# Digital Glucose Modelling

## Bachelor Thesis

by

D. van de Pol
B.R. Menkveld

Student numbers: D. van de Pol - 5646863
B.R. Menkveld - 5786290

Thesis committee:
Dr. F. A. Cardoso, TU Delft, supervisor
Dr. I. Ercan, TU Delft
Dr. Ir. R. F. Remis, TU Delft

TU Delft

# Abstract

A common method of diabetes treatment is monitoring of blood glucose using a continuous glucose monitor(CGM). These monitors come in different forms, but are often uncomfortable because they penetrate the skin with needles. For this project, an alternative, smaller sensor is explored for which the design process is centred. This sensor is inserted into the dermis layer of the skin, measuring interstitial fluid glucose and should be less invasive and more comfortable to the user. The project is divided into two parts, one discussing the sensor reading circuit, and the other the digital processing of sensor information. This report is on the digital processing, and will convert sensor data into blood glucose concentration estimates. To achieve this, various mathematical models and filter techniques are implemented and evaluated. The subsequent system is subdivided into modules, each performing a task in the total process line. The report gives a general introduction to the broad topic and eventually outlines the structure and motivation behind these models. In the end a Kalman filter and biophysical blood-interstitial fluid dynamics model are shown to improve performance over a linear model in an in silico test environment. This work represents a step towards a working glucose sensor which can be used in CGM solutions, with the potential application in both clinical and at-home diabetes management.

# Preface

This report is part of a broader project aimed at designing a working glucose sensor for a continuous glucose monitoring (CGM) system. From a provided Sensible[1] sensor by the supervisor, from which assumptions are made. The team is tasked to design and build both the analogue and digital components of such CGM. While the analogue circuitry is responsible for capturing signals related to interstitial fluid (ISF) glucose levels, this report will have its focus on the digital processing pipeline. This project was conducted as an Electrical Engineering Bachelor's Thesis.

Menkveld appreciates how useful their minor of Mathematics Master courses in Bologna was for this project. They were able to use their newly gained proof writing skills and numerical method knowledge to develop a purpose-made algorithm for modelling a yet unfamiliar biophysical process, which they perceived as a motivating challenge. Menkveld is also specifically interested in the subject of the project, as they hope to specialise in the biomedical field. They would like to give special thanks to their partner Azad, for comforting them when the project reached its stressful finalisation stage.

D. van de Pol
B.R. Menkveld
Delft, June 2025

# Contents

# 1 Introduction

## 1.1. Background

Diabetes mellitus is a chronic metabolic disorder characterised by insufficient insulin production or insulin resistance. According to the WHO [2], over 830 million people had diabetes in 2022, significantly higher from the 200 million in 1990. And in 2021, diabetes was the direct cause of 1.6 million deaths. This presents a significant burden on healthcare systems worldwide. Accurate and continuous glucose monitoring (CGM) is critical for effective diabetes management, allowing patients and clinicians to make informed therapeutic decisions in real-time.

Traditional blood glucose monitoring techniques, such as finger-prick tests, provide discrete glucose measurements with limited temporal resolution. These methods are invasive, often painful, and lack the frequency necessary to detect rapid fluctuations in glucose levels. Continuous glucose monitoring systems, however, offer a transformative improvement over traditional techniques as they measure glucose levels in the interstitial fluid (ISF) continuously, typically every few minutes. While only measuring inside the skin, as opposed to having to do deeper blood draws. These CGMs provide a dynamic view of glucose trends and enable more responsive management strategies such as closed-loop insulin delivery (artificial pancreas) and help patient management of glucose, preventing life-threatening situations as extreme glucose fluctuations, hypo- and hyperglycemia. Currently, more than 9 million people use such CGMs worldwide for diabetes management [3].

The sensor at the heart of CGMs is often based on enzymatic electrochemical principles, where glucose concentration is indirectly inferred from a measured electrical signal, such as current resulting from applying a working potential (chronoamperometry), resulting in the oxidation of glucose. However, translating these signals into reliable estimates of blood glucose requires signal processing with the use of models and filters to relate these signals to glucose concentration and additionally compensate for physiological delays, sensor shift, and noise. In this context, digital signal processing and mathematical modelling become indispensable for ensuring accuracy and usability.

### 1.1.1. Interstitial Fluid

Ribet [4] stipulates the role of interstitial fluid (ISF), which makes up a quarter of the water contents of the human body and can be found in virtually any intertissular space in the body. In addition to water, the ISF contains salts, glucose, small proteins, lipids, amino acids, fatty acids, hormones, neurotransmitters, and other small bio-molecules.

According to Ribet [4], because of ISF's similarity with blood plasma, ISF is gaining growing interest as a monitoring matrix. Moreover, he says, "In the hypodermis, inhomogeneity in size of the adipocytes and scarcer capillary distribution detrimentally affect the local ISF analyte concentrations, whilst in the epidermis, blood capillaries are not present at all. As a consequence, the dermis is the ideal measurement location within the human skin."

The sensor that will be used will reach only the dermal layer of the skin; therefore, the dynamics between the blood glucose and the ISF must be considered if one is to monitor the users current glucose concentrations. This report will describe a biochemical model which describes the kinetics between ISF glucose and blood glucose.

## 1.2. Monitoring

According to Ribet [4] there a three important considerations:

1. Damage is caused by large oscillations rather than constant hypo- or hyperglycemia. Therefore, it is important to measure consistently over time, not miss oscillations, and act quickly.
2. Regular pricking for direct blood glucose measurements (for monitoring or calibration) exposes patients to a higher risk of infections;
3. Pain and discomfort may often result in reduced patient compliance.

To achieve widespread distribution of care, the following requirements must be met:

1. Reduce the monetary cost;
2. Minimise the invasiveness and discomfort;
3. Improve reliability.

It is thus important to maximise patient safety and comfort. Therefore, there is a need for reliable and accurate CGMs which are as least invasive as possible. Additionally, typical glucose levels and conditions can be found in Appendix A.

## 1.3. Performance measure

The Parkes error grid, also known as the consensus error grid, is a measure developed to assess the performance of new blood glucose monitors (BGMs). The new BGM's blood glucose concentration(BG) values are plotted against a reliable BGM sensor. Points inside the A and B regions are precise enough to be used for therapeutic decisions, but points in the other regions can have undesired consequences for therapeutic decisions, like insulin delivery. The ISO 15197:2013[5] standard requires 99% of measurements to be within the A and B regions for the BGM to be reliable enough for therapeutic use[5]. Both type 1 and type 2 diabetes have their own Parkes error grid. Additionally, at least 95% of BGM results have to be within $\pm$15 mg/dl at glucose concentrations $<$100 mg/dl and within $\pm$15% at $\geq$100 mg/dl compared to the true glucose values.

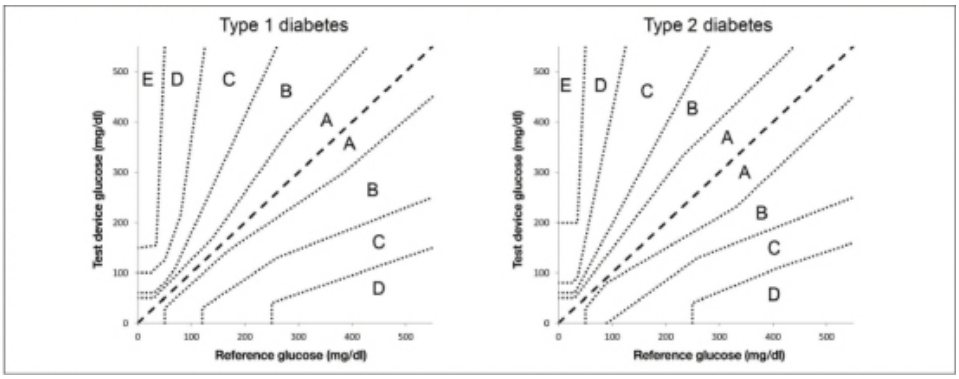


Figure 1.1: Parkes error grid

Another popular performance measure is the Mean Absolute Relative Difference(MARD)[6]. While the Parkes error grid measures the reliability of a BGM, MARD is a measure of mean accuracy. MARD can be calculated using the following equation.

$$MARD = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{x}_i - x_i}{x_i}\right|$$

# 2

# Programme of Requirements

## 2.1. Mandatory requirements

Mandatory requirements are requirements that the product should always, at the very least, comply with. These requirements can be subdivided into functional requirements and non-functional requirements. Functional requirements are things the product has to do, and non-functional requirements are the qualities or attributes which the product must have.

### 2.1.1. Non-Functional requirements

- The sensor must be able to meet the glucose accuracy standard. (ISO 15197:2013)
- The sensor should be able to communicate wirelessly.

### 2.1.2. Functional requirements

- The sensor must correlate sensor readings to blood glucose.
- The sensor must be able to communicate between the in-person device and the external device.
- The sensor must transmit data every 5 minutes to a paired external device.
- The sensor must allow calibrations at regular intervals, e.g., every 8 hours.

## 2.2. Trade-off requirements

These are criteria which should be complied with as much as possible.

- The sensor should be usable in a closed-loop system for an artificial pancreas.
- The sensor should be as real-time as possible, meaning measurement delay should be as minimal as possible.
- Battery life should be preserved as much as possible.
- The sensor should be able to store up to 12 hours of measurements in case of connection loss.

## 2.3. Objective

For digital processing, the purpose is to relate the sensor output readings to an estimated glucose reading. To realise this, the following targets should be realised with the PoR in mind.

- Find an appropriate way to measure the sensor output.
- Translate these readings into glucose levels in the interstitial fluid (ISF).
- Relate these ISF glucose readings to blood glucose levels.

# 3

# System

This report will discuss how sensor readings are converted into BG measurements. This system can be divided into the following three blocks:

- The Bluetooth module is responsible for transmitting and receiving sensor readings from the sensor to the computer. On the sensor side, it is integrated with the microcontroller, which handles the control signals of the sensor circuit. This module is discussed in chapter 7.
- A sensor-ISFG module. This is responsible for the conversion of the transmitted output readings to an estimated interstitial fluid glucose, as the sensor resides inside the skin, in the interstitial fluid. This must be an appropriately calibrated model. Methods to determine ISFG will be described in Chapter 5.
- An ISFG-BG module. This module models the dynamics between BG and ISFG to estimate BG based on ISFG measurements. This will be described in Chapter 6.

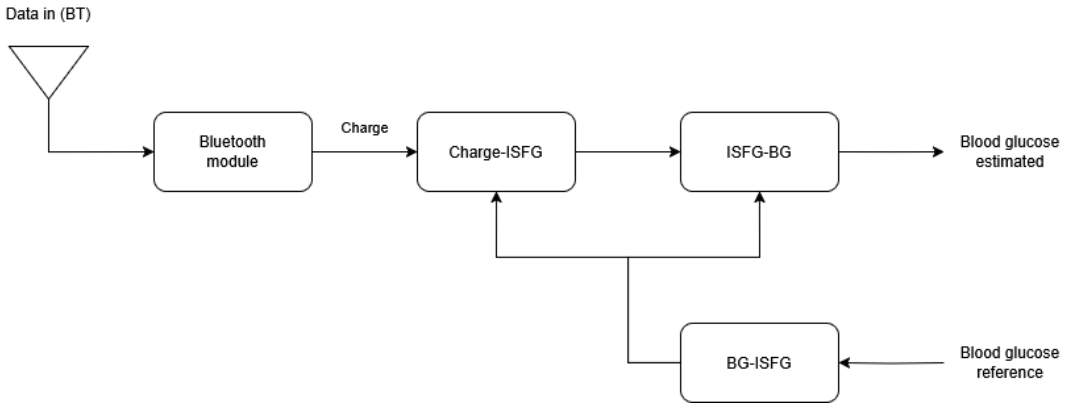


Figure 3.1: Block diagram of the total system for the digital part.

# 4

# Current

## 4.1. Current measurements

In chronoamperometry, a fixed potential step is applied at the working electrode, and the resulting current $i(t)$ is monitored. Under diffusion control at a planar electrode, current follows the Cottrell equation: [7]

$$i(t) = \frac{nFAC_0D^{1/2}}{\pi^{1/2}t^{1/2}} \tag{4.1}$$

where n is electrons transferred per molecule, F Faraday's constant ($C/mol$), A electrode area ($cm^2$), D diffusion coefficient ($cm^2/s$) and C bulk glucose concentration ($mol/cm^3$). Thus $i(t) \propto C_0 \cdot t^{-1/2}$.
In practice, the very early current (t->0+) includes a non faradic capacitive current component due to captive charges between the electrode and analyte. This current decays by 1/t [7]. One usually ignores the first few milliseconds [7], or wait at least $5R_uC_d$, with $R_u$ the uncompensated resistance (solution resistance between working electrode and reference electrode) and $C_d$ the double-layer capacitance. [[8], p. 264] and analyse the later diffusion-controlled decay.
Because i(t) is non-steady per Cottrell equation, glucose concentration can be extracted by several approaches;

1. Choose a fixed-time current. Choose a time $t_0$ after the initial charging current spike. According to Cottrell, $i(t) \propto C_0 \cdot t^{-1/2}$. Meaning $i(t_0)$ is directly proportional to glucose concentration. In theory, measuring $i(t_0)$ against a linearly calibrated model should yield a glucose concentration.

