LIÈGE université
School of Engineering

MSc. in Computer Science and Engineering

Automated glucose regulation system

Project of linear control systems

Ivan Klapka
Simon Bernard
Adrien Schoffeniels

December 18, 2019

Contents

1 Control problem and introduction

1.1 Topic

Our topic is the control of blood sugar concentration.

1.2 Context

We would like to create a system regulating the blood sugar concentration for people affected by a type 1 diabetes. These people have their pancreases that do not produce any insulin such that they need to inject themselves what they need. Our linear control system would do it by itself, using an insulin pump, and basically replacing exactly what the pancreas does for a non-diabetic person.

1.3 Control problem diagram

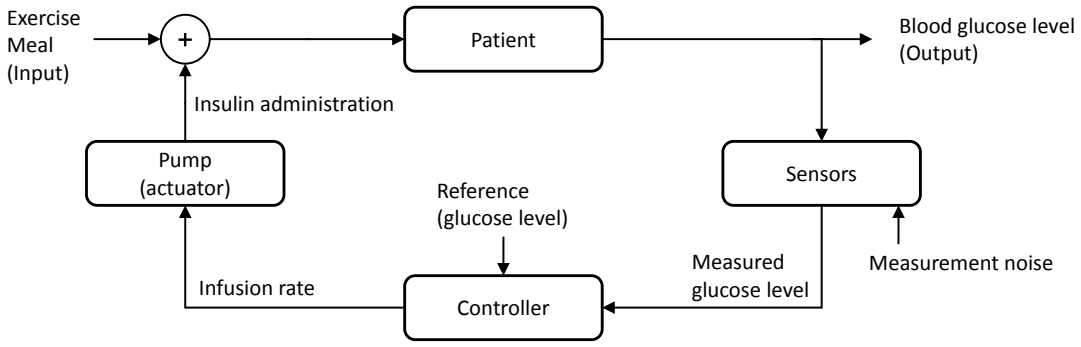

Figure 1 – Closed loop control diagram for glucose regulation

1.4 Control problem description

Utility of the controller Injecting insulin to type 1 diabetic patients to replace the role not taken by their pancreas

System to be controlled Patient's glucose concentration in blood

Inputs into the system Meal (we neglect the other inputs such as exercice,...)

Outputs of the system Blood glucose level

Reference Healthy glucose level (about 90mg/dL of blood)

Sensors Subcutaneous sensor (measure sugar in interstitial fluid)

Actuators Insulin pump

Constraints and Limitations Avoid hypoglycemia and hyperglycemia and inject insulin at a realistic rate.

2 Open loop system

2.1 Detailed schematic of the open loop system

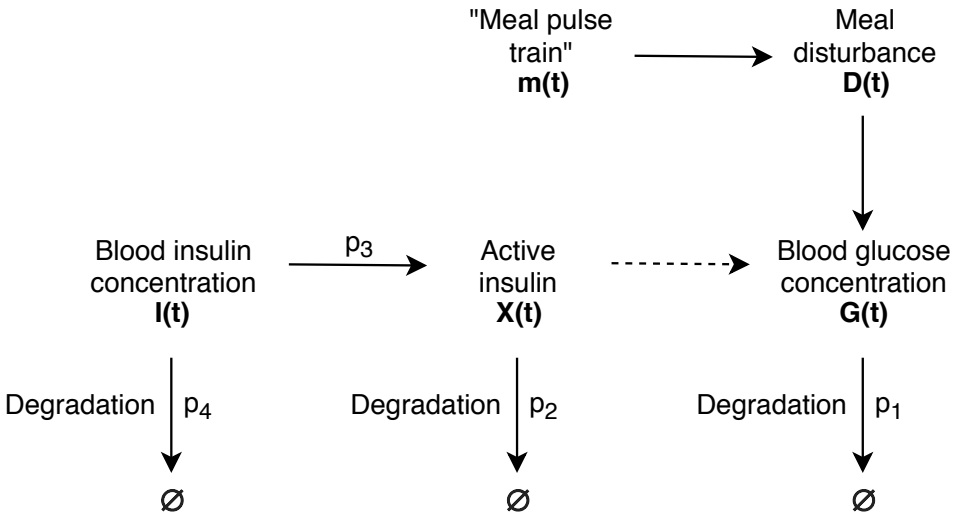

Figure 2 – Schematic of the open loop system

2.2 State-space representation

The model we will use is a coupling between Bergman's minimal models with modifications and additions done by Friis-Jensen E. [1] In our case, the input of exogenous insulin is not already considered. We also suppose as a simplification that we can measure directly in the blood the glucose concentration (as suggested by the teacher).

The model is described by the following system :

[1] Friis-Jensen E., "Modeling and Simulation of Glucose Insulin Metabolism", technical university of Denmark informatics and mathematical modeling, (2007). (Section 2.2.3.2)

$$\begin{cases} \dot{G}(t) &= -\left(p_1 + X(t)\right) G(t) + p_1 G_b + D(t) \\ \dot{X}(t) &= -p_2 X(t) + p_3 \left(I(t) - I_b\right) \\ \dot{I}(t) &= -p_4 I(t) + \frac{u(t)}{V_I} \\ \dot{D}(t) &= -d_{rate} D(t) + m(t) \\ Y(t) &= G(t) \end{cases}$$

As we consider $I_b = 0$, we get :

$$\begin{cases} \dot{G}(t) &= -\left(p_1 + X(t)\right) G(t) + p_1 G_b + D(t) \\ \dot{X}(t) &= -p_2 X(t) + p_3 I(t) \\ \dot{I}(t) &= -p_4 I(t) + \frac{u(t)}{V_I} \\ \dot{D}(t) &= -d_{rate} D(t) + m(t) \\ Y(t) &= G(t) \end{cases}$$

Unfortunately, our system is non-linear due to the $X(t)G(t)$. We need to linearize it around a fixed point. As we aim a glucose concentration of 90mg/dL, we will take it as our fixed point (and 0 for other variables while no values are required for them). By the way, G_b being the basal blood concentration and so assuming that it also is 90mg/dL, it turns out that our fixed point is also an equilibrium point ($G_e = G_b = 90, X_e = 0, I_e = 0, D_e = 0, m_e = 0$).
After the linearization we obtain :

$$\begin{cases} \dot{\delta G}(t) &= -\left(p_1 + X_e\right) \delta G(t) - G_e \delta X(t) + \delta D(t) \\ \dot{\delta X}(t) &= -p_2 \delta X(t) + p_3 \delta I(t) \\ \dot{\delta I}(t) &= -p_4 \delta I(t) + \frac{\delta u(t)}{V_I} \\ \dot{\delta D}(t) &= -d_{rate} \delta D(t) + \delta m(t) \\ Y(t) &= \delta G(t) \end{cases}$$

Which can be simplified, using the null equilibrium value X_e:

$$\begin{cases} \dot{\delta G}(t) &= -p_1 \delta G(t) - G_e \delta X(t) + \delta D(t) \\ \dot{\delta X}(t) &= -p_2 \delta X(t) + p_3 \delta I(t) \\ \dot{\delta I}(t) &= -p_4 \delta I(t) + \frac{\delta u(t)}{V_I} \\ \dot{\delta D}(t) &= -d_{rate} \delta D(t) + \delta m(t) \\ Y(t) &= \delta G(t) \end{cases}$$

We notice that we use as a new output the variation around 90mg/dL of glucose concentration in blood.
We can now build our ABCD system with those new equations.

$$
\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}
$$

$$
= \begin{cases} \begin{pmatrix} \delta\dot{G}(t) \\ \delta\dot{X}(t) \\ \delta\dot{I}(t) \\ \delta\dot{D}(t) \end{pmatrix} = \begin{pmatrix} -p_1 & -G_e & 0 & 1 \\ 0 & -p_2 & p_3 & 0 \\ 0 & 0 & -p_4 & 0 \\ 0 & 0 & 0 & -d_{rate} \end{pmatrix} \cdot \begin{pmatrix} \delta G(t) \\ \delta X(t) \\ \delta I(t) \\ \delta D(t) \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{V_I} \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \delta m(t) \\ \delta u(t) \end{pmatrix} \\ Y(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \delta G(t) \\ \delta X(t) \\ \delta I(t) \\ \delta D(t) \end{pmatrix} + \begin{pmatrix} 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \delta m(t) \\ \delta u(t) \end{pmatrix} \end{cases}
$$

All the parameters used in the equations are listed in Table 1.

Parameter	Unit	Description
G(t)	[mg/dL]	Blood glucose concentration
X(t)	[1/min]	The effect of active insulin ("remote insulin")
I(t)	[mU/L]	Blood insulin concentration
D(t)	[mg/dL/min]	Meal disturbance function
m(t)	$[\text{mg/dL/min}^2]$	Pulse train when meals are taken
G_b	[mg/dL]	Basal blood glucose concentration
I_b	[mU/L]	Basal blood insulin concentration
p_1	[1/min]	Glucose clearance rate independent of insulin
p_2	[1/min]	Rate of clearance of active insulin (decrease of uptake)
p_3	$[L/(\text{min}^2\text{m U})]$	Increase in uptake ability caused by insulin.
p_4	[1/min]	Decay rate of blood insulin.
d_{rate}	[1/min]	Decay rate of the meal disturbance
V_I	$[L]$	Volume of blood
$u(t)$	$[mU/min]$	Injected insulin (controllable input)

Table 1 – Description of the parameters of the equations

NB : We color the $/min$ in red because we realised we missed it. Unfortunately, this was too late (10th of December) to change all our project. This means that all discussions we thought we made over the derivative of u(t) was in fact on the acceleration of the pump instead of its rate. We know that this small change can make huge difference. However, we hope you will not penalize us too much as the the practical approach of designing under constraint as been done, even though it was not the good one.
Moreover, this does not change the rightness of the blood glucose concentration. In addition, we (hopefully) made also a mistake (on the unit) on what we thought to be the normal rate, taking something much smaller. This means that the rate we get is still in

an accepting range.
This also means that some discussion/graph in the report could be wrong. We will do our best to correct everything, but if a mistake in formulation remains, it could be from that.

The numerical values used for the different parameters are listed in Table 2. Most of them are directly taken from Friis-Jensen E's modified model, except for p_1 which has been tuned by simulating the system in order to have a realistic scenario, and V_I which was not realistic.

Parameter	Value
G_b	90
p_1	0.00058735
p_2	0.028344
p_3	$5.035 \cdot 10^{-5}$
p_4	0.3
d_{rate}	0.05
V_I	6

Table 2 – Numerical values used for the different parameters

2.3 Constraints, assumptions, limitations

- A healthy person normally has a blood glucose concentration at about 70 to 110 mg/dL. As an exact value will be required, we will assume 90mg/dL as target value[2].
- Blood glucose level should stay between 60 mg/dL (hypoglycemia) and 270 mg/dL (hyperglycemia).
- We focus our attention on diabetes of type 1.
- Exercises and physical activities will not be taken into account in this model.
- We assume basal blood insulin concentration (I_b) to be equal to 0, the reasoning is that a type 1 diabetic is not able to achieve a basal blood insulin concentration.
- We assume that our model is not able to produce glucagon, so we will avoid as much as possible negative values of insulin.
- As said in the previous section, we made mistake on units. This means that we have a constraint on the acceleration of the pump that can not be greater than 0.66mu/min^2 (the true value for the maximum rate is about 500mu/min^3, which will, as we will see later, be also respected).

[2] `https://www.diabetes.co.uk/diabetes_care/blood-sugar-level-ranges.html`
[3] `https://www.sps.nhs.uk/wp-content/uploads/2018/05/Insulin-pump-table-May-2018.pdf`

- The meal disturbance (uncontrollable input) consist in a train of impulsion, each impulsion correspond to a meal the patient has ingested and the value of the peak is proportional to the size of the meal (how much glucose will be extracted from the meal). Each impulsion ranges from a scale from 1 to 10, for example a 1 would be a small snack with little glucose and a 10 would be a large and sugar abundant meal. Our goal is to be able to able to response to a impulse of up to size 12. We also assume our meals to be taken at least two hours apart from each other and a maximum of 3 meals per day.

2.4 System simulations without controller

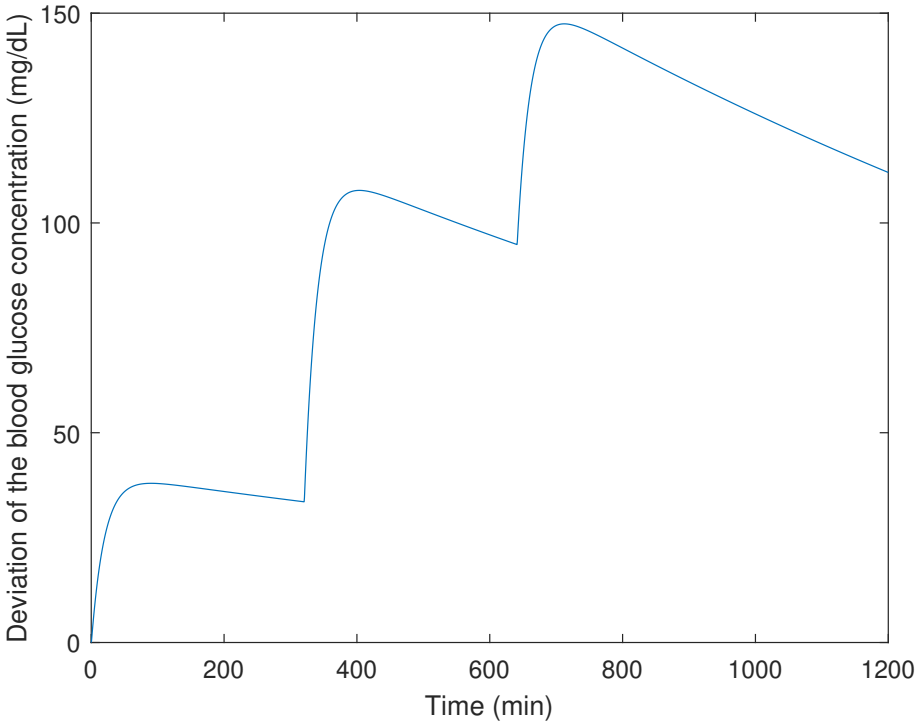

Figure 3 – Simulation of a complete day with 3 meals (of size 8, 12 and 10) for the open-loop system

A simulation of a complete day with 3 meals for the open-loop system is shown in Figure 3. As one can see, the system seems to be stable but it reacts very slowly, which is the expected result since there is no insulin at all in the system. As a result, the blood glucose concentration explodes, reaching 350 mg/dL, which is way higher than the hyperglycemia level and becomes very dangerous for the patient. Indeed such high values will disrupt the whole human body (for example the plasma's pH will be unbalanced which can be lethal if not treated).

In order to avoid this type of scenario, we will design a controller associated to a controllable input which will inject insulin in the blood when needed in order for the system to be way more reactive. This will be done in the next section.

