Meal Management in Closed Loop

V.0.2

For most T1D loopers, meals are the major problem for keeping glucose in range.

This paper looks at the principle challenges and available solutions from a lay user perspective. It builds on the published learnings of looping pioneer Dana Lewis, and other related literature. It also reflects own experiences, and insights shared on the internet platforms of the international looper communities.

Focus is on open source "DIY" looping systems, notably those using the **oref(1) algorithm (OpenAPS, AAPS, Trio, iAPS)**, The author is type1 diabetic for many years, and looper for several years. He uses AAPS.

Both his education in technical chemistry and his nautical hobbies helped him understand the meal control problem. Hopefully, this proves helpful for communicating about it.

Throughout this paper, we make frequent references to sections of his e-books on Github:

- For Hybrid Closed Looping: https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings
- On Full Closed Looping (for advanced users): https://github.com/bernie4375/FCL-potential-autoISF/blob/FCL-e-book/00_Introduction_FCL-book.V%203.1.pdf

While principal challenges are the same, **iOS Loop** users are confronted with elevated importance of all carb-related inputs. Loop&Learn has two videos out on bolusing for meals:

- https://www.youtube.com/watch?v=zs67r0pkgQM and https://www.youtube.com/watch?v=4_5K9zH9CFc

Commercial systems are much less well described regarding how exactly they work, and come with far less options to tune. Contact the supplier for any specific questions.

Numerous references are made to what other open source loopers have published. We all cannot guarantee correctness of all content, or applicability to other patients, and assume no liablity. In line with our #WeAreNotWaiting philosophy, we like to encourage critiques and debates so we all can deepen our knowledge in moving forward..

Please use everything that is said only to trigger thoughts for cautious self-responsible conclusions. Seek further clarification in the open-source looper communities, and, most importantly, with your doctor.

A **glossary** of terms used by loopers you find here: https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings/blob/HCL-.-settings-main-repo-(pdf)/00_Glossary_V%2C4.5.pdf

For **studies** published on safety and efficacy of open source looping, see here: https://openaps.org/outcomes/

Already in late 2021, one of the globally most renowned medical journals published an international consensus paper, encouraging our doctors to embrace open-source loops as viable options for their patients, besides the increasing number, but more limited nature, of commercial loops: K.Braune et al, Lancet Diabetes Endocrinol 2021 https://doi.org/10.1016/S2213-8587(21)00267-9

Topics overview

We look at our topic, meal management in closed loop, in four main sections.

First, the principal challenge when approaching meal time is laid out. We look into the time-patterns of insulin activity and of carb absorption, to gain some understanding of **„sluggish" delayed control**. The key parameters loops use to navigate our blood sugar through the disturbances created by meals are presented. For the further discussion, a **3 phase model** of meal management is laid out.

The **second** section will be about how to reasonably interact with the looping algorithm in the **phase before meals;** and about the **early meal phase**, where, in hybrid closed looping, our meal **bolus** is highly active and forces the loop to retreat into a co-management role.

The **third** section discusses the **late phase** of a meal. This is when the bolus initiated by the user fades out, while more late carbs are being absorbed. Effects from fat and protein complicate the picture. Now the looping algorithm must increasingly take over. We focus here on the two main algorithms as used in the open-source systems. To enable the algorithm to take best care, the user must give appropriate inputs about the meal, and select appropriate settings in her/his looping system. Also, tuning some parameters may be required from time to time. We know it is sometimes tempting to interfere with the loop, so we will touch on the pros and cons of doing this, as well.

The **fourth** section describes **simplified meal management** when working with the **„oref(1)"** algorithm. It principally can work without carb inputs for meals.

- This enables a very much simplified „carb counting" in hybrid closed looping.
- Moreover, full closed looping (no boli and no carb inputs by the user) is a possibility if certain criteria are met.

- Several options are also available that radically simplify looping by some form of meal announcement (MA)

It is not advisable to jump right into advanced methods. There is great benefit from first establishing a working hybrid closed loop. Understanding and mastering meal management is a core requirement for this.

Table of contents

Simplified solutions when looping with oref(1) -> Meal.Mgt.section.4

- General solutions toolbox
- Good (enough) practice suggestion for oref(1) loops in HCL
- Meal announcement (MA) methods
- Full closed looping (No carb inputs, no boli)
- Weight control
- Closing remarks

Delayed control

One of the core challenges in type 1 diabetes management is to provide our body the suitable amount of insulin for any meal. While „historically“ the focus was on roughly providing a balance without incurring severe hypos, or prolonged hyperglycemias, the advent of continuous glucose monitors was, literally, an eye opener to most of us:

It allows us, every 5 minutes, to have a look how carb absorption and insulin activity work out to result in the glucose curve we see developing:

When we eat, it takes around 10 minutes until the very first „fast“ carbs provide glucose to our cells (as also experienced in hypo treatment). Depending on quantity, but also kind of diet, the glucose-elevating effect from a meal can be uneven, and last for hours. Fat stretches absorption further out, and often requires insulin boosts due to short-term insulin resistance.

