

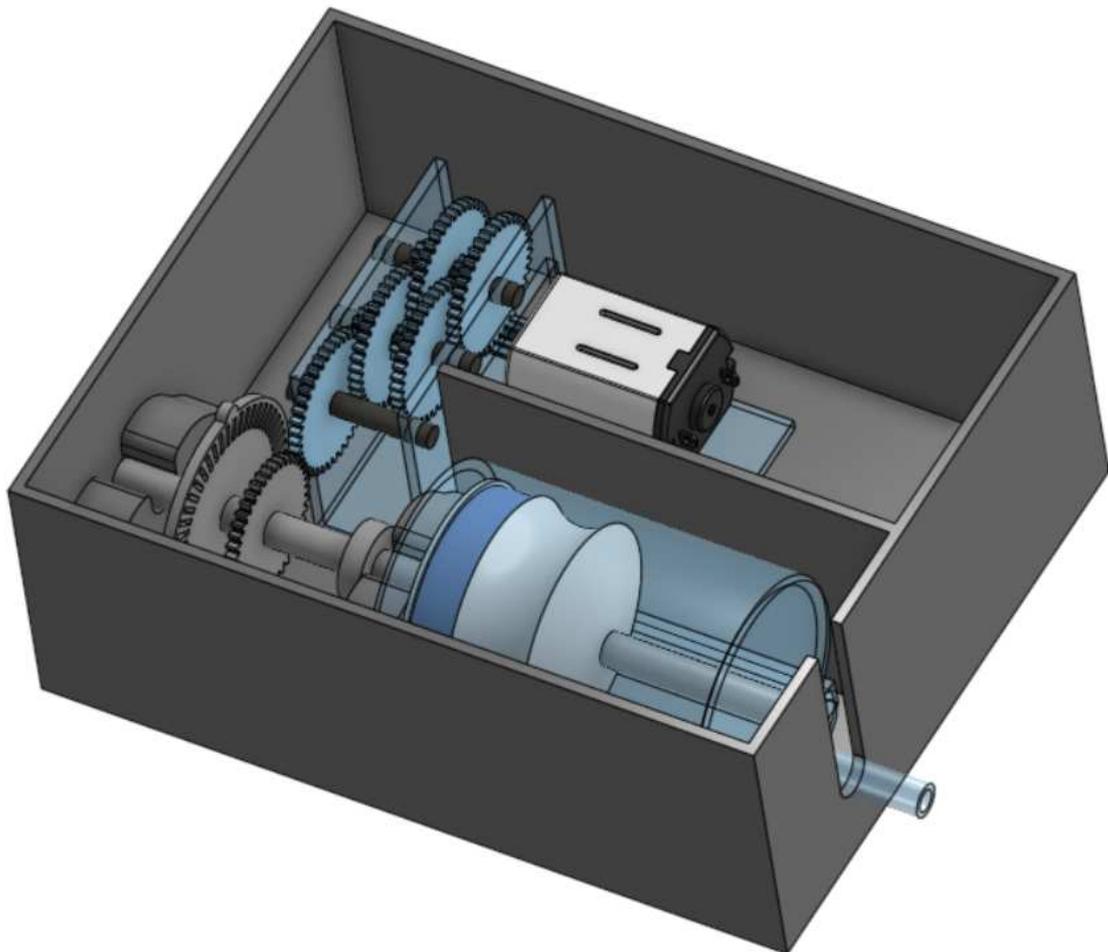
COFFIN Alice

FAHEM Amina

Février 2018

Rapport de projet de fin d'études

La pompe à insuline automatique



Introduction

Ce projet de fin d'études a pour but premier d'émuler le fonctionnement complet d'un pancréas, cela permettrait à des patients atteints de diabète de vivre sans se préoccuper de leur maladie. Le système se chargerait alors de réguler la glycémie afin de le maintenir à l'équilibre comme chez un individu sain (homéostasie).

Le corps humain est une formidable machine capable de s'autoréguler complètement, seulement dans certains cas, notamment celui du diabète, ce dernier ne parvient pas à s'équilibrer, ce qui peut causer des dommages irréversibles.

Afin de nous immerger dans ce projet pluridisciplinaire, il nous a fallu comprendre la complexité du corps humain et les mécanismes liés à la maladie en plus des connaissances techniques apprises et mises en œuvre lors de ce projet.

Pour réaliser cela, nous allons suivre les étapes de réalisation d'un projet. Nous devons commencer par réaliser le modèle de notre pompe, ensuite il faudra travailler sur la commande puis pour finir s'intéresser au modèle réel, le prototype ainsi que la partie hardware.

Dans ce rapport, nous ferons une présentation générale du projet, son contexte, l'état de l'art et le cahier des charges. Nous reprendrons ensuite le modèle analytique de la pompe réalisé ainsi que sa commande. Nous allons ensuite exposer les systèmes mis en place pour émuler la glycémie de manière à pouvoir tester notre modèle dans la partie qui suit et d'aboutir à la validation de ce dernier. Nous allons ensuite terminer ce rapport en expliquant le prototype réalisé et ses perspectives d'amélioration.

Sommaire

Introduction.....	2
Sommaire	3
1. Préface du projet.....	4
1.1 Le contexte	4
1.2 Etat de l'art	6
1.3 Cahier des charges.....	9
2. La pompe avec Matlab/Simulink.....	11
2.1 Modèle analytique.....	11
2.2 Test boucle ouverte	17
2.3 Commande du débit	19
2.4 Commande de la quantité injectée	20
3. Emulation de la glycémie	23
3.1 Comment émuler la glycémie ?	23
3.2 Sous Simulink.....	24
3.3 Sous Arduino.....	27
4. Modèle complet avec Matlab/Simulink.....	29
4.1 Calcul des doses à injecter.....	29
4.2 Validation du modèle total.....	32
5. Prototype.....	35
5.1 Le modèle 3D.....	35
5.2 Mesures et capteurs	36
6. Les problèmes rencontrés.....	38
Conclusion	39
Bibliographie	40

1. Préface du projet

1.1 Le contexte

Le diabète :

Le diabète est une maladie se traduisant par un taux de sucre (glucide) trop élevé dans le sang. Ce taux peut être causé par deux problèmes, soit le pancréas ne sécrète pas assez d'insuline (l'hormone qui régule le glucose dans le sang) soit l'insuline est mal utilisée par l'organisme. C'est pour cela qu'il existe deux types de diabète.

Le diabète de type 1, aussi appelé diabète insulino-dépendant, est une maladie auto-immune sans cause connue. Dans ce cas, les anticorps du malade attaquent et détruisent les cellules du pancréas fabriquant l'insuline. Lorsque 90% des cellules sont détruites, le patient est atteint de diabète.

Le diabète de type 2 représente 90% des diabètes et est dû en grande partie à une surcharge pondérale et à la sédentarité. Ce diabète apparaît en deux temps, tout d'abord le pancréas fabrique toujours de l'insuline mais un phénomène d'insulino-résistance apparaît, le corps devient résistant à l'insuline, pour palier à ce problème le pancréas fabrique de plus en plus d'insuline pour maintenir le taux de glucose à un niveau normal. A cause de sa sur-utilisation, le pancréas se fatigue et produit de moins en moins d'insuline, c'est l'insulino-déficience.

Selon l'Organisation Mondiale de la Santé, le nombre des personnes atteintes de diabète dans le monde est passé de 108 millions en 1980 à 422 millions en 2014 soit une multiplication par quatre du nombre de malade en 35 ans.

D'après l'Institut de Veille Sanitaire, en 2015, 3.3 millions de personnes étaient atteintes de diabète en France, soit 5% de la population.