One can also observe that the glucose concentration deviation doesn't go below 0. This is the consequence of the linearization around the concentration of 90mg/dL. This is not really a problem since the goal of the system is to decrease the glucose concentration when a meal is taken, since we can not directly add glucose in the system (the glucose rise is the disturbance of the system).

2.5 State-space representation analysis

Before designing the controller (and the observer), we first need to verify if our system is stable, controllable and observable. Therefore we can write our system using the ABCD state-space representation to directly apply the formulas derived in the theoretical lectures. Our system can be represented by the following ABCD system :

$$A = \begin{pmatrix} -p_1 & -G_e & 0 & 1 \\ 0 & -p_2 & p_3 & 0 \\ 0 & 0 & -p_4 & 0 \\ 0 & 0 & 0 & -d_{rate} \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{V_I} \\ 1 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 0 & 0 \end{pmatrix},$$

(a) Stability

In order to study the stability of our system, the inputs are set to zero (controllable and non-controllable). We thus have

$$\dot{x} = Ax$$

Where x are the states of our system. The system stability is determined by the eigenvalues of A. They are given by :

$$\begin{aligned} |\lambda \mathbb{I} - A| &= \begin{vmatrix} \lambda + p_1 & G_e & 0 & -1 \\ 0 & \lambda + p_2 & -p_3 & 0 \\ 0 & 0 & \lambda + p_4 & 0 \\ 0 & 0 & 0 & \lambda + d_{rate} \end{vmatrix} \\ &= (\lambda + p_1)(\lambda + p_2)(\lambda + p_4)(\lambda + d_{rate}) = 0 \end{aligned}$$

$$\Rightarrow \begin{cases} \lambda_1 = -p_1 = -0,00058735 \\ \lambda_2 = -p_2 = -0,028344 \\ \lambda_3 = -p_4 = -0,3 \\ \lambda_4 = -d_{rate} = -0,05 \end{cases}$$

Since all the eigenvalues of the matrix are negative, the system is stable.

One can also notice that the two parameters that influence badly the behavior of our system are p_1 and p_2 (since they are small, they lead to the slow reactivity of the system). This information will help us to design the controller in the next section.

(b) Controllability

In order to study the controllability of the system, the controllability matrix W_R is computed. For a system with four states, it is defined as follows:

$$W_R = \begin{pmatrix} B & AB & AAB & AAAB \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 0 & 1 & 0 & -d_{rate} - p_1 & -\frac{G_b p_3}{V_I} & p_1^2 + p_1 d_{rate} + d_{rate}^2 & \frac{p_3 G_b (p_1+p_2+p_4)}{V_I} \\ 0 & 0 & 0 & \frac{p_3}{V_I} & 0 & -\frac{p_3(p_2+p_4)}{V_I} & 0 & \frac{p_3(p_2^2+p_2 p_4+p_4^2)}{V_I} \\ 0 & \frac{1}{V_I} & 0 & -\frac{p_4}{V_I} & 0 & \frac{p_4^2}{V_I} & 0 & -\frac{p_4^3}{V_I} \\ 1 & 0 & -d_{rate} & 0 & d_{rate}^2 & 0 & -d_{rate}^3 & 0 \end{pmatrix}$$

Since this matrix is full rank, the system is controllable. This means that the states of the system can be modified acting on the controllable input. Since the matrix B has two non-zero values, there are two actuators needed to control the state. As we assume the meals and thus the rise of glucose concentration to be the disturbance, we only need one actuator to control the system (since our goal is to decrease the glucose concentration when it increases).

(c) Observability

To study the observability of the system, the observability matrix W_O is computed. For a system with four states, it is defined as follows:

$$W_O = \begin{pmatrix} C & CA & CAA & CAAA \end{pmatrix}^T$$
$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ -p_1 & -G_b & 0 & 1 \\ p_1^2 & G_b(p_1+p_2) & -G_b p_3 & -d_{rate} - p_1 \\ -p_1^3 & -G_b(p_1^2 + p_1 p_2 + p_2^2) & G_b p_3 (p_1 + p_2 + p_4) & p_1^2 + p_1 d_{rate} + d_{rate}^2 \end{pmatrix}$$

Since this matrix is full rank, the system is observable. This means that we can infer the states of the system by measurements over time of its input and output. Since C has one non-zero value, we need one sensor, measuring the glucose concentration.

3 Controller in time domain

3.1 State-feedback controller

In the case of the state feedback controller, the controlled input is given by the following equation :

$$u = -Kx + k_r r = -\begin{pmatrix} k_1 & k_2 & k_3 & k_4 \end{pmatrix} \begin{pmatrix} \delta G(t) \\ \delta X(t) \\ \delta I(t) \\ \delta D(t) \end{pmatrix} + k_r r$$

By replacing this equation in

$$\dot{x} = Ax + Bu$$

we obtain:

$$\dot{x} = (A - BK)x + k_r Br$$

Since our system is linearized around the equilibrium point, the reference input k_r is null a each time. We thus obtain

$$\dot{x} = (A - BK)x$$

We want our system to be stable and to have a good reactivity to disturbances (meals) so we will choose the eigenvalues to guarantee the good behavior of our closed-loop system. To do so we will use the `place()` function available in Matlab that computes the values of K to have a system with such eigenvalues.

When we studied the stability of the open-loop system, we highlighted the fact that there were two eigenvalues that influence badly the reactivity of the system. Therefore, we start by fixing two eigen values of our closed-loop system to be the same as the one that doesn't influence badly the behavior of the open-loop system :

$$\begin{cases} \lambda_1 = -p_4 = -0,3 \\ \lambda_2 = -d_{rate} = -0,05 \end{cases}$$

For the two other eigenvalues, we need to tune them by doing simulations using the `place()` function until we obtain the desired result.

We ended with the following two eigenvalues :

$$\begin{cases} \lambda_3 = -0,02 \\ \lambda_4 = -0,03 \end{cases}$$

For such eigenvalues, we have the following K matrix :

$$K = \begin{pmatrix} -0,2264 & 750,0778 & 0,1264 & -0,3510 \end{pmatrix}$$

Figure 4 shows the simulation of a complete day with 3 meals (see fig 27) for the system with the controller that we just designed. One can see that the controlled system is a lot more reactive than the open-loop system. Moreover, the blood glucose concentration even stays in the recommended range (see section 2.3).

One can also highlight by looking at Figure 5 and Figure 6 that the insulin injection rate and acceleration are totally acceptable with the limitations of the insulin pump considered (see section 2.3).

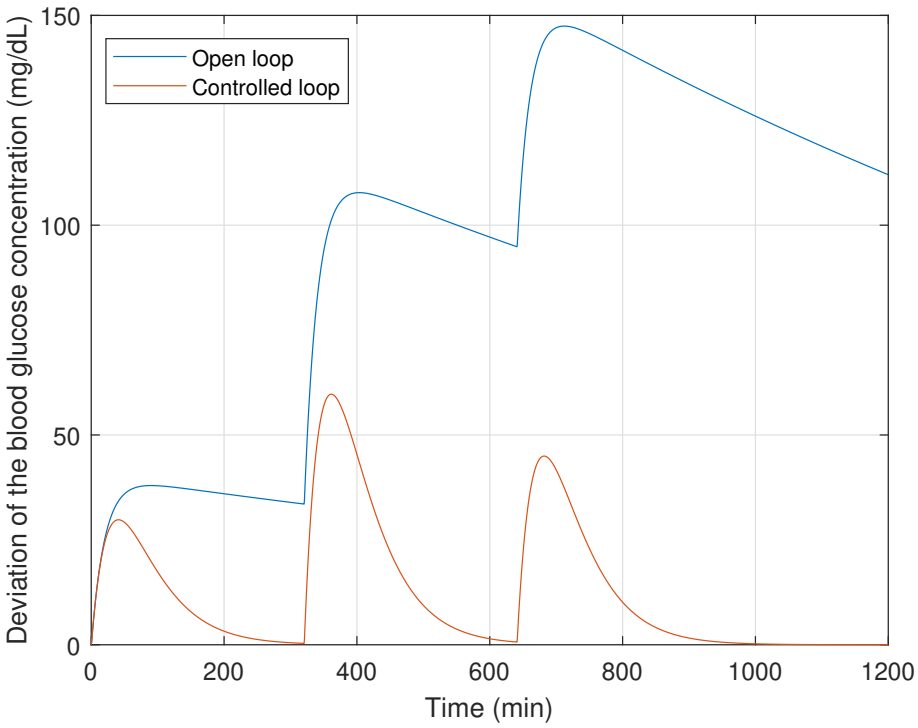

Figure 4 – Deviation of the blood glucose concentration for input at fig 27

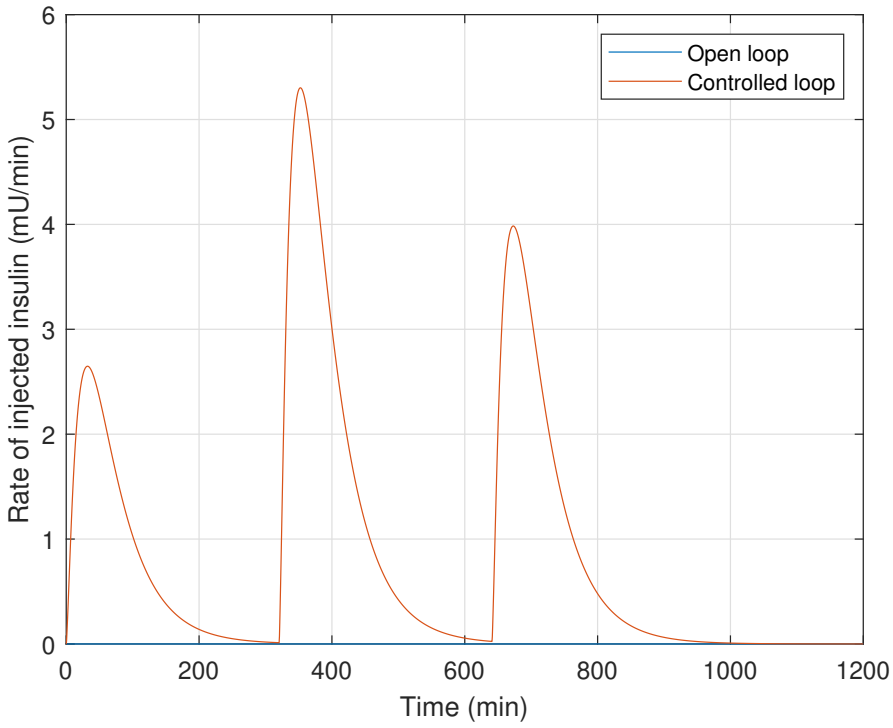

Figure 5 – Rate of the injected insulin for input at fig 27

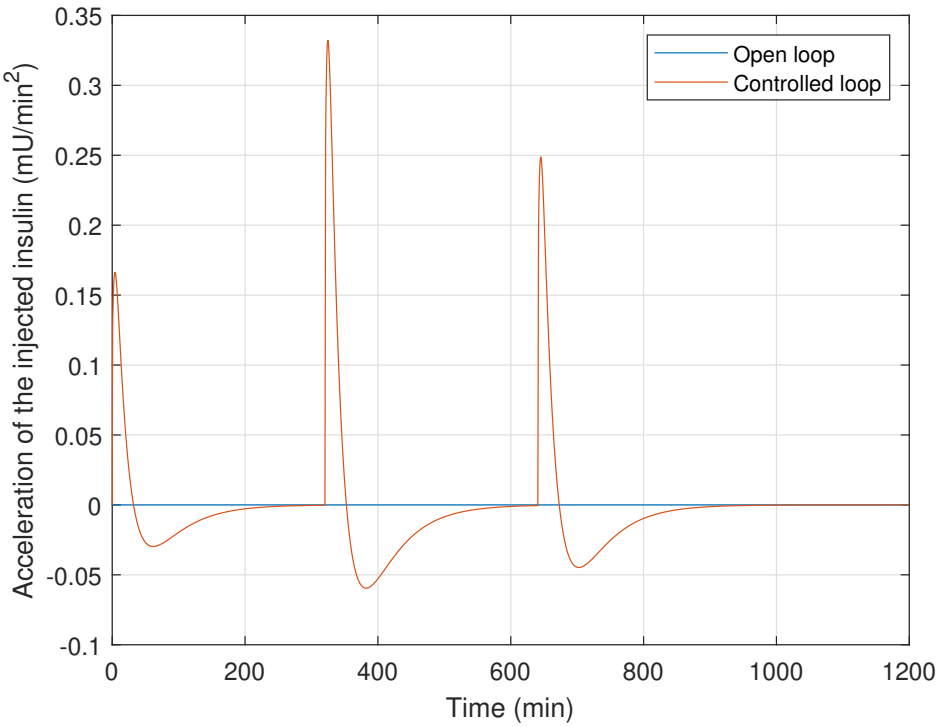

Figure 6 – Acceleration of the injected insulin for input at fig 27

3.2 Observer

Since the system is observable, we can construct an observer that estimates the state of the system using its input and output :

$$\begin{aligned}\frac{d\hat{x}}{dt} &= A\hat{x} + Bu + L(y - C\hat{x}) \\ &= (A - LC)\hat{x} + Bu + Ly\end{aligned}$$

This time, we would like to design the observer in order to have the state estimate converging fast toward is real value. Therefore, we will use the function `place()` once again to fix the eigenvalues until we find a good result when we simulate the system. This function will give us the L matrix needed to have such eigenvalues with the associated behaviour. Since we want our observer to have faster dynamics than the system itself, we place the poles (associated to the eigenvalues) that influence the most the system much farther to the left for the observer. So once again we start by fixing two eigenvalues to be the same as the one that doesn't influence badly the behavior of the open-loop system :

$$\begin{cases}\lambda_1 = -0,3 \\ \lambda_2 = -0,05\end{cases}$$

Then, as for the controller, we tune the other eigenvalues, keeping in mind that we need to place the poles associated to them farther to the left. After a few simulations, we found the following eigenvalues (5 times the one of the open-loop system) :

$$\begin{cases} \lambda_3 = -0,03 \\ \lambda_4 = -0,15 \end{cases}$$

The following graph was simulated with the initial condition of meal size at 8 for the real state and 7 for the others.