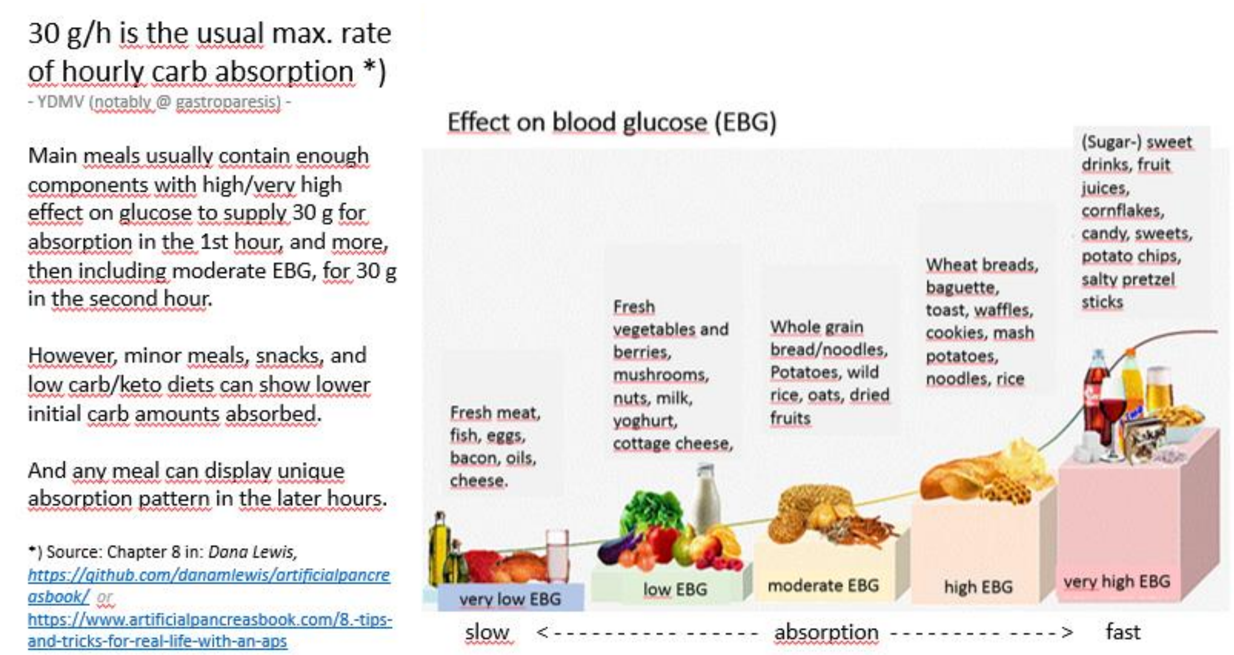
Insulin administered needs at least 20 minutes to get into our cells and produce a first small effect (lowering tissue glucose only after a couple of more minutes). The unfolding activity is characteristic for the insulin in use (time-to-peak, duration of action (DIA); more see: https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings/blob/HCL-.-settings-main-repo-(pdf)/Insulins_DIA%20and%20other%20settings_V.3.0.pdf).

For all the insulin given by the user or by the loop, these activity curves stack up, and make up a complex pattern of insulin activity. **It is problematic to supply additional activity for any moment of extra need** (as everybody knows from „rage“ bolussing when values are high: First not much happens, until max. activity of the new bolus is approached; then, for hours, further activity is likely to drive glucose towards a hypo, despite the loop countering with zero-temping).

Constantly balancing carb digestion with insulin activity is the task. It is really a mission-impossible, given the sluggish nature of anything we already did (eat, past insulin additions) or could do (further insulin additions, withdrawing basal, adding more carbs).

Meal Management therefore is a very difficult control problem - very much like boating -. Foresight, patience, and experience are needed to maneuvre on a good-enough course. And a good algorithm will have some of this modelled-in, or could also have some self-learning capabilities (like Autotune, Autosens in open-source systems, or non-disclosed machine learning in some of the newer commercial closed loops).

