3. Description of autoISF features V 1.1

Please note that with autoISF 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

Related case studies:

Links to more case studies or detailed docu on special topics: See section 3.9

3.1 Overview

autoISF can be used to refine the workings of your **Hybrid Closed Loop.**
If you use autoISF for Hybrid Closed Loop, you exclusively can do so by studying the documents linked in this section 3, all available from Github/ga-zelle (repo autoISF and repo APS-what-if).

> Note that the **apk to build** your autoISF variant of AAPS (and the installation instructions for it) is elsewhere: https://github.com/T-o-b-i-a-s/AndroidAPS/; for Trio see , and iAPS see:https://github.com/mountrcg/iAPS

autoISF allows also to build a top performing **Full Closed Loop**.

> FCL is the sole topic in all other sections of this FCL e-book. https://github.com/bernie4375/FCL-potential-autoISF-research

-

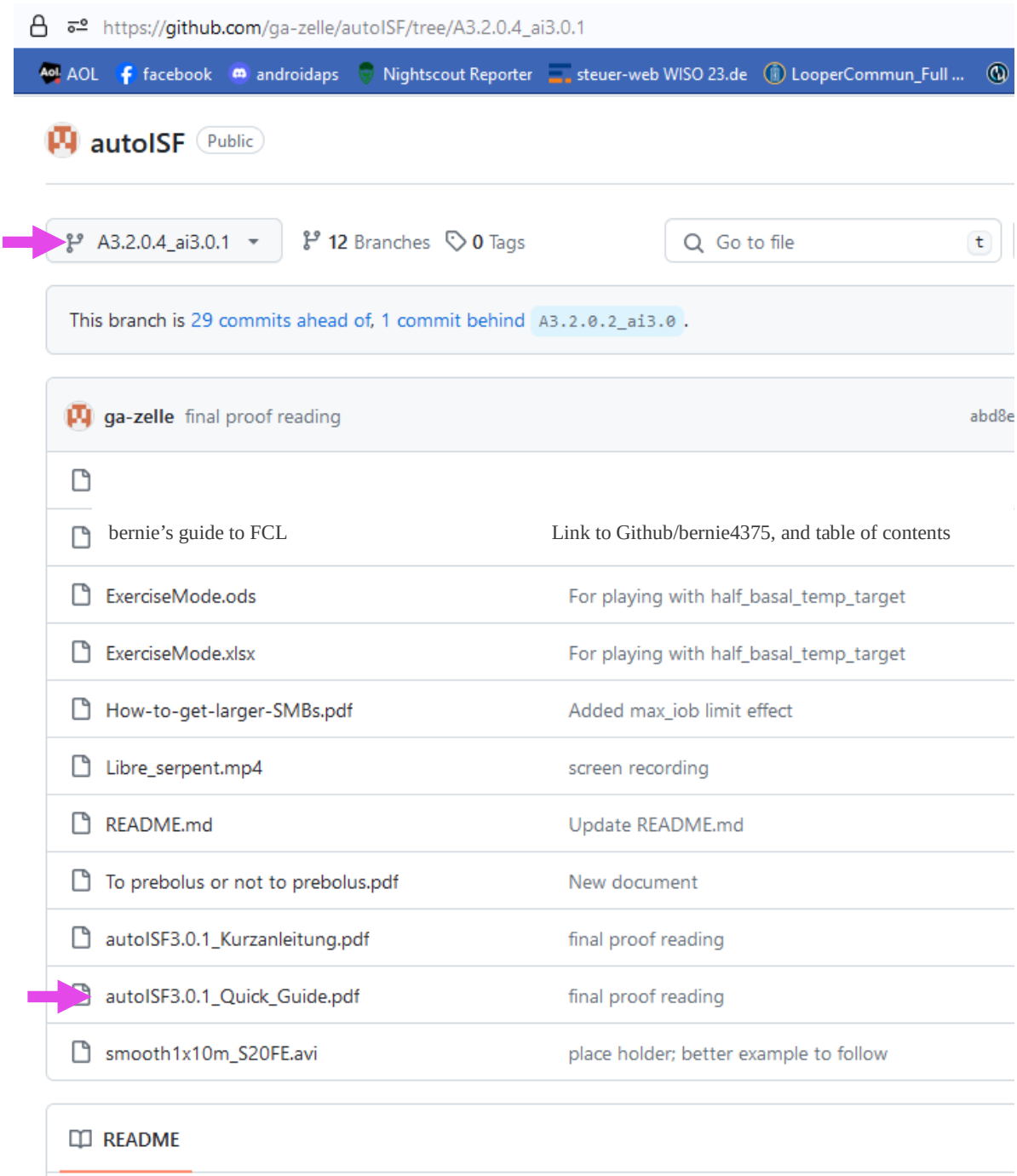
The **general workings of autoISF** are best characterized with the chart that is included in section 4.1. . It sketches which of the autoISF parameters have a key role of managing your bg curve, in its characteristic different post-meal stages.