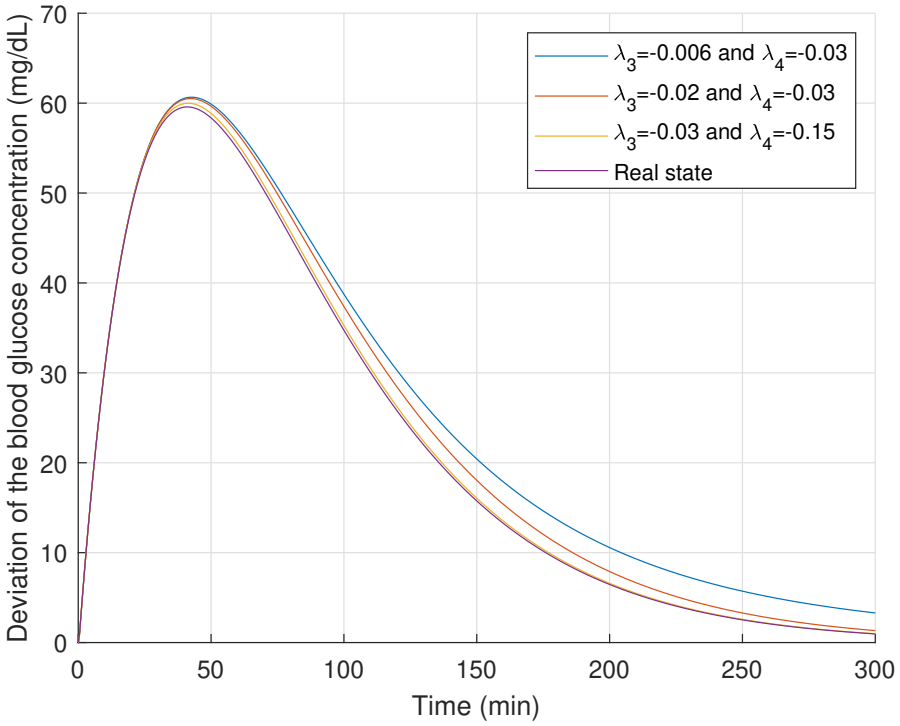

Figure 7 – Output of the system for different poles placements of A-LC

We can see on the graph above that when choosing to place the poles at $\lambda_3 = -0.03$ and $\lambda_4 = -0.15$ we converge very quickly to the real state as desired.

3.3 Simulations and discussion

(a) Response to a reference variation

First we compute k_r thanks to the following formula (with D=0),

$$k_r = \frac{-1}{C(A - BK)^{-1}B} = -0.2383$$

We then plot the deviation of the blood glucose concentration and see that as expected we converge towards the new reference.

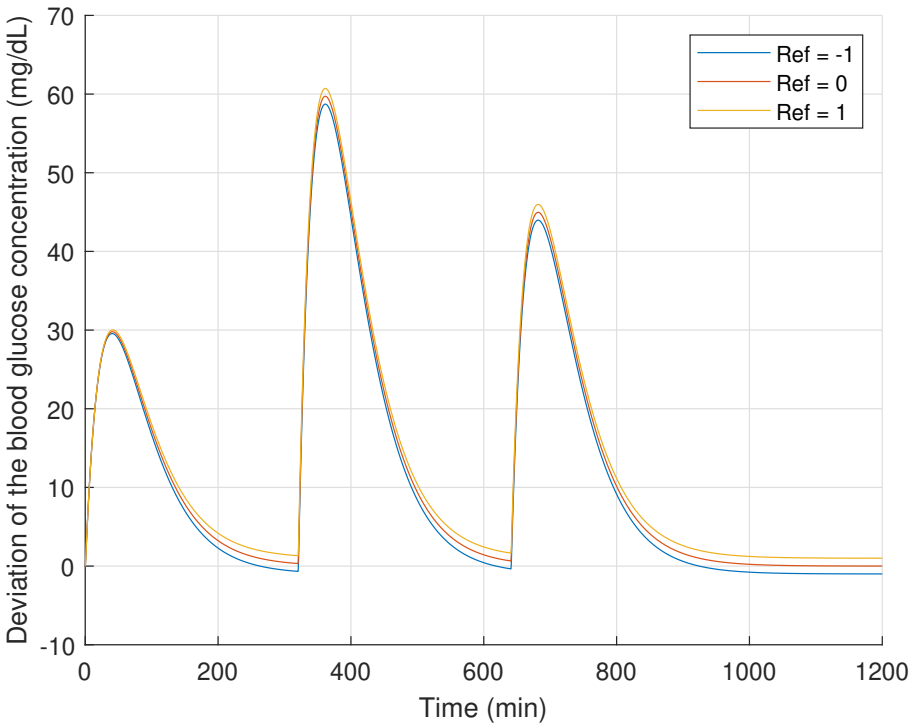

Figure 8 – Deviation of the blood glucose concentration (mg/dL) for different reference

(b) Response to a perturbation

We simulate the scenario that a patient ingest a lot of food in a relatively short period of time(2 meals of size 8 and 10 spaced between 2 hours). As we can see it behaves as expected, respecting all our constraints.

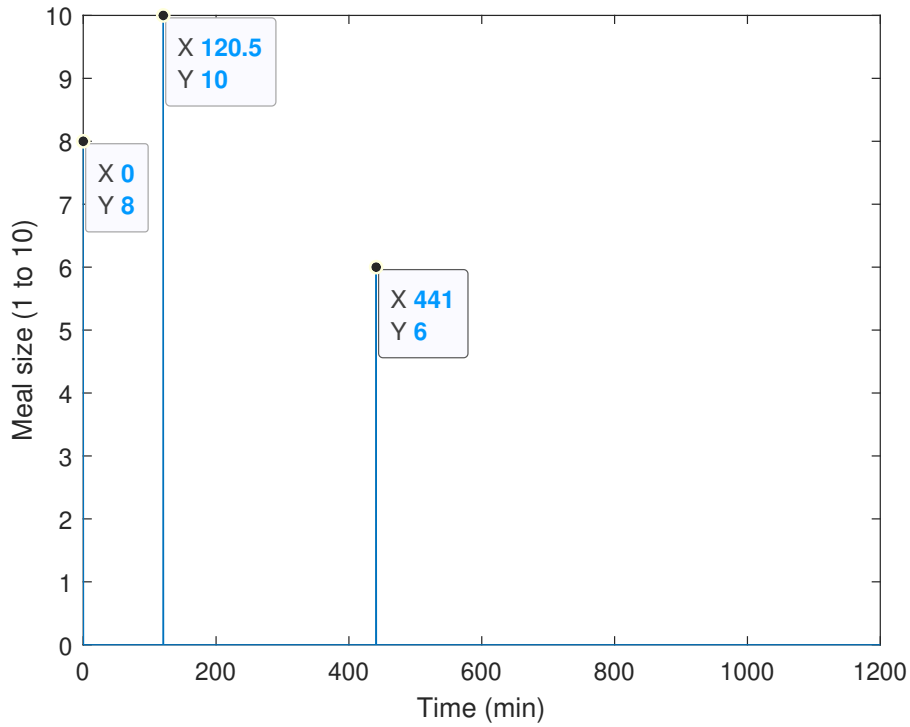

Figure 9 – Meal disturbance impulsion train

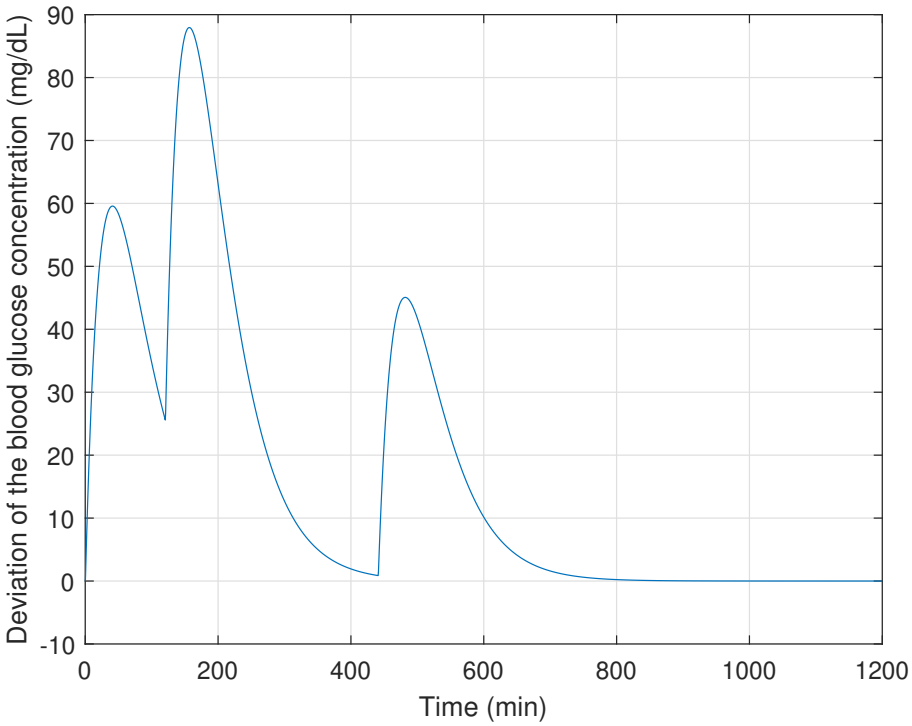

Figure 10 – Deviation of the blood glucose concentration for input at Fig 9

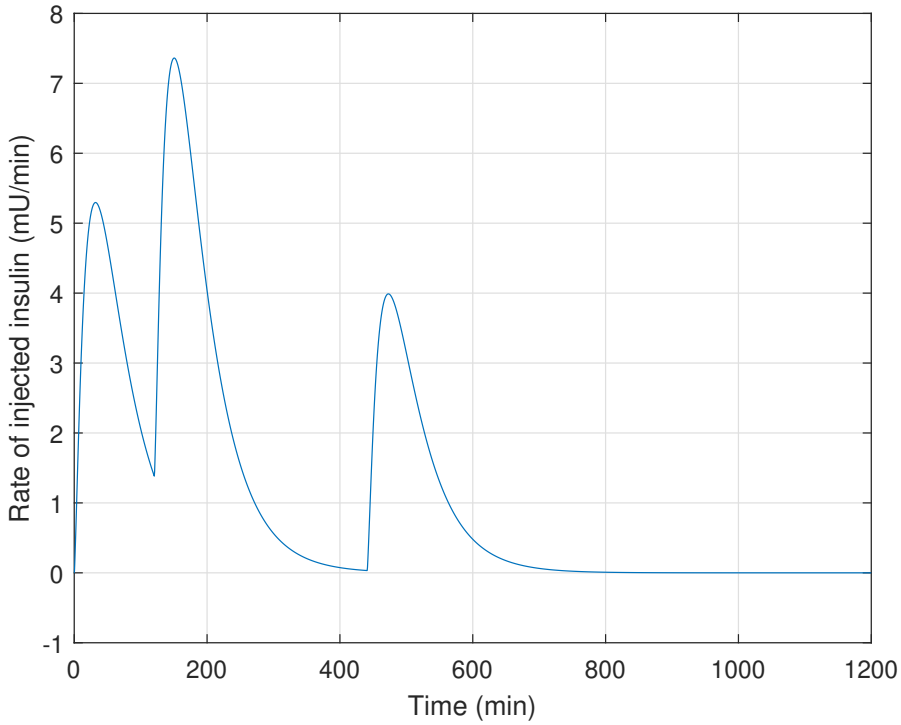

Figure 11 – Rate of injected insulin for input at Fig 9

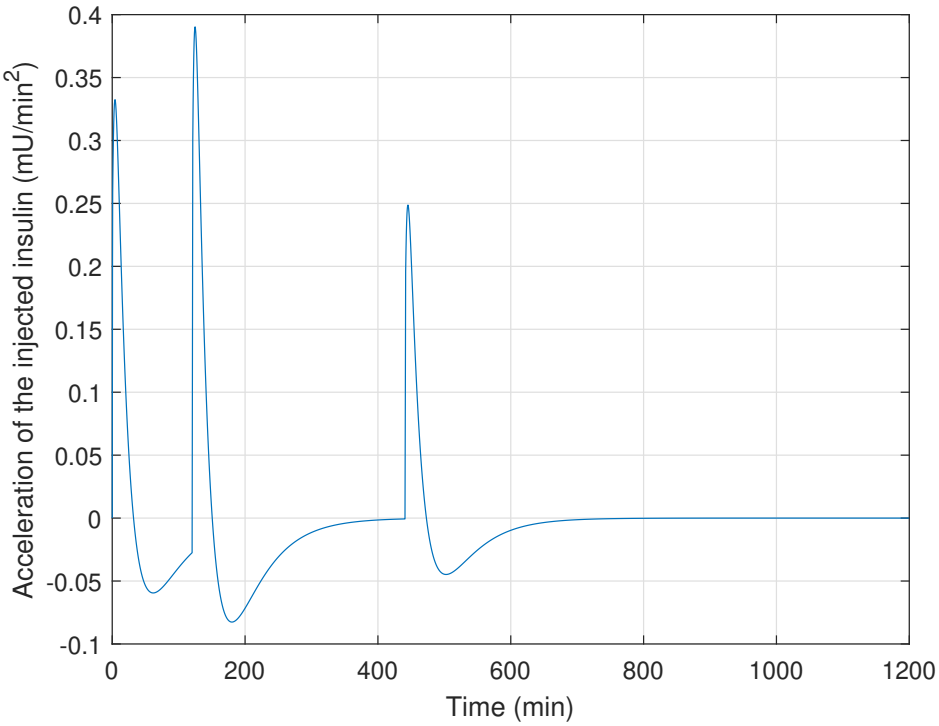

Figure 12 – Acceleration of injected insulin for input at Fig 9

(c) Noise

To simulate noise we added a random uniformly chosen numbers ranging[4] from [-2, 2] to the output of our system.

In this case, our system does not behave in a good way. First we can see that there is no noise in the glucose concentration, which is not a good behavior. Indeed, this means that our system reacts to the noise and counters it, while we would want it to not be considered by our controller. We can also see that the acceleration of our pump exceeds our constraint. A noise filter would thus be mandatory to implement the time-domain controller.