Carb absorption



The fastest absorbing foods have a very high (fast) effect on blood glucose. So this is what we need to prevent or resolve a hypoglycemia. But as component of a meal this makes glucose control very difficult (possible really only with a suitable pre-bolussing time).

However, even the slowest absorbing foods can be challenging to deal with. Typically, they are absorbed too late for a meal bolus to take effective care. Notably fats can also interfere with absorption speed and insulin requirements for carbohydrates.

Also, there is a maximum hourly capacity: 30 g is the usual maximum hourly carb absorption (YDMV, notably with gastroparesis).

Main meals often contain enough components with high or very high effect on glucose, to supply 30 g for absorption in the 1st hour, and more, then including moderate EBG, for 30 g in the second hour.

So, for the fast insulins (Fiasp or Lyumjev) that we use for looping, our initial bolus usually should cover 60g.

However, low carb/keto diets can show lower initial carb amounts absorbed (as is the case for minor meals or snacks).

Any meal can display unique absorption pattern in the later hours

Carb input for fat and protein

Proteins and fats can be counted as grams of „extended" Carbs. These „e-"Carbs also require insulin, and are entered for hours 3 -~7 after the meal (without giving a bolus for it)

„Real" carbs that were <u>not</u> absorbed while the user bolus was very active are added to these e-Carbs.

There are two approaches how to count protein and fat:

(1) Calculation of e-Carb equivalents via kcal from fat and protein.
As caloric information on components on your plate is not easy figured out, in daily T1D life mostly the other route is taken:

(2) Using factors 50-60% per gram of protein, and 10-20% per gram of fat.

Actually, the % applied for fat is discussed quite controversially:

- There is an additional effect from fat (and also from fibre) to slow down absorption of "real carbs", so it may seem that fats translate to a higher % into late carbs.
- Often, fats create complications several hours after a meal through **intermittent insulin resistance** at high glucose values (we come back to this in section 3 on late phase of meal).

Important here is, that the reasonable conversion factor depends on the used strategy (A or B) to deal with these effects:

- Strategy (A): Apply a **bigger %** for the conversion, and in consequence **exaggerate carbs** from fats, and this way also gets them covered with more insulin. For iOS Loop, this method (A) is preferable.
- Strategy (B) is to use the suggested **low factor** of 10 – max 20% for fats; **plus** apply **stronger ISF** temporarily, only when glucose is stuck on high values. This method (B) is preferable for oref(1) loops, which do not react directly on grams of „e"-carbs entered, but even have special features to address temp. insulin resistance (see section 3).

Insulin on bord (iob)

Insulin on bord shows how much insulin is available for activity in the upcoming hours.

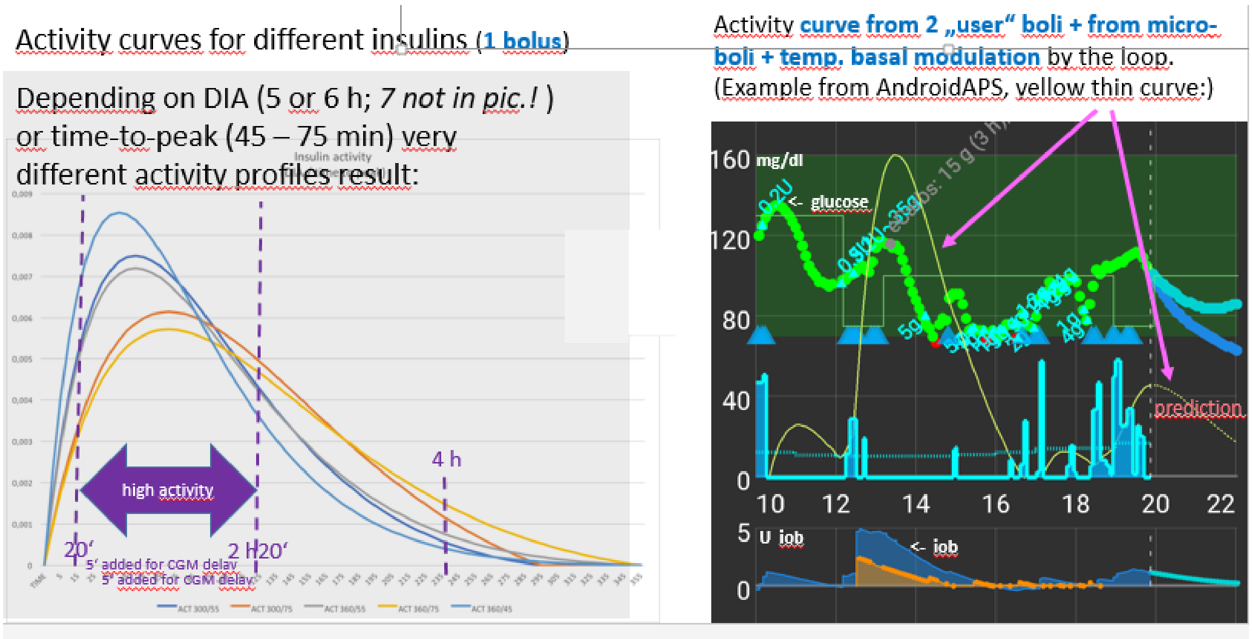
Ideally (like in AAPS), iob is defined as insulin *above the basal needs* as defined in profile. That way, it is always clear how much is there for correction-to-target, or for more carbs becoming absorbed.