2. Apply coulometry. Compute the total charge $Q(t) = \int_0^\tau i(\tau)d\tau$ over the transient.

$$Q(t) = \int_{t_0}^{\tau} \frac{nFAC_0D^{1/2}}{\pi^{1/2}t^{1/2}}\,dt = 2nFAC_0D^{1/2}\sqrt{\frac{\tau - t_0}{\pi}} \tag{4.2}$$

   By computing the total charge for a given time interval, glucose concentration can be extracted since $Q \propto C_0$

3. Curve fitting. Fit the entire current transient to a theoretical model that contains the concentration as a parameter. And using this fitted curve, extract a glucose concentration.

4. Steady-state or plateau current. Ideally, under diffusion control, the Cottrell equation applies, and i(t) approaches 0. However, in some applications of chronoamperometry, the transient may reach a quasi-steady current instead, just as in continuous glucose monitoring, as new glucose tends to diffuse towards the sensor from its surroundings. For circular microelectronics, the current is given by: [9]

$$I_{ss} = 4nFDC_0a \tag{4.3}$$

   With a, the radius of the electrode. Some continuous glucose monitoring systems use the steady-state current as a reference to determine the glucose concentration. Bard, Faulkner, and White [8, p. 805]

## 4.2. Measuring

It is important to choose how the decaying current will be measured and if there are any other unwanted effects on the current.
Choosing a single point at a fixed time for every measurement is highly prone to noise, and measuring the whole current and curve fit is too exhaustive, as measurement would take too long, and with a limited battery, this would be too power-intensive. Also, waiting for steady-state current can take long, for the Sensible sensor used as reference, this would settle at 90% after 300 seconds. [10]. Instead, the decision was made that the current gets integrated analogously in a limited time window, with the use of an integrator and microprocessor controlling the time window. The next step would be to decide when to start with the integration and how long this time window would be.
As explained earlier, the running current does not follow Cottrell's equation completely. At the beginning, the current is dominated by a charging current after which the Cottrell behaviour follows, but eventually reaches a steady-state current. Both the Cottrell and steady-state currents are proportional to the glucose concentration. Per equations, 4.1, 4.3. However, the charging current is given by

$$i = \frac{E}{R_s} e^{-t/R_s C_d} \tag{4.4}$$

This current is not proportional to glucose concentrations, and thus, when measuring, one should avoid the influence of the charging current.
With $R_s$, the solution resistance between the working and reference electrode. $C_d$ this capacitance of the working electrode and $E$ the applied potential. $R_s$ can be calculated by the following for ultra microelectrodes (UME),[8]

$$R = \frac{1}{4\pi k r_0} \tag{4.5}$$

with $k$ the solution conductivity for typical ISF $k = 2.0 S/m^2$[11], and the electrode area of the sensor is $A = 0.0306 mm^2$ by approximating this as a circular electrode yields $r_o = 9.87 \cdot 10^{-5} m$. Filling in equation 4.5 yields $R = 403\Omega$. Typical UME sensors have a capacitance between 10-40$\mu F/cm^2$. Taking the highest value and multiplying it by the electrode area yields $C = 12.2nF$. Ultimately the time constant is $\tau = RC = 4.92\mu s$. By filling in equations 4.4, 4.1, 4.3 for a glucose concentration of $80 mg/dL$ and diffusion coefficient for ISF in skin given by $D = 2.64 \cdot 10^{-6} cm^2/s$ [12]. Gives the following figure;

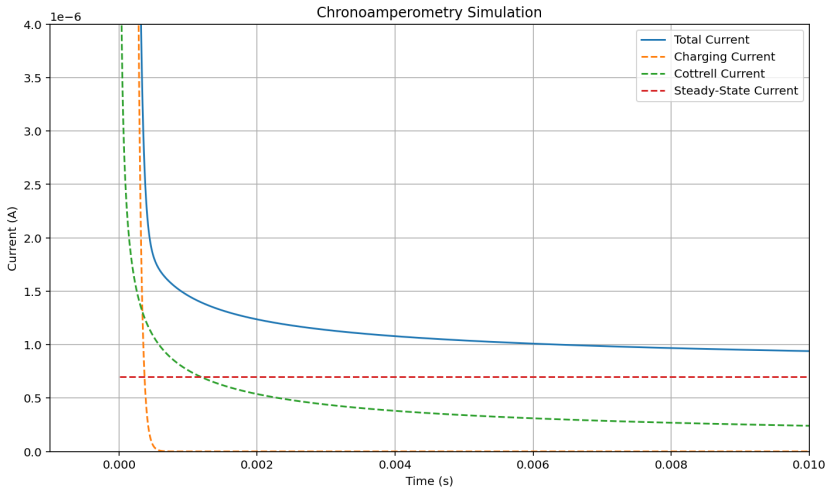


Figure 4.1: Charging, Cottrell and steady state current contributions

To conclude, the current will be measured by integrating the current over a time window. As said in the previous section, during measuring, one should typically wait $10RC$ or a few microseconds before doing any meaningful measurements, as at the beginning, there is a charging current which is not proportional to glucose concentrations. From the plot and from the calculations $10RC = 49.2\mu s$, the effect of this current is short-lived. In reality, this effect may persist longer, but waiting a few seconds before starting the integrator should be sufficient enough and this could be experimentally fine-tuned.

For now, the decision is made to start the integrator 2 seconds after the sensor is turned on, with an integration window of 20 seconds. The exact implementation can be found in section 7

# 5

# Charge-ISF Glucose model

As the current of the sensor is getting integrated by an integrator analogy, for a fixed time window. The total sensor circuit has an output charge. This charge measurement should be made with the use of an appropriate model that is related to the measured glucose concentration.

## 5.1. Linear Model

CGM sensors measure glucose concentration indirectly, by measuring the current produced by the glucose-oxidase electrochemical reaction. When calibrating a sensor, an estimation will be made of a mathematical relationship that converts measured current or, in this case, measured charge to glucose concentration. (Adapted from G. Acciaroli et al. [13]). Let x(t) be the glucose concentration profile, y(t) the charge and f() the function of parameters $\mathbf{P} = [p_1, p_2, ..., p_n]$ that relates x(t) to y(t). And e(t) is the measurement error.

$$y(t) = f(\mathbf{P}, x(t)) + e(t) \tag{5.1}$$

A numerical value $\hat{\mathbf{P}}$ of the calibration vector can be provided using parametric estimation techniques. This step can be continuously repeated each time a new blood glucose reference is available, which consequently updates $\hat{\mathbf{P}}$. From the estimated $\hat{\mathbf{P}}$ and a measured current a glycemic profile $\hat{x}(t)$ can be obtained by inverting the function f().

$$\hat{x}(t) = f^{-1}(\hat{\mathbf{P}}, y(t)) + e(t) \tag{5.2}$$

The choice of f() is critical. It has to be invertible and must describe the relationship between charge and glucose concentration. [13] The most common and simplest calibration function adapted is a first-order time-independent linear function, with $\mathbf{P} = [s, b]$. With s, sensor sensitivity and b, baseline. This linear assumption will be sufficient in this case, as the provided sensor,r as described in [1], has a linear relationship between current and glucose levels. Respectively, this will result in;

$$y(t) = f(\mathbf{P}, x(t)) = s \cdot x(t) + b + e(t) \tag{5.3}$$

When, for instance, two blood glucose references are available at times $t_1$ and $t_2$. Numerical values of $\hat{s}$ and $\hat{b}$ can be found using so called two point calibration;

$$\begin{cases} \hat{s} = \dfrac{y(t_2) - y(t_1)}{x(t_2) - x(t_1)} \\ \hat{b} = y(t_2) - \dfrac{y(t_2) - y(t_1)}{x(t_2) - x(t_1)} \cdot x(t_2) \end{cases} \tag{5.4}$$

In general, when multiple pairs of electrical charge and blood glucose samples are available at time $t_i$ $(i = 1, 2, ..., N)$. Linear regression is used to fit the sensitivity and baseline to the data. The numerical determination of the model parameters can be determined by minimising the residual sum of squares.

$$[\hat{s}, \hat{b}] = \underset{s,b}{\arg\min} \sum_{i=1}^{N} e(t_i)^2 = \underset{s,b}{\arg\min} \sum_{i=1}^{N} (y(t_i) - \hat{y}(t_i))^2 \tag{5.5}$$

with $y(t_i)$ the blood glucose measurement and $\hat{y}(t_i) = s \cdot x(t_i) + b$ the model prediction. However, it is also possible to apply calibration using one-point calibration to measure the sensor's sensitivity using one pair of blood glucose and sensor charge readings. Assuming the baseline (y-intercept) were to be known (usually b = 0 [14]), the following can be used;

$$s = (y - b)/x \tag{5.6}$$

Finally, the calibrated glucose profile is obtained from the measured charge with the established calibration parameters by the following: [13]

$$\hat{x}(t) = \frac{y(t) - \hat{b}}{\hat{s}} \tag{5.7}$$

One major problem with this linear approach is that it does not account for the delay between blood glucose and sensor signal. (delay BG and ISF) as it is influenced by the BG-to-ISF dynamics. One should either add another model describing this ISF-BG relationship or use models which can directly relate sensor readings to blood glucose.

### 5.1.1. Sensor drift

The various methods of calibrating the data over all capillary blood glucose measurements performed define the accuracy of the sensor. However, one thing overlooked in the calibrations is the underlying drift of the sensors. If there is a specific relationship between the sensor signal and the glucose levels and no other factors were to affect the signal resulting in drift, in the event of the sensor signal showing linear drift (= change of the sensor sensitivity of time due to the degradation of the enzymes on the electrode) the linear relationship explained earlier between current and glucose concentration can be expanded by including a drift component $+ct$ with c the time-independent drift factor.[15]. However, besides expanding the linear relationship, frequent recalibration is required as "Recalibration of the CM system at a certain time interval is mandatory with most types of sensor, as it is virtually impossible to predict the drift of the sensor signal" per Lodwig and Heinemann [15].

## 5.2. Kalman Filter

An alternative implementation, which should better handle noise and uncertainties a different method than the linear model can be used to estimate the glucose measurements. Namely, a Kalman filter.

A Kalman Filter (KF) is an algorithm that provides estimates of unknown variables (states) in a dynamic system using a series of noisy measurements over time. It is optimal for linear systems with Gaussian noise, providing a recursive method to minimise the mean-squared estimation error. A KF is widely used in areas which use noisy sensors, like in control systems, robotics, signal processing and in this case also for medical sensing. At its core, a Kalman Filter assumes a system can be modelled using a state transition model, which tells how the internal state evolves. And a measurement model, how the observed data relate to the internal state.

### 5.2.1. Kalman Filter Theory

A discrete-time linear system is typically described by the following equations:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{5.8}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{5.9}$$

Where $\mathbf{x}_k$ is the state vector at time $k$, $\mathbf{F}$ is the state transition matrix which models how the system evolves from time $k-1$ to $k$, $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q})$ is the process noise, $\mathbf{z}_k$ is the measurement, $\mathbf{H}$ is the measurement matrix mapping the true state to the observed space, $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$ is the measurement noise.
The algorithm has two phases:

Prediction

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1} \tag{5.10}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q} \tag{5.11}$$

$\hat{\mathbf{x}}_{k|k-1}$ is the predicted state estimate, $\mathbf{P}_{k|k-1}$ is the predicted error covariance.

Update

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T \left(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}\right)^{-1} \tag{5.12}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}\right) \tag{5.13}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1} \tag{5.14}$$

$\mathbf{K}_k$ is the Kalman gain, $\hat{\mathbf{x}}_{k|k}$ is the updated state estimate, $\mathbf{P}_{k|k}$ is the updated error covariance.
Essentially during the prediction, given the filters internal state $\hat{\mathbf{x}}_{k-1|k-1}$ and error covariance $\mathbf{P}_{k-1|k-1}$, a prediction is made for the next state $\hat{\mathbf{x}}_{k|k-1}$ and error covariance $\mathbf{P}_{k|k-1}$ for a given time step. After which, an update (correction) is made.
The Kalman gain $\mathbf{K}_k$ essentially tells how much weight the filter gives to new measurements versus the current predicted estimate when updating the state. If measurements are noisy ($\mathbf{R}$ is high), this will result in a lower Kalman gain, which results in the filter relying more on its prediction. If predictions are uncertain ($\mathbf{P}_{k|k-1}$ is large), this will result in a higher Kalman gain, giving more weight to measurements. $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ are the new estimates, combining both prediction and measurements based on the Kalman gain.

### 5.2.2. Unscented Kalman Filter (UKF)

The Kalman Filter is optimal only for linear systems. When the process or measurement models are non-linear, the filter may yield poor estimates. The Unscented Kalman Filter (UKF) is an extension that addresses this by handling non-linear state transitions and non-linear measurement functions without the need for linear approximations as is done in Extended Kalman Filters (EKF). The UKF uses the Unscented Transform to better estimate the mean and covariance of a random variable undergoing a non-linear transformation.
Instead of linearising the system, the UKF deterministically samples sigma points around the mean, propagates them through the non-linear functions, and reconstructs the posterior distribution [16].

### 5.2.3. Application

State Representation We model the state as:

$$\mathbf{x} = \begin{bmatrix} G \\ \dot{G} \\ a \end{bmatrix} \tag{5.15}$$

Where; $G$ is the blood glucose concentration (mg/dL), $\dot{G}$ is the rate of change of glucose, $a$ is the sensor gain factor, which may drift over time.