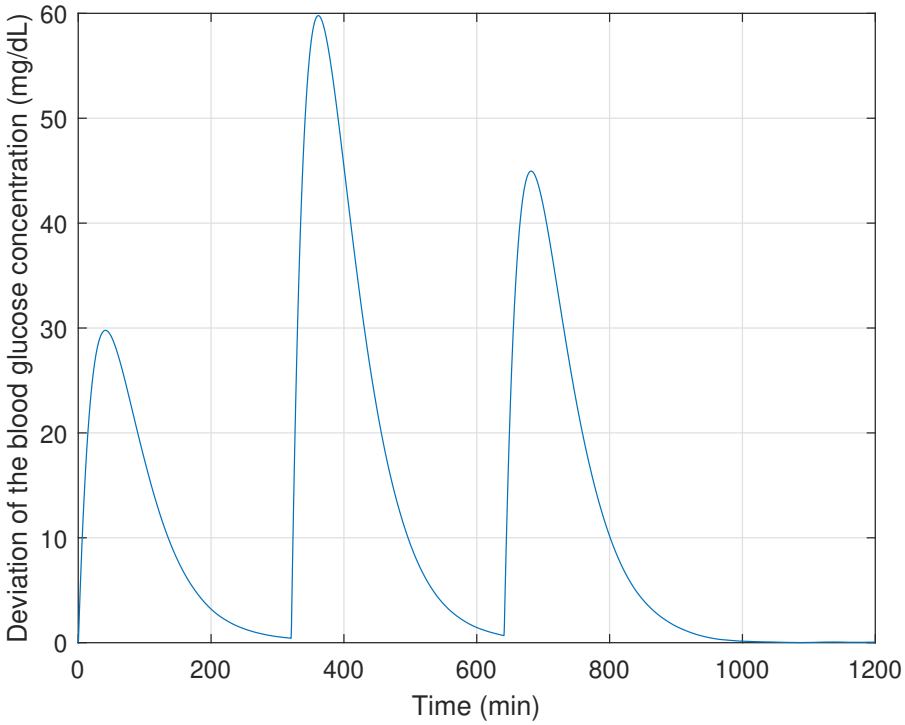

Figure 13 – Deviation of the blood glucose concentration with noise

[4]The range is set to correspond to the precision of a glucose sensors working around 40 mg/dL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4455380/`

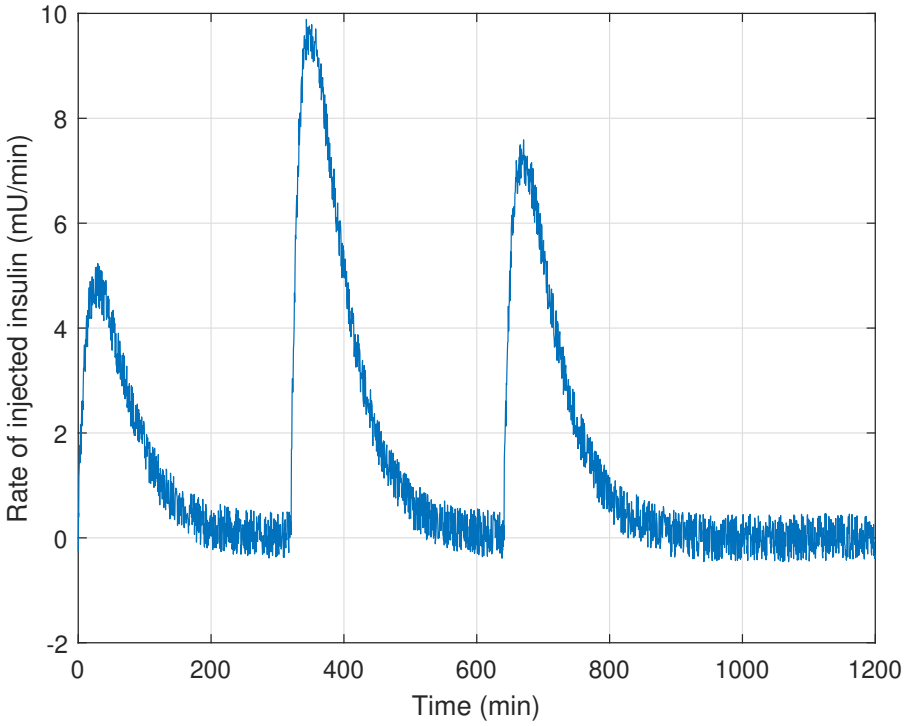

Figure 14 – Rate of injected insulin with noise

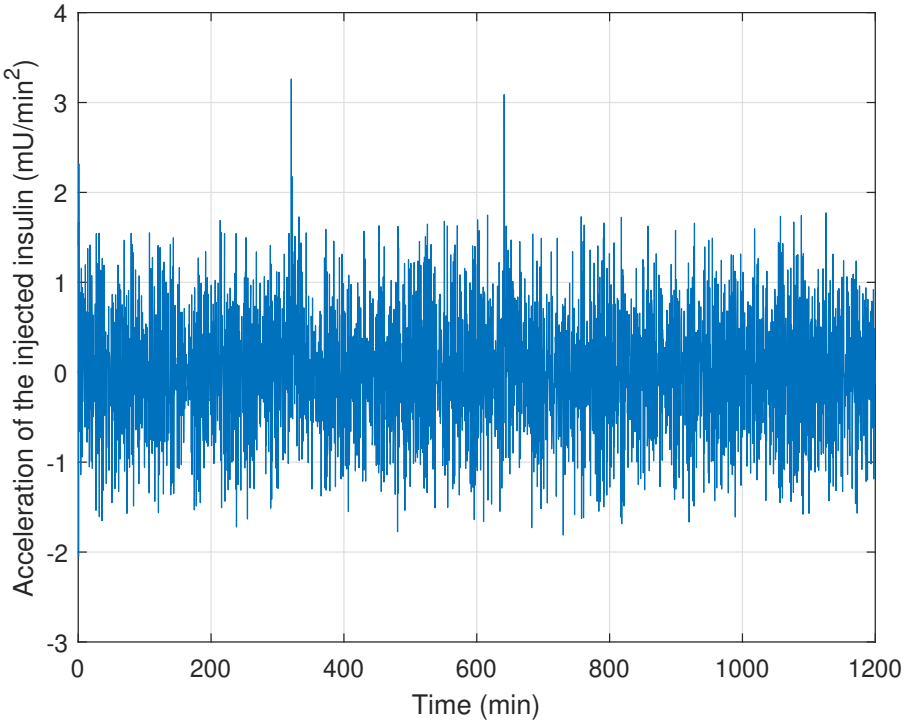

Figure 15 – Acceleration of injected insulin with noise

4 Controller design in frequency domain

Note: For every graph and legend in this section, please consider the mentions "rad/s" to be instead "rad/min".

4.1 Transfer function H(s)

(a) Computation

In the frequency domain the transfer function is defined as follows :

$$H(s) = \frac{Y(s)}{U(s)}$$

The state space representation is written as follows :

$$\begin{cases} sX(s) = AX(s) + BU(s) \\ Y(s) = CX(s) + DU(s) \end{cases}$$

Using these equations and injecting these in the transfer function definition, we get

$$H(s) = C(s\mathbb{I} - A)^{-1}B + D$$

This equations give us two different transfer functions, one for each input.

For the disturbance (meal), we have

$$H_1(s) = \frac{1}{s^2 + 0.05059s + 2.937 \cdot 10^{-5}}$$

For the controllable input (insulin), we have

$$H_2(s) = \frac{-0.0003776}{s^3 + 0.3289s^2 + 0.008696s + 4.994 \cdot 10^{-6}} = P(s)$$

(b) Bode diagram

As one can see on the graph below, the system is amplifying low frequencies while attenuating high frequencies. It is also shifting the phase until it reaches -270 °. It is stable but it is not converging fast enough as we would like to avoid hyperglycemia.

The first step is choosing the crossover frequency, which we decided should be about 0,1 Hz (ω_c = 0,6283 rad/s). This is due to the fact that the blood glucose is a slow system and it does not require a high frequency to operate. We see that we will need to filter out the high frequency to get rid of the noise.

The second step is to increase the phase margin around our crossover frequency as a way to account for potential delay. In addition, we need to avoid static error as well as to be robust to load disturbance.

For that we will design different components of C(s) to shape L(s) into an acceptable form.

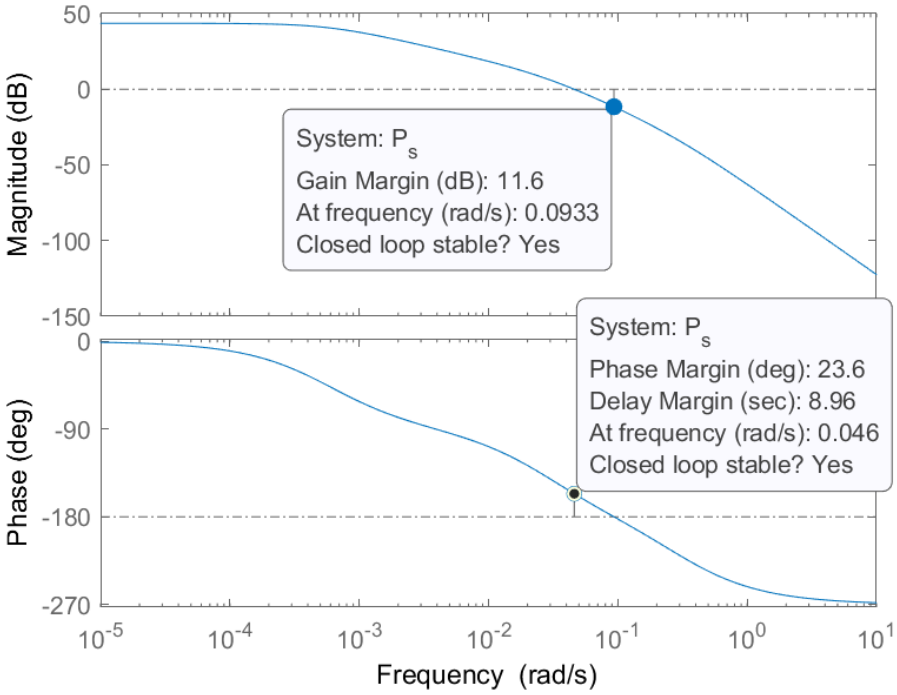

Figure 16 – Bode diagram of the transfer function $-P(s)$

4.2 Loop Shaping

Note: When we add a new component we consider all the previous ones to be inside our system. This mean that the plots done for that component represents the system for that component with all the previous ones.

(a) Gain

The first component of the controller is the gain and since the output of our system (glucose concentration) is decreasing when the input (insulin) is increasing our gain has to be negative (-K). We do this in order to have negative feedback instead of a positive feedback and to still be able to use the Nyquist convention as usual. Since a minus is now added to the controller, **the next bode diagrams of L(s) will theoretically be of be of** $L(s) = -P(s)C(s)$.

Now we need to find the value of K, so we plot the output for different values of K (and a meal size of 10 at time 0).

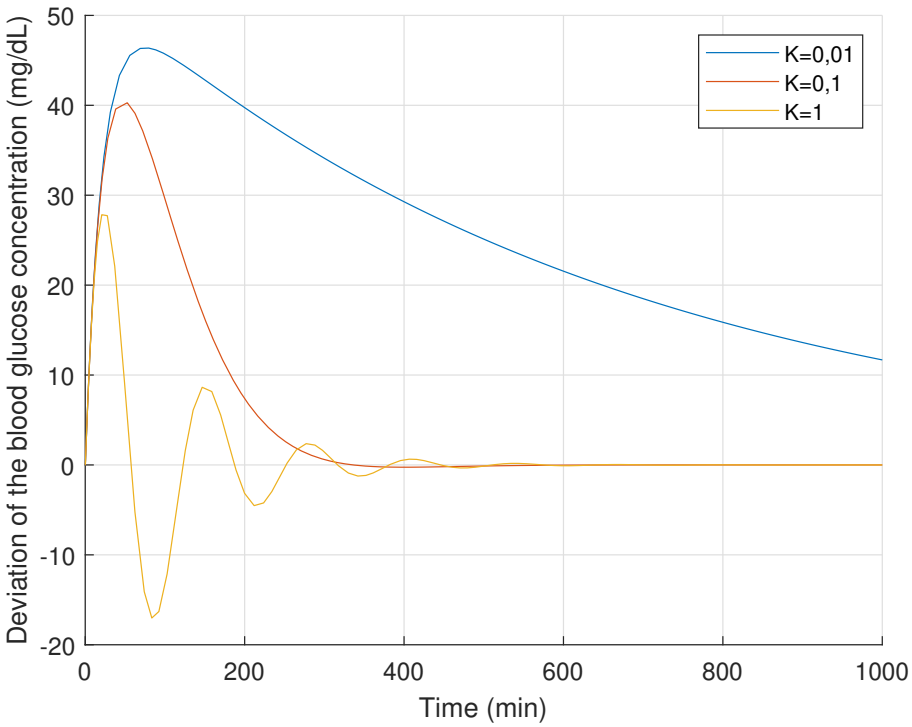

Figure 17 – Output of the system for different gains

We do the same for the rate of injected insulin,

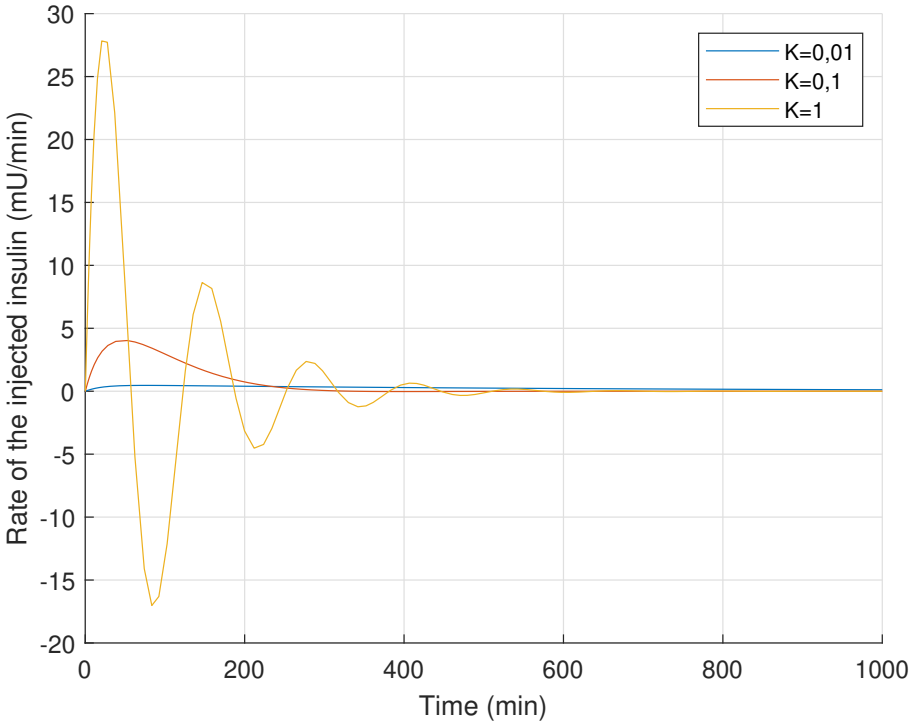

Figure 18 – Rate of injected insulin for different gains

In the end we chose the gain K = 0,1 because it seems to be a good compromise as the glucose concentration stabilise quickly enough without the rate of injected insulin oscillating[5]. It also as an acceptable acceleration of insulin delivery. For the gain we thus have,

$$C_1(s) = -K = -0,1$$

(b) The lead compensator

The lead compensator is the second component of our system, it was designed to increase the phase margin around our cross over frequency (at the cost of increasing the amplitude of higher frequency). Here is the transfer function of a lead compensator,

$$G_L(s) = G_{L_0} \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}}$$

And below is the bode plot of $L(s)$ with different values of the frequency ω_z at which we start increasing the phase margin. The frequency ω_p is where the phase return back to what it was before. The distance between these frequencies is set to $\omega_p = 10 \cdot \omega_z$ so the frequency range is narrow and allows for a precise increase in the phase margin.