- Negative iob then indicates that less than basal need is on bord. This will not occur at meal time.
- Posititive iob indicates there is insulin available to cover more expected carb absorption, or to correct high glucose values. Therefore, the iob should „match" the insulin need for unabsorbed carbs (which can be calculated as =cob/IC factor), plus for corrections (which can be calculated via = (bg – bg_target) /ISF factor).

However, the iob information cannot tell when/how strong this insulin actually provides activity. Therefore an „overall match for the next couple of hours" of iob with cob might not be good enough.

Fortunately, the loop records all insulin deliveries, and calculates the **aggregate activity** curve. This enables predictions for every 5-minute segment in the future. PLUS the loop can calculate, how insulin above (or also below) basal need it gives (withholds), works out „on top" of that activity pattern, increasing (or decreasing) available „power" every 5 minutes in the future. (It immediately adds fully to iob, but that is only the trivial part of the story).

Insulin activity



The **graph on the left** shows for various types of Insulin the time of peak activity, and how it decreases after the peak depending on the set DIA (duration of insulin action; unfortunately, the chart does not include the important 7 h DIA curves). We can see (dotted lines) that active insulin amout can differ by factor of 2 after 15 minutes, or up to factor 3 at 4 hours. So it is very important to give the loop the correct data on the insulin in use.

Beginner loopers often select the DIA too short: One consequence is, that several hours after dinner, their loop is falsely „told“ the bolus is already very weak, practically gone, and will give more correction insulin than would be the case with a longer set DIA. So, it is not necessarily a too strong nightly ISF that can lead to low glucose values!

Refer to https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings/blob/HCL-.-settings-main-repo-(pdf)/Insulins_DIA%20and%20other%20settings_V.3.0.pdf for more on DIA and insulins for looping in general.

The rapid insulins we prefer for looping (blue and gray curves) show a time window with high activity between 20 minutes and 2 hours and 20 minutes after the bolus was given. (Slower insulins would show a longer albeit on average weaker period of high activity, with onset shifted by ~15 minutes, too).

For the meal bolus in hybrid closed loop, it is very important to picture how many grams of carbs will become absorbed in this time window.

With realistic inputs for time-to-peak and DIA, the loop can exactly calculate, for past AND for future hours, how much insulin was or will be active in each 5 minute segment. This is very powerful for the loop‘s decision whether and how much insulin activity to add - or to withdraw via zero-temping - going forward.

The **graph on the right** (main screen AAPS example) shows how the aggregate insulin activity (thin yellow curve) developed between 10 and 20 h, plus how it will develop in the next 2 hours. The prediction assumes, no extra insulin above basal need is given from 20:00h on.