Model Equations

Process function $f(\cdot)$:

$$G_{t+1} = G_t + \dot{G}_t \cdot \Delta t \tag{5.16}$$
$$\dot{G}_{t+1} = \dot{G}_t \tag{5.17}$$
$$a_{t+1} = a_t \tag{5.18}$$

Measurement function $h(\cdot)$:

$$z_t = a_t \cdot G_t \tag{5.19}$$

The measurement model is nonlinear due to the multiplication of two states ($a$ and $G$). This nonlinearity justifies the use of a UKF over a standard KF. This measurement function is used to relate glucose concentration to charge. As the output of the sensor is a charge reading.

Implementation Details We implement the filter in Python using the `filterpy` library by Roger R. Labbe Jr. [17]. Key components:

- `MerweScaledSigmaPoints` for sigma point generation.
- `UnscentedKalmanFilter` class for UKF filtering logic.
- Custom functions `fx()` and `hx()` for process and measurement models.
- Tuning parameters: initial state, process noise covariance $\mathbf{Q}$, and measurement noise covariance $\mathbf{R}$.

The goal of the filter is to track glucose over time, with the current state of $G$, it predicts $G_{t+1}$ given $\dot{G}$ and during the update stage, it will give the best estimate given this prediction and measurement. Also, the filter had the additional support for periodic calibration (e.g., when a true glucose value is known). This is done by setting the latest $G_t$ to the reference value and its error covariance $P_t$ close to 0. Which will weigh this reference point heavier. The filter implementation can be found in Appendix C.1.

# 6

# Blood-ISF Glucose concentration Model

The sensor of the CGM in this paper is inserted into the skin, where it measures the glucose in the ISF. However, the ISF Glucose concentration(ISFG) is not the same as the Blood Glucose concentration(BG). The ISFG exhibits a time lag of approximately 5 to 6 minutes compared to BG in overnight fasted healthy adults [18], but the full BG-ISFG dynamics consists of more than just the time lag[19]. The purpose of the CGM is to give an estimate of the BG to the user, so a conversion should be made from ISFG to BG. The blood-ISF glucose concentration model aims to model the dynamics between blood glucose and ISF glucose, allowing BG to be determined through measured ISFG. The BG is often determined using a linear model directly from a characteristic of the Cortrell equation to the BG[20]. This can result in lower accuracy because it doesn't factor in the more complicated dynamics caused by the underlying biophysical processes. The goal of the model described in this section is to account for these biophysical processes and demonstrate higher accuracy and reliability compared to a linear model.

## 6.1. Linear Alternative

A common method to determine BG from an ISF glucose sensor is by using a linear model between a Cottrell characteristic value and BG. In this paper, the total charge of a time window over the Cottrell equation is used as described in section 4. This charge is then used to determine the ISFG. For the linear model, this ISFG can then be linearly related to BG using 2 calibration values as described below.
$x$: ISFG value.
$y(x)$: BG value for $x$.
$(x_0, y_0)$, $(x_1, y_1)$: ISFG-BG calibration values. BG is recorded using a reliable BG monitor like a blood-prick device.
The linear model proposes that $y(x) = ax + b$.
The constant $a$ can be determined using estimator $\hat{a} = \frac{y_0 - y_1}{x_0 - x_1}$. The constant $b$ can be determined using the estimator $\hat{b} = y_0 - \hat{a}x_0$.
The BG can then be determined using $\hat{y}(x) = \hat{a}x + \hat{b}$.

## 6.2. T. Shi model

T. Shi, et al. described a model between the ISFG and BG based on the biophysical processes at play[19]. The model consists of three parts. The gradient-driven blood-ISF transport, the pressure-driven blood-ISF transport, and the ISF diffusion.

### Gradient driven blood-ISF transport

The gradient-driven transport is the transport of glucose from the blood to the ISF due to the difference between their concentrations. The formula simplified to use only the blood glucose concentration by using the extraction of glucose for relating the ISFG to the BG, which gives the fraction term in the equation. This is related to the blood flow velocity $F$ and the permeability of glucose through the

capillary wall $P$. The gradient-driven transport equation is given by 6.1 with $A$ as the diffusion area.

$$C_{fick}(t) = C_{blood}(t)\left(1 - \frac{\left(1 - e^{-PA/F}\right) \times F}{P}\right) \tag{6.1}$$

### Pressure driven blood-ISF transport

The other type of transport through the capillary wall is pressure-driven. Starling's principle states that fluid is exchanged between the capillary and ISF due to pressure differences. This is modelled using equation 6.2.

$$C_{starling}(t) = \frac{J_{gain}}{J_{gain} + J_{loss}} C_{blood}(t) + \frac{J_{loss}}{J_{gain} + J_{loss}} C_{isf}(t) \tag{6.2}$$

$$J_{gain} = K_f \left( \int_{P_a}^{\sigma\pi_c} \frac{P_c - P_a}{P_v - P_a} \cdot dP_c \right)$$

$$J_{loss} = K_f \left( \int_{\sigma\pi_c}^{P_v} \frac{P_v - P_c}{P_v - P_a} \cdot dP_c \right)$$

### ISF diffusion

The ISF diffusion is the diffusion of glucose from the capillary wall, from the blood-ISF transport, through the ISF. The diffusion is governed by Fick's first law, given in equation 6.9. The ISF is assumed homogeneous at $t = 0$, $C_{isf}(x, 0) = C_{isf0}$. If the concentration at $x = 0$ is then fixed at $C_{blood-isf}(t)$ the solution of the system becomes equation 6.4, assuming $C_{blood-isf}$ is constant. erf is the Gauss error function. However, the constant source assumption can influence accuracy and the model can not be used in this form to make BG estimations from ISFG measurements, so it was decided to develop an alternative method that works for BG estimation.

$$C_{blood-isf}(t) = C_{starling}(t) R_{starling} + C_{fick}(t)(1 - R_{starling}) \tag{6.3}$$

$$C_{isf}(x, t) = C_{blood-isf}(t) - \left(C_{blood-isf}(t) - C_{isf0}(t)\right) \operatorname{erf}\left(\frac{x}{2\sqrt{Dt}}\right) \tag{6.4}$$

### Combined transport

In order to simplify the glucose transport through the capillary wall, the transport is reduced to the form in equation 6.5, where the transport is linearly related to the BG and ISFG at the capillary wall with the coefficients $K_b$ and $K_i$ respectively. These coefficients are defined in equations 6.6 and 6.7.

$$C_{blood-isf}(t) = K_b C_{blood}(t) + K_i C_{isf}(t) \tag{6.5}$$

$$K_b = R_{starling} \frac{J_{gain}}{J_{gain} + J_{loss}} + (1 - R_{starling}) \left(1 - \frac{\left(1 - e^{-PA/F}\right) \times F}{P}\right) \tag{6.6}$$

$$K_i = R_{starling} \frac{J_{loss}}{J_{gain} + J_{loss}} \tag{6.7}$$

## 6.3. ISF Diffusion

The diffusion in the interstitial fluid is described by two equations, the continuum equation with source, equation 6.8, and Fick's first law on diffusion, equation 6.9. These equations can be combined to form equation 6.10. There are two walls containing the ISF, the capillary wall and the ISF boundary, at $x = 0$ and $x = x_{bound}$ respectively. There is no gradient-driven transport through the walls, giving $J_{gradient}\big|_{x \in \{0, x_{bound}\}} = 0$.

$$\frac{\partial n}{\partial t} + \frac{\partial J_{gradient}}{\partial x} = J_{source} \tag{6.8}$$

$$J_{gradient} = -D \frac{\partial n}{\partial x} \tag{6.9}$$

$$\frac{\partial n}{\partial t} = D\frac{\partial^2 n}{\partial x^2} + J_{source} \tag{6.10}$$

With:

| | | |
|---|---|---|
| $J_{gradient}(x,t)$ | : | gradient driven transport at $(x,t)$. |
| $n(x,t)$ | : | glucose concentration at $(x,t)$. |
| $J_{source}(x,t)$ | : | The external glucose influx at $(x,t)$. |
| $D$ | : | The diffusion coefficient of the interstitial fluid. |

## Discretisation

To make computing easier, the domain will be discretised and numerical methods will be applied for solving the differential equations.
Let us define $X = \{x \in \mathbb{Z} \,:\, 0 \leq x \leq l\}$ with $l \in \mathbb{N}$. Then $n(x,t) \equiv n_{x,t}$ denotes the glucose concentration at $(x,t)$ with $x \in X$ and $t \in \mathbb{N} \cup \{0\}$. We define the spatial derivative as follows

$$\left.\frac{\partial y}{\partial x}\right|_{x,t} \equiv y^x_{x,t} = y_{x+\frac{1}{2},t} - y_{x-\frac{1}{2},t} \tag{6.11}$$

So that the second derivative becomes

$$\left.\frac{\partial^2 y}{\partial x^2}\right|_{x,t} \equiv y^{xx}_{x,t} = y_{x+1,t} + y_{x-1,t} - 2y_{x,t} \tag{6.12}$$

This is known as the Midpoint method.

The time derivative is defined

$$\left.\frac{\partial y}{\partial t}\right|_{x,t} \equiv y^t_{x,t} = \frac{y_{x,t+1} - y_{x,t}}{T} \tag{6.13}$$

So that $y_{x,t+1} = y_{x,t} + Ty^t_{x,t}$, which is known as Explicit Euler method. There are a few reasons why an explicit method was chosen over an implicit one.

- An implicit method would make the model more complicated, especially the Source Determination described in the next section, due to the dependency of $n_{x,t}$ on its same-time neighbours $n_{x+1,t}$ and $n_{x-1,t}$.
- Explicit methods are generally faster than implicit methods.
- Due to the low frequency of BG fluctuations, explicit methods will be stable if a small enough time step $T$ is chosen.

By applying these rules on equations 6.8 and 6.9, we get

$$n_{x,t+1} = n_{x,t} + T\left(Dn^{xx}_{x,t} + J_{x,t}\right) \tag{6.14}$$

If we let the walls be at $x = -\frac{1}{2}$ and $x = l + \frac{1}{2}$ we get

$$Dn^{xx}_{x,t} = -J^x_{x,t} = -(J_{x+\frac{1}{2},t} - J_{x-\frac{1}{2},t}) = J_{x-\frac{1}{2},t} - J_{x+\frac{1}{2},t}$$

$$Dn^{xx}_{0,t} = J_{-\frac{1}{2},t} - J_{\frac{1}{2},t} = -J_{\frac{1}{2},t} = Dn^x_{\frac{1}{2},t} = D(n_{1,t} - n_{0,t}) \tag{6.15}$$

$$Dn^{xx}_{l,t} = J_{l-\frac{1}{2},t} - J_{l+\frac{1}{2},t} = J_{l-\frac{1}{2},t} = -Dn^x_{l-\frac{1}{2},t} = -D(n_{l,t} - n_{l-\frac{1}{2},t}) \tag{6.16}$$

Using this set of rules, it is possible to simulate the diffusion if the initial condition $n_{x,0}$ and the source $J_{x,t}$ are known.

| Parameter | Symbol | Value |
|---|---|---|
| arteriolar pressure | $P_a$ | 35 mmHg |
| venular pressure | $P_v$ | 15 mmHg |
| reflection coefficient | $\sigma$ | |
| capillary and interstitial oncotic pressure | $\pi_c = \pi_i = \pi$ | |
| | $\sigma\pi$ | 25 mmHg |
| filtration coefficient | $K_f$ | 0.978 mL/min/mmHg |
| surface area | $A$ | 1 unit |

Table 6.1: Constant parameter values in T. Shi model[21]

| Parameter | Symbol | Value |
|---|---|---|
| glucose membrane permeability | $P$ | $15 \sim 30$ mL/100g/min/m$^2$ |
| ISF diffusion coefficient | $D$ | $1 \sim 3 \times 10^{-10}$ m$^2$/s |
| ratio pressure driven transport | $R_{starling}$ | $0 \sim 0.2$ |
| blood flow velocity | $F$ | $1 \sim 3$ mL/100g/min |
| | | $5 \sim 15$ mL/100g/min |

Table 6.2: Person/activity specific parameter ranges in T. Shi model[21]

### Source Determination

The source in the biophysical model is of the form $J_{s,t} = K_b f_t + (K_i - 1)n_{s,t}$ with $s = 0$ the location of the source. $J_{x,t} = 0 \; \forall x \neq s$. The function $f_t$ is the blood glucose concentration.
As described in appendix D the concentration at the sensor $x = k$ at time $t_1$ can be described by the initial condition $n_{x,t_0}$ and $f_t \; \forall \; t_0 \leq t \leq t_1$ with equation 6.17. For simplicity's sake a variable $f_{t_0-t_1}$ is defined as seen in equation 6.18. This is the value of $f_t$ if it were a constant concentration.
By setting the initial condition to $n_{x,t_0} = 0 \; \forall \; x \in X$, and $f_t = 1 \; \Rightarrow f_{t_0-t_1} = 1$, equation 6.18 simplifies to $n_{k,t_1} = B$, this can be used to determine the constant $B$.

$$n_{k,t_1} = \sum_{i \in X} a_i n_{i,t_0} + \sum_{j=t_0}^{t_1} b_j f_j \tag{6.17}$$

$$n_{k,t_1} = \sum_{i \in X} a_i n_{i,t_0} + B f_{t_0-t_1} \tag{6.18}$$

If the constant $B$ and initial condition $n_{x,t_0}$ are known, $f_{t_0-t_1}$ can be determined using equation 6.19. $n_{k,t_1}$ is the measured sensor concentration with unknown $f_t$, and $\bar{n}_{k,t_1}$. $\bar{n}_{k,t_1}$ is the sensor concentration with $f_t = 0$ and initial condition $\bar{n}_{x,t_0} = n_{x,t_0}$, which can be obtained through simulation. This error can then be used to determine $f_{t_0-t_1}$.
Under the assumption that the blood glucose concentration does not have significantly high-frequency components, $f$ can be linearly approximated locally, so that $f_{t_1} - f_{t_0} \approx a(t_1 - t_0)$. Then by taking $\hat{f}_{(t_1+t_0)/2} = f_{t_0-t_1}$, $f_{t_1}$ can be extrapolated $f_{t_1} = 1.5 f_{t_0-t_1} - 0.5 f_{t_0}$.

$$\varepsilon = n_{k,t_1} - \bar{n}_{k,t_1} = B f_{t_0-t_1} \tag{6.19}$$

## 6.4. Parameters

T. Shi, et al. also discussed the values of the parameters of their model in another paper[21]. The constant parameters are given in Table 6.1. Other parameters have a range of possible values, which can be person and/or activity-specific. These are given in table 6.2. The two most influential of these parameters are $P$ and $F$.