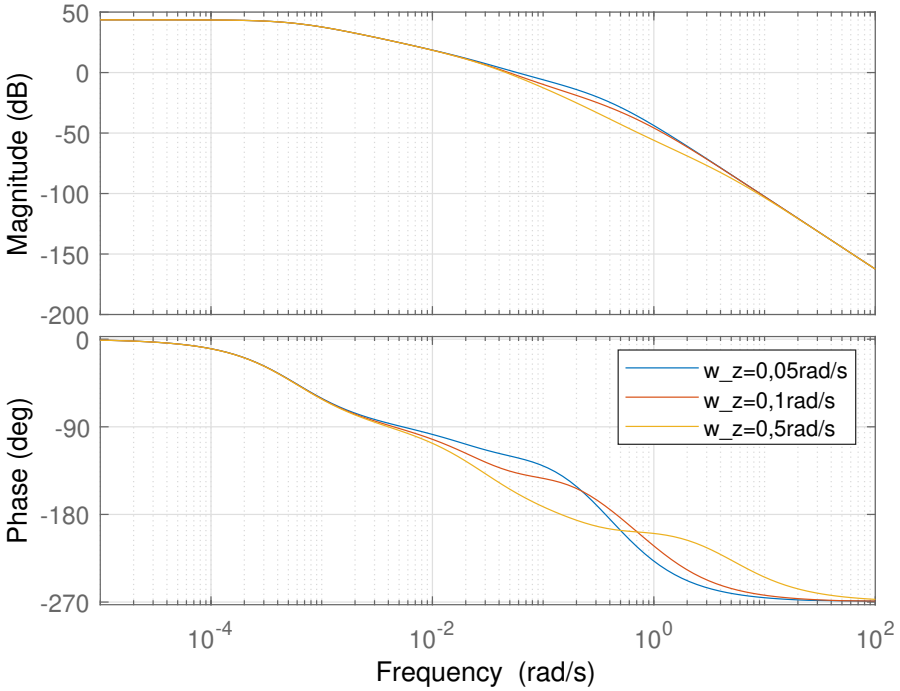

Figure 19 – Bode diagram of $L_(s)$ for different frequencies ω_z

[5]In order to remain coherent with the real word, we want to avoid oscillating as much as possible. This is because it is theoretically impossible to remove injected insulin. In the real word this effect could be associated with the ones of the glucagon as it increases the concentration of blood glucose. While this effect is not really significant for small amounts we decided best to avoid it as much as possible

Here are the simulations (with the meal input seen at fig 27),

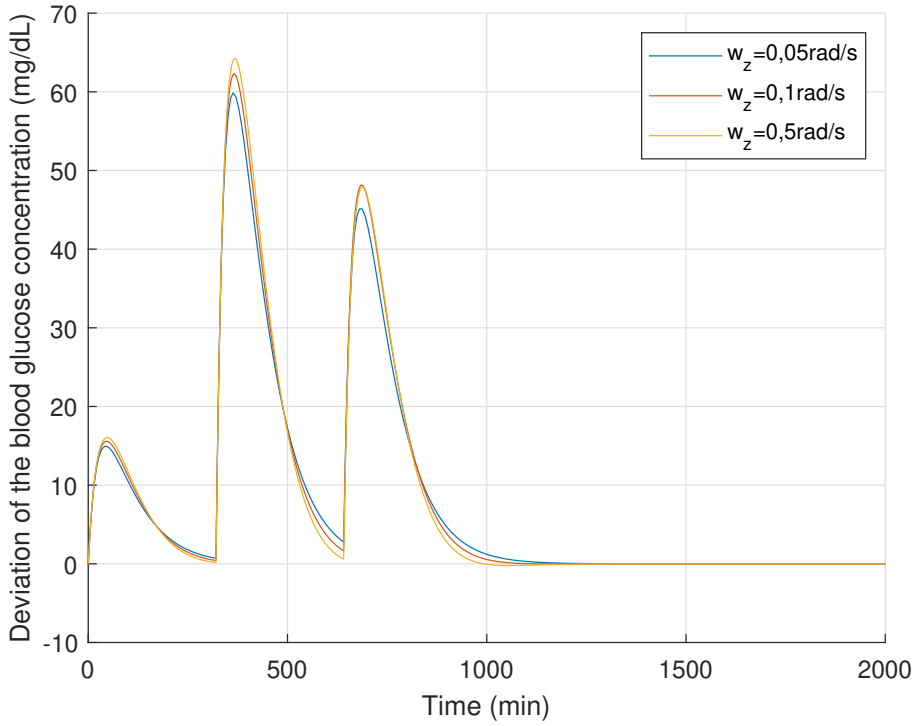

Figure 20 – Output of the system for different frequencies ω_z

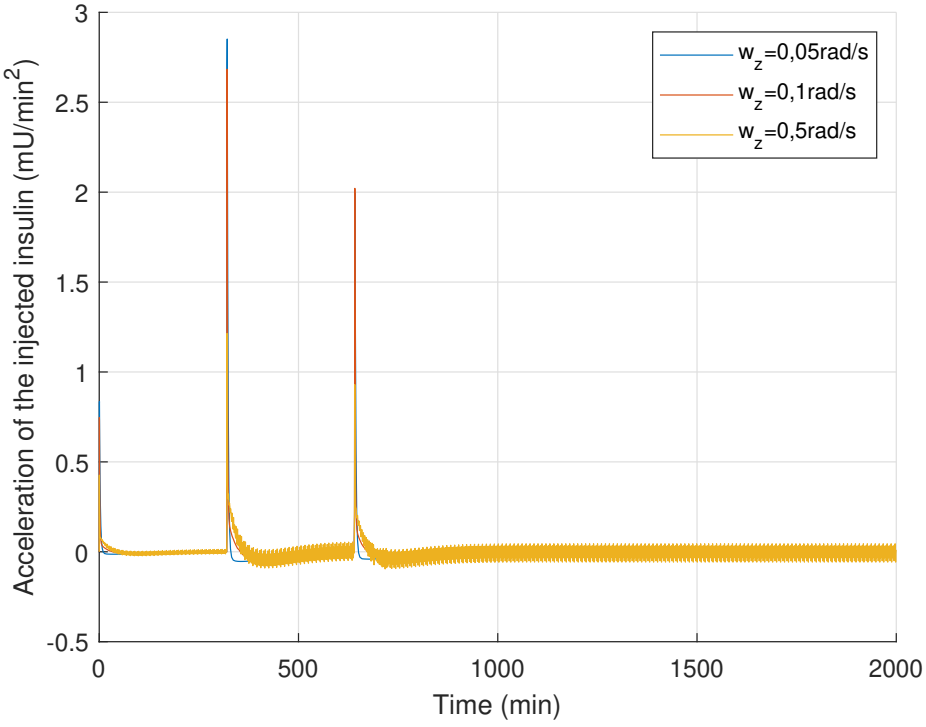

Figure 21 – Acceleration of the injected insulin for different frequencies ω_z

The best compromise seems to be a frequency $\omega_z = 0{,}1$ rad/s (and $\omega_p = 1$). This is because the acceleration of the injected insulin is not oscillating too much and because the maximum is not too high. But we still have to reduce it as the constraint has to be below 0.66 mU/min^2, next we will thus use a low pass filter to reduce the peak.

Finally the transfer function for the lead compensator is,

$$C_2(s) = \frac{1 + \frac{s}{0{,}1}}{1 + s}$$

(c) The low pass filter

Our third component is the low pass filter. It is used to attenuate high frequencies (at a cost of a lost in phase for high frequencies). Our goal is to filter out the noise and decrease as much as possible the oscillations in the acceleration of the injected insulin. For that we have to decrease the frequencies beyond our crossover frequency of 0,1 Hz (0,682 rad/s). Here is the transfer function of a low pass filter,

$$G_{HF}(s) = G_{HF_0} \frac{1}{\frac{s}{\omega_{hf}} + 1}$$

And below is the bode plot of $L(s)$ with different values of cut-off frequencies for the low pass filter.

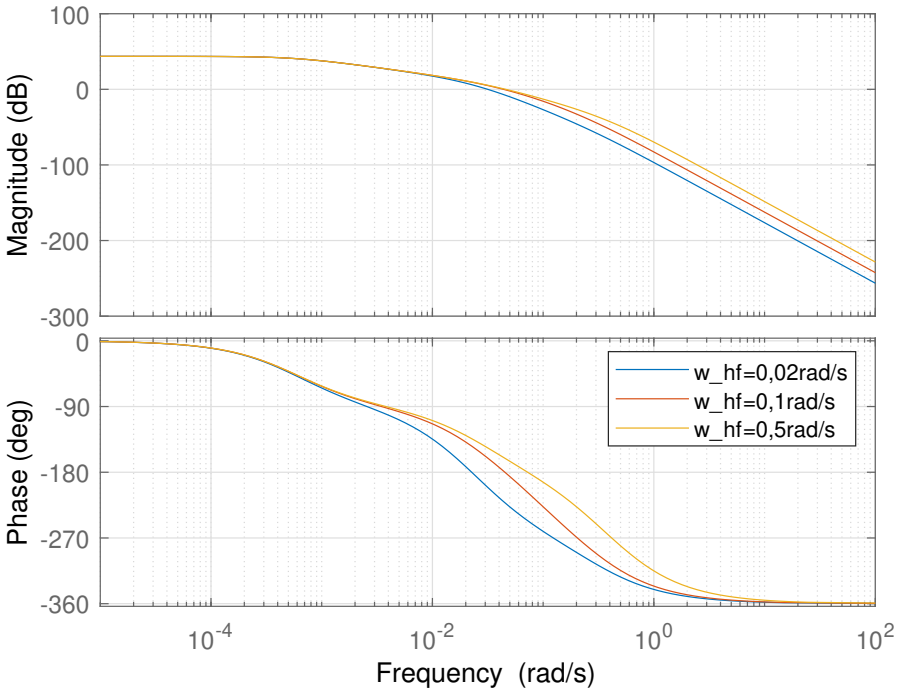

Figure 22 – Bode diagram of $L_(s)$ for different cut-off frequencies

Here is the simulation (with the meal input seen in fig 27),

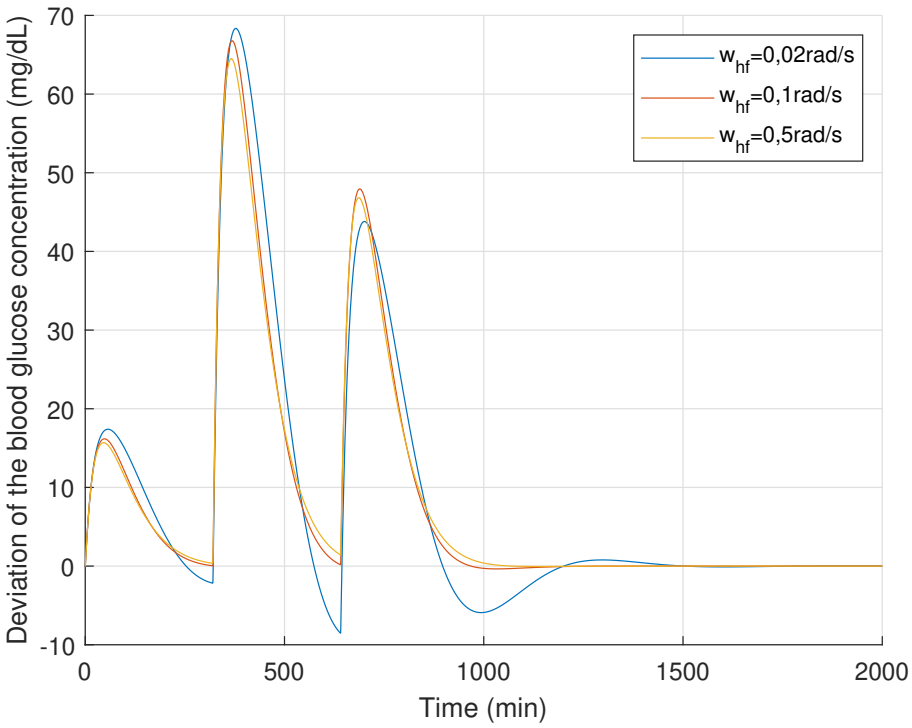

Figure 23 – Output of the system for different cut-off frequencies

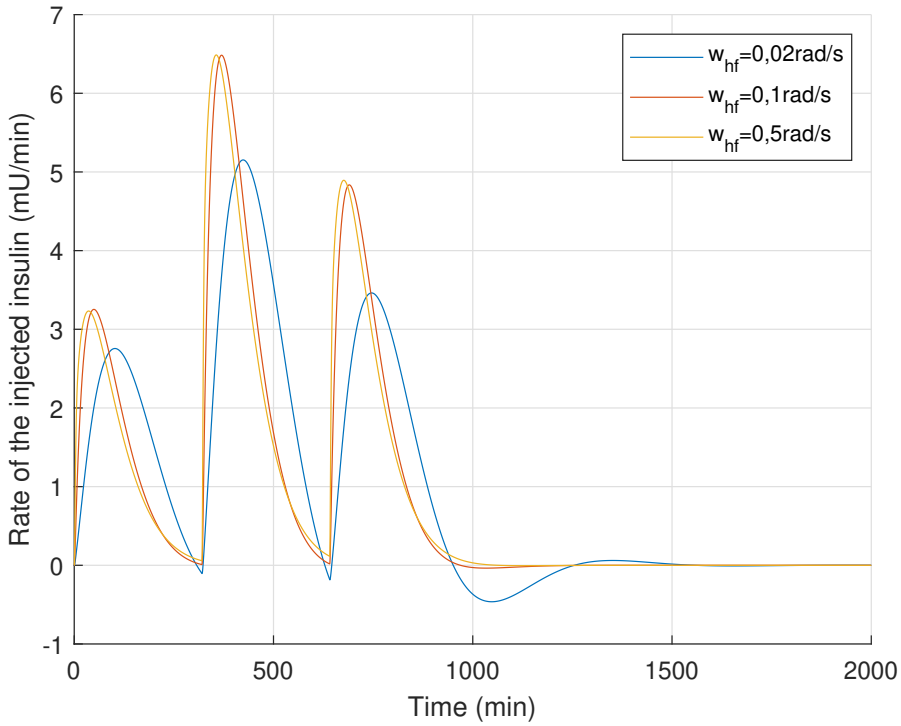

Figure 24 – Rate of the injected insulin for different cut-off frequencies

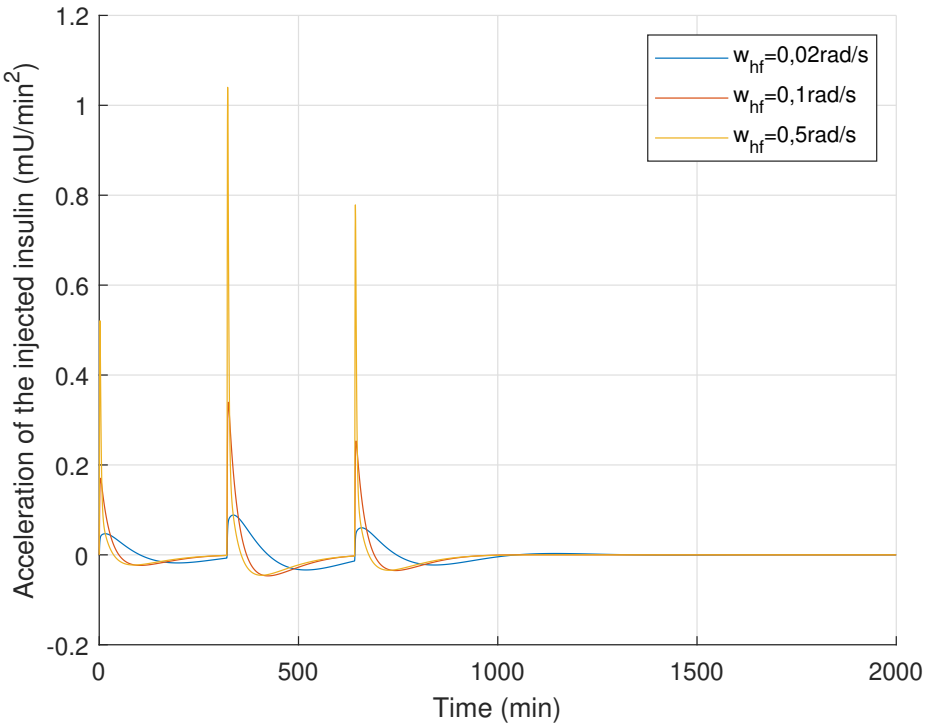

Figure 25 – Acceleration of the injected insulin for different cut-off frequencies

The cut-off frequency $w_{hf} = 0,1$ fits the most to our needs as the rate of injected insulin does not go under zero (and thus we avoid unnatural behaviour of the pump). We also could not pick a higher frequency as the peak in the acceleration would have been too high (above 0,66 mU/min^2).