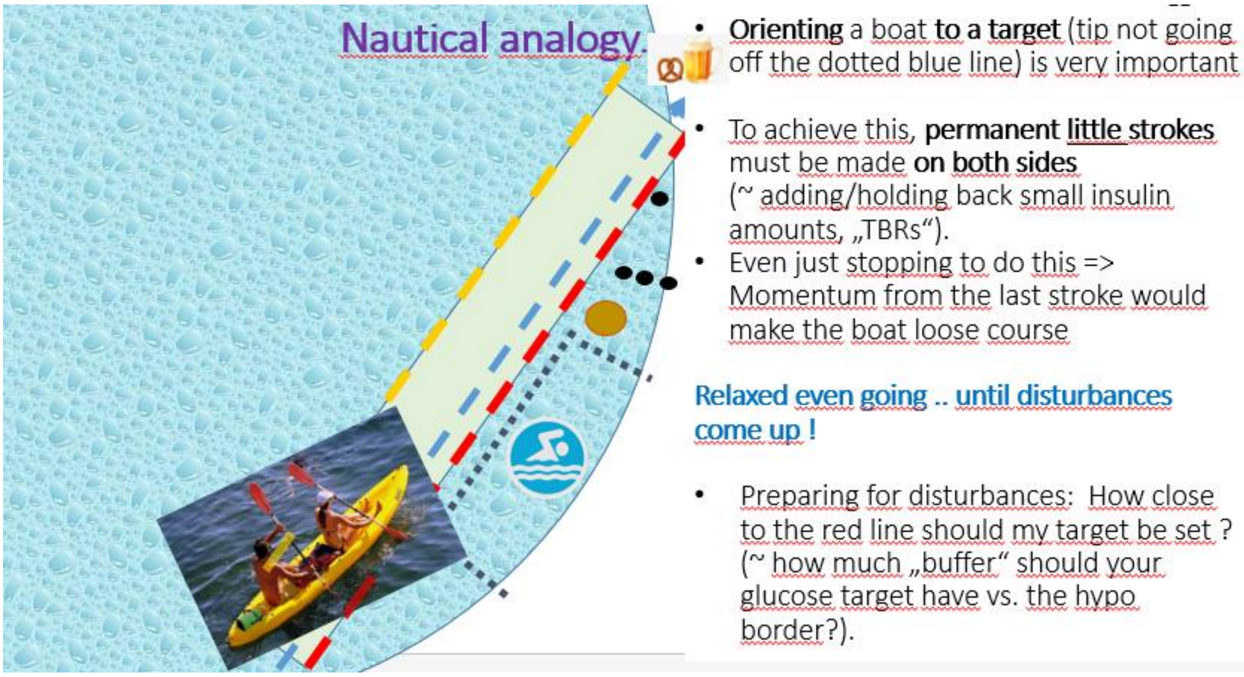
After a big bolus was given at 12:30h, we see in the smaller bottom graph an iob maximum of about 5 units. But – it is not active much, so the green curve in the top graph, the glucose, keeps rising!

It is the (thin yellow) insulin activity curve that tells us better where glucose will be headed.

Sluggish delayed control: Nautical analogy

Our key problem is, that anything we do, just starts to become effective in a 15 to 30 minutes time frame, and smears in its effects over a couple of hours

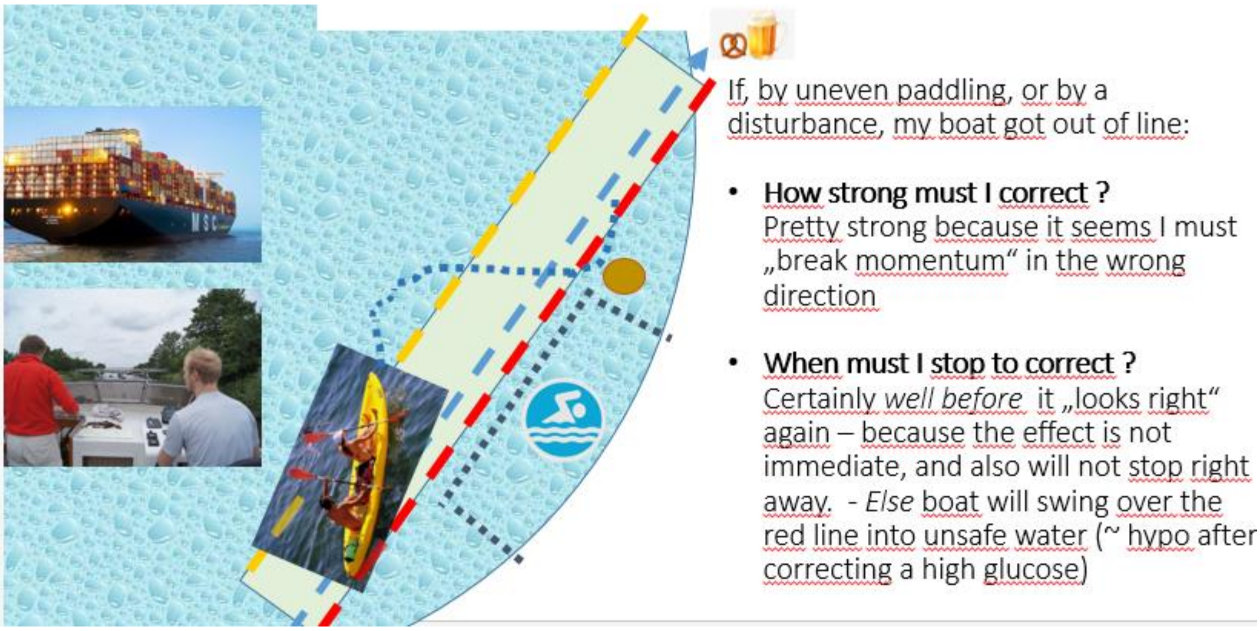
On a paddling tour to the beer garden we could learn a lot about glucose control and looping:



When paddling, it is very important to always orient the tip of the boat towards the target, and, to achieve this, always slight paddling is needed, alternating on the right and the left. (~ In our case, the loop does this fine-tuning by elevating and reducing the temporary basal rates (TBR) all the time).