A comprehensive description of autoISF is the devloper's Quick Guide here:

- https://github.com/**ga-zelle/autoISF**/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

See screenshot below for current content.

Always watch out to use the most up-to-date ?/ latest version (branch in the Github repo) /?



Note: The developer provides some materials also in German language in (t)his Github repo.

3.2 ISF adaptation flowcharts

autoISF calculates every 5 minutes (and more often if you use Libre3) an ISF (called sens) to use in place of your profile_ISF (profile.sens, which remains an important anchor point).

autoISF 3.0.1 ff users on Android can see on the 1st page of their **SMB tab,** how these calculations (and how their individual settings re. profile, safety limits, but also set TT etc.) determine sens and SMB size.

Set of flowcharts describing calculation of sens (the concluded effective ISF to use, as in SMB tab:)

- **page 1 – 6** of the **Quick Guide**: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

:

3.3 Exercise mode and dynamic iobTH

autoISF is geared towards aggressive treatment. However, in an exercise context, it is desireable to have built-in features that allow manage situations with much less typical insulin need.

autoISF has several special features to address this, which all are described here:

Exercise mode:

- on **page 7** of the **Quick Guide**: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

Dynamic iobTH:

- is explained under the headline “internal automation for iobTH” on **page 9** of the **Quick Guide**: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf
- Calculators to determine how half-basal exercise target, set TT influence sens and iobTH, (in .xls or odt format), here: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/ExerciseMode.xlsx and https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/ExerciseMode.ods

  Consult Section 6 of this FCL e-book for more guidance to find appropriate exercise-related settings for your favorite types of exercise

SMB delivery ratio:

- on **page 10** of the **Quick Guide**: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

Even/odd target (for SMB on/off)

- see **page 11** of the **Quick Guide**: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

Loop power level characterization in the SMB tab

- is explained see **page 12** of the **Quick Guide**: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

The ai % indicator underneath the Autosens % on the AAPS screen

- is explained also on**page 12** of the **Quick Guide**: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

3.4 Automation options with autoISF parameters

autoISF provides AAPS users an expanded set of Conditions and Actions to choose from, when setting up an Automation.

autoISF parameters available in Automations (as Condition, and/or Action) are described:

- on **page 11** of the **Quick Guide**: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

Caution: In FCL with AutoISF, please do not rush into setting up lots of Automations to “fine tune” (this is discussed in detail in section 4.6-4.7, and section 5.). Rather, first try to do a good job following (in FCL) the sequence as laid out in the FCL e-book.

3.5 Activity Monitor

autoISF also comes with an Activity Monitor. You can calibrate it to your personal sensitivity swings as they may relate to stepcount, or to periods of total in-activity.