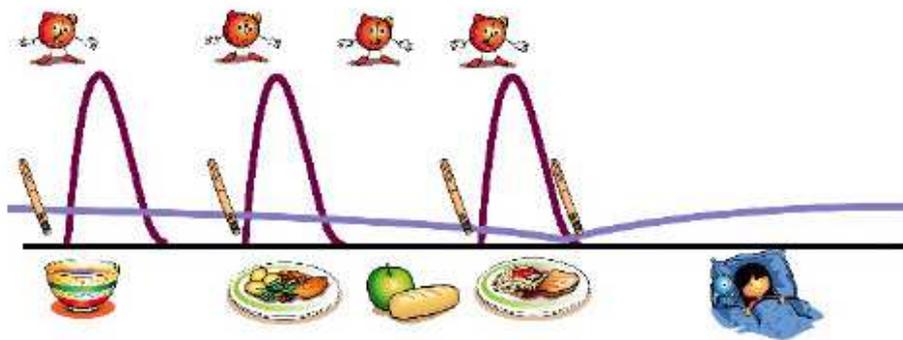
Le traitement du diabète :

Il y a différents traitements en fonction du type et de l'avancement du diabète. Durant la première phase du diabète de type 2, il existe des médicaments sous forme de pilules à avaler au cours des repas, ce sont des médicaments qui aident l'insuline à mieux fonctionner et donc à contrôler le taux de glycémie.

Dans les autres cas, on utilise l'insulinothérapie. On injecte de l'insuline synthétique dans le sang à différents moments de la journée pour conserver un taux de glycémie à un niveau normal. Il y a différentes façons de s'injecter de l'insuline : la seringue, le stylo, le pistolet et la pompe à insuline.

Pour une personne non diabétique, son taux d'insuline varie au cours de la journée en fonction de la quantité de sucre dans le corps. Il s'élève au moment des repas pour compenser les apports en glucide le temps de la digestion et est constant le reste du temps.

Pour une personne malade, cette régulation ne se fait plus, il faut donc s'injecter de l'insuline pour le compenser. Il existe deux types d'insuline : l'insuline lente qui commence à faire effet 1h après l'injection et pendant 24h et l'insuline rapide qui fait effet 15min après l'injection et pendant 3h. Lorsqu'ils n'utilisent pas de pompe à insuline, les diabétiques s'injectent une certaine quantité d'insuline lente une fois par jour (injection basale) et de l'insuline rapide avant chaque repas (injection bolus). Cela permet de recréer les variations classiques d'insuline dans le corps.



Lorsqu'il utilise une pompe à insuline, le diabétique n'utilise que de l'insuline rapide qui s'injecte dans le corps tout au long de la journée à partir d'une courbe entrée en paramètre. La pompe permet de se piquer que tous les 2/3 jours (lors de l'installation du cathéter). Mais quel que soit le type d'injection, une personne diabétique doit vérifier son taux de glycémie 3 à 4 fois par jour pour corriger les quantités d'insuline à injecter car il n'y a pas de schéma fixe, en fonction de la nourriture et de l'activité de la journée les injections sont différentes d'un jour à l'autre. Ces vérifications et ajustements en continu sont très contraignant pour le patient.

Les pompes à insuline :

Pour libérer les diabétiques de la contrainte de plusieurs piqûres quotidiennes, la pompe à insuline a été créée. Le système se porte à la ceinture et est relié au corps par un cathéter installé lors de la recharge de la pompe (tous les 2 ou 3 jours en général).

Cette pompe injecte au cours de la journée toutes les doses nécessaires en insuline rapide (un débit basal tout au long de la journée et du bolus pour corriger la glycémie au moment des repas). Le réglage se fait par le patient tous les jours en fonction des doses calculées avec son médecin et avec des corrections en fonction de ses prédictions sur la journée à venir (s'il va faire du sport, il faut diminuer les doses habituelles pour ne pas tomber en hypoglycémie).

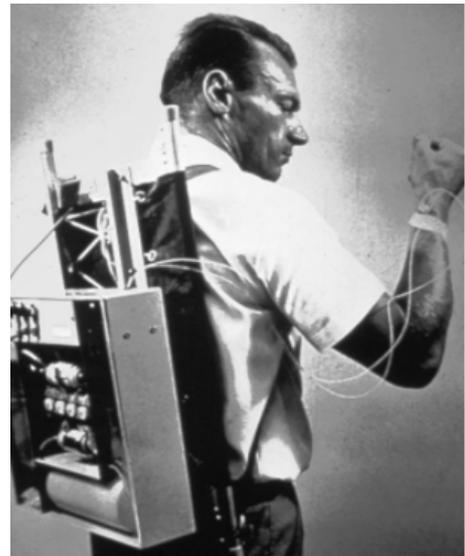
Le patient est libéré de certaines contraintes mais il doit toujours surveiller sa glycémie plusieurs fois par jour car le système est en boucle ouverte, il n'y a pas de retour du taux de la glycémie sur la pompe. Un système totalement autonome serait très utile pour les malades.

1.2 Etat de l'art

Historique

Dans les années 1960, la première pompe à insuline a été réalisée par le Docteur Arnold Kadish en Californie. Elle se portait comme un sac à dos. Quelques années plus tard, les premiers capteurs à insuline sont mis sur le marché par les laboratoires MILES ensuite racheter par Bayer (1969).

Parallèlement, la première transplantation de pancréas a eu lieu à l'Université du Minnesota (1966). Les docteurs ont alors essayé de guérir le diabète par transplantation. Aujourd'hui, la greffe de pancréas est une opération réalisée partout dans le monde.



En 1978, Baxter lance la première commercialisation de la pompe à insuline alors appelée AutoSyringe (inventée par Dean Kamen). Ce fut la première pompe programmable utilisable à domicile par les patients.

En 2004, la première pompe programmable sans fil, l'OmniPod est commercialisé.

Aujourd'hui

De nombreuses pompes à insuline existent sur le marché. Seulement, ce dispositif médical reste peu répandu du fait de son prix très élevé (3000 euros minimum).

Les principales entreprises réalisant ce genre de dispositif sont Medtronic, Roche, Novalab. Les derniers modèles proposent des moyens de surveillance et d'acquisition de données en temps réel.



La pompe à insuline Animas (Novalab) a un système de SGC (Surveillance du Glucose en Continu) intégré. Ce système permet de connaître la quantité de bolus encore actif dans l'organisme, cela permet de prendre de meilleure décision quant à l'administration de l'insuline. Il existe également des pompes connectées au capteur de glycémie afin de contrôler les taux du patient.

D'autres sociétés ont fait le pari de la discrétion en essayant de minimiser les dimensions de la pompe de manière à ce qu'elle soit discrète pour le patient qui la porte.



Ce fut le cas pour OmniPod qui s'est engagé à créer une pompe discrète tout en assurant sa robustesse et ses performances avec des pompes à changer tous les 3 jours. Cette pompe est connectée à un PDM (Personal Diabetes Manager) sans fil qui gère l'administration de

l'insuline, récupère les données du glucomètre intégré, calcule l'insuline active dans l'organisme et peut également suggérer au patient une dose de bolus à injecter.

Cellnovo a souhaité réaliser une pompe à insuline discrète mais qui serait en plus rechargeable de manière à avoir un impact le plus minime possible sur l'environnement.

Problématique de ces systèmes

Ces systèmes sont uniquement commandés en boucle ouverte : il n'y a aucun contrôle de la glycémie par la pompe. C'est pour cela que le médecin est obligé de réaliser un contrôle fréquent de l'utilisation de la pompe et de donner des directives aux patients quant à la prescription et l'administration de l'insuline. Malgré le fait que la pompe est connectée au capteur, il n'influe en rien sur l'administration du médicament mais permet uniquement de réaliser un suivi. Pour que le calcul de la dose soit interactif de manière à réaliser un contrôle en boucle fermée, il faut faire une validation clinique. C'est l'objet de certaines recherches énoncées par la suite.

Thématique de recherche

Le projet de pancréas artificiel pour lutter contre le diabète a commencé en 2011 avec un groupe de diabétologue. Ce projet est également appelé « Insulinothérapie en boucle fermée », il consiste à faire communiquer le capteur de glycémie et la pompe à insuline grâce à une intelligence artificielle.

