autoISF algorithm
new_parameter   delta_ISF                   1.0         ### overwrite autoISF algorithm
new_parameter   dura_ISF                    1.0         ### overwrite autoISF algorithm
```

Here *new_parameter xxx_ISF* just replaces what autoISF would normally calculate internally. It is a general method of you calculate the various *xxx_ISF* factors yourself in the VDF to try your own algorithms. After saving the file as *only_AAPS.vdf* and running the emulator it is time to inspect the results and compare them with the original results for the xxx_ISF factors, InsReq, SMB, and TBR. First let us look at the PDF-file to compare results visually. No deviations between *orig* and *emul* can be identified. Also, after the meal the time window shows meal carbs (COB increasing) and bolus delivered (IOB increasing). Please note that manual or wizard initiated boli are not shown in the graph as individual events because they cannot be changed by the emulator anyway. What can be detected, however,is the follwing increase in IOB.



To be on the safe side you should also look at the numeric comparison as listed on screen during the emulation or even better load the CSV-file into the spreadsheet. Here is an extract with some scrolling and hiding of rows and columns:

	A	B	E	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO
1			bg	auto	acce	bg	pp	delta	dura	final				Ins.	Ins.	max	max				
2		UTC	accel	sens	ISF	ISF	ISF	ISF	ISF	ISF	ISF	ISF	ISF	Req.	Req.	bolus	bolus	SMB	SMB	TBR	TBR
3	id	ime		emul	emul	emul	emul	emul	emul	emul	orig	prof	emul	orig	emul	orig	emul	orig	emul	orig	emul
37	33	11:46:17	126	1	1	1	1	1	1	1	20,2	20,2	20,2	0	0	0	0	0	0	0	0
38	34	11:51:18	130	1	1	1	1	1	1	1	20,2	20,2	20,2	1,38	1,38	0	0	0,6	0,6	0,64	0,64
39	35	11:56:19	121	1	1	1	1	1	1	1	20,2	20,2	20,2	0	0	0	0	0	0	0	0
40	36	12:01:20	104	1	1	1	1	1	1	1	20,2	20,2	20,2	0	0	0	0	0	0	0	0
41	37	12:06:21	104	1	1	1	1	1	1	1	20,2	20,2	20,2	0	0	0	0	0	0	0	0
42	38	12:11:22	104	1	1	1	1	1	1	1	20,2	20,2	20,2	-0,06	-0,06	0	0	0	0	1,93	1,93
43	39	12:16:09	96	1	1	1	1	1	1	1	20,2	20,2	20,2	0	0	0	0	0	0	0	0
44	40	12:21:17	91	1	1	1	1	1	1	1	20,2	20,2	20,2	0	0	0	0	0	0	0	0
45	41	12:26:18	89	1	1	1	1	1	1	1	50,4	50,4	50,4	0	0	0	0	0	0	0	0
46	42	12:31:19	97	1	1	1	1	1	1	1	50,4	50,4	50,4	1,19	1,19	0	0	0,5	0,5	3,47	3,47
47	43	12:36:19	95	1	1	1	1	1	1	1	50,4	50,4	50,4	0	0	0	0	0	0	0	0
48	44	12:46:22	80	1	1	1	1	1	1	1	50,4	50,4	50,4	0	0	0	0	0	0	0	0
49	45	12:51:22	90	1	1	1	1	1	1	1	50,4	50,4	50,4	-0,04	-0,04	0	0	0	0	1,07	1,07
50	46	12:52:45	90	1	1	1	1	1	1	1	50,4	50,4	50,4	0	0	0	0	0	0	0	0
51	47	12:52:50	90	1	1	1	1	1	1	1	50,4	50,4	50,4	0	0	0	0	0	0	0	0
52	48	12:56:19	118	1	1	1	1	1	1	1	50,4	50,4	50,4	2,4	2,4	0	0	0	0	3,47	3,47
53	Minimum:		80	1	1	1	1	1	1		20,2	20,2	20,2					0	0		
54	Maximum:		155	1	1	1	1	1	1		50,4	50,4	50,4					1,4	1,4		
55	Totals:																	8,5	8,5	1,72	1,72

All xxx_ISF factors are set to 1.0, all *orig* versus *emul* numbers are identical and as sort of a cross check the numbers in cell range AL53:AO55 for min, max and total SMB and TBR insulin are identical.