### Glucose membrane permeability

The glucose membrane permeability $P$ has a negative correlation with the BG-ISFG error. A higher $P$ results in more glucose absorption into the ISF, and thus an ISFG that closer resembles the BG. A lower $P$ on the other hand, increases the difference between ISFG and BG. The permeability is constant in a specific person and sensor location[22](as cited in [21]), this allows it to be calibrated.

### Blood flow velocity

The blood flow velocity $F$ is positively correlated with the BG-ISFG error. A higher $F$ results in a lower ISFG compared to the BG. $F$ is dependent on the activity of the person. If the person is active, the blood flow velocity is high, $5 \sim 15$ mL/100g/min, and if the person is inactive, the blood flow velocity is low, $1 \sim 3$ mL/100g/min. This can cause problems for the model due to changes in the blood flow velocity of the individual throughout the day. An input device that can be used for determining the blood flow velocity, like a heart rate monitor, can be used to correctly adjust the model to these changes. This is, however, outside of the scope of this project, so the model should take this variation in mind while being constrained to the glucose sensor only.
This problem is tackled by assuming a low $F$ to ensure an accurate BG estimate when the patient is at rest. The model should also be stable, in the sense that after a period of activity, it should return to giving accurate BG estimates when the patient is at rest again.

## 6.5. Implementation

The methods described in the previous sections are used to estimate the BG. Algorithm 1 describes the way these methods are implemented.
First, the entire ISF is assumed to be in equilibrium and have a homogeneous glucose concentration equal to the first sensor glucose measurement.
Then the $K_b$ is calibrated. This is done using an initial BG value. It is assumed that $J_{s,t_0} = 0$ due to the equilibrium, which gives $K_b = (1 - K_i)n_{k,t_0}/f_{t_0}$. $K_i$ is calculated using a $R_{starling}$ guess, described later in this section.
Then, for each incoming measurement, the timestep between the measurements is first determined.
The constant $B$ as described in equation 6.18 is then calculated for this specific timestep.
A simulation is then run with $f_t = 0$, and the error with the measured value is then used to determine $f_{t_{i-1}-t_i}$ and $f_{t_i}$ using equation 6.19 and extrapolation.
$f_{t_i}$ is then recorded and sent back as the estimated BG, and $f_{t_{i-1}-t_i}$ is used to update the initial condition.
All simulations are performed using algorithm 2 using the rules defined in section 6.3 and 6.2

Algorithm 1: Sensor Glucose(SG) to BG Algorithm

Input: Initial BG $f_{t_0}$ and measured SG $n_{k,t} : t \in \{t_0, t_1, ...\}$
Output: List of BG estimates $f_t : t \in \{t_0, t_1, ...\}$
$\tilde{n}_{X,t_0} = n_{k,t_0}$;
$K_b = (1 - K_i)n_{k,t_0}/f_{t_0}$;
foreach SG measurement $i$, $n_{k,t_i}$ do
  $\Delta t = t_i - t_{i-1}$;
  $B = \text{simulate}(0, \Delta t, 1)[k]$;
  $\tilde{n}_{X,t_i} = \text{simulate}(\tilde{n}_{X,t_{i-1}}, \Delta t, 0)$;
  $f_{t_{i-1}-t_i} = (n_{k,t_i} - \tilde{n}_{k,t_i})/B$ ;
  $f_{t_i} = 1.5 f_{t_{i-1}-t_i} - 0.5 f_{t_{i-1}}$;
  $\tilde{n}_{X,t_i} = \text{simulate}(\tilde{n}_{X,t_{i-1}}, \Delta t, f_{t_{i-1}-t_i})$;

The model has a number of parameters which need a value to run the algorithm. The first one is Fick's diffusion constant for ISF, $D$. This value is between $1 \sim 3 \cdot 10^{-10}$ $\text{m}^2\text{s}^{-1}$ so a value of $2 \cdot 10^{-10}$ $\text{m}^2\text{s}^{-1}$ is assumed for the algorithm. This has to be converted to the unit $\Delta x^2\text{s}^{-1}$ to be used on equation 6.14, with $\Delta x$ the spatial step length. $r = \Delta x \cdot (l + 1)$ is the maximal diffusion depth which is assumed to be $40 \cdot 10^{-6}$m[21]. $(l + 1) = 25$ is the number of spatial steps. $D$ can then be converted using $D_{\Delta x^2\text{s}^{-1}} = D_{\text{m}^2\text{s}^{-1}} / \left(\frac{r}{l+1}\right)^2 = \frac{625}{8} = 78.125$ $\text{m}^2\text{s}^{-1}$.
In order to ensure stability $T$ must be chosen so that $DT$ is sufficiently bellow 1, $DT < 1 \Rightarrow T < D^{-1} = 0.0128$ s. $T = 0.005$ s was chosen and verified to be stable on all of the dataset.
$R_{starling}$ was chosen to be $0.1$ in the middle of its expected range $0 \sim 0.2$, and $k$ the sensor location is taken to be $80\%$ of the total diffusion distance, so $20\Delta x$[21].

Algorithm 2: Simulation algorithm for constant BG

```
Function simulate( initial_condition[], simulation_time, f )
    n[] ← initial_condition[];
    t ← 0;
    for t < simulation_time do
        buffer ← new array[l + 1];
        buffer[0] = n[0] + T(D(n[1] − n[0]) + K_b f + (K_i − 1)n[0]);
        buffer[l] = n[l] + TD(n[l − 1] − n[l]);
        foreach x ∈ (0, l) do
            buffer[x] = n[x] + TD(n[x + 1] + n[x − 1] − 2n[x]);
        n[] ← buffer[];
        t ← t + 1;
    return n[]
```

Something to note is that the $K_b$ is calibrated once at the start of operation. $K_b$ is a function of $R_{starling}$, $P$, and $F$. Whereas $P$ and $R_{starling}$ are person-specific constants, $F$ is not constant but activity-dependent. It is therefore important that calibration happens while the user has a representative blood flow, which will be when the user is awake but not physically active. Calibrating during activity can introduce errors due to $K_b$ changing. This could be reliably fixed by introducing a second sensor that can be used to estimate the blood flow, like a heart rate monitor, but this is outside of the project scope.

## 6.6. Dataset

In order to test the system on real BG and ISFG dynamics, it was decided to use a publicly available dataset, due to their readily availability. Datasets containing regular BG recordings and unprocessed ISFG recordings were hard to find, due to many trials being kept private and public trials often using processed ISFG recordings in the form of CGM. It was therefore decided to use a BG dataset and generate data using the model to acquire ISFG data. The selected dataset is the Replace-BG dataset[23], a clinical trial run on T1D patients to determine if regular BGM measurements are needed in supplement to using CGM. The dataset contains CGM recordings with a market-approved sensor, which allows it to be used as reference BG values for evaluating other CGMs according to the ISO standard[5]. Due to the computational time required to generate and test data, a selection of 1474 datapoints of 29 different individuals from the dataset was made to be used for testing the model.

### ISFG data generation

To obtain ISFG sensor data, data was generated using the simulation algorithm 2. The biophysical parameters were picked randomly for each individual using a uniform distribution within the ranges specified in section 6.4. For the bloodflow each individual had a 50% chance to have a high bloodflow $P(1 \leq F < 3) = P(5 \leq F < 15) = 0.5$, the exact bloodflow was then decided uniformly $\{ F \mid 5 \leq F < 15\} \sim U([5, 15\rangle)$, $\{ F \mid 1 \leq F < 3\} \sim U([1, 3\rangle)$.

The BG is made continuous to avoid inducing any artificial time delay. This is done through linear interpolation, so that $f_{t_0+\delta t} = f_{t_0} + \delta t \frac{(f_{t_1} - f_{t_0})}{(t_1 - t_0)}$. The initial ISFG is assumed homogeneous and in equilibrium, $n_{x,t_0} = \frac{K_b f_{t_0}}{1 - K_i} \; \forall \; x \in X$.

# 7

# Microcontroller

The microcontroller is responsible for managing the control signals of the sensor circuit. It reads the sensor circuit output, and sends this to the computer for digital processing. This chapter will not cover the sensor circuit itself, but will discuss the control of the sensor, and the communication with the computer. The sensor itself is described in the report of the other project subgroup, Guijt and van Ham[24].

## 7.1. Procedure

The sensor circuit has two control signals, one for turning the sensor on/off, the other for turning the integrator on/off. The sensor needs to turn on to start the glucose reaction, starting the Cottrell equation. The integrator should turn on during the integration window for determining the charge. The parameters are discussed in chapter 4 and the implementation is given in this section.
Every 5 minutes, a measurement is made, and each measurement takes 23 seconds. During the measurement the sensor will be turned on, and after 2 seconds the charge integrator will turn on. The integration window is 20 seconds, after which the measurement is made, the integrator is turned off, and after another second the sensor is turned off. This is described in the procedure below.

Procedure Measurement Loop

```
Repeat
    sensorOn();
    Wait(2s);
    integratorOn();
    Wait(20s);
    doMeasurement();
    integratorOff();
    Wait(1s);
    sensorOff();
    Wait(4min 37s);
```

When the measurement is made it should be send to the computer. If the microcontroller is disconnected however, the measurement should be stored so that it can be send when the microcontroller reconnects with the computer. This is needed to allow the user to disconnect without losing data, which may occur if their phone dies, the distance is too much, or due to other connection issues, and should therefore be accounted for. The current implementation can store the measurements for at least 12 hours in the case of a disconnected sensor.

Procedure Measurement handling

```
Function doMeasurement()
    measurement = readAnalogPin();
    if connected then
        sendMeasurement(measurement);
    else
        addToStack(measurement);
```

Procedure Reconnection Event

```
Function onReconnection()
    foreach measurement in stack do
        sendMeasurement(measurement);
    clearStack();
```

## 7.2. Test Setup Implementation

For testing the complete system a Bluetooth connection was first established and then measurements were send to the computer directly as shown in the procedure bellow. A start signal could be send from the computer to the microcontroller in order to start a measurement. Other command were added as well to be able to turn on/off the sensor and integrator from the computer. This was done to make debugging easier. As a back-up measure the microcontroller was also able to do direct current measurements, in which the integrator is skipped and measurements are done at the input of the integrator 10 times with a wait of 2 seconds after each measurement.

Procedure Test Setup

```
initialise();
waitForConnection();
Repeat
    if startsignal then
        sensorOn();
        Wait(2s);
        integratorOn();
        Wait(20s);
        sendMeasurement(measurement);
        integratorOff();
        Wait(1s);
        sensorOff();
```

# 8 Results

## 8.1. Charge-ISFG Model

### 8.1.1. Linear results

The linear model as implemented in B.1 is first tested with an arbitrary glucose profile, noise-free, baseline at 100mg/dL, with some glucose increases to mirror breakfast, lunch, and dinner. With a sensor drift of 2% per day, which was found to be the maximum drift per day for a Freestyle Navigator CGM [25]. The goal is ultimately to see how the model performs in a dynamic situation.

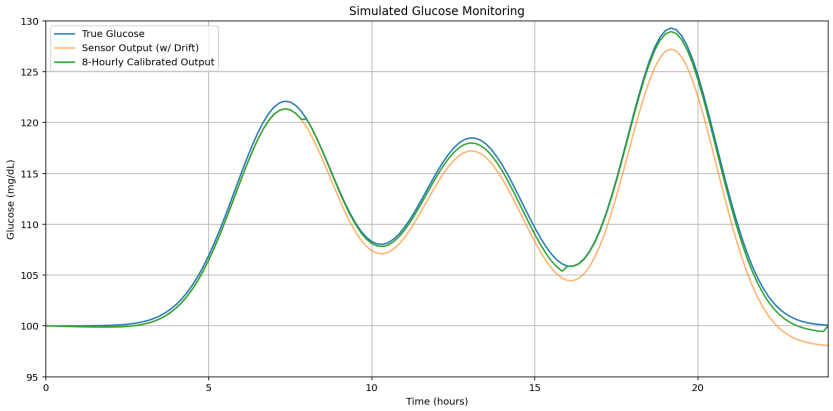


Figure 8.1: Linear model with sensor drift

Not surprisingly, Figure 8.1 shows the sensor starts to deviate from the 'truth' due to sensor drift. However, with a regularly calibrated sensor, in this case, an 8-hourly calibration shows better overall performance, clearly showing the need for regular calibrations if a sensor's drift were to be unknown especially since sensor drift can differ from sensor to sensor and person to person as biological factors can also impact the sensor sensitivity.