Le dispositif est alors composé de 3 parties : une partie acquisition en continu de la glycémie, une seconde partie qui calcule les doses à injecter et qui donne des ordres à la troisième partie, une pompe à insuline connectée.

Ce projet a débuté en 2011 et des études cliniques ont été réalisées en 2014 sur une trentaine de patient. Le concept avait été alors validé. La compétition est maintenant rude après le lancement de plusieurs études sur patient dans les conditions de vie habituelle. En Amérique (Université de Virginie), en Angleterre (Cambridge) ou encore en France les premiers tests en temps réel sont lancés. Cependant, aucun des systèmes n'est prêt à être commercialisé.

Diabeloop espère commercialiser son premier pancréas artificiel en 2018.

Parallèlement, ce projet intéresse aussi les écoles d'ingénieur telles que Sup'biotech où un groupe d'élèves a imaginé un système de type implant. Ce concept a été présenté au concours Coup2boost (Dassault Système) qu'ils ont gagné. Le groupe a également été nommé pour le Concours des Jeunes Ingénieurs organisé par l'Usine Nouvelle.

1.3 Cahier des charges

Le travail que nous avons à effectuer durant le projet est le suivant : nous devons concevoir une pompe à insuline totalement automatisée afin d'émuler le fonctionnement complet du pancréas de manière à réguler le taux de sucre dans le sang (glycémie). Pour ce faire, il nous a été demandé différentes choses.

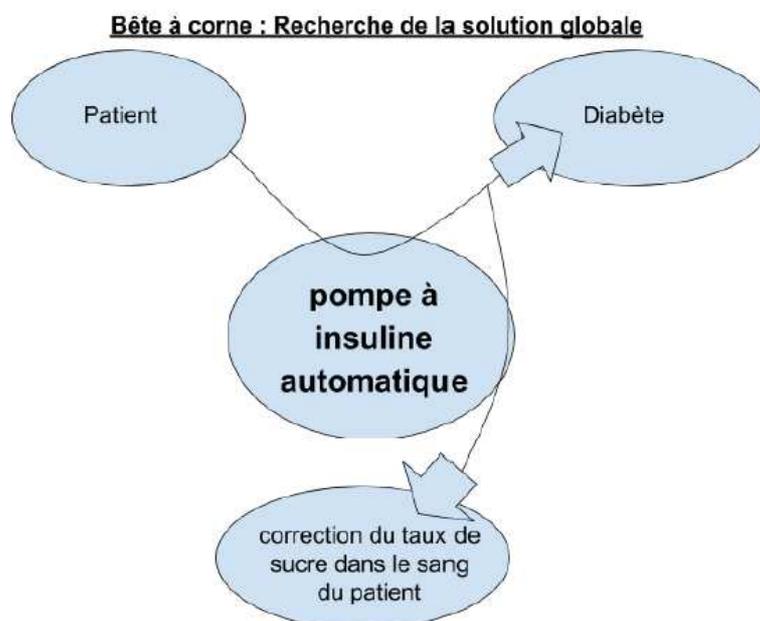
Tout d'abord, il faut modéliser le système mécatronique de la pompe en utilisant Matlab/Simulink. Il est également demandé de réaliser une commande de manière à ne pas avoir de dépassement de la consigne tout en gardant un temps de réponse correcte.

Nous devons également gérer le débit basal d'insuline qui est injecté tout au long de la journée. Il faudra également réaliser un modèle de calcul qui permettra de calculer grâce à la glycémie les quantités d'insuline à injecter.

En parallèle, nous allons devoir réaliser un prototype de la pompe pour pouvoir, au final, tester la même commande qu'en simulation et voir si le comportement est semblable. De plus, il est nécessaire d'assurer la robustesse et la fiabilité du système en mettant en place des dispositifs de sécurité sur le prototype.

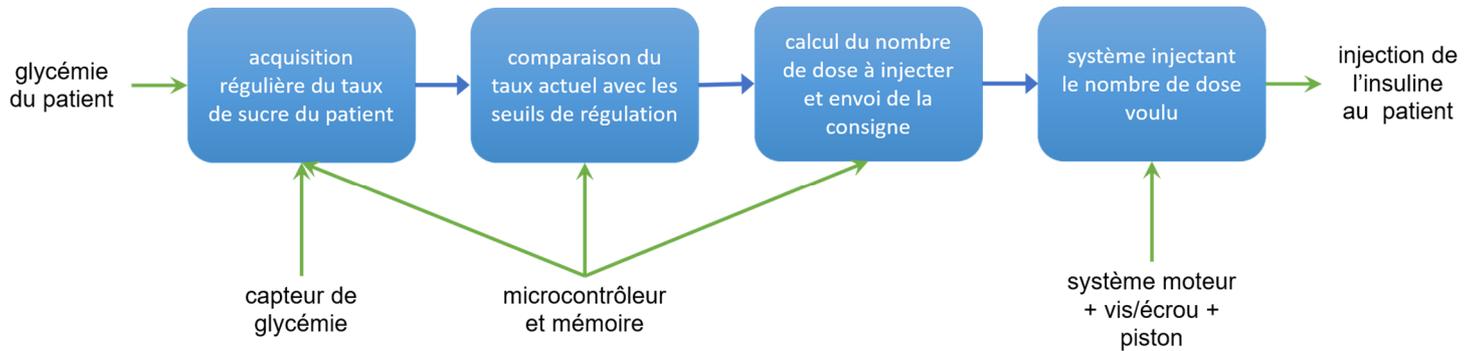
Pour ces tests, le taux de glycémie sera émulé car nous ne testerons notre pompe sur personne, il est donc inutile de rechercher un capteur de glycémie pour le début de ce projet qui sera un projet long, il est donc nécessaire de réaliser une émulation de cette glycémie de manière à valider le modèle, il faut également réaliser un dispositif dans lequel on implantera l'émulation afin de réaliser des tests temps réel sur le prototype.

Une pompe à insuline automatique permettrait de corriger automatiquement la glycémie d'un patient atteint de diabète :



Pour commencer ce projet, il faut comprendre ce qu'est le diabète et comment le traiter. A partir de là, il est possible d'avoir une vue générale du système que nous allons réaliser.

Il avait été décidé lors d'un stage sur ce projet que le système serait à vis sans fin avec le déplacement du piston réalisé par un écrou sur cette vis. Le système sera donc le suivant :

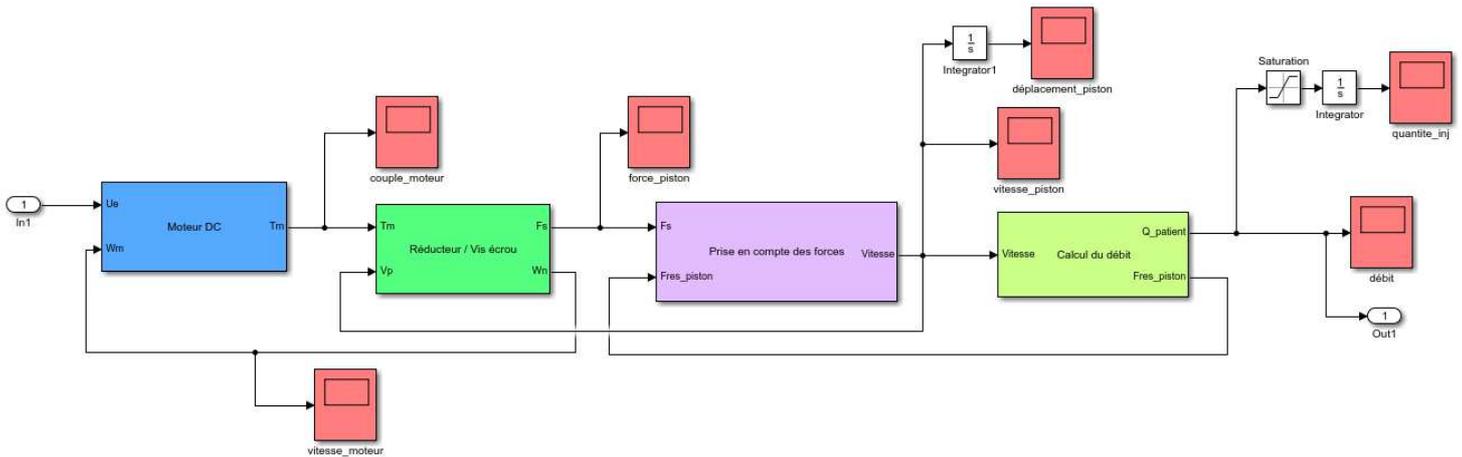


Comme il n'existe pas de loi mathématique concernant le taux de glucide dans le sang, la régulation ne sera pas simple à réaliser. La quantité de sucre ingérée et son temps de dispersion dépend des aliments digérés.