We are free to select our target. (~ Many loopers select a very low goal. Beginners sometimes think to „lure" the loop in the desired direction with this measure. However, it is generally a good idea to have some room for disturbances, to avoid hectic moves and alarm fatique. Also experienced loopers could be more relaxed, and utilize more options, if not „always" near the red line.

Also, everybody should be aware of the performance of the CGM system (eventually depending on body site, how long in use etc.).



Seasoned navigators learned over years how to deal with sluggish control of their vessel.

However, not knowing enough about it, makes many maneuvers dangerous.

In type1 diabetes we face very similar challenges. Neither the insulin nor the food (glucose) we take will immediately be active in our body. And their effects will be very uneven, but last for hours.

We face a threefold challenge:

1) We must tell our loop certain basic parameters that it needs to keep us in a certain range
2) We must watch and analyze our loop occasionally, in order to learn how to tune certain parameters for better performance in the next days
3) We must become experienced „sugar-captains"

What we advocate here is:

- To give the loop ever improving parameters to stay better in range („tuning")
- To give the loop decisive information for its job (e.g. announcing a meal, specifying the absorption time for some components of it, etc.)
- To know also about LIMITATIONS of the system in use
- To be able to recognize a MISFUNCTION of the system, and have a plan B for that scenario.

Generally, it is important to remain calm, analytic, open to learn, and in good spirit, to make it better next time.

To remain in the target range is the goal. How to achieve this more and more, and also around challenging meals, is our core topic

Basic parameters the loop needs for control

We already saw that the kinetic characteristics (time-to-peak and DIA) of the insulin in use are very important for the loop.
We tell our loop our basal rate to cover our hourly „baseline" insulin need w/o eating, stress, or activity (see also box next page).

DIY loops always calculate the iob (insulin on bord) which says how much above (or below) basal need is available (e.g. to deal with carbs in the next hours). **)

The Carb Ratio IC *(g/U)* describes for how many grams of carb one unit of insulin suffices. More see: https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings/blob/HCL-.-settings-main-repo-(pdf)/IC%20(carb%20ratio)_V.3.1.pdf

In iOS Loop, the absorption times of carbohydrates are also a very important input for each meal. AAPS primarily concludes on carb absorption from the glucose curve (details see section 1.2 in "IC…pdf", linked above)

The sensitivity factor ISF, expressed in *(mg/dL)/U or in (mmol/L)/U,* describes by how much a unit of insulin (from the available iob) will bring down high glucose. This describes our sensitivity to insulin, and usually differs between times-of-day. More see: https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings/blob/HCL-.-settings-main-repo-(pdf)/ISF%20determination_V.3.33.pdf

Important: Depending on your age, sex, lifestyle, health status/medications used, insulin sensitivity can also have "longer waved" swings. Make sure you calibrate for your personal meal management strategy strictly for what is YOUR average normal day.

That will provide a solid basis also for days of altered insulin sensitivity. The open source loop systems offer a myriad of automated and manual options to adjust to (auto-detected, or "communicated") situations with elevated resistance or sensitivity:

- Modifying the glucose target creates counter-momentum for expected disturbances.
- Modulated basal rate and factors („%profile"-changes, „overrides") alter the „aggressiveness" of the loop in special situations (e.g. exercise),
- Moreover, open-source systems allow the user to define *Automations* for the loop to act differently when pre-defined conditions occur. Automations are a fully integrated feature in AAPS (programmable on the smartphone within seconds), and an available