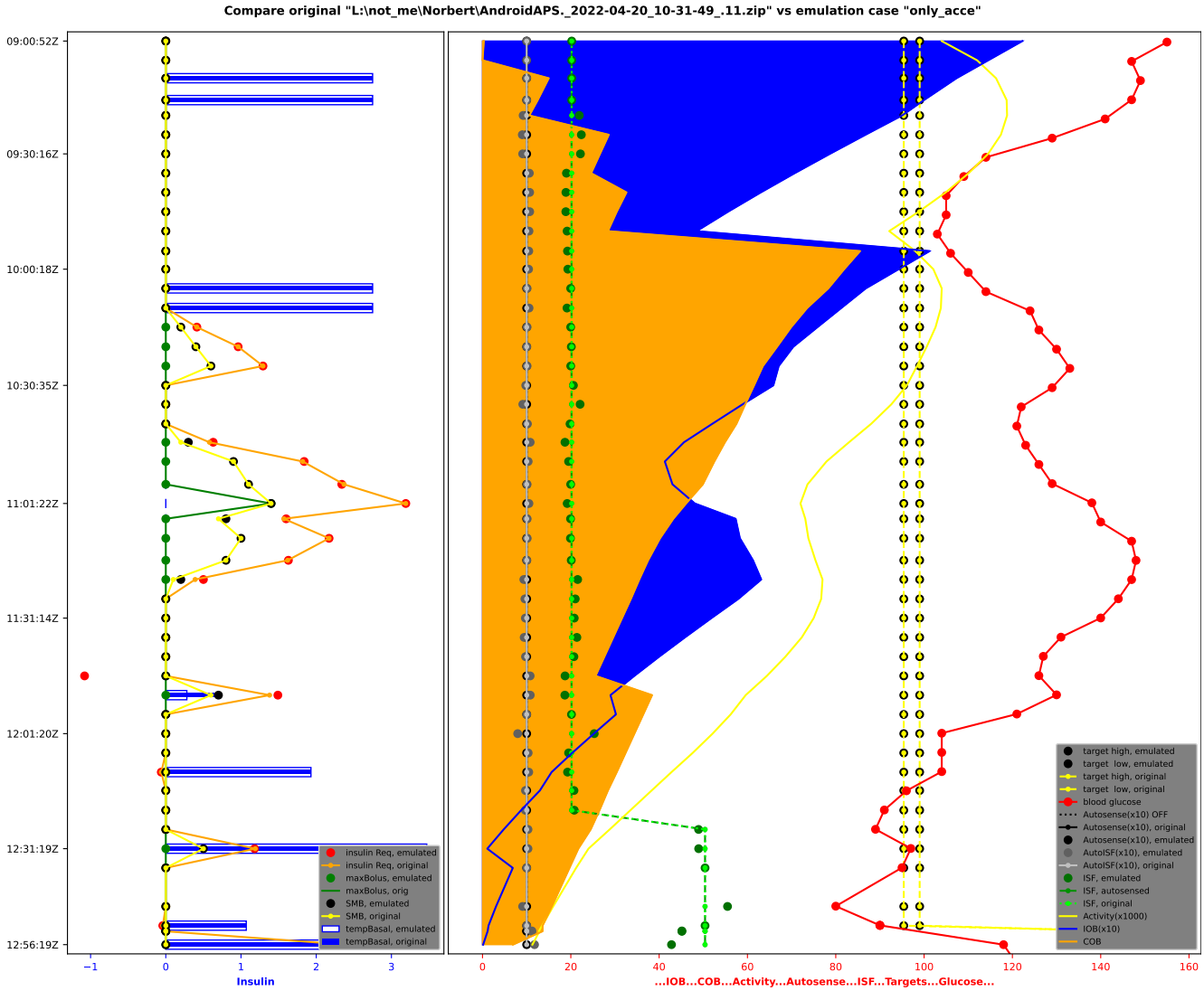
2. Check the impact of acce_ISF

First you need to allow autoISF and define its minimum and maximum adaptations just like you do for Autosense. For this example I use the standard settings recommended in the instructions (https://github.com/ga-zelle/autoISF/blob/A3.1.0.3_ai2.2.8/autoISF2.2.8_Quick_Guide.pdf). In that PDF you also find the meanings, spellings and the like. For *acce_ISF* to be activated you need to provide the weights for phases of accelerating and decelerating glucose values and these replace the fixed assignment of acce_ISF=1.0 from the previous section. The final *only_acce.vdf* looks like this:

```
profile          enable_autoISF          True         ### switch it on
profile          autoISF_max             1.2          ### recommended initial setting
profile          autoISF_min             0.7          ### recommended initial setting

profile          bgAccel_ISF_weight      0.02         ### recommended initial setting
profile          bgBrake_ISF_weight      0.02         ### recommended initial setting
new_parameter    bg_ISF                  1.0          ### overwrite autoISF algorithm
new_parameter    delta_ISF               1.0          ### overwrite autoISF algorithm
new_parameter    dura_ISF                1.0          ### overwrite autoISF algorithm
```

The result of the emulation is here:



The dark green dots in the right frame show slight strengthening when the downward trend is reduced or even inverted. In analogy, weakening of ISF happens when an upward trend slows down or even turns into a downward trend. This is typical feature, namely the acceleration recognises a change in trend very early even while it is overlaying a steep rise or fall. In the left half you see the resulting slight increases in insReq and SMB at some points in time. Most of the time the stronger ISF did not