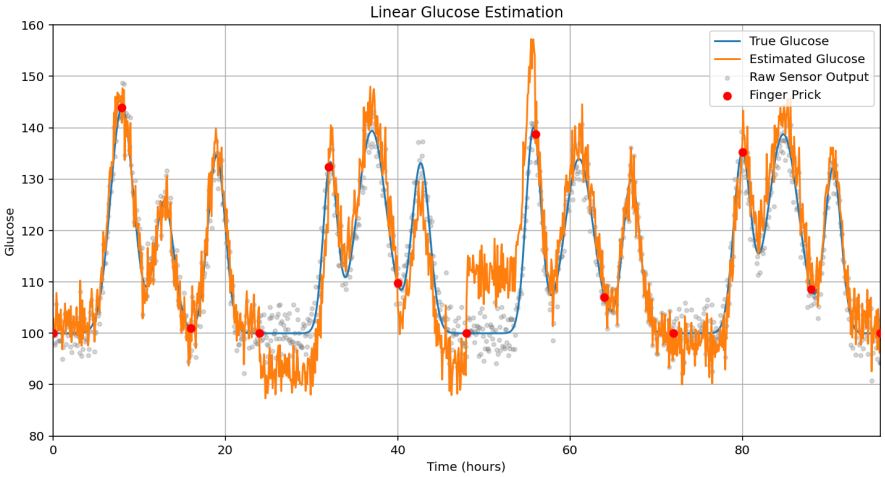


Figure 8.2: Linear model with noise and sensor drift

Figure 8.2 shows a plot for 4 days with added measurement noise and also added noise to the 'true glucose' reference points for which an one-point calibration is used. This is to reflect that conventionally, only one blood finger-prick is used, and they are also noisy themselves. An additive Gaussian normal distributed noise $X \sim N[0, 6]$ was used. Translating to a rough $\pm 12 mg/dL$ maximum deviation for 95% of measurements. This noise is chosen as extreme, as glucose sensors per ISO should be within 15% for glucose values below 100mg/dL and this noise variance approaches this. It shows the estimated glucose profile is quite noisy and its performance can degrade quite if a reference point used for calibration is noisy itself. For this particular setup, a Mean Absolute Relative Difference (MARD) of 4.14% was achieved.
In all, this means using a linear model on raw, noisy data and calibrating using noisy reference data is not optimal and can lead to poor performance. There is a need for an additional filter or a different implementation to handle the raw measurements.

### 8.1.2. Kalman results

The test data used for the Kalman filter is the same as the data used mentioned earlier for the linear model in section 8.1.1. Since the added normal Gaussian noise is $X \sim N[0, 6]$, which is the simulated sensor noise. The measurement noise covariance R is set to $\sigma^2 = 36$ The process noise covariance to Q = $[1, 1, 0.000005]$, this is empirically determined.

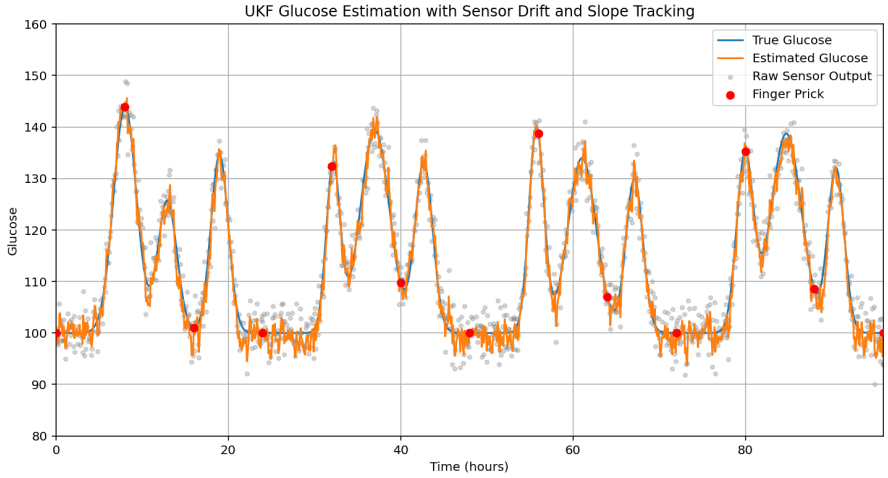


Figure 8.3: Kalman glucose estimation with Gaussian $N \sim [0, 6]$ noise, daily 2% sensor drift.

Figure 8.3 clearly shows a better estimated glucose performance over the simple linear model in Figure 8.2. The resulting MARD is 1.75% compared to the MARD of 4.14% for the linear model. A Kalman filter, if tuned, handles noisy measurements better. In Appendix B.2 more additional plots such as relative error and evolution of sensor sensitivity over time are available. Also in Appendix C.2, these plots for the Kalman filter can be found. Comparing both the relative error of the two models shows Kalman performing better, as well as better in sensor sensitivity tracking.

## 8.2. ISFG-BG Model

### Dataset and diffusion

In order to verify the data generation process and diffusion, the BG data of one of the individuals is put into the data generation algorithm and plotted in Figure 8.4. In order to get a better look at the diffusion within the model, the resolution of the ISFG is increased by sampling the sensor ISFG 4 times the frequency as the BG. Even though the BG is provided to the algorithm linearly interpolated between its actual samples, in the same way as the BG is plotted, the ISFG is visibly smoothed. This shows that the diffusion acts as a low-pass filter, inducing a time delay as described in chapter 6.

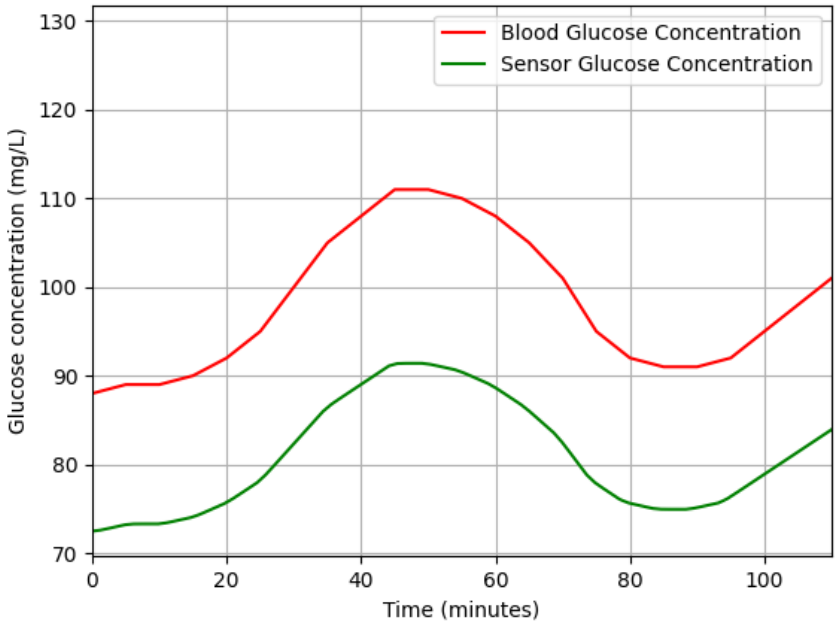


Figure 8.4: Sample of the generated ISFG from BG with increased resolution

## Linear and biophysical model test

The ISFG was used to estimate the BG on the generated dataset with the biophysical model and the linear model as described in 6. The linear model used the first and last BG value as calibration points, and the biophysical model only the first BG value. The resulting performance is shown in Table 8.1, and the data points are plotted against the ISO margins in Figures 8.5 and 8.6. It can be seen that all samples fall within the A and B regions of the Consensus error grid. For the 95% allowed error margin, the linear model has 10 times more measurements that don't meet it. The Mean Absolute Relative Difference(MARD) of the biophysical model is less than half that of the linear model.

| | Biophysical | Linear |
|---|---|---|
| MARD | 0.89% | 2.00% |
| ISO 95% | 99.80% | 98.03% |
| ISO 99% | 100.00% | 100.00% |

Table 8.1: Performance of the biophysical and linear models

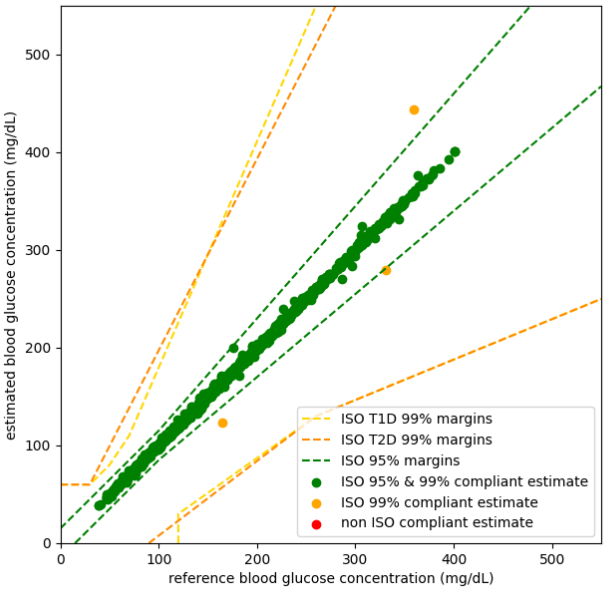


Figure 8.5: Biophysical model data points scatter plot with ISO margins

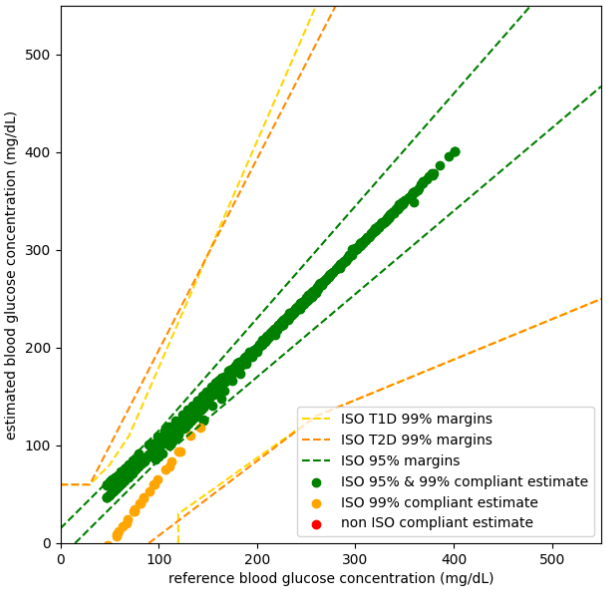


Figure 8.6: Linear model data points scatter plot with ISO margins

## 8.3. Total digital System

After exploring different ways to relate sensor readings to ISF glucose, such as a simple linear approach compared to a Kalman filter, and comparing a linearised approach versus the T. Shi model to relate ISF glucose to blood glucose. The total digital system will be tested by combining these blocks. The performance using a Kalman Filter and the biophysical model will be tested, and the accuracy evaluated. Also, the system performance will be evaluated using a linear approach in both blocks, allowing a comparison between the two setups. The dataset used is the blood glucose datasets described in 6.6. From this dataset, a charge is ultimately simulated with added Gaussian white noise of $X \sim N[0, 6]$, this to mimic sensor noise as perfect conditions are not noise-free. Ultimately, the total digital system performance is tested.

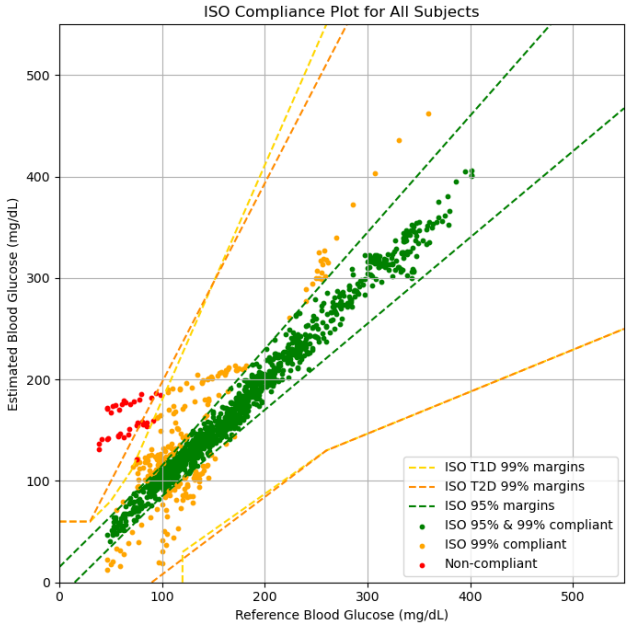


Figure 8.7: Linear model for Sensor-ISFG and linear ISFG-BG.