2. La pompe avec Matlab/Simulink

2.1 Modèle analytique

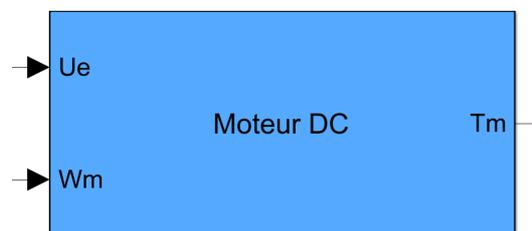
Le modèle analytique permet de simuler le fonctionnement de la pompe. On intègre tous les phénomènes physiques qui entreraient en compte dans un système réel (frottement, inertie, ...).



En entrée du système se trouve la tension d'alimentation du moteur, en sortie on trouve le débit d'insuline. A partir du débit, il est ensuite possible de calculer la quantité d'insuline injectée. Il suffit d'intégrer le débit.

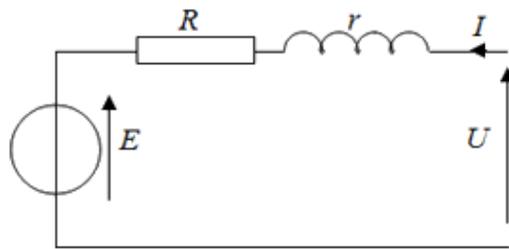
Le modèle se compose de quatre sous parties qui sont détaillées ci-dessous.

Le moteur :



Un moteur est un générateur de couple commandé en tension.

On le schématise par :



Les équations du moteur sont les suivantes :

$$E = U + R * I + L * p I$$

Donc

$$\frac{I}{U - E} = \frac{1}{R + Lp}$$

La force électromotrice est donnée par

$$\omega_0 * K_e = E$$

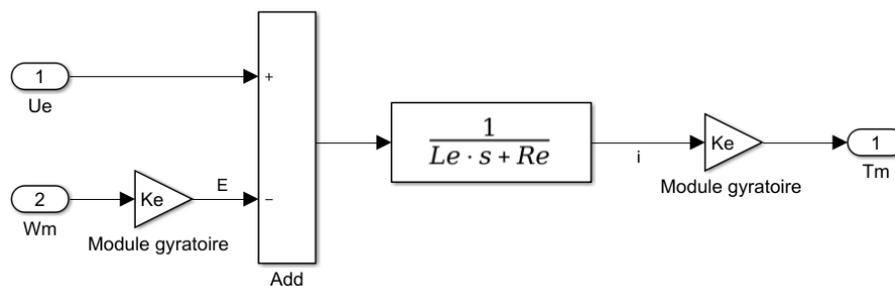
Et on sait que la puissance électromagnétique s'écrit

$$P_{em} = E * I = \omega_0 * T_m$$

Donc

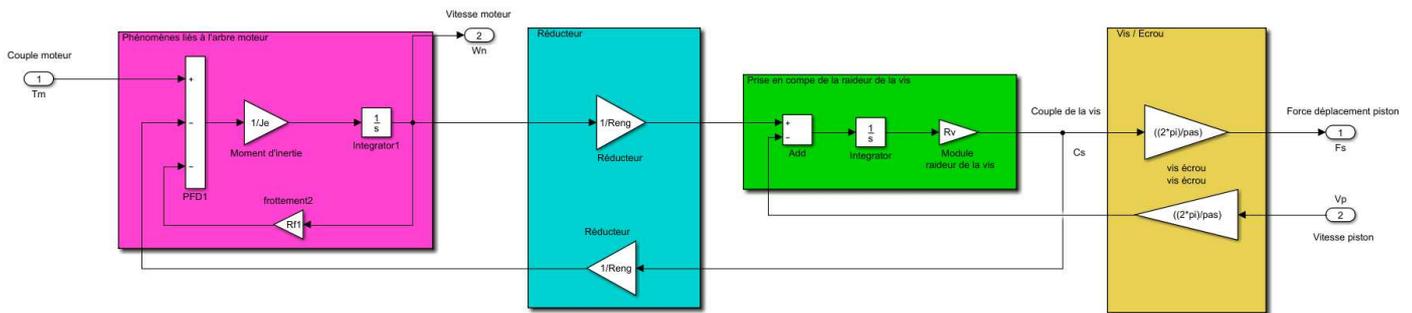
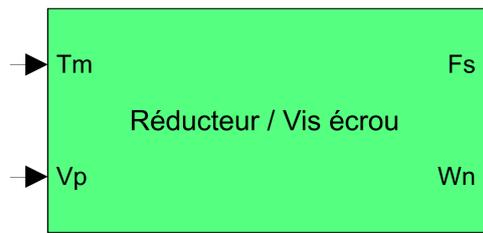
$$T_m = I * K_e$$

On modélise ces équations par :



Après différents tests, on a choisi $R_e=0.5 \Omega$, $L_e=0.1 \text{ H}$ et $K_e=0.139 \text{ rad/s}$

Réducteur / Vis écrou :



Différents phénomènes sont à prendre en compte : l'inertie et les frottements de l'arbre moteur ; la raideur de la vis

Lorsque l'on utilise un réducteur, on utilise l'équation suivante :

$$T_m = J * p * W_m + rf * W_m + n * T_c$$

Avec

J = inertie

rf = frottement de l'arbre moteur

n = réduction (=1/Reng)

Donc

$$W_m = J * \frac{1}{p} * (T_m - W_m * rf - n * T_c)$$

La vitesse en sortie du réducteur est de

$$W_{red} = \frac{1}{R_{eng}} * W_m$$

Dans notre système, la réduction est de Reng = 1024

La prise en compte de la raideur (K) :

$$T = K * \theta$$

Or

$$\theta = \int W$$

Donc

$$T = K * \int W$$

Pour notre système on a donc :

$$T_c = K * \int (W_{red} - W_{piston})$$

La raideur de la vis est assimilé à la raideur d'une poutre

$$K = \frac{A * E}{L}$$

Avec A la section de la vis, E le module de Young de l'acier inoxydable qui est de 203 et L la longueur de la vis.