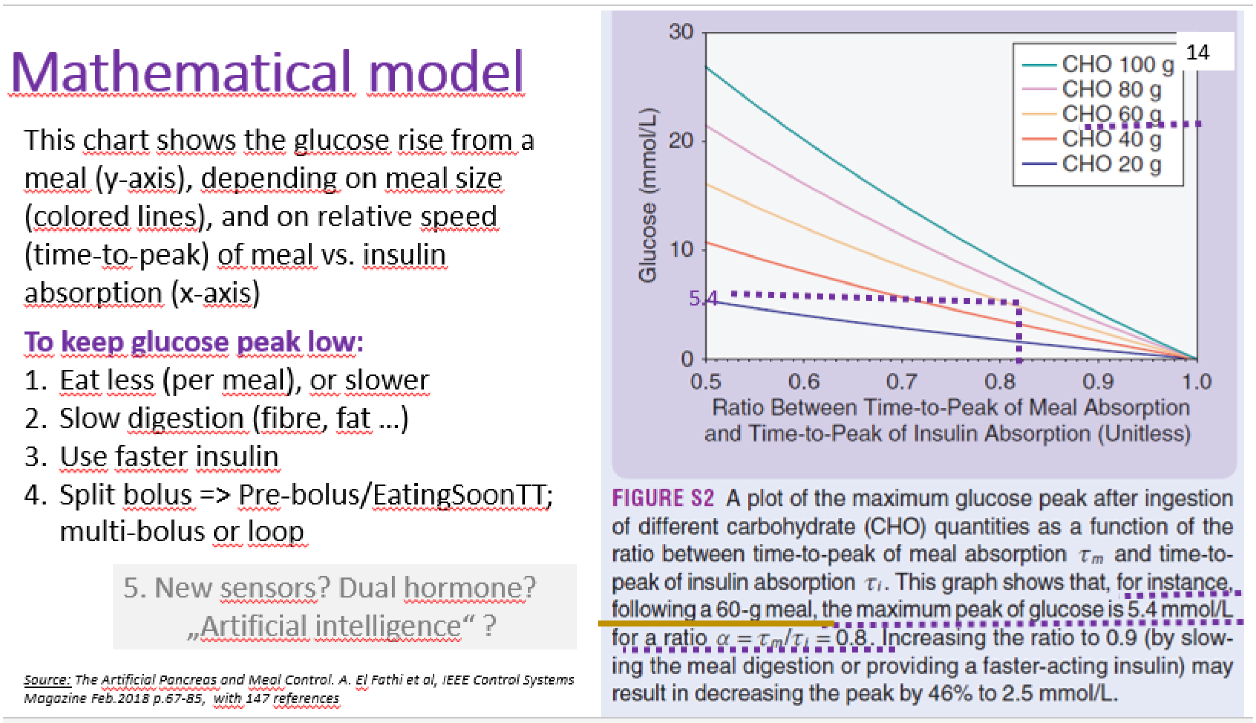
add-on option in iOS Loop (via extra software like IFTTT or Automate!), and also in Trio and iAPS (via “middleware”).

- Some loops analyze changes in insulin sensitivity, and can automatically adjust (Autotune, Autosens, Motion Monitor-based auto-adjust, dynamicISF, autoISF, machine learning).

> Contoured “circadian” basal (and factors) debate:
>
> Pro flat: It works OK to go from a flat (in case of doubt, low) basal, and „invest“ care into validating different IC and ISF factors for several times during the the day and night. *(A contoured basal, but single ISF and IC, is likely worse - and this is what Autotune leads some „trusting“ loopers to do!).*
>
> Pro circadian: Factors AND basal run in typical „biorhythm“ patterns*). The loop has best-possible data if it exactly knows which part of insulin is basal requirement each hour, and hence what the „true“ iob is that is available for managing meals and for bg corrections. Also, if loop does not work (system failure), the set profile basal is truly what keeps you stable.
>
> Con making a big deal out of this question: As insulin sensitivity is affected by many factors (stress, exercise, infection), a lot of more things (like also: does your loop have an Autosens feature etc) is playing into this debate, and everybody must really decide how much she/he is willing to invest, and which result is „good enough“. After all, the loop looks, calculates, every 5 minutes, and CAN correct for things that are a bit „off“.
>
> *) 24 hour circadian patterns change strongly during childhood – see here: https://www.facebook.com/groups/AndroidAPSUsers/permalink/3091340191087377 or also in section 5. of: https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings/blob/HCL-.-settings-main-repo-(pdf)/ISF%20determination_V.3.33.pdf

Mathematical model

Modelling the course of carb absorption and of insulin activity, the effects on glucose peak heights can be quantified . (reference: https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings/blob/HCL-.-settings-main-repo-(pdf)/The%20Artificial%20Pancreas%20and%20Meal%20Control.pdf)



The graph shows on the y axis peak over baseline (in mmol/L), and on the x-axis the relative speed of insulin absorption to carb absorption. (Carb absorption is faster, therefore all values are under 1.0).