lead to more SMB because the meal bolus and SMBs in the original run covered that need. Details can be investigated in the CSV-File:

	A	B	E	F	L	Y	AD	AE	AF	AG	AH	AI	AL	AM	AN	AO
1			bg	bg		acce	final				Ins.	Ins.				
2		UTC	accel	brake		ISF	ISF	ISF	ISF	ISF	Req.	Req.	SMB	SMB	TBR	TBR
3	id	ime			iob	emul	emul	orig	prof	emul	orig	emul	orig	emul	orig	emul
21	17	10:25:32	133		6,71	1,01	1,01	20,2	20,2	20	1,29	1,29	0,6	0,6	0	0
22	18	10:30:35	129	129	6,59	0,98	0,98	20,2	20,2	20,6	0	0	0	0	0	0
23	19	10:35:31	122	122	5,89	0,91	0,91	20,2	20,2	22,1	0	0	0	0	0	0
24	20	10:40:37	121		5,2	1,02	1,02	20,2	20,2	19,8	0	0	0	0	0	0
25	21	10:45:24	123		4,56	1,08	1,08	20,2	20,2	18,7	0,58	0,63	0,2	0,3	0	0
26	22	10:50:25	126		4,13	1,03	1,03	20,2	20,2	19,5	1,82	1,84	0,9	0,9	0	0
27	23	10:56:22	129		4,3	1,01	1,01	20,2	20,2	20	2,37	2,34	1,1	1,1	0	0
28	24	11:01:22	138		4,81	1,05	1,05	20,2	20,2	19,2	3,19	3,19	1,4	1,4	0	0
29	25	11:05:21	140		5,73	1,01	1,01	20,2	20,2	19,9	1,57	1,6	0,7	0,8	0	0
30	26	11:10:26	147		5,83	1,01	1,01	20,2	20,2	19,9	2,17	2,17	1	1	0	0
31	27	11:16:10	148		6,12	1	1	20,2	20,2	20,1	1,62	1,63	0,8	0,8	0	0
32	28	11:21:10	147	147	6,31	0,94	0,94	20,2	20,2	21,5	0,39	0,5	0,1	0,2	0	0
33	29	11:26:14	144	144	5,79	0,96	0,96	20,2	20,2	21	0	0	0	0	0	0
34	30	11:31:14	140	140	5,18	0,97	0,97	20,2	20,2	20,8	0	0	0	0	0	0
35	31	11:36:15	131	131	4,58	0,94	0,94	20,2	20,2	21,4	0	0	0	0	0	0
36	32	11:41:17	127	127	4	0,97	0,97	20,2	20,2	20,7	0	0	0	0	0	0