In the end we ended up with the following low pass filter,

$$C_3(s) = \frac{1}{\frac{s}{0,1} + 1}$$

4.3 Recap and constraints analysis

Now that we have the main design of our controller we need to see if it is compatible with our constraints. This part will also be used as a recap of the entire system and will provide details about the different choices we made.

First we may look at the bode diagram of L(s) with our completed controller,

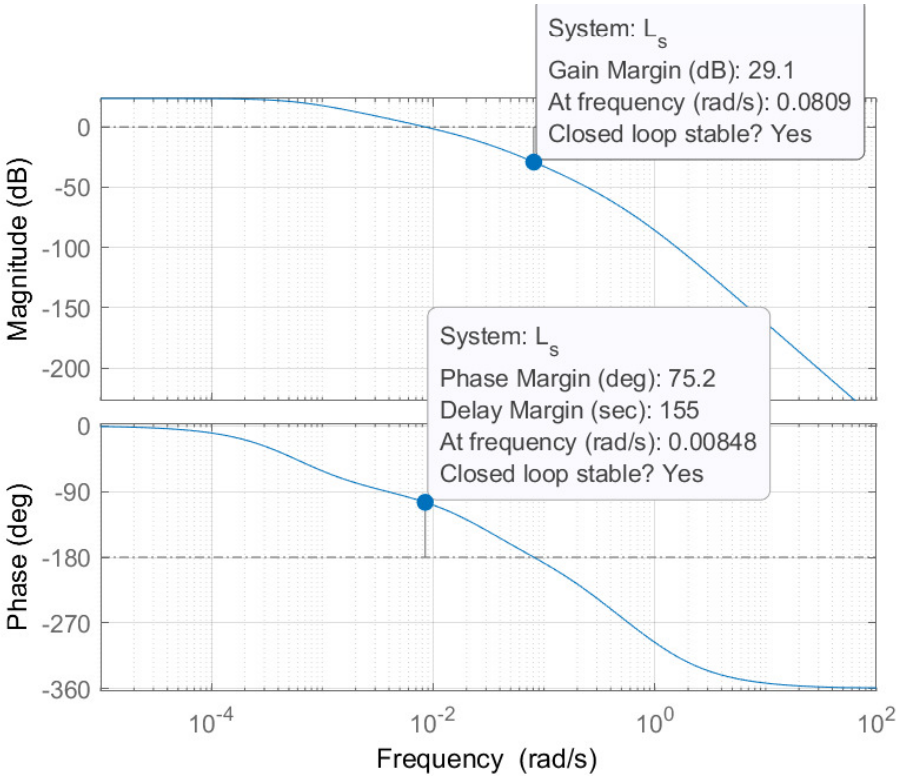

Figure 26 – Bode diagram of $L_(s)$

The bode diagram shows us that we achieved what we were hoping for: a low attenuation around the crossover frequency as well as a phase margin higher than 60°.

Now we look at the different values of blood glucose concentration, the rate and acceleration of the injected insulin (with the input seen in fig 27).

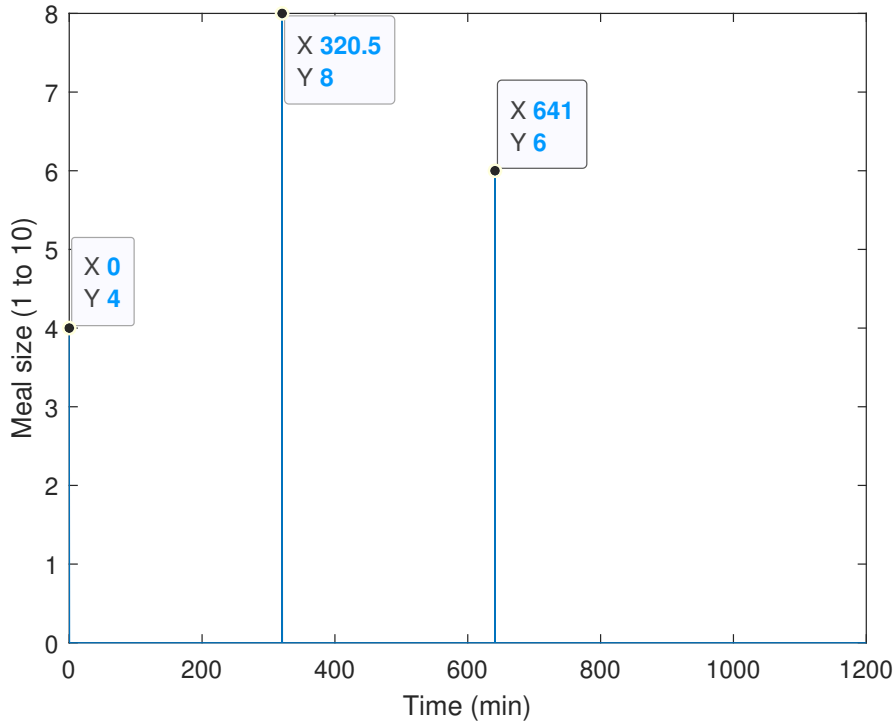

Figure 27 – Meal disturbance impulsion train

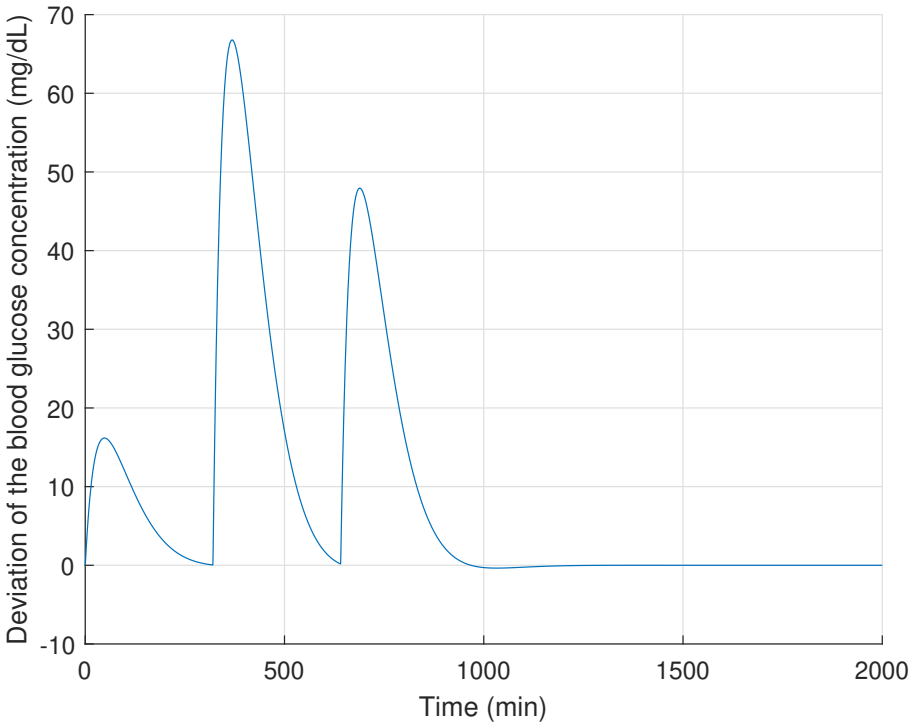

Figure 28 – Deviation of the blood glucose concentration of the controlled system

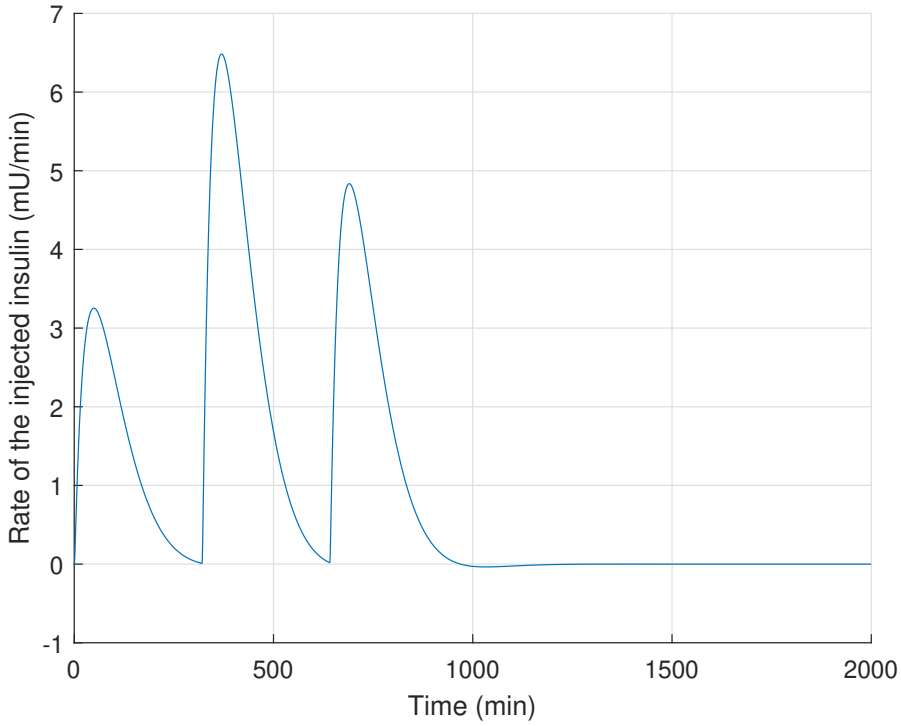

Figure 29 – Rate of the injected insulin of the controlled system

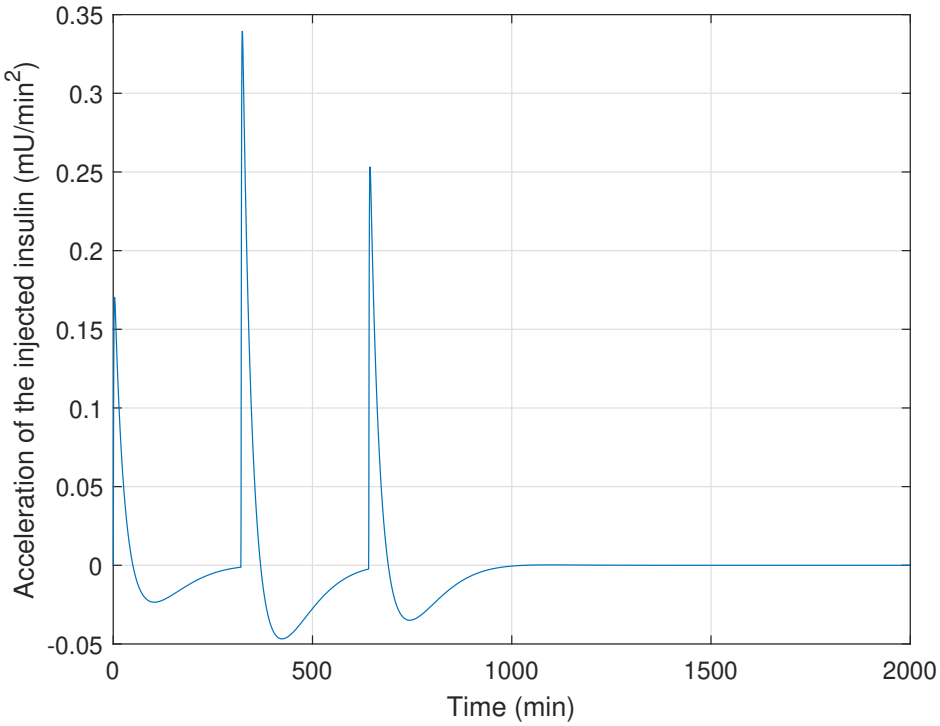

Figure 30 – Acceleration of the injected insulin of the controlled system

We can see that we have achieved our objective the blood glucose concentration remains between 60 mg/dL (hypoglycemia) and 270 mg/dL(hyperglycemia) (the deviation remains between -30 and 180 mg/dL). The pump is also functional as the *acceleration of the pump* is under 0,66 mU/min^2 (and the rate is largely below the maximum of 500mU/min). We also managed to avoid having negative values for the injection rate.

4.4 Gang of four

The best way to see the impact of the gang of four on our system is to write the equations involving their impact on y and μ.

$$y = T(s) \cdot r + PS(s) \cdot d + S(s) \cdot n$$

$$\mu = CS(s) \cdot r + T(s) \cdot d + CS(s) \cdot n$$

We will now derive from that the expected behavior of each of them.

(a) Sensitivity

The equation for the sensitivity function is given by

$$S(s) = \frac{1}{1 + L(s)}$$

where all the poles have a negative real part for our L(s) (to ensure its stability).

Here is its bode diagram :

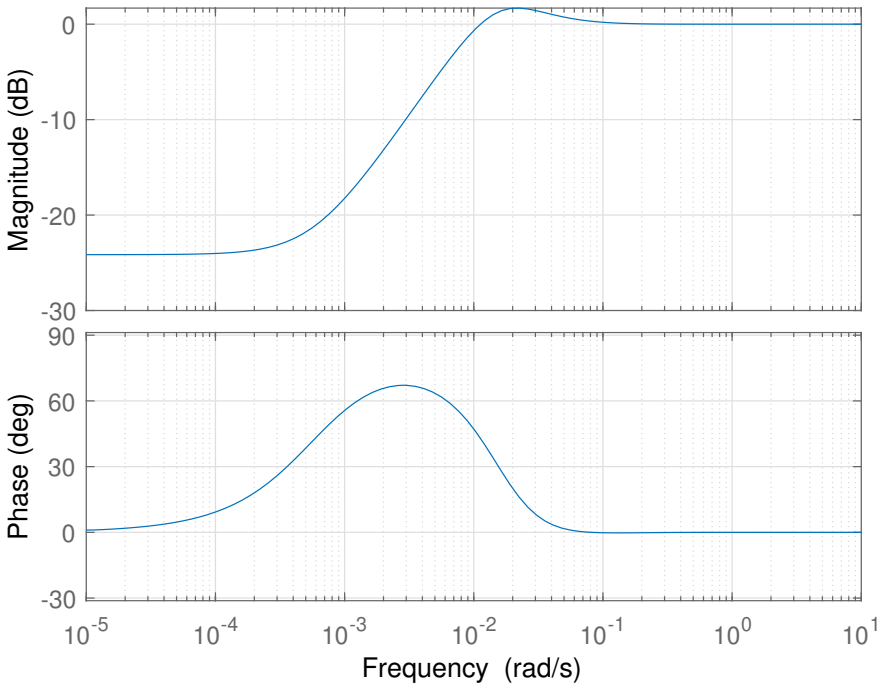

Figure 31 – Bode diagram of the sensitivity function S

We want 2 things from S(s). First we want it to be small for low frequencies in order to attenuate the disturbance in the output, then we want it to be equal to 1 (or 0 in dB) in the noise frequencies in order to not try to act on them. Both effects can be seen on our bode diagram above.