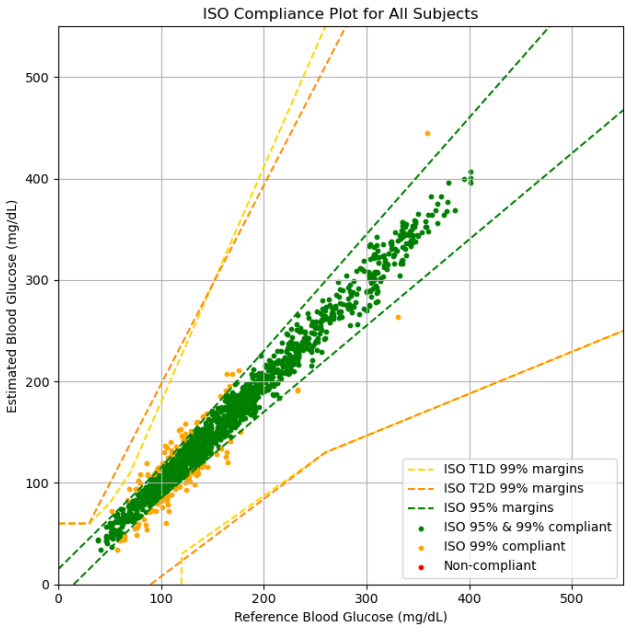


Figure 8.8: Kalman and biophysical model.

| | Linear | Proposed |
|---|---|---|
| MARD | 12.01% | 6.27% |
| ISO 95% | 83.04% | 92.95% |
| ISO 99% | 97.56% | 100.0% |

Table 8.2: Performance of the biophysical and linear models

In Figure 8.7 the Parkes error grid can be seen, it shows most measurements being within the ISO99. However, there is a significant amount outside of the stricter ISO95, and even tens of measurements outside of both. Many of the measurements are especially at the lower glucose values quite spread and all over the place. The total MARD, ISO95, and ISO99 can be found in Table 8.2. For the non-optimal linearised approach, this results in a MARD of 12.01%, an ISO95 of 83.04% and an ISO99 of 97.56%.

In Figure 8.8, the Parkes error grid of the optimised approach is applied to the mentioned dataset. The overall spread of all the measurements is better, and most measurements are within the ISO95 zone, with a few just outside of the ISO99 zone. Nonetheless, this corresponds to all measurements being within the A-B region of the Parkes error grid, meaning the measurements are reliable enough for therapeutic decisions. Also, compared to the linear approach, the MARD of 6.27% is lower and ISO95 of 92.95%, ISO99 100% is higher.

## 8.4. Total CGM Test

The next final step is to test whether the digital block integrates well with the analogue part of the total project. For this, the total sensor will be tested in phosphate-buffered saline (PBS) solutions, using three gold electrodes, different from the earlier-mentioned Sensible sensors. Additionally, since these measurements are made in vitro opposed to in vivo, no meaningful conclusion can be made about the performance of the ISFG-BG model, as only buffered solutions are used, which don't mimic the blood-ISF glucose dynamics. However, it still allows the total system to be tested up to the first part of the digital component, Charge to ISFG.
However, instead of glucose concentrations, dopamine and rhodamine concentrations were used because the glucose solutions were not delivered on time. Which, if for the potentiostat in the analogue part, the correct working potential is applied, it should be fine to test the system's performance. Instead of a glucose sensor, it would be a dopamine or rhodamine sensor.

| Concentration | Value | Voltage [V] |
|---|---|---|
| PBS with sensor off | 509 | 0.15V |
| 600 mg/dl Dopamine in PBS | 1456 | 0.42V |
| PBS normal measurement | 1452 | 0.42V |
| 600 mg/dl Rhodamine in PBS | 1455 | 0.42V |

Table 8.3: Voltages at the output of the ADC as measured (in chronological order) for various concentrations of dopamine and rhodamine in PBS.

The following measurement results were achieved in Table 8.3. These results show the sensor output of the integrator, which is all inconclusive. This is because of issues with the integrator, as the amplifier got saturated during the integration window too quickly. Consequently, due to errors in the sensor output, it was not possible to test the performance of the digital system.

# 9

# Conclusion & Discussion

## 9.1. Conclusion

### Current

Chapter 4 describes the current in chronoamperometry, the current driven by the Cottrell equation, which eventually decays into the steady-state current, both are proportional to the initial glucose concentration and can be used for measurements. However, at the start, the current is mostly driven by a charging current, as the electrode and analyte act as a capacitor which gets charged. However, this effect is short-lived, and measurements can be made quite quickly. The decision was made to integrate the current over a time window of 20 seconds, 2 seconds after initially applying the necessary potential, starting the oxidation of glucose. Resulting in a charge reading which should be proportional to the ISF glucose concentration.

### Charge-ISFG

Consequently, in chapter 5, two methods of relating charge to ISF glucose were mentioned. A simple linear approach and a Kalman filter. For both, the performance was tested with a simulated glucose profile with additive Gaussian normal disturbed noise $X \sim N[0, 6]$ and a sensor drift of 2%. From the results, it was noticed that the Kalman filter, a filter which is often used with noise sensors, handles both sensor drift and noise, and noisy glucose references better. Having a MARD of 1.75% compared to 4.14% for the linear approach. From this, it's clear to use a Kalman filter in estimating ISF glucose from sensor readings, as it handles noise well.

### ISFG-BG model

In chapter 6, a biophysical model describing the kinetics between ISF glucose and blood glucose, a modified version of the Shi model[19] was proposed. This model was compared to a linear model between ISFG and BG.
Both the linear and the biophysical models, on their own, meet the ISO standard. The biophysical model has more points within the 95% margins, 99.80% compared to 98.03% in the linear model. The MARD is lower for the biophysical model as well, with 0.89% compared to 2.00%. Since the ISFG-BG model will be combined with other modules to form a full system, a low MARD is desirable, since errors can be amplified through the other modules, resulting in measurement error outside of the acceptable margins. The biophysical model can therefore be considered a better option over the linear model in this context. It is important to note that the blood flow is taken to be constant in the dataset.

### Total Digital

Both the Charge-ISFG and ISFG-BG models were tested together in the following combinations: one with both linear models and one with the Kalman and biophysical model. Results indicated that the linear approach was inferior compared to the proposed configuration. Using the Kalman in combination with the biophysical model resulted in measurements being all within the A-B zone of the Parkes error grid, with a MARD of 6.27%, ISO95 of 92.95% and ISO99 of 100.0%. While using both linear models

resulted in various measurements being outside the Parkes error grid, with worse MARD, ISO95, and ISO99 values, respectively, being 12.01%, 83.03% and 97.56%.

### Full System

Lastly, an in-vitro test was attempted for the whole sensor, that is, the analogue part in connection with the digital part, to see the overall performance of the sensor for various buffers with different concentrations; however, these tests were inconclusive as the experiments failed, meaning it was not possible to test whether the digital implementation would have worked in the total integrated CGM system test.

## 9.2. Discussion

The results and consequent conclusion reveal that the use of a Kalman filter and biophysical approaches is superior compared to using simple linear approaches. The used dataset for the total digital system test had added noise, which approached the boundaries of the ISO95 standard; as a result, the total system just came short of completely complying with ISO95. However, this added noise explored the limits of acceptable noise, and most likely, the sensor output noise is lower, especially as the analogue subgroup designed the circuit to have noise as low as possible. However, due to the inability to perform the total sensor tests. The true known noise is undefined as for now. However, assuming that the noise is most likely less than the added noise. The used digital approach would be most likely satisfactory for blood glucose sensors, as performance would comply with the set ISO standard for blood glucose sensors.

It is important to note that the generated data assumes blood flow velocity to be constant, ignoring the fact that the blood flow varies throughout the day with different activities. The blood flow velocity could be modelled for the data generation, but this can only be completely verified using recorded in vivo BG-ISFG dynamics.

Also, whether the current implementation of them would work in the total CGM system tested in vivo is unknown, as for this project, in vivo tests were not allowed, and the simple sensor performance in vitro could not be properly conducted due to functionality errors in the analogue part of the system with the integrator. There were problems with the analogue circuitry of the integrator, as this saturated too quickly. There were also other problems with this circuit, which are out of the scope of this report, and additionally, the integration window for the sensor could have been too long, but due to time constraints, it was not possible to address all these problems. In all, due to problems with the analogue circuitry to utilise the sensor, the digital pipeline could not be tested for the total CGM system. Instead, this report is limited to evaluating the digital system to available datasets and simulations.

## 9.3. Future works

Simple simulations and open-source blood glucose datasets to evaluate the ISFG-BG model show good results and are in compliance with the set ISO standards for blood glucose sensors. However, there is a clear need for testing of the complete CGM system both in vivo and in vitro. Especially setting up trials to test the complete sensor in individuals is important, as biological processes in the body can affect a sensor greatly, which is not easily mimicked with in vitro tests and in silico generated ISF data. Also, the exact measurement noise of the analogue circuitry used should be tested to properly tune the digital part to it, like the Kalman filter. As discussed in chapter 6, the blood velocity flow could be adjusted using a second sensor such as a heart rate monitor. This would allow more accurate functioning during periods of high activity, possibly allowing for the usage in sports.

# Bibliography


[1] F. Ribet, G. Stemme, and N. Roxhed, "Real-time intradermal continuous glucose monitoring using a minimally invasive microneedle-based system," en, Biomed. Microdevices, vol. 20, no. 4, p. 101, Dec. 2018.

[2] World Health Organization, Diabetes, https://www.who.int/news-room/fact-sheets/detail/diabetes, Fact sheet published 14 November 2024; accessed 15 June 2025, 2024.

[3] S. K. Garg, "Past, present, and future of continuous glucose monitors," Diabetes Technology & Therapeutics, vol. 25, no. S3, S1–S4, 2023, Published online 12 June 2023; PMID: 36749142. DOI: 10.1089/dia.2023.0041. [Online]. Available: https://www.liebertpub.com/doi/10.1089/dia.2023.0041.

[4] F. Ribet, "Integrated microsystems for continuous glucose monitoring, interstitial fluid sampling and digital microfluidics," Ph.D. dissertation, KTH, Micro and Nanosystems, 2020.

[5] "In vitro diagnostic test systems Requirements for blood-glucose monitoring systems for self-testing in managing diabetes mellitus," International Organization for Standardization, Geneva, CH, Standard, 2013.

[6] G. Freckmann, J. Mende, S. Pleus, et al., "Mean absolute relative difference of blood glucose monitoring systems and relationship to ISO 15197," en, J. Diabetes Sci. Technol., vol. 16, no. 5, pp. 1089–1095, Sep. 2022.

[7] Analytical Sciences Digital Library. "Chronoamperometry potential step methods." Accessed: 2025-04-29. (2023), [Online]. Available: https://chem.libretexts.org/Bookshelves/Analytical_Chemistry/Supplemental_Modules_(Analytical_Chemistry)/Analytical_Sciences_Digital_Library/Courseware/Analytical_Electrochemistry%3A_The_Basic_Concepts/04_Voltammetric_Methods/A._Basics_of_Voltammetry/01_Potential_Step_Methods/a)_Chronoamperometry.

[8] A. J. Bard, L. R. Faulkner, and H. S. White, Electrochemical Methods: Fundamentals and Applications, 3rd ed. Wiley, 2023, Published 13 September 2023.

[9] V. S. Conceição, D. P. M. Saraiva, G. Denuault, and M. Bertotti, "Calibration-free analysis with chronoamperometry at microelectrodes," Analytical Chemistry, vol. 96, no. 37, pp. 14 766–14 774, 2024, Published: September 3, 2024. DOI: 10.1021/acs.analchem.4c01645. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11411494/#:~:text=In%20coulometry%2C%20one%20of%20the,sample%20is%20consumed%20and%20thus.

[10] F. Ribet, G. Stemme, and N. Roxhed, "Ultra-miniaturization of a planar amperometric sensor targeting continuous intradermal glucose monitoring," en, Biosens. Bioelectron., vol. 90, pp. 577–583, Apr. 2017.

[11] J. G. Stinstra, S. Shome, B. Hopenfeld, and R. S. Macleod, "Modeling the passive cardiac electrical conductivity during ischemia," Medical & Biological Engineering & Computing, vol. 43, no. 4, pp. 474–482, 2005. DOI: 10.1007/BF02344724. [Online]. Available: https://www.sci.utah.edu/~macleod/papers/jeroen/stinstra-mbec05.pdf.

[12] E. Khalil, K. Kretsos, and G. B. Kasting, "Glucose partition coefficient and diffusivity in the lower skin layers," Pharmaceutical Research, vol. 23, no. 6, pp. 1227–1234, 2006. DOI: 10.1007/s11095-006-0141-9. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/16715366/.

[13] G. Acciaroli, M. Vettoretti, A. Facchinetti, and G. Sparacino, "Calibration of minimally invasive continuous glucose monitoring sensors: State-of-the-art and current perspectives," Biosensors, vol. 8, no. 1, p. 24, 2018. DOI: 10.3390/bios8010024. [Online]. Available: https://www.mdpi.com/2079-6374/8/1/24.

[14] B. W. Bequette, "Continuous glucose monitoring: Real-time algorithms for calibration, filtering, and alarms," Journal of Diabetes Science and Technology, vol. 4, no. 2, pp. 404–418, 2010, Published: March 1, 2010. DOI: 10.1177/193229681000400222. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC2864177/.

[15] V. Lodwig and L. Heinemann, "Continuous glucose monitoring with glucose sensors: Calibration and assessment criteria," Diabetes Technology & Therapeutics, vol. 5, no. 4, pp. 572–586, 2003. DOI: 10.1089/152091503322250596. [Online]. Available: https://www.researchgate.net/publication/9079542_Continuous_Glucose_Monitoring_with_Glucose_Sensors_Calibration_and_Assessment_Criteria.

[16] A. Kastsiukavets, Kalman filter, extended kalman filter, unscented kalman filter, https://medium.com/@kastsiukavets.alena/kalman-filter-extended-kalman-filter-unscented-kalman-filter-dbbd929f83c5, Accessed: 2025-06-11, 2017.

[17] R. R. L. Jr., Kalman and bayesian filters in python, https://filterpy.readthedocs.io/en/latest/, Online book and Python library. Accessed: 2025-06-11, 2014.

[18] A. Basu, S. Dube, M. Slama, et al., "Time lag of glucose from intravascular to interstitial compartment in humans," Diabetes, vol. 62, no. 12, pp. 4083–4087, 2013, Published Nov 16, 2013. DOI: 10.2337/db13-1132. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC3837059/.