37	33	11:46:17	126		3,44	1,08	1,08	20,2	20,2	18,7	0	-1,08	0	0	0	0
38	34	11:51:18	130		2,9	1,08	1,08	20,2	20,2	18,6	1,38	1,49	0,6	0,7	0,64	0,28
39	35	11:56:19	121		3,02	1	1	20,2	20,2	20,1	0	0	0	0	0	0
40	36	12:01:20	104	104	2,52	0,8	0,8	20,2	20,2	25,3	0	0	0	0	0	0
41	37	12:06:21	104		2,03	1,03	1,03	20,2	20,2	19,5	0	0	0	0	0	0
42	38	12:11:22	104		1,56	1,05	1,05	20,2	20,2	19,3	-0,06	-0,06	0	0	1,93	1,93
43	39	12:16:09	96	96	1,3	0,97	0,97	20,2	20,2	20,7	0	0	0	0	0	0
44	40	12:21:17	91	91	0,88	0,97	0,97	20,2	20,2	20,8	0	0	0	0	0	0
45	41	12:26:18	89		0,48	1,03	1,03	50,4	50,4	48,9	0	0	0	0	0	0
46	42	12:31:19	97		0,11	1,03	1,03	50,4	50,4	49	1,19	1,18	0,5	0,5	3,47	3,47
47	43	12:36:19	95		0,69	1	1	50,4	50,4	50,4	0	0	0	0	0	0
48	44	12:46:22	80	80	0,31	0,91	0,91	50,4	50,4	55,5	0	0	0	0	0	0
49	45	12:51:22	90		0,14	1	1	50,4	50,4	50,4	-0,04	-0,04	0	0	1,07	1,07
50	46	12:52:45	90		0,12	1,12	1,12	50,4	50,4	45,2	0	0	0	0	0	0
51	47	12:52:50	90		0,12	1,12	1,12	50,4	50,4	45,2	0	0	0	0	0	0
52	48	12:56:19	118		0,01	1,18	1,18	50,4	50,4	42,8	2,4	2,9	0	0	3,47	3,47
53	Minimum:		80			0,8		20,2	20,2	18,6			0	0		
54	Maximum:		155			1,18		50,4	50,4	55,5			1,4	1,4		
55	Totals:												8,5	8,9	1,72	1,69

The cells highlighted in grey show cases where increased *insReq* led to more SMB and row 55 shows the sum of all those SMBs. The summed up numbers need to be taken with care because the emulator has no loop capability, i.e. changes at time x have no impact on the situation at time x+5min but the earlier a change happens the larger the impact downstream. Use your experience and judgement. To get the extra dose you wish that early on as soon as the acceleration starts to show.