Equation du système vis écrou :

$$T_s = \frac{pas}{2 * \pi} * F_p$$

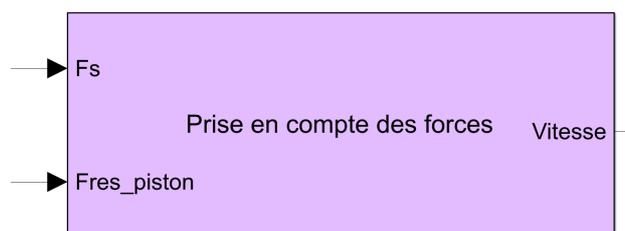
Donc

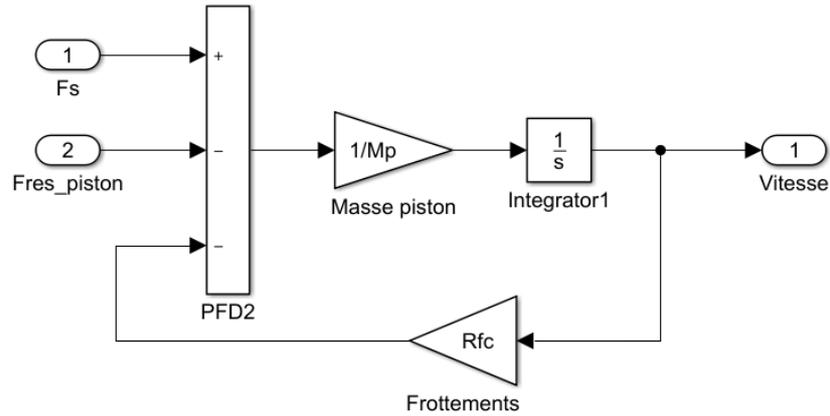
$$F_p = T_s * \frac{2 * \pi}{pas}$$

Et

$$W_{piston} = V_{piston} * \frac{2 * \pi}{pas}$$

Prise en compte des forces / masse du piston :





Application du principe fondamental de la dynamique en translation :

$$\sum F = M * \frac{dV}{dt}$$

Donc

$$V = \frac{1}{M_p} \int F_{sys}$$

Avec

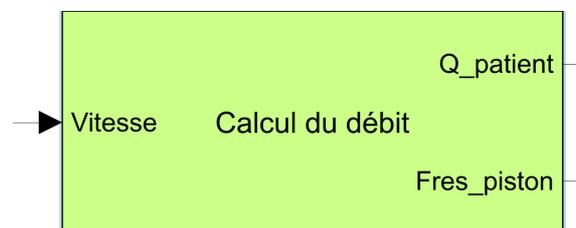
$$F_{sys} = F_p - F_f - F_{res}$$

Les forces données par le moteur, de frottement et de résistance du liquide sur le piston.

$$F_f = R_c * V$$

La masse est calculée par notre modélisation 3D sur Onshape.

Calcul du débit / partie hydraulique :



Le débit d'insuline injectée au patient se calcule de la façon suivante à partir des pressions dans la pompe et du sang du patient :

$$Q_{patient} = \frac{1}{R_{or}} (P_{cyl} - P_{patient})$$

La pression dans la pompe :

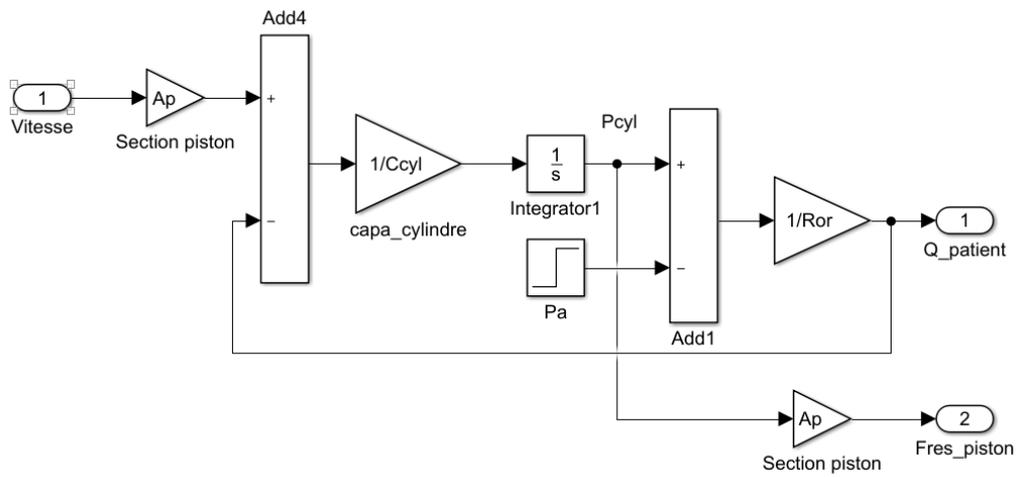
$$P_{cyl} = \frac{1}{C_{cyl}} \int (Q_{pompe} - Q_{patient})$$

Le débit donné par la pompe dépend de la vitesse de déplacement du piston et de la section du piston :

$$Q_{pompe} = V * A_p$$

C'est ici que l'on calcule la force de résistance du liquide nécessaire plus haut

$$F_{res} = P_{cyl} * A_p$$



2.2 Test boucle ouverte

Vérification de la stabilité du système

Pour vérifier la stabilité du système, nous récupérons les équations d'états du système à partir du fichier Simulink. Nous avons trouvé une fonction Matlab permettant de calculer le critère de Routh et ainsi la stabilité du système.

Les résultats du script Verifications.m :

```
Routh-Hurwitz Table:
a =
1.0e+15 *
0.0000    0.0024    2.2693
0.0000    0.4626    2.9042
0.0006    2.2584         0
0.4626    2.9042         0
2.2544         0         0
2.9042         0         0

Number Of Right Poles = 0

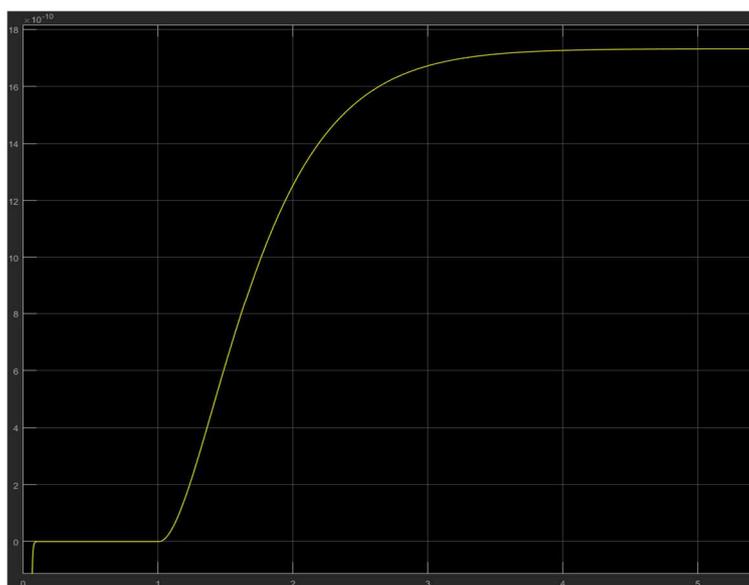
Given Polynomials Coefficients Roots :
ROOTS =
1.0e+06 *
-0.0000 + 1.5394i
-0.0000 - 1.5394i
-0.0002 + 0.0000i
-0.0000 + 0.0000i
-0.0000 - 0.0000i

Le système est stable
```

Tests en boucle ouverte

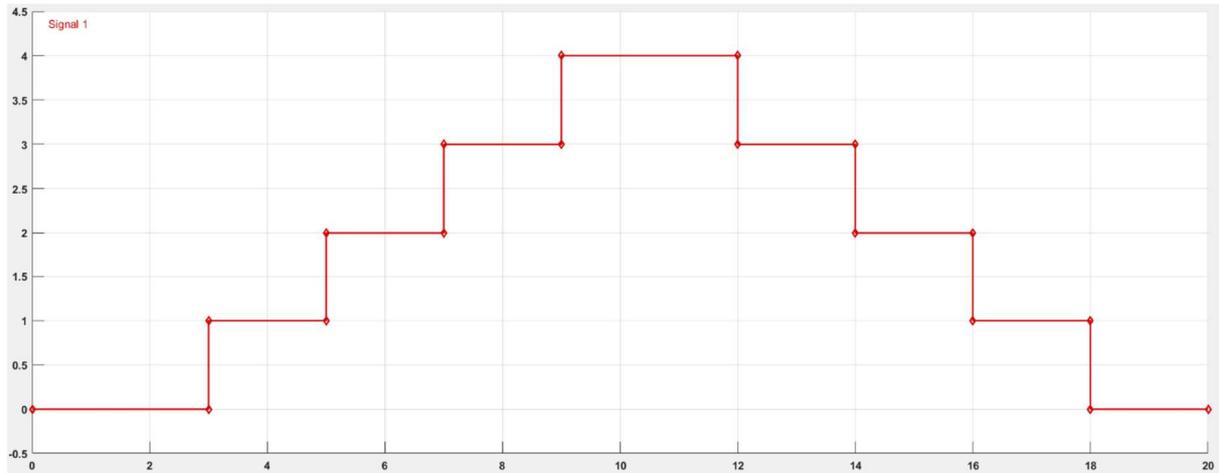
Nous avons fait quelques tests en boucle ouverte.