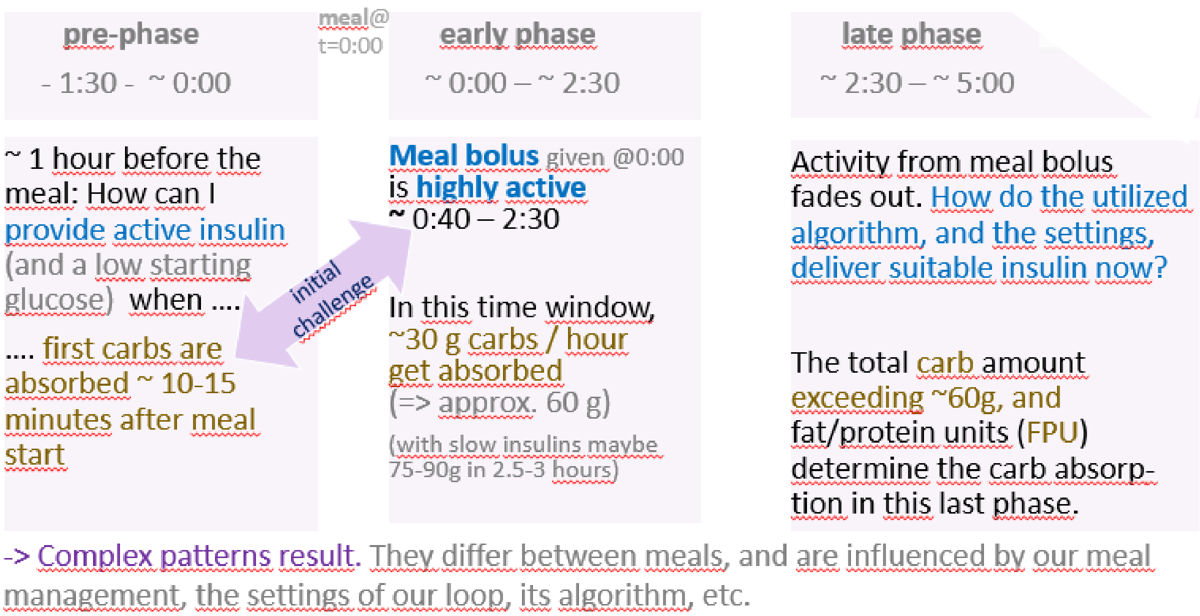
The model allows to calculate how post-prandial glucose peaks can be reduced via:

1. Eating less (per meal; see colored curves), or slower
2. Slower digestion (fibre, fat …). This would move the RATIO of insulin/carb absorption peak on the x-axis to the right.
3. Using faster insulin. Also this would move the RATIO of insulin/carb absorption peak on the x-axis to the right.
4. Splitting bolus => Pre-bolus/EatingSoonTT; multi-bolus, or loop.

New developments like dual hormone systems, or faster responding sensors, might also have potential to lower glucose peaks.

3 meal phases in Hybrid Closed Looping

In Hybrid Closed Looping, we see three phases to each meal that require special attention:



Often we face an unbalanced situation right after meal start, with strongly rising glucose levels. Or, when pre-bolussing, also the opposite could happen, a tendency towards hypoglycemia right around meal start. It is therefore suggested to seek appropriate action already in a **pre-phase**, with the goal to provide active insulin (and a low starting glucose) when first carbs will be absorbed ~ 10-15 minutes after meal start

This marks the beginning of the **early** meal **phase**, in which a constant stream of carbs is absorbed, while the meal bolus develops, and sustains, high activity. In the following main section on the early phase, we look into the options how and when to set the meal bolus.

In the **late phase** of meals, activity from meal bolus fades out. The total carb amount exceeding ~60g, and fat/protein units (FPU) determine the carb absorption in this last phase. In a later chapter we will look how the utilized looping algorithm, and the settings, deliver suitable insulin then.

Dynamic carb absorption

oref(1) “UAM” loops do not require carb inputs. (And in case they are given, they use them only for one of several predictions). Oref(1) loops always put more trust into their own calculations of carb decay, based on development of iob and glucose.

This is explained in section 1.2 of*:* https://github.com/bernie4375/HCL-Meal-Mgt.-ISF-and-IC-settings/blob/HCL-.-settings-main-repo-(pdf)/IC%20(carb%20ratio)_V.3.1.pdf

Other references to the oref(1) algo see:
https://androidaps.readthedocs.io/en/latest/Usage/COB-calculation.html?highlight=cob%20calculation#cob-calculation

https://androidaps.readthedocs.io/en/latest/CROWDIN/de/Usage/COB-calculation.html#erkennung-fehlerhafter-cob-werte

Dana Lewis: https://de.loopercommunity.org/uploads/short-url/grgfNjbECtyau5nwtQH8xhaMRyI.pdf