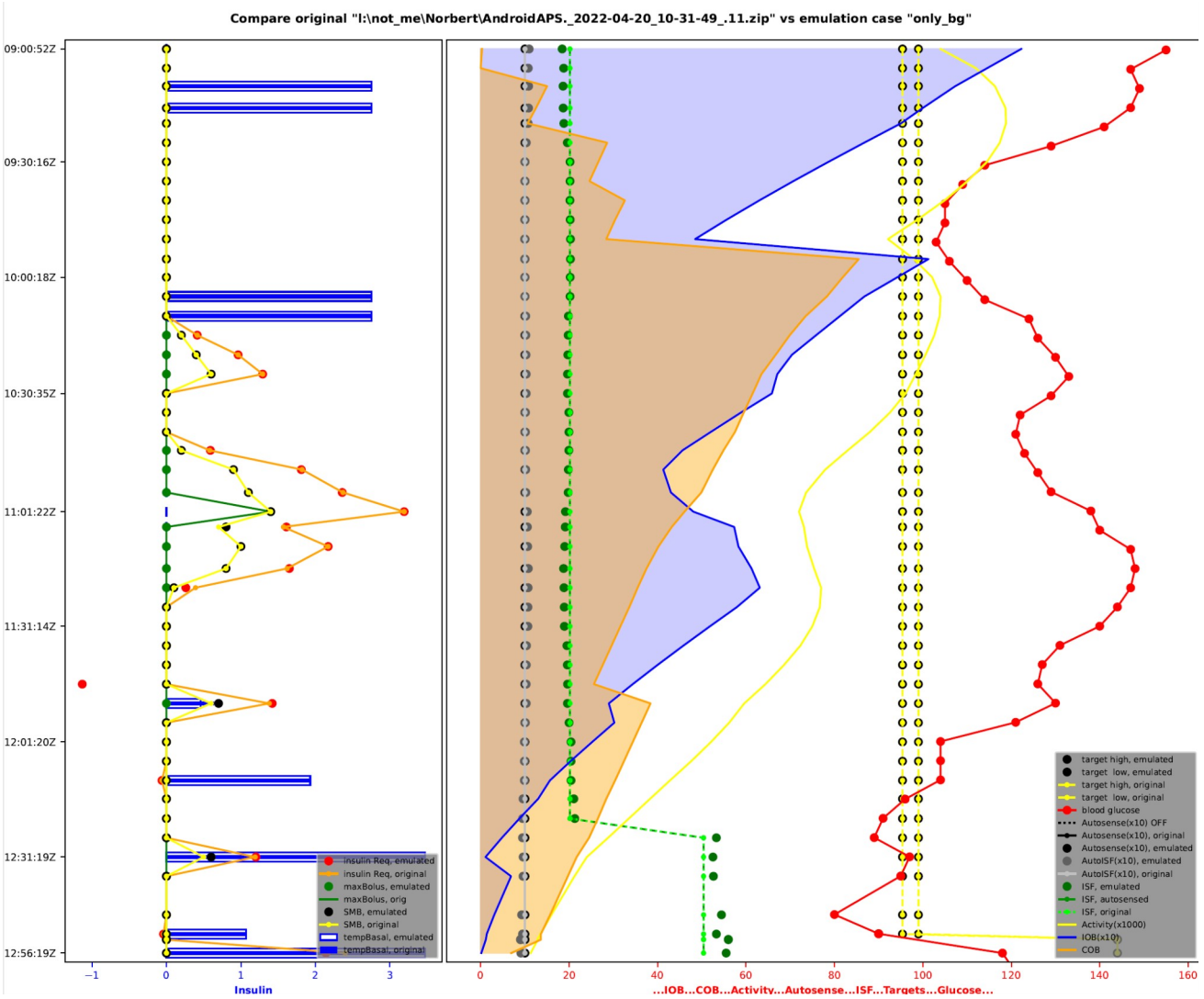
3. Check the impact of bg_ISF

As you might have guessed the previous acce_ISF is effectively disabled by reintroducing the fixed assignment from the noChange case and instead replacing the bg_ISF assignment by its weights. This is what the *only_bg.vdf* looks like:

```
profile          enable_autoISF           True        ### switch it on
profile          autoISF_max              1.2         ### recommended initial setting
profile          autoISF_min              0.7         ### recommended initial setting

new_parameter    acce_ISF                 1.0         ### overwrite autoISF algorithm
profile          higher_ISFrange_weight   0.2         ### recommended initial setting
profile          lower_ISFrange_weight    0.2         ### recommended initial setting
new_parameter    delta_ISF                1.0         ### overwrite autoISF algorithm
new_parameter    dura_ISF                 1.0         ### overwrite autoISF algorithm
```

The result of the emulation is here:



Although the weights were picked cautiously you can already see the ISF strengthening slightlyat high bg and weakening at bg below target.