Tout d'abord, avec un step de 10V à 1s en entrée on retrouve en sortie le débit suivant :

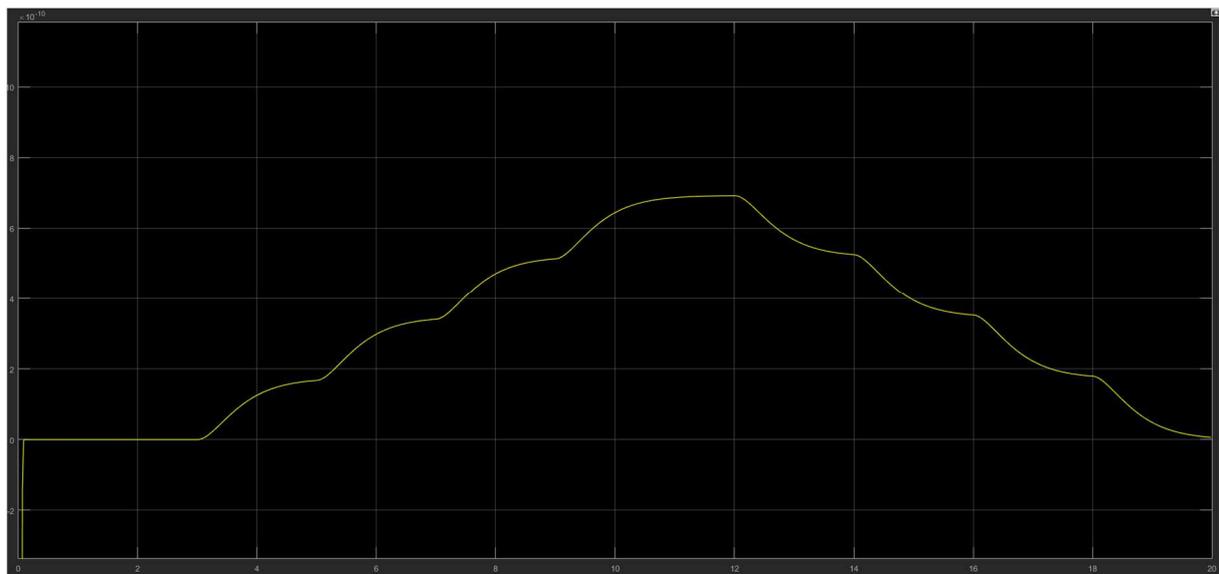


Le débit est de $17e-10$ m³/s avec un temps de réponse de 4s.

Nous avons également utilisé une commande de tension variable :

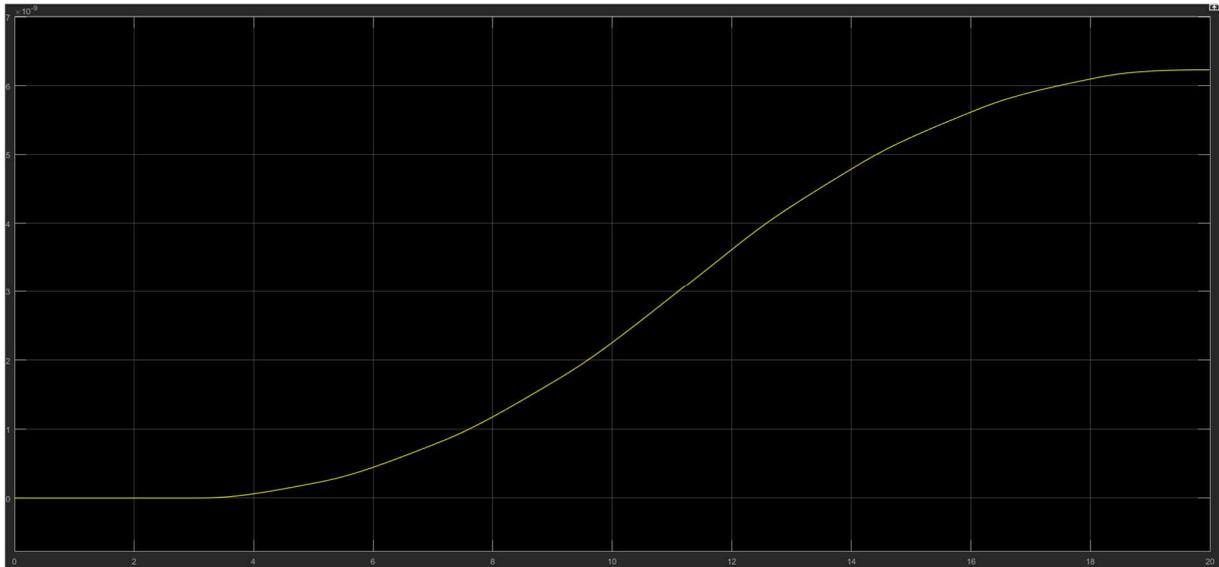


Ce qui donne un débit variable :



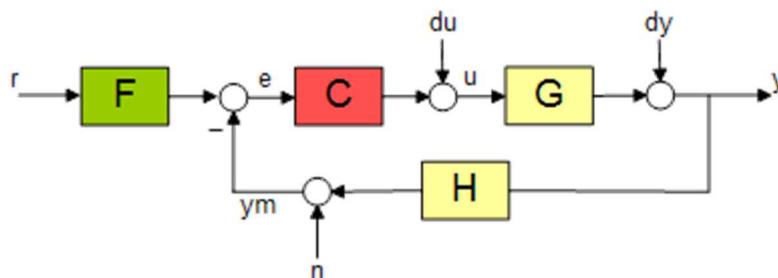
A cause du temps de réponse vu plus haut, le débit d'injection ne se stabilise jamais avant que la tension de commande change.

A partir du débit, on récupère la quantité d'insuline injectée :

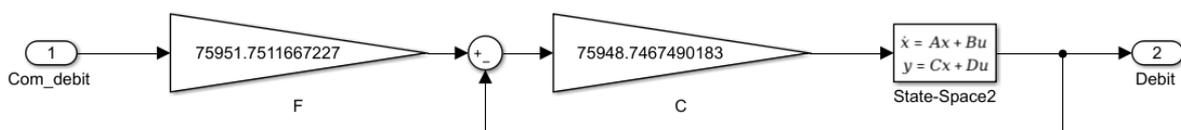


2.3 Commande du débit

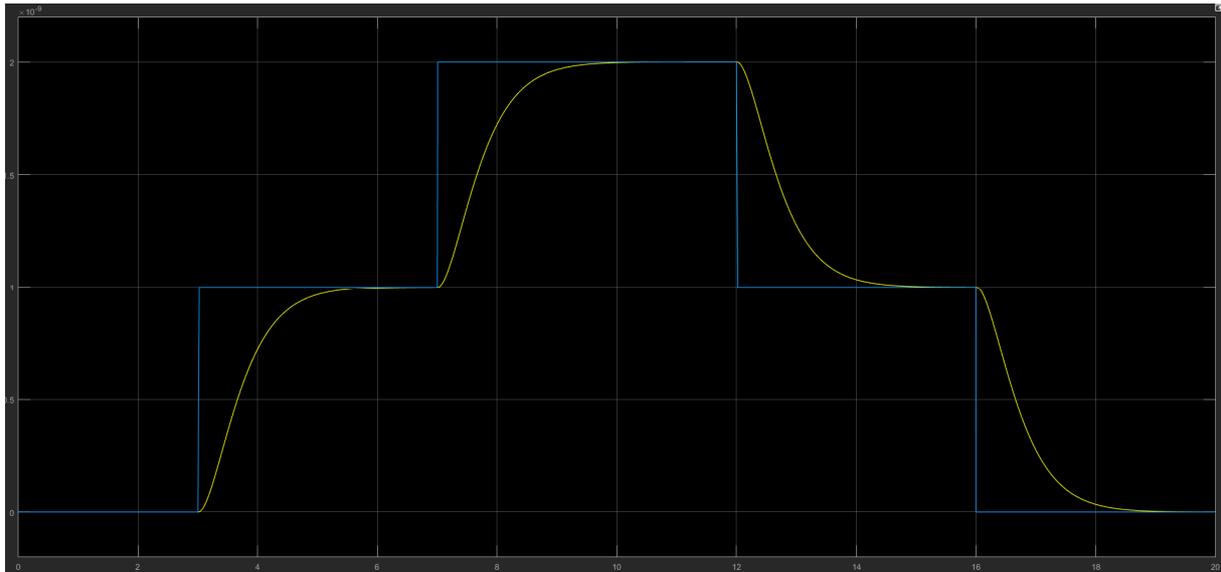
Pour réaliser la commande en débit nous avons utilisé l'app Matlab « Control System Tuner » qui permet de donner le PID ou les gains (correcteur) nécessaires au bon fonctionnement du système en boucle fermée.