Activity monitor description:

- see **page 8** of the **Quick Guide**: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

3.6 Using 1-minute CGM (Libre3)

1 minute Libre3 data use in autoISF:

- go to **page 13** of the **Quick Guide**: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

Especially if you go for FCL: The issue over-arching your hope for avg maybe 2 minutes earlier clues from the CGM must be: Solid non-jittery performance (see section 1.4).

3.7 Additional parameters in autoISF (18), and recommended start settings

In an older version you may see (17), as iobTH% was missing in the list. Default setting for this should be around 50 (more see FCL e-book section 2.4).

The table in Attachment 1 of the Quick Guide gives an overview of additional settings if you operate autoISF to its full potential.

The default, and recommended start of tuning suggestions in this table are made for Hybrid Closed Loopers.
For FCL, please consult this FCL e-book

Table showing all autoISF parameters w/ default settings see: all autoISF parameters see: …

- on **page 14** of the **Quick Guide**: https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

autoISF settings screen in AAPS/Preferences (inside the OpenAPS SMB menue), see:

- last page of the **Quick guide** = **page 15** : https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/autoISF3.0.1_Quick_Guide.pdf

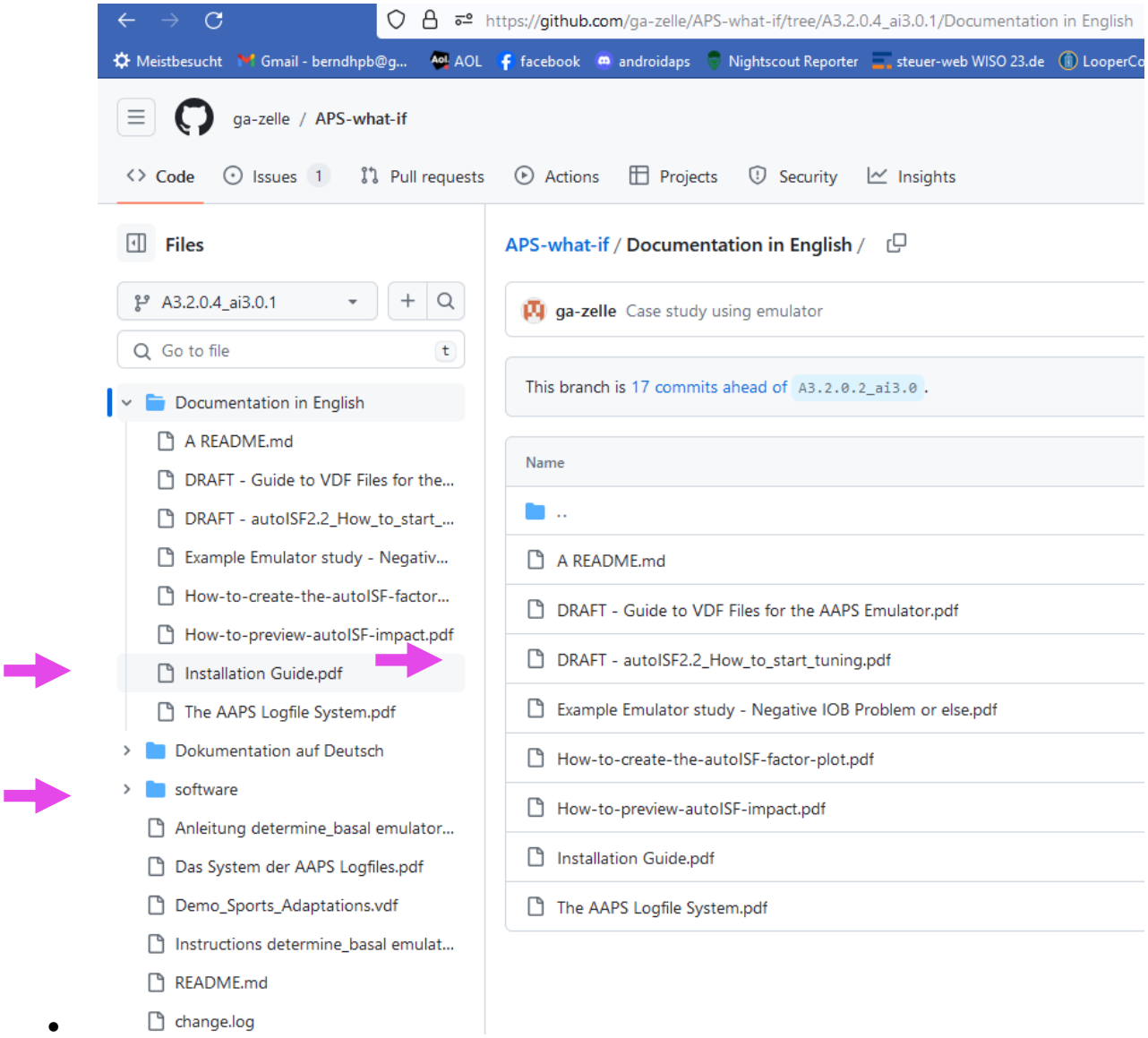
3.8 Emulator for AAPS logfile analysis

The links given in section 3.8 are numbered for easier referencing in other text.

It can be impractical to real-time inspect the SMB tab (or take screenshots for later inspection). To determine which of your settings should be changed for better performance, the autoISF developer provides an extra tool, the Emulator. It is described in another repo:

1).Emulator documentation

- https://github.com/**ga-zelle/APS-what-if**/tree/A3.2.0.4_ai3.0.1/Documentation%20in%20English (watch for latest version branch)



-

2).Emulator installation guide see:

- https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/Installation%20Guide.pdf

FCL e-book sections 10 (PC) and 11 (phone) offer additional installation guidance.

3).How to start tuning guide for HCL – (consult this FCL e-book additionally if you go FCL), see:

- https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/DRAFT%20-%20autoISF2.2_How_to_start_tuning.pdf

FCL e-book sections 10 (PC) and 11 (phone), plus associated case studies, offer additional guidance for interpretation and tuning, with focus on application in Full Closed Loop.

This “emulator” tool does not require building an apk.
Go to “software” and download the needed (mostly python) files. Then follow installation guide(s).

4).Software download for PC and Android phone here:

- https://github.com/ga-zelle/APS-what-if/tree/A3.2.0.4_ai3.0.1/software

Specifically, there are these examples of .vdf files for what-if investigations offered to download (for use, or for customization):

5).

6).https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Demo_Sports_Adaptations.vdf

7).

Note that tuning your settings for Full Closed Loop is a very difficult project in which you should follow the sequence of sections 1 -6 of this e-book.
Especially in your “section 4 phase”, the Emulator is a great tool to use (refer to sections 10 and 11),

The emulator can also be used for AAPS SMB+UAM without (or with only a few) autoISF features utilized

3.9 Links to related case studies or other detailed documents

The links given in section 3.9 are numbered for easier referencing in other text.

1).reserved for =case study 3.1:
(link)#

2).To pre-bolus or not to pre-bolus = case study 3.2:

https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/To%20prebolus%20or%20not%20to%20prebolus.pdf

3).Analyzing a negIOB situation with the emulator = case study 3.3
https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/Example%20Emulator%20study%20-%20Negative%20IOB%20Problem%20or%20else.pdf

4).reserved for =case study 3.4:
(link)#

5).reserved for =case study 3.5:
(link)#

6).How to get larger SMBs
https://github.com/ga-zelle/autoISF/blob/A3.2.0.4_ai3.0.1/How-to-get-larger-SMBs.pdf

7).How to pre-view autoISF impact
https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/How-to-preview-autoISF-impact.pdf

8).How to create the autoISF factors plot
https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/How-to-create-the-autoISF-factor-plot.pdf

9).Guide to vdf files for emulator
https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/DRAFT%20-%20Guide%20to%20VDF%20Files%20for%20the%20AAPS%20Emulator.pdf

10).How to