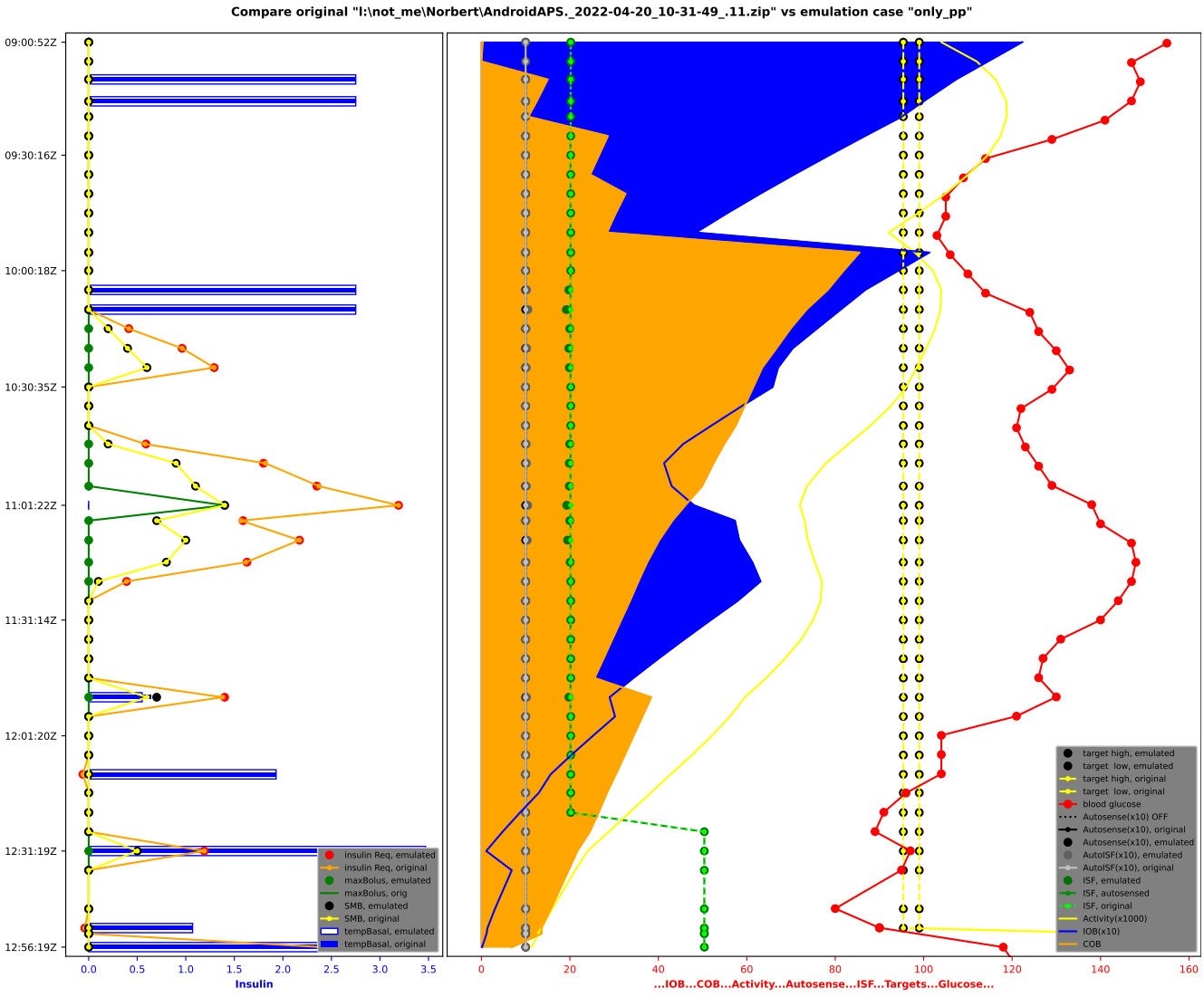
4. Check the impact of pp_ISF

Next option would be to look at the impact of delta_ISF. However, it is very similar to pp_ISF which is of special interest to T1Ds with gastroparesis. With that user group in mind there is the usual weight factor but also a time window to define how long after a meal the effect is active. Using the recommended initial settings the *only_pp.vdf* looks like this:

```
profile          enable_autoISF          True         ### switch it on
profile          autoISF_max             1.2          ### recommended initial setting
profile          autoISF_min             0.7          ### recommended initial setting

new_parameter    acce_ISF                1.0          ### overwrite autoISF algorithm
new_parameter    bg_ISF                  1.0          ### overwrite autoISF algorithm
profile          pp_ISF_weight           0.005        ### recommended initial setting
profile          pp_ISF_hours            6            ### recommended initial setting for gastroparesis
#new_parameter   delta_ISF               1.0          ### leave out as it would overwrite pp_ISF
new_parameter    dura_ISF                1.0          ### overwrite autoISF algorithm
```

The result of the emulation is here:



Although the weights were picked cautiously you can already see the ISF strengthening slightly when bg is rising around 10:00UTC and 11:00 UTC. Also there is a small effect at 11:51UTC.

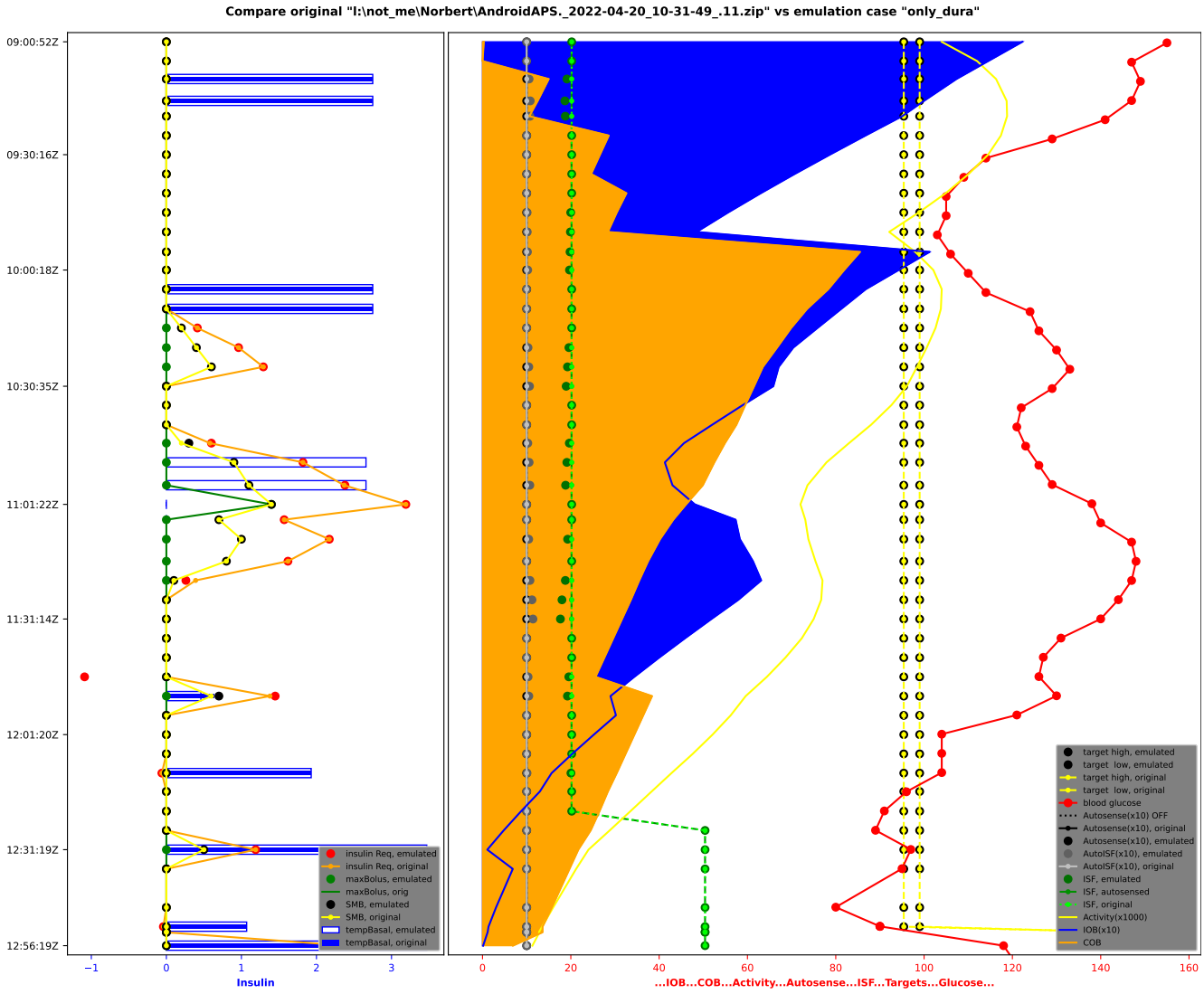
5. Check the impact of dura_ISF