[19] T. Shi, D. Li, G. Li, Y. Zhang, K. Xu, and L. Lu, "Modeling and measurement of correlation between blood and interstitial glucose changes," Journal of Diabetes Research, vol. 2016, no. 1, p. 4 596 316, 2016. DOI: https://doi.org/10.1155/2016/4596316. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2016/4596316. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2016/4596316.

[20] G. Acciaroli, M. Vettoretti, A. Facchinetti, and G. Sparacino, "Chapter 9 - calibration of cgm systems," in Glucose Monitoring Devices, C. Fabris and B. Kovatchev, Eds., Academic Press, 2020, pp. 173–201, ISBN: 978-0-12-816714-4. DOI: https://doi.org/10.1016/B978-0-12-816714-4.00009-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128167144000090.

[21] T. Shi, Y. Zhang, and L. L. and, "Effect of physiological parameters on glucose microcirculation compartmental model in glucose monitoring," Biotechnology & Biotechnological Equipment, vol. 32, no. 4, pp. 1047–1052, 2018. DOI: 10.1080/13102818.2017.1413595. eprint: https://doi.org/10.1080/13102818.2017.1413595. [Online]. Available: https://doi.org/10.1080/13102818.2017.1413595.

[22] T. Koutny, "Using meta-differential evolution to enhance a calculation of a continuous blood glucose level," Computer Methods and Programs in Biomedicine, vol. 133, pp. 45–54, 2016, ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2016.05.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260716300128.

[23] G. Aleppo, K. J. Ruedy, T. D. Riddlesworth, et al., "REPLACE-BG: A randomized trial comparing continuous glucose monitoring with and without routine blood glucose monitoring in adults with well-controlled type 1 diabetes," en, Diabetes Care, vol. 40, no. 4, pp. 538–545, Apr. 2017.

[24] C. N. Guijt and J. J. van Ham, "Potentiostat-Based Analogue Circuit for Wearable Glucose Sensors," TU Delft, 2025.

[25] U. Hoss, E. S. Budiman, H. Liu, and M. P. Christiansen, "Continuous glucose monitoring in the subcutaneous tissue over a 14day sensor wear period," Journal of Diabetes Science and Technology, vol. 7, no. 5, pp. 1210–1219, 2013. DOI: 10.1177/193229681300700511. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3876365/.

# A

# Bio-signal

## A.1. Typical glucose levels

| Condition | Glycaemic thresholds |
|---|---|
| Hypoglycemia | • Fasting plasma glucose < 70 mg/dl (3.9 mmol/L) |
| Normal | • Fasting plasma glucose 70-100 mg/dl (3.9-5.6 mmol/L)<br>• 2-h plasma glucose* < 140 mg/dl (7.8 mmol/L) |
| Diabetes | • Fasting plasma glucose ≥ 126 mg/dl (7.0 mmol/L) or<br>• 2-h plasma glucose* ≥ 200 mg/dl (11.1 mmol/L) or<br>• HbA1c ≥ 6.5% |
| Impaired glucose tolerance (IGT) | • Fasting plasma glucose < 126 mg/dl (7.0 mmol/L) and<br>• 2-h plasma glucose* 140-200 mg/dl (7.8-11.0 mmol/L) |
| Impaired fasting glucose (IFG) | • Fasting plasma glucose 110-125 mg/dl (6.1-6.9 mmol/L) and<br>• (if measured) 2-h plasma glucose* < 140 mg/dl (7.8 mmol/L) |
| Gestational diabetes (GDM) | • One or more of the following:<br>• Fasting plasma glucose 92-125 mg/dl (5.1-6.9 mmol/L)<br>• 1-h plasma glucose** ≥ 180 mg/dl (10.0 mmol/L)<br>• 2-h plasma glucose 153-199 mg/dl (8.5-11.0 mmol/L) |

# B

# Linear

## B.1. Linear Class

```python
import numpy as np
from datetime import datetime
from typing import List, Tuple, Optional

class GlucoseSensor:
    def __init__(self, calibration_data: Optional[List[Tuple[float, float]]] = None) -> None:
        self.calibration_data = calibration_data or []
        self.a = 0
        self.b = 0
        self.history = []
        self._calibrate()
        self.cal_type = 1 # '1' one-point, otherwise set not to 1

    def add_reference_data(self, charge_coulombs: float, glucose_levels: float) -> None:
        self.calibration_data.append([charge_coulombs, glucose_levels])
        self._calibrate()

    def clear_calibration_data(self) -> None:
        self.calibration_data.clear()
        self.a = 0
        self.b = 0

    def _calibrate(self):
        if not self.calibration_data:
            return

        currents, glucose_levels = zip(*self.calibration_data)
        if self.cal_type == 1:
            current, glucose = self.calibration_data[-1]
            if current == 0:
                raise ValueError("Current cannot be zero in one-point calibration.")
            self.a = current / glucose
            self.b = 0 #current-glucose
        else:
            x = np.array(glucose_levels[-2:])
            y = np.array(currents[-2:])
            self.a, self.b = np.polyfit(x, y, 1)
            print(self.a, self.b)

    def calculate_glucose(self, charge_coulombs: float, time_seconds=1) -> float:
        if self.a == 0 and self.b == 0:
            raise ValueError("Model parameters are zero. Provide calibration data to enable
                calculation.")
        if time_seconds <= 0:
            raise ValueError("Time must be positive.")

        current = charge_coulombs / time_seconds
        glucose = (current - self.b) / self.a
```

```
48          self.history.append((datetime.now(), glucose, self.a, self.b))
49          return glucose
50
51      def get_history(self) -> List[Tuple[datetime, float]] | None:
52          return self.history
```

## B.2. Linear Test Plots

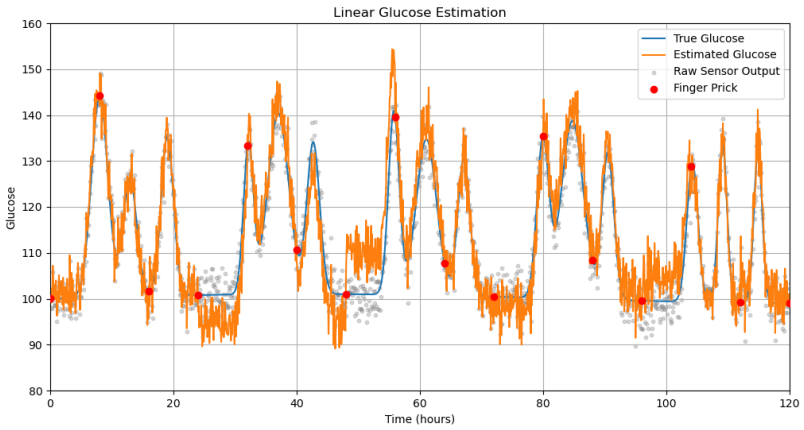


Figure B.1: 5 day period, 2% daily drift, Gaussian $N \sim [0, 6]$

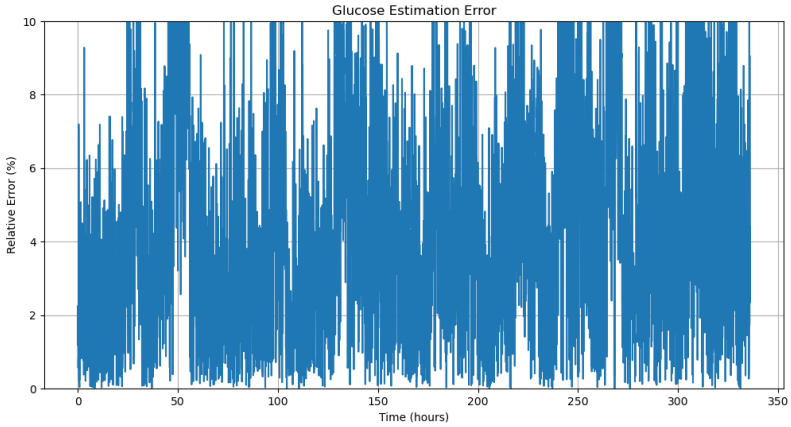


Figure B.2: 14 day period relative error , 2% daily drift, Gaussian $N \sim [0, 6]$

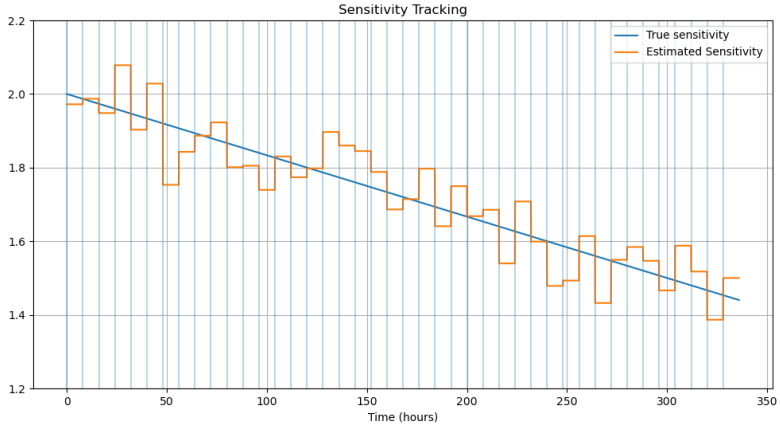


Figure B.3: 14 day period sensor sensitivity , 2% daily drift, Gaussian $N \sim [0, 6]$

# C Kalman

## C.1. Kalman Class

```python
import numpy as np
from datetime import datetime
from typing import List, Tuple, Optional
from filterpy.kalman import UnscentedKalmanFilter as UKF
from filterpy.kalman import MerweScaledSigmaPoints

class GlucoseUKF:
    def __init__(self,
                 initial_glucose: float = 100.0,
                 initial_a: float = 0.0,
                 dt: float = 1.0,
                 process_noise: Tuple[float, float, float] = (1.0, 1.0, 0.0),
                 measurement_noise: float = 0):
        self.dt = dt
        self.points = MerweScaledSigmaPoints(n=3, alpha=0.1, beta=2.0, kappa=0)
        self.ukf = UKF(dim_x=3, dim_z=1, fx=self.fx, hx=self.hx, dt=dt, points=self.points)

        # Initial state: [Glucose, Glucose change rate, sensor gain 'a']
        self.ukf.x = np.array([initial_glucose, 0.0, initial_a])
        self.ukf.P *= 100

        self.ukf.Q = np.diag(process_noise)
        self.ukf.R = np.array([[measurement_noise]])

        self.history = []

    @staticmethod
    def fx(x, dt):
        G, G_dot, a = x
        G_new = G + G_dot * dt
        return np.array([G_new, G_dot, a])

    @staticmethod
    def hx(x):
        G, _, a = x
        return np.array([a * G])

    def update(self, sensor_output: float, dt: Optional[float] = None) -> float:
        if dt is not None and dt != self.dt:
            self.dt = dt
            self.ukf.dt = dt

        self.ukf.predict()
        self.ukf.predict()
        self.ukf.predict()
        self.ukf.update(np.array([sensor_output]))

        est_glucose = self.ukf.x[0]
```

```
49          self.history.append((datetime.now(), est_glucose, self.ukf.x.copy()))
50          return est_glucose
51
52      def calibrate(self, true_glucose: float) -> None:
53          self.ukf.x[0] = true_glucose
54          self.ukf.P = np.diag([1e-10, self.ukf.P[1,1], self.ukf.P[2,2]])  # very low
                uncertainty in glucose after calibration
55
56      def get_current_estimate(self) -> Tuple[float, float, float]:
57          """Returns the current estimate of [Glucose, Glucose_dot, Gain a]."""
58          return tuple(self.ukf.x)
59
60      def get_history(self) -> List[Tuple[datetime, float, np.ndarray]]:
61          return self.history
```

## C.2. Kalman Test Plots

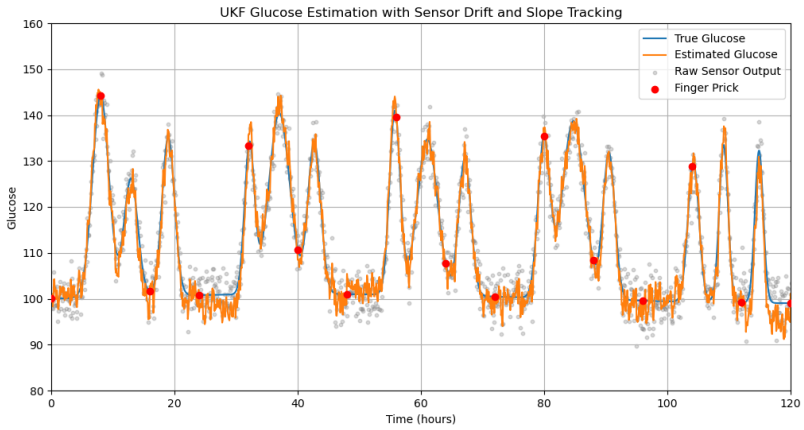


Figure C.1: 5 day period, 2% daily drift, Gaussian $N{\sim}[0,6]$

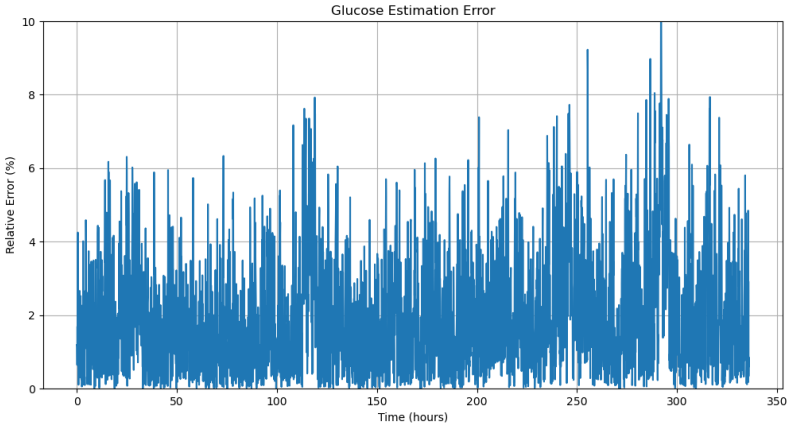


Figure C.2: 14 day period relative error , 2% daily drift, Gaussian $N{\sim}[0,6]$