Pour le correcteur de débit, nous avons donné les équations d'état de la pompe sous forme système à G et nous avons récupéré les gains F et C pour un « Tracking of step commands » sans dépassement.



Avec une commande variable de débit on trouve le graphique suivant :



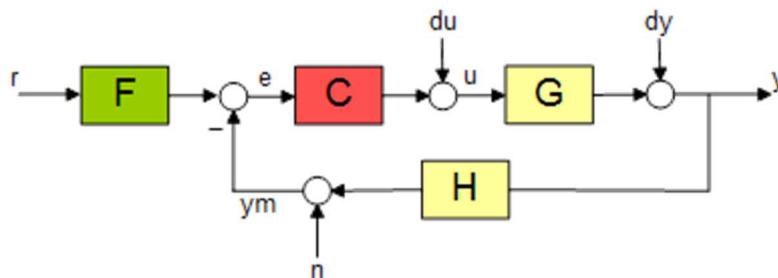
Bleu : la commande de débit / Jaune : le débit de sortie

On voit que le débit suit bien la commande sans dépassement comme nous le voulions.

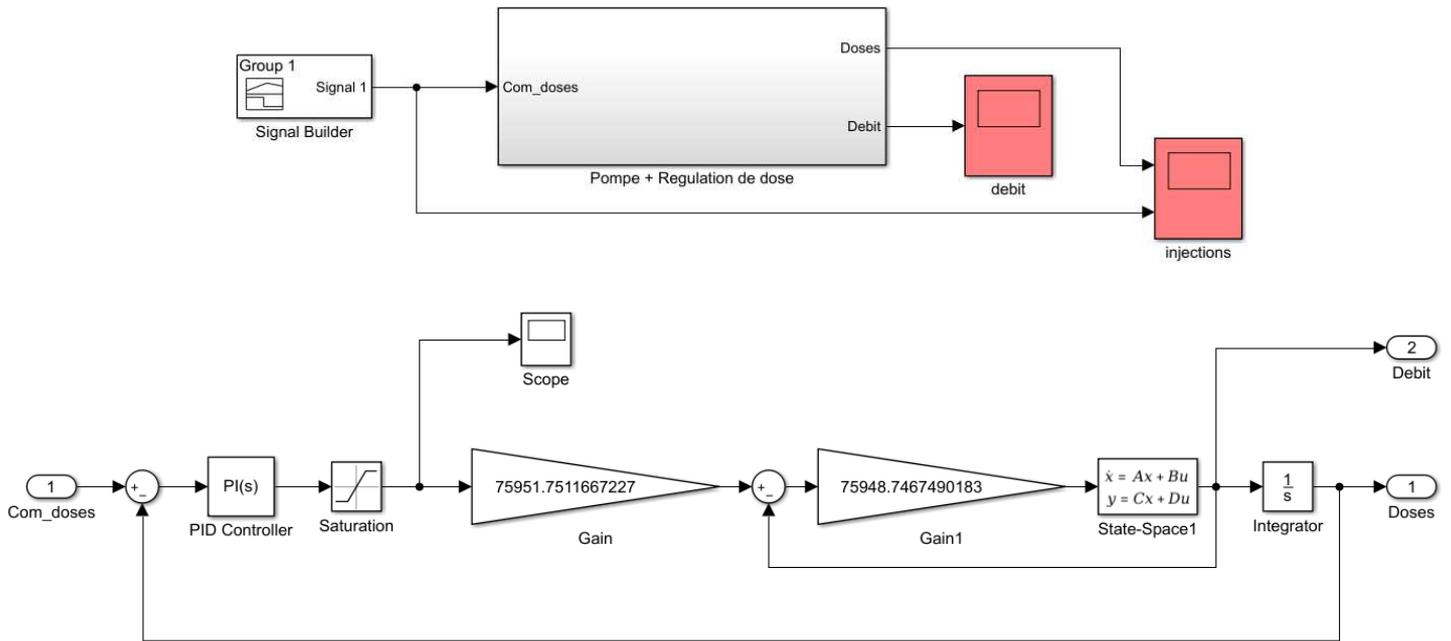
2.4 Commande de la quantité injectée

On souhaite commander la pompe en quantité d'insuline à injecter. Pour cela il faut un nouveau contrôleur qui régule l'injection des doses et donc le débit. On limite le débit d'injection à $2e-9$ m³/s ce qui permet d'injecter une dose d'insuline ($1e-8$ m³) en minimum 5s. On ajoute donc une saturation à $2e-9$ sur la commande de débit.

On utilise la même app que précédemment pour trouver le PID.

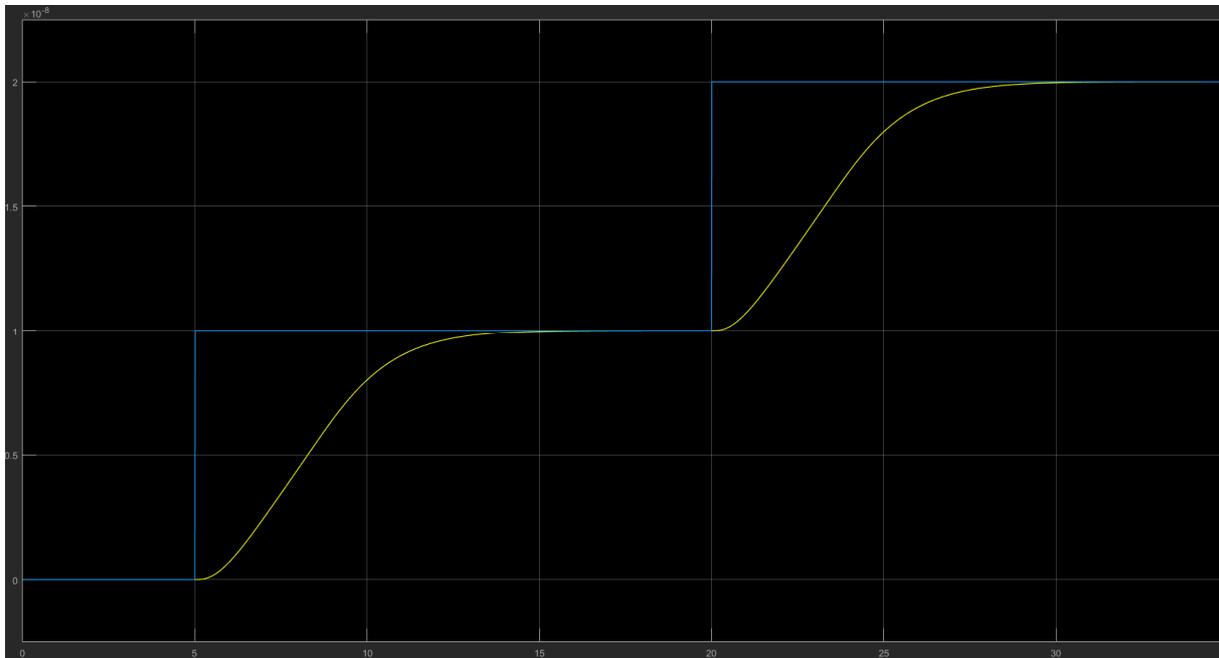


Nous avons cette fois-ci utilisé le système régulé précédent avec l'intégration pour récupérer la quantité injectée comme système G et nous ne recherchons que le C sous forme PI en plaçant F à 1.