Just defining *dura_ISF_weight* would normally work already. But in the example used here there are carbs present during the whole time window and therefore *enable_dura_ISF_with_COB* must be activated to see the effect. The corresponding VDF-file looks like this:

```
profile          enable_autoISF           True        ### switch it on
profile          autoISF_max              1.2         ### recommended start setting
profile          autoISF_min              0.7         ### recommended start setting

new_parameter    acce_ISF                 1.0         ### overwrite autoISF algorithm
new_parameter    bg_ISF                   1.0         ### overwrite autoISF algorithm
new_parameter    delta_ISF                1.0         ### overwrite autoISF algorithm
profile          dura_ISF_weight          0.6         ### 3 times recommended start setting
profile  enable_dura_ISF_with_COB         True        ### COB present in example
```

The result of the emulation is here:



From the graph it becomes clearer why in this case I used 3 times the recommended initial setting for the weight. It better demonstrates how dura_ISF is strengthening linearly with its duration, especially leading up to 11:31UTC. The message which would appear in the SMB-tab at that time would read

> dura_ISF adaptation is 1.14 because ISF 20.2 did not do it for 30 m

and is listed in the result file *2022-04-20T09.only_dura.txt*.

6. Closing remarks

You can copy and paste the VDF-files from this document and run them for your own projects. Here I did not run any combination of several features. It would be easy by combing VDF-file extracts but as beginners you should focus on one effect after the other. The general experience is that acce_ISF dominates the reaction of the loop