(b) Complementary sensitivity

The equation for the complementary sensitivity function is given by

$$T(s) = \frac{L(s)}{1 + L(s)}$$

where all the poles have a negative real part for our L(s) (to ensure its stability).

Here is its bode diagram :

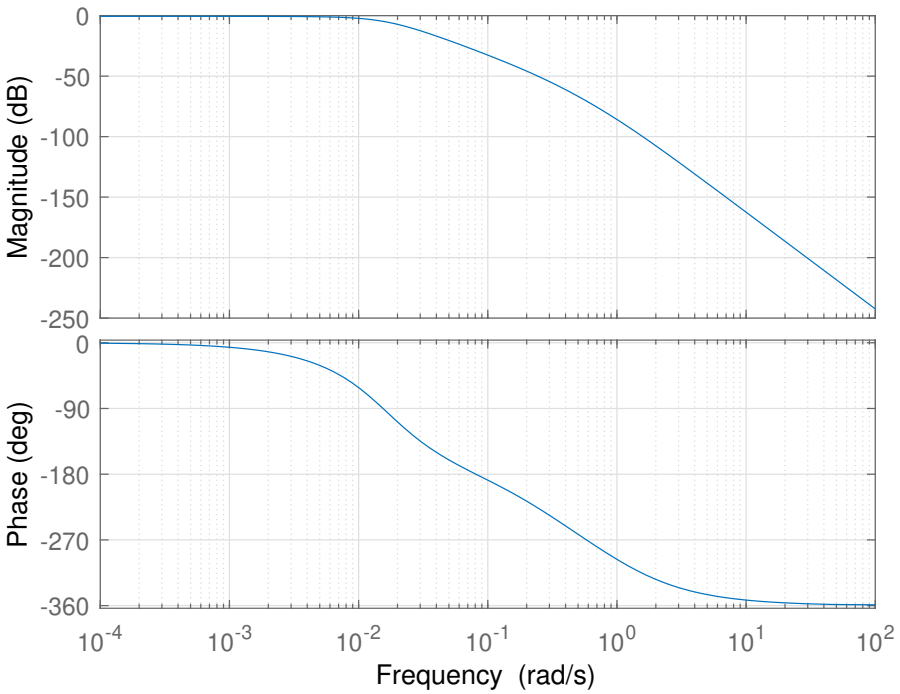

Figure 32 – Bode diagram of the complementary sensitivity function T

For this one, we want it to be close to 1 for low frequencies in order to have the disturbance impacting our input and to remove static error. This is the case in our T(s).

(c) Load sensitivity

The equation for the load sensitivity function is given by

$$PS(s) = \frac{P(s)}{1 + L(s)}$$

where all the poles have a negative real part for our L(s) (to ensure its stability).

Here is its bode diagram :

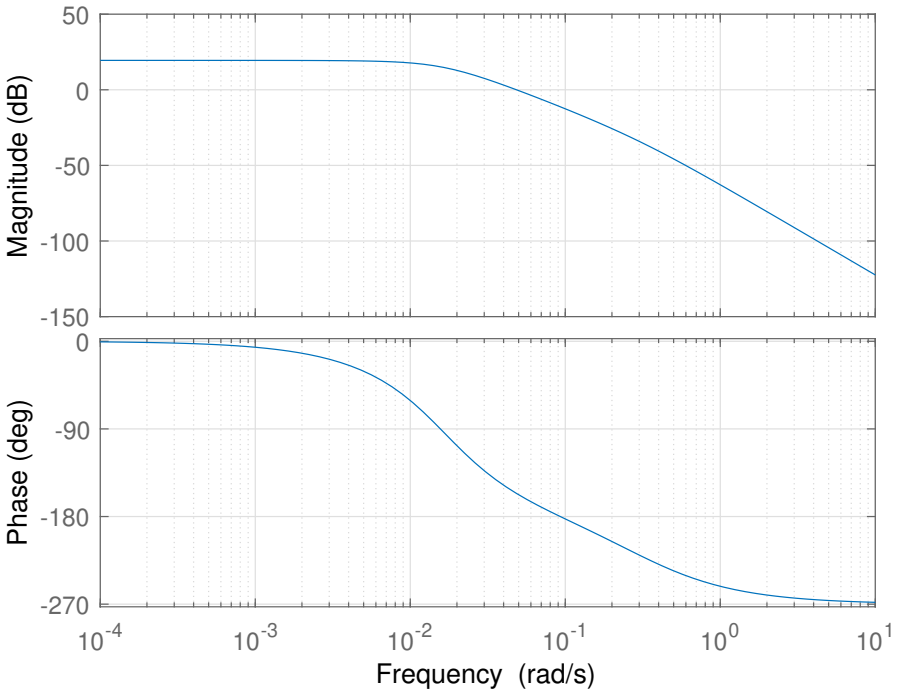

Figure 33 – Bode diagram of the load sensitivity function PS

Even though the S(s) is small for low frequencies, PS(s) seems to be high. This is probably due to the slow reactivity of our system. We attenuate it to acceptable glucose level, but peaks remain in the controlled system.

(d) Noise sensitivity

The equation for the noise sensitivity function is given by

$$CS(s) = \frac{C(s)}{1 + L(s)}$$

where all the poles have a negative real part for our L(s) (to ensure its stability).

Here is its bode diagram,

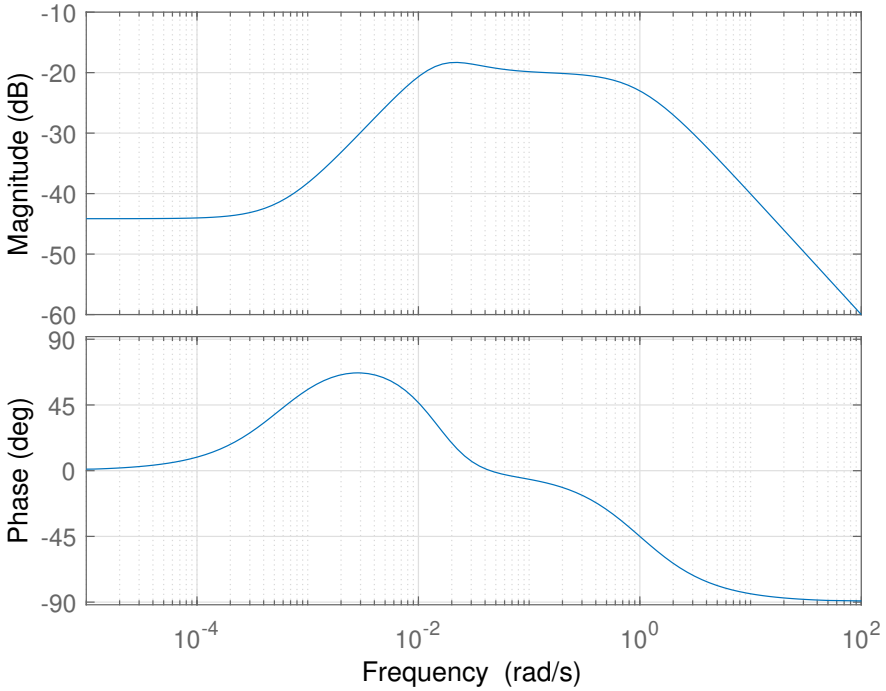

Figure 34 – Bode diagram of the noise sensitivity function CS

We want it to be small for high frequencies, in order to avoid our input to be impacted by the noise. As our system does not need continuous inputs to be stable around 90mg/dL (glucose in blood), we also want it to be small at 0. Both requirements are satisfied in our noise sensitivity.

4.5 Delays

The expression for the delay is given by

$$delay(s) = e^{-t_d s}$$

We chose different delays, added the component to our system and got the followings plots (Figures 35 and 36).

We can see that for the $t_d = 180$ min, the system becomes unstable as the pole $L(j\omega) = -1$ is encapsulated in the Nyquist plot. We can clearly see this effect on the output of the system as it is oscillating and constantly increasing. For $t_d = 60$ min the system is still stable but oscillating a lot more than before.

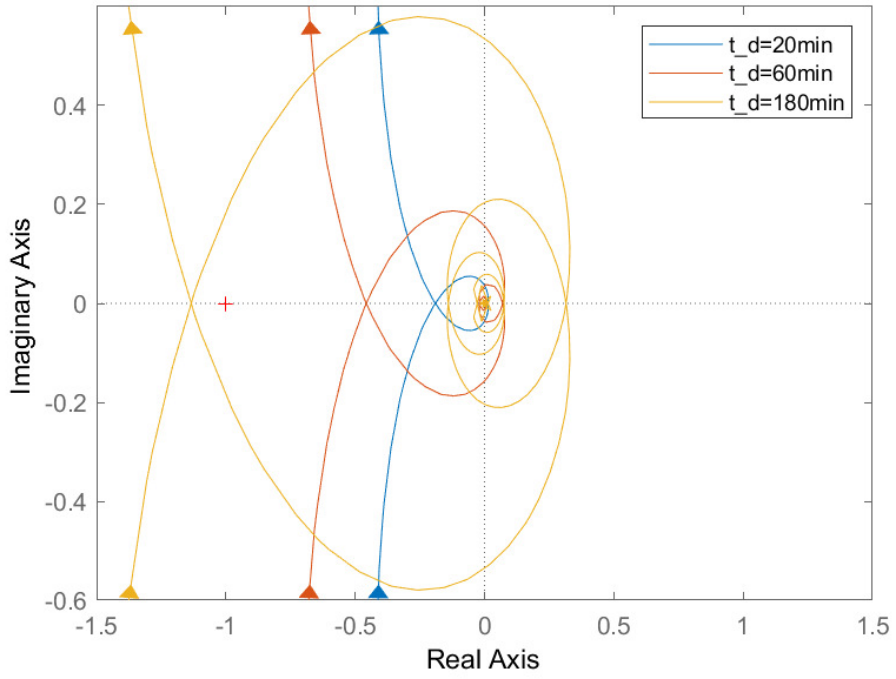

Figure 35 – Nyquist plot of the system with different delays

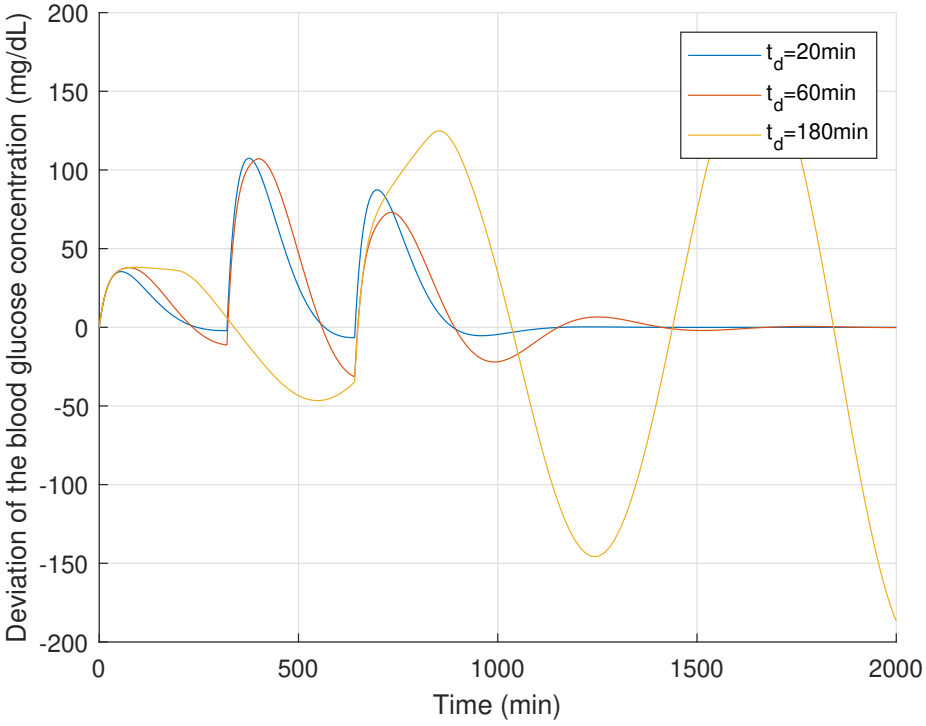

Figure 36 – Output of the system with different delays

4.6 Noise

To simulate noise we added a random uniformly chosen numbers ranging[6] from [-2, 2] to the output of our system.

We can see in the following simulations that our system remains stable but ends ups with very high oscillations especially in the acceleration of the pump. In order to correct this behaviour we would have to find a way to filter out the noise.