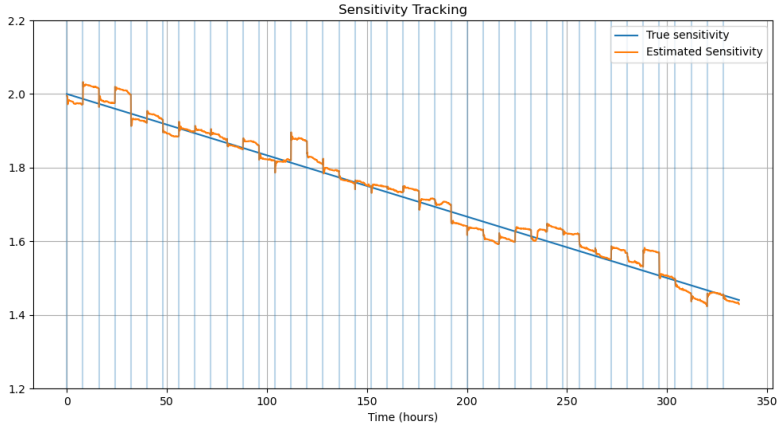


Figure C.3: 14 day period sensor sensitivity , 2% daily drift, Gaussian $N{\sim}[0,6]$

# D

# Source Determination Theory

## D.1. Domain

### Discrete

In order to simplify things, and be able to numerically approach the problem it was chosen to discretize the domain.
Let $x, t \in \mathbb{N} \cup \{0\}$ be a spatial index and time index, so that $n_{x,t}$ is the concentration at those indexes. We define the spatial derivative as follows

$$\frac{\partial y}{\partial x}|_{x,t} \equiv y^{x}_{x,t} = y_{x+\frac{1}{2},t} - y_{x-\frac{1}{2},t} \tag{D.1}$$

So that the second derivative becomes

$$\frac{\partial^2 y}{\partial x^2}|_{x,t} \equiv y^{xx}_{x,t} = y_{x+1,t} + y_{x-1,t} - 2y_{x,t} \tag{D.2}$$

This is known as the Midpoint method.

The time derivative is defined

$$\frac{\partial y}{\partial t}|_{x,t} \equiv y^{t}_{x,t} = \frac{y_{x,t+1} - y_{x,t}}{T} \tag{D.3}$$

So that $y_{x,t+1} = y_{x,t} + T y^{t}_{x,t}$, which is known as Explicit Euler method.

### Diffusion

The equations governing the system in continuous time and space are defined as

$$\frac{\partial n}{\partial t}(x,t) = -\frac{\partial J_{gradient}}{\partial x}(x,t) + J_{source}(x,t) \tag{D.4}$$

$$J_{gradient}(x,t) = -D\frac{\partial n}{\partial x} \tag{D.5}$$

And can be combined to form

$$\frac{\partial n}{\partial t}(x,t) = D\frac{\partial^2 n}{\partial x^2}(x,t) + J_{source}(x,t) \tag{D.6}$$

In discrete domain this would look like

$$n^{tt}_{x,t} = D n^{xx}_{x,t} + J_{x,t} \tag{D.7}$$

With $J_{x,t} = J_{source}(x,t)$.

### Boundaries

The system is contained between two walls at $x = -\frac{1}{2}$ and $x = l + \frac{1}{2}$, for which $J_{gradient} = 0$ since there is no flow through the walls.

$$Dn^{xx}_{x,t} = -J^{x}_{x,t} = -(J_{x+\frac{1}{2},t} - J_{x-\frac{1}{2},t}) = J_{x-\frac{1}{2},t} - J_{x+\frac{1}{2},t}$$

For $x = 0$ this means

$$Dn^{xx}_{0,t} = J_{-\frac{1}{2},t} - J_{\frac{1}{2},t} = -J_{\frac{1}{2},t} = Dn^{x}_{\frac{1}{2},t} = D(n_{1,t} - n_{0,t})$$

And for $x = l$

$$Dn^{xx}_{l,t} = J_{l-\frac{1}{2},t} - J_{l+\frac{1}{2},t} = J_{l-\frac{1}{2},t} = -Dn^{x}_{l-\frac{1}{2},t} = -D(n_{l,t} - n_{l-\frac{1}{2},t})$$

### Source

The source we're interested in is concentration-driven, and is of form

$$J_{source}(x,t) = K(f(x,t) - n(x,t)) \tag{D.8}$$

Here $f(x,t)$ is a desired concentration, which $n(x,t)$ will tend to. In discrete domain the source is

$$J_{x,t} = K(f_{x,t} - n_{x,t}) \tag{D.9}$$

We are most interested in a single source which will then be denoted $J_{s,t} \equiv J_t = K(f_t - n_{s,t})$ with $s$ the location of the source.

## D.2. Theorem

### Linearity

Let $X = \{0, 1, \dots, l\}$ be the set of integers between 0 and $l$, inclusive. The source at point $j$ and time $m$ is $J_{j,m}$. Then $\forall\ t_0 < t_1,\ i \in X$ we have

$$n_{i,t_1} = \sum_{j \in X} a_j n_{j,t_0} + \sum_{j \in Y} \sum_{m=t_0}^{t_1 - r_j} b_{j,m} J_{j,m} \tag{D.10}$$

With $Y = \{\ x \in X\ :\ |i - x| \le t_1 - t_0 - 1\}$ and $r_j = |i - j| + 1$ being limits to the speed of information induced by the model. $a_j, b_{j,m} \in \mathbb{R}$ are coefficients.

### Single Source

Let $J_{j,m} = 0\ \forall\ j \neq s$ and $J_{s,m} \equiv J_m$. If $r = |k - s| + 1 \ge t_1 - t_0$, then

$$n_{k,t_1} = \sum_{j \in X} a_j n_{j,t_0} + \sum_{m=t_0}^{t_1 - r} b_m J_m \tag{D.11}$$

With $b_m$ coefficients.

### Error

Let $n_{x,t}$ be a model with $J_{s,t} = K(f_m - n_{s,t})$, and $J_{x,t} = 0\ \forall\ x \neq s$.
Let $\bar{n}_{x,t}$ be the same model, but with $f_m = 0$ such that $\bar{J}_{s,t} = -K\bar{n}_{s,t}$, and $\bar{n}_{x,t_0} = n_{x,t_0}\ \forall\ x \in X$. Then for $t > t_0$

$$\varepsilon = n_{k,t_1} - \bar{n}_{k,t_1} = \sum_{m=t_0}^{t_1 - r} b_m f_m \tag{D.12}$$

With $b_m$ coefficients.

# D.3. Proof

## Linearity Lemma

We know that the spatial derivatives of the gradient-driven transport is given by the following equations.

$$n_{x,t}^{xx} = n_{x+1,t} + n_{x-1,t} - 2n_{x,t}, \quad \forall\, x \notin \{0, l\}$$

$$n_{0,t}^{xx} = n_{1,t} - n_{0,t}$$

$$n_{l,t}^{xx} = n_{l-1,t} - n_{l,t}$$

These can all be simplified to the form

$$n_{x,t}^{xx} = \sum_{i \in X} a_i n_{i,t}$$

With $a_i$ being the coefficients corresponding to the formula coefficients of above.
Applying this to the diffusion equation gives us

$$n_{x,t}^{t} = D \sum_{i \in X} a_i n_{i,t} + J_{x,t}$$

So that

$$n_{x,t+1} = n_{x,t} + T\left( D \sum_{i \in X} a_i n_{i,t} + J_{x,t} \right)$$

Which can be rewritten to the equation below by redefining the coefficient

$$n_{x,t+1} = \sum_{i \in X} a_i n_{i,t} + T J_{x,t} \tag{D.13}$$

Let us now substitute this equation into the $n_{i,t}$ term

$$\begin{aligned} n_{x,t+1} &= \sum_{i \in X} a_i \left( \sum_{j \in X} b_j n_{j,t-1} + T J_{i,t-1} \right) + T J_{x,t} \\ &= \sum_{i \in X} a_i \sum_{j \in X} b_j n_{j,t-1} + \sum_{i \in X} a_i T J_{i,t-1} + T J_{x,t} \\ &= \sum_{i \in X} c_i n_{i,t-1} + \sum_{i \in X} \sum_{m=t-1}^{t} d_{i,m} J_{i,m} \end{aligned}$$

If we then substitute $n_{i,t}$ in equation D.13 with that same equation $v - 1$ times, we get

$$n_{x,t} = \sum_{i \in X} c_i n_{i,t-v} + \sum_{i \in X} \sum_{m=t-v}^{t} d_{i,m} J_{i,m}$$

The definition of $\frac{\partial n}{\partial x^2}$ as seen earlier in this section limits the spread of information to one spatial step per time step. $n_{x,t}$ is only a function of $n_{y,t-1}$ if $|x - y| \leq 1$, and $J_{y,t-1}$ if $|x - y| \leq 0$.
Generally, $n_{x,t}$ depends on $n_{y,t-w}$ if $|x - y| \leq w$ and for $J_{y,t-w}$ this is $|x - y| \leq w - 1$ due to induction.
For our purpose we only want to look at the dependency on $J$.

Doing a change of variables $t \to t_1$ and $v \to t_1 - t_0$ we get

$$n_{x,t_1} = \sum_{i \in X} c_i n_{i,t_0} + \sum_{i \in X} \sum_{m=t_0}^{t_1} d_{i,m} J_{i,m} \tag{D.14}$$

$d_{i,m} = 0$ if $n_{x,t_1}$ is independent of $J_{i,m}$. This is true for $J_{y,t_1-w}$ if $|x-y| > w-1$.
$m = t_1 - w \;\Rightarrow\; w = t_1 - m$, so

$$\begin{aligned} |x-i| &> w-1 \\ |x-i| &> t_1 - m - 1 \\ m &> t_1 - |x-i| - 1 \\ m &> t_1 - r_i \end{aligned}$$

We can therefore say that for $r_i = |x-i|+1$, $d_{i,m} = 0 \quad \forall\, m > t_1 - r_i$. As a result

$$\sum_{m=t_0}^{t_1} d_{i,m} J_{i,m} = \sum_{m=t_0}^{t_1-r_i} d_{i,m} J_{i,m} \tag{D.15}$$

$\sum_{m=t_0}^{t_1-r_i} d_{i,m} J_{i,m} = 0$ if $d_{i,m} = 0 \;\forall\; t_1 - r_i \le m \le t_0$, which is true if $t_0 > t_1 - r_i$. Which gives us $r_i > t_1 - t_0 \;\Rightarrow\; |x-i| > t_1 - t_0 - 1$. If we then define a set of $i \in X$ for which this holds, $Y = \{\, x \in X \;:\; |i-x| \le t_1 - t_0 - 1\}$, then we can reduce the term.

$$\sum_{i \in X} \sum_{m=t_0}^{t_1-r_i} b_{i,m} J_{i,m} \;\Rightarrow\; \sum_{i \in Y} \sum_{m=t_0}^{t_1-r_i} b_{i,m} J_{i,m} \tag{D.16}$$

By combining equations D.14, D.15, D.16 we get the linearity lemma D.10.

### Single Source

By setting $J_{j,m} = 0 \;\forall\; j \neq s$ and $J_{s,m} \equiv J_m$. Let $r = |k-s| + 1 \ge t_1 - t_0$ and define $b_{i,m} \equiv b_m$. The linearity lemma D.10 gives

$$n_{k,t_1} = \sum_{j \in X} a_j n_{j,t_0} + \sum_{m=t_0}^{t_1-r} b_m J_m \tag{D.17}$$

## D.4. Application

For practical reasons the source is assumed to be constant over the time window between measurements. This means that theorem D.12 simplifies to

$$\varepsilon = \hat{n}_{k,t_1} - \bar{n}_{k,t_1} = Bf \tag{D.18}$$

With $k$ being the point of measurement, $\hat{n}$ the measured value and $t_1 - t_0$ the window between measurements.
The source coefficient can be determined by simulation with

- $J_t = K(1 - n_{s,t})$
- $n_{x,0} = 0 \;\forall\; x$
- the system parameters $T, D, K$ being the same as the final system we want to use it for

Then $n_{k,t_1-t_0} = B$
To determine the source using measurements follow the next steps.

1. Let $n_{x,t_0} = \hat{n}_{k,t_0}$
2. Measure $\hat{n}_{k,t_1}$
3. Simulate to $\bar{n}_{k,t_1}$ with $\bar{J}_t = -K\bar{n}_{s,t}$

4. Calculate $f_{t_0 \to t_1}$ using error
5. Simulate using $J_t = K(f_{t_0 \to t_1} - n_{s,t})$
6. Update initial condition based on simulation
7. Repeat from step 2