to bg changes and occasionally dura_ISF attacks stubborn highs. In the example time window there is no obvious situation that needs to be improved. But those 4 hours provided phases where the potential impact of all 4 autoISF effects could be demonstrated.

One next step for you could be to stick with plain AAPS but install and run the emulator on the AAPS phone with your favourite autoISF settings. This would tell you by speech output whenever a higher or lower bolus would be given by autoISF with these settings. Similar to the old days of open looping you can then decide whethter applying that extra bolus – or just a part of it – appears atrtactive. That way you can build confidence in the more aggressive settings and eventually switchover to autoISF with those new settings.

The logfiles were from a setup using mmol/l units instead of mg/dl which I use and tested before. That triggered a few changes in the emulator such that the two resulting TXT-files can be compared more easily. One confusing observation was that in the mmol/l world targets seem to be reported in the SMB-tab in mmol/l and predicted BG values in mg/dl.

Running all the preceding examples was an interesting exercise for me. I knew using the emulator on logfiles from plain AAPS without autoISF would work with providing special parameters in the VDF-file to compensate for them missing in the original run. That takes only a few iterations and it should be easy for me. But how about adding those checks right in the source code itself and make life easier for everyone? Well, that is what I did while running all these five case studies.

Status: 8.Mar.2023 @ 20:34