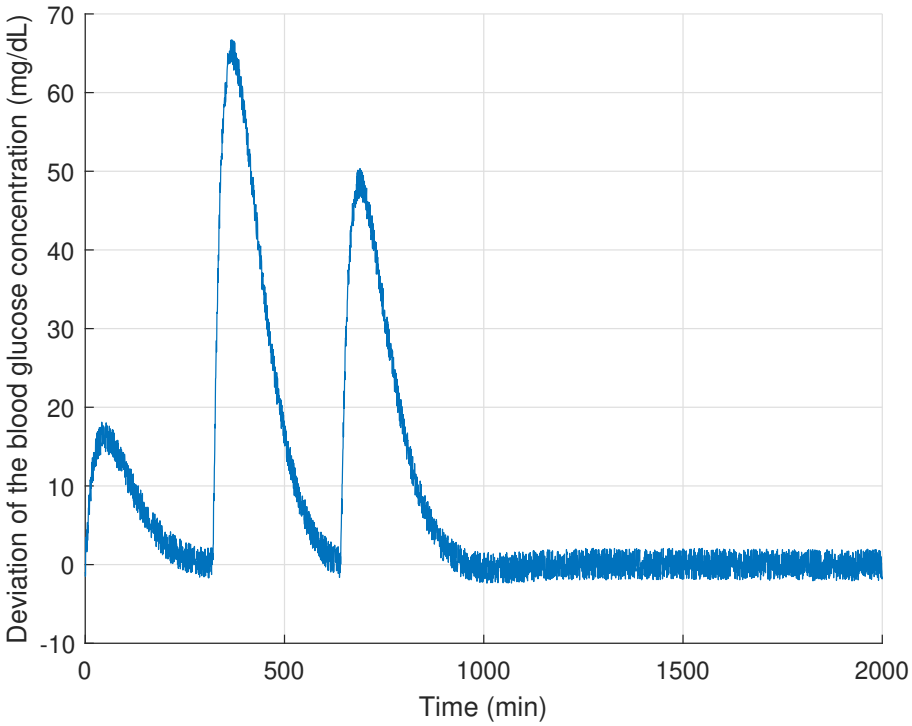

Figure 37 – Deviation of the blood glucose concentration of the system with noise

[6]The range is set to correspond to the precision of a glucose sensors working around 40 mg/dL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4455380/`

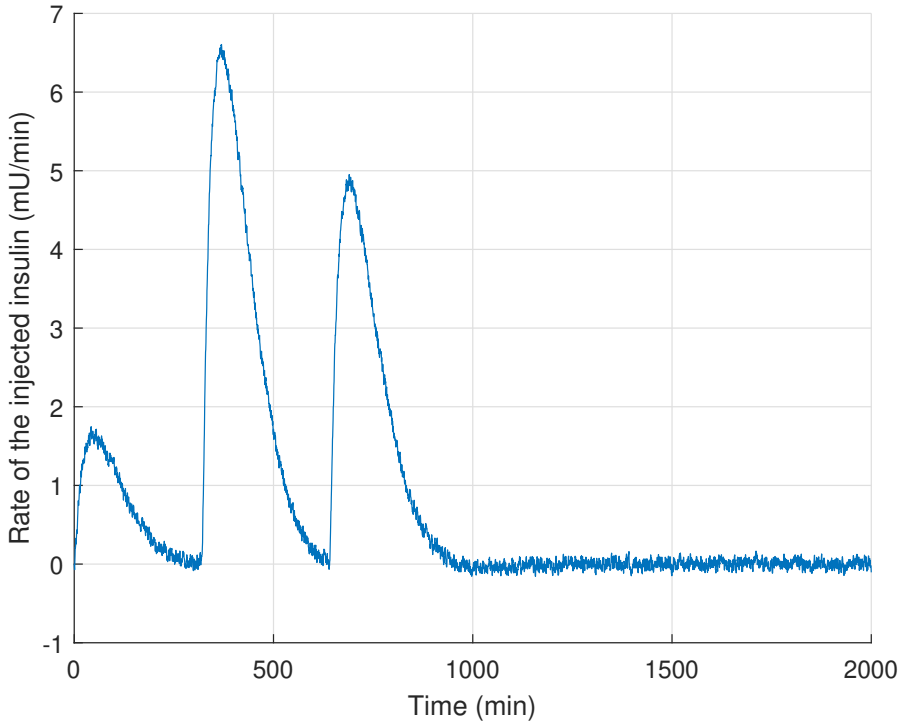

Figure 38 – Rate of the injected insulin of the system with noise

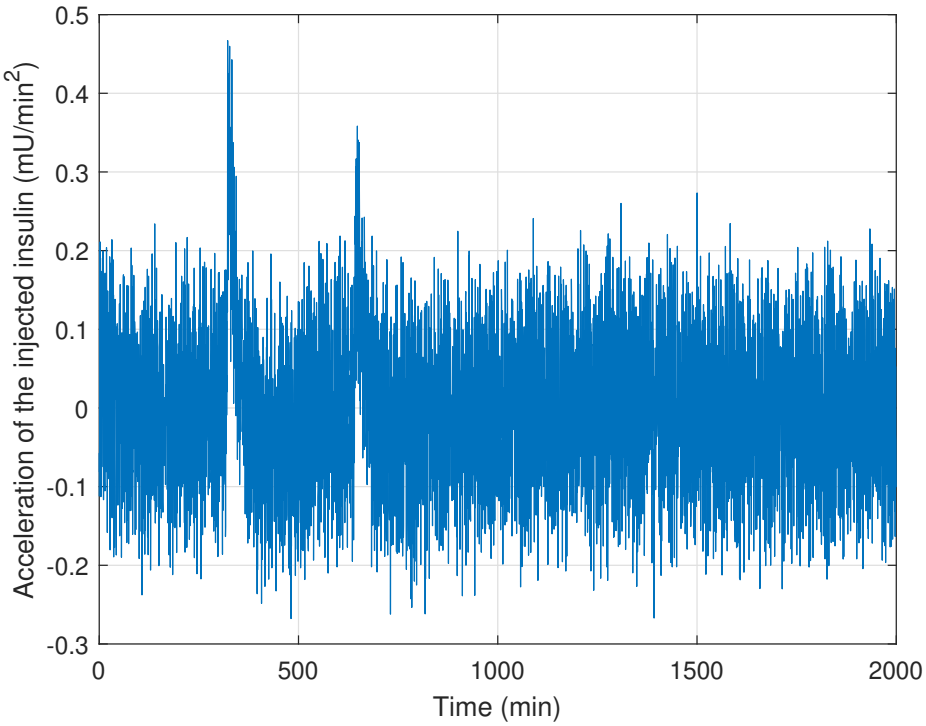

Figure 39 – Acceleration of the injected insulin of the system with noise

4.7 Feedforward discussion

Implementing a feedforward could be beneficial but seems infeasible in practise. Feedforward in the blood glucose for diabetic patient problem implies knowing that a person will take a meal before he actually does. This would allows to react quicker and deliver insulin before a rise in glucose could be sensed in the blood. The problem is that the only way to obtain that information would be to ask the user to indicate when he plans on eating. This would not only be annoying for the user as it would also be dangerous. If a user ends up by missing a meal he planned to eat, the insulin pump will preemptively inject insulin. This will induce a decrease of glucose concentration and lead to hypoglycemia as the glucose to counteract was never ingested. Moreover, the user would also have to know what will be the glucose contribution of the meal, which is not obvious in the everyday life.

5 Conclusion

5.1 Comparison between time domain and frequency domain

The time domain has the advantage of being more intuitive than the frequency domain. As we use the ABCD matrices for our system we easily get a sense of what is in relation to what. The observer for example is relatively easy to assemble as we know the different signals and how the matrix interacts with them.

On the other hand, the frequency domain is a bit more complicated to understand but it allows for a deeper tuning. It has lots of different ways to give us insights of what influences the system. The gang of four for example is a great tool for understanding the behaviour of our system and there is not quite something equivalent in the time domain. The gang of four gives us information about the noise sensibility and load disturbance, somethings that we need to simulate in the time domain in order to gain information.

A last point to highlight is the noise, which is much easier to deal with in the frequency domain as it is a frequency-oriented phenomenon. However, adding a filter after the sensor is always a possibility to solve the problem in time domain.

In the end both methods have the strengths and weaknesses and both yield significant results as we able to achieve our goals with both of them. Combining the two offers a great perspective on the system and enables accurate design decisions to be made.

5.2 General discussion

To conclude we can say that the design of this controller was a success. We managed to respect all of our constraints as well as having realistic values for all the different states of the system. Although the model would not be suited for real life applications as the model we chose is a large simplification of the real behaviour (few equations, constant assimilation rate of the meals and so on). It has allowed us to understand the fundamentals of controllable systems.

We think that if this technology was to be developed, it would greatly benefit people with diabetes. The major constraints at the moment are the elaboration of a realistic

enough model as well as the development of accurate sensors for the glucose that could work in real time. If these obstacles were to be overcome, we would surely see diabetic people using insulin pumps and thus push the humanity a step further into the field of transhumanism.

6 Appendix

6.1 Added info and figures

Graphs of the systems with maximum disturbance possible, 3 meals of size 12 taken at 2 hours apart from each other. We can see that the system is still performing inside our constraints but we feel like this in not a typical behaviour so we didn't judge necessary to include it inside the main sections.

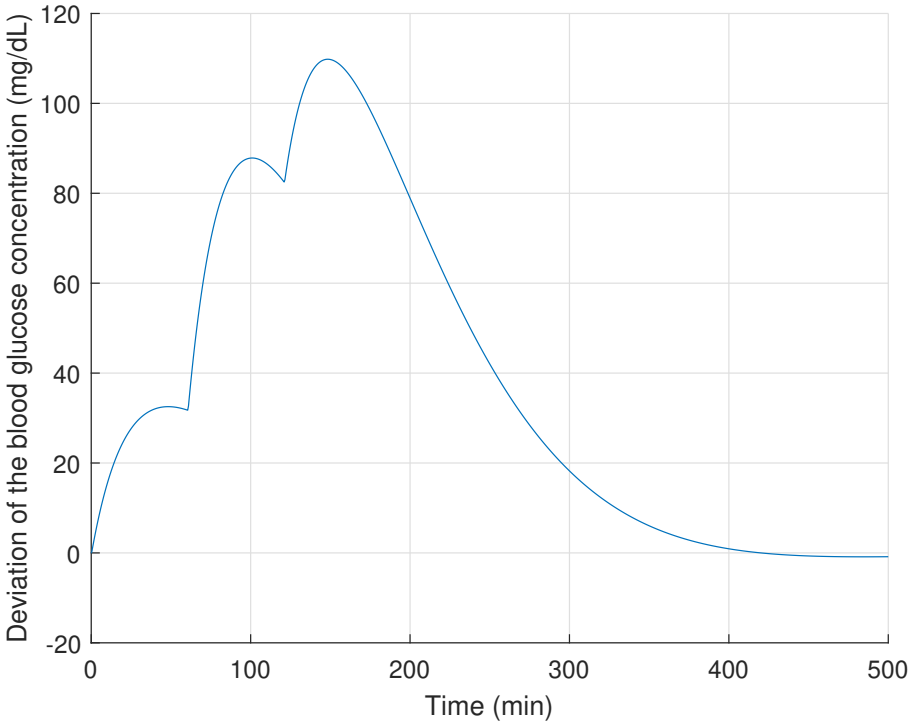

Figure 40 – Deviation of the blood glucose concentration of the controlled system

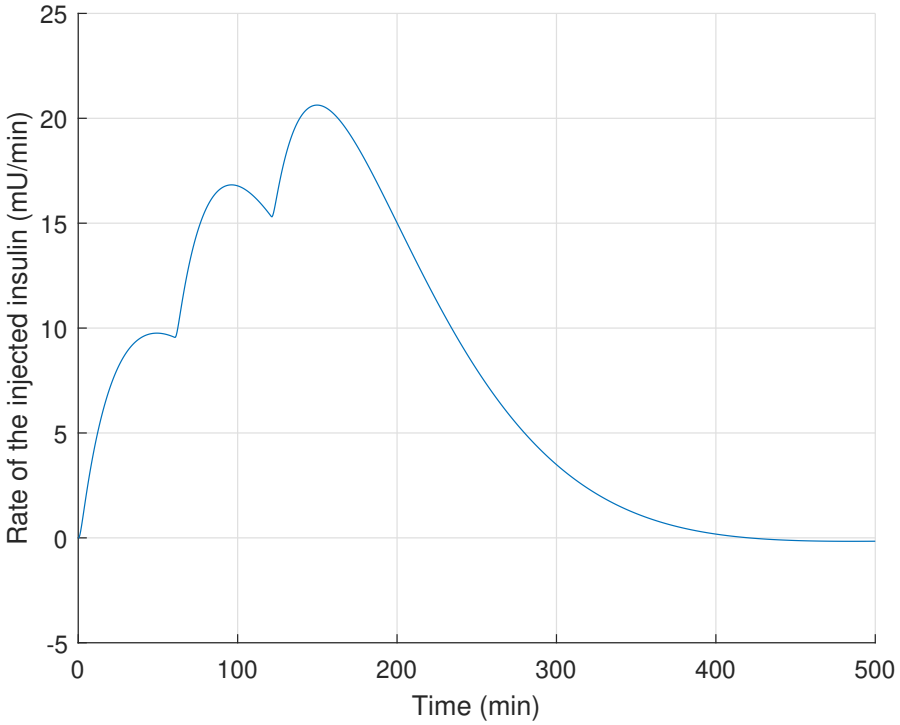

Figure 41 – Rate of the injected insulin of the controlled system

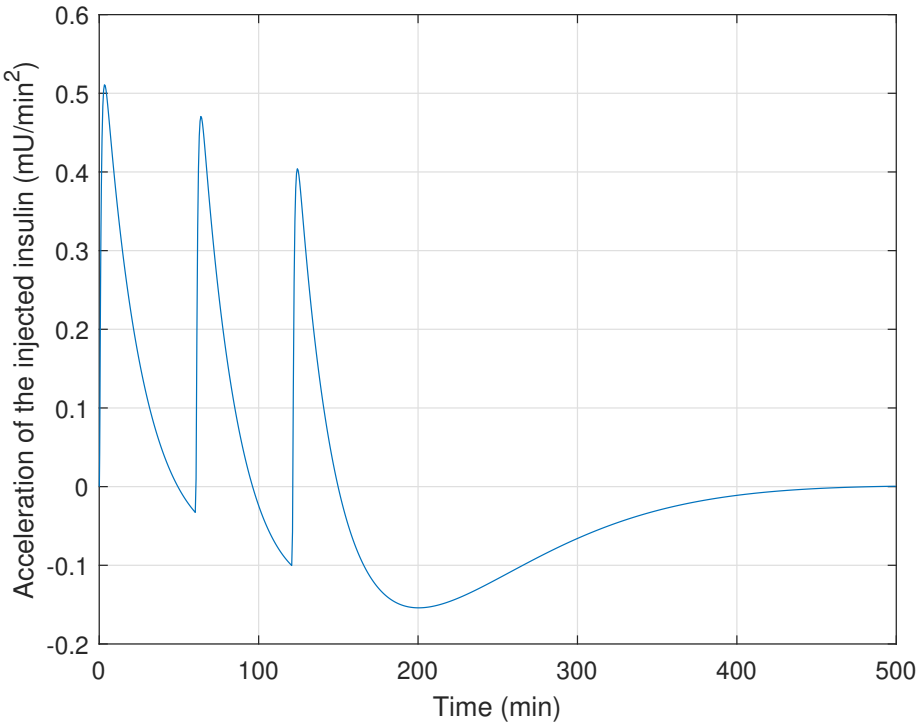

Figure 42 – Acceleration of the injected insulin of the controlled system

6.2 Bibliography

- Theoretical course given by Mr Drion (https://sites.google.com/site/gdrion25/teaching/syst0003)
- Practical session given by Ms Coutisse
- http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/5312/pdf/imm5312.pdf
- https://www.sps.nhs.uk/wp-content/uploads/2018/05/Insulin-pump-table-May-2018.pdf
- https://en.wikipedia.org/wiki/Hypoglycemia
- https://en.wikipedia.org/wiki/Hyperglycemia
- https://www.diabetes.co.uk/diabetes_care/blood-sugar-level-ranges.html
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4455380/
- http://www.cds.caltech.edu/~murray/amwiki/index.php/Main_Page
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4455380/