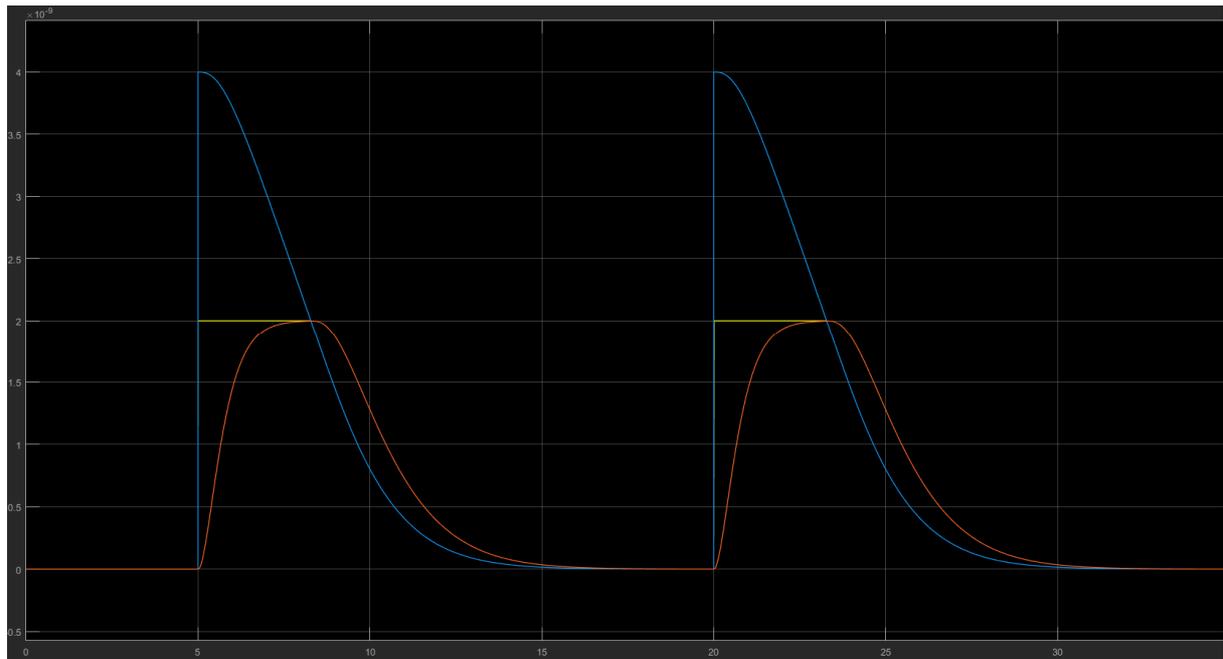


Une dose d'insuline est de $1e-8$ m³, ici on injecte une dose à 5s puis une seconde à 20s. La commande du nombre de dose doit être l'addition du nombre de dose injectée depuis la mise en marche plus le nombre à injecter.

Courbes :



Bleu : commande de quantité / Jaune : quantité injectée



Bleu : commande de débit en sortie du PI / Jaune : commande de débit après la saturation / Rouge : débit en sortie de la pompe

On peut voir que comme nous le souhaitons, la quantité injectée atteint bien la consigne. Il faut 10s pour injecter une dose, ce qui correspond à notre demande (temps d'injection supérieur à 5s). On voit également que la saturation fonctionne parfaitement.

Après ces différentes étapes de modélisation et de commandes, nous avons sous Simulink le modèle de la pompe commandable en quantité d'insuline à injecter et qui retourne le débit d'injection ainsi que la quantité réellement injectée.

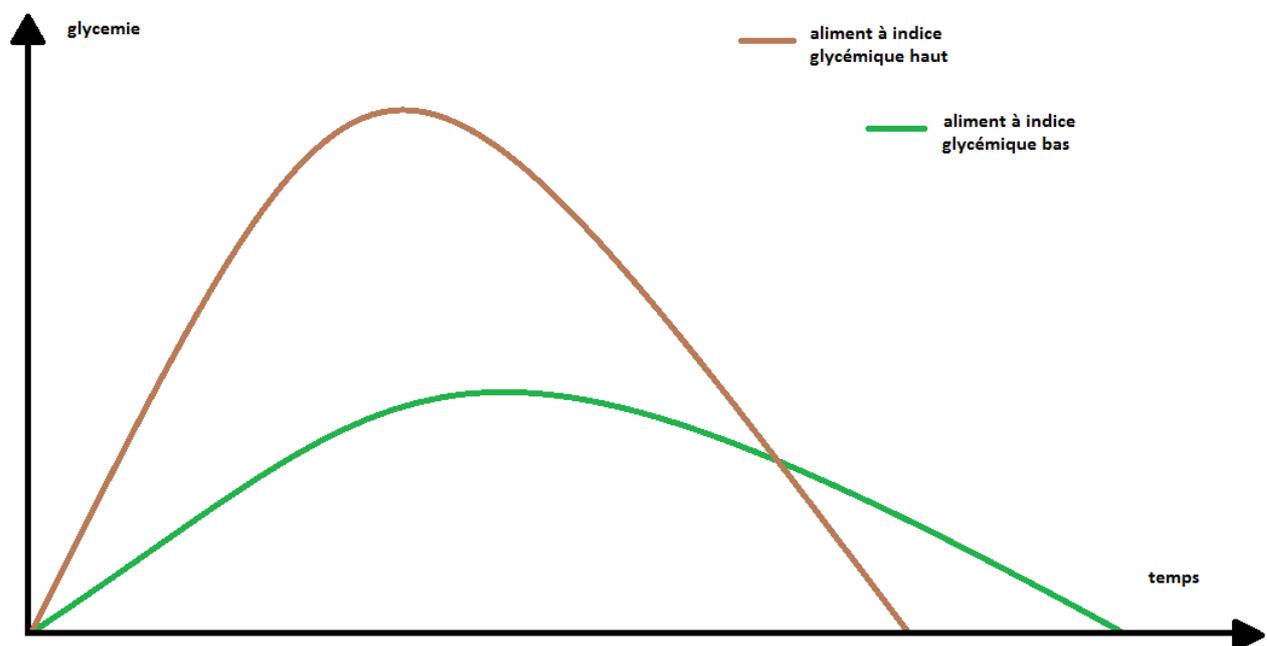
3. Emulation de la glycémie

3.1 Comment émuler la glycémie ?

Afin de pallier au manque de capteur de glycémie et de cobaye (ce qui empêche de réaliser des tests en temps réel) nous avons réalisé un dispositif d'émulation du comportement de la glycémie. Pour ce faire, nous avons étudié la cinétique de ce taux sanguin de manière à se rapprocher le plus possible du comportement réel.

La glycémie est le taux de glucose présent dans le sang, c'est la caractéristique surveillée chez les diabétiques car il ne s'autorégule pas chez eux (c'est l'homéostasie).

Lorsque l'on ingère un aliment, la glycémie augmente en fonction du taux glucidique de ce que l'on a mangé. En règle générale, lorsque l'on consomme des glucides, notre glycémie augmente. L'augmentation dépend de l'indice glycémique de l'aliment ingéré. Si son indice est élevé, la glycémie augmentera très rapidement et pour une personne non diabétique elle diminuera rapidement également. Inversement si l'indice de l'aliment est petit, la glycémie augmentera très lentement et pour une personne non diabétique elle diminuera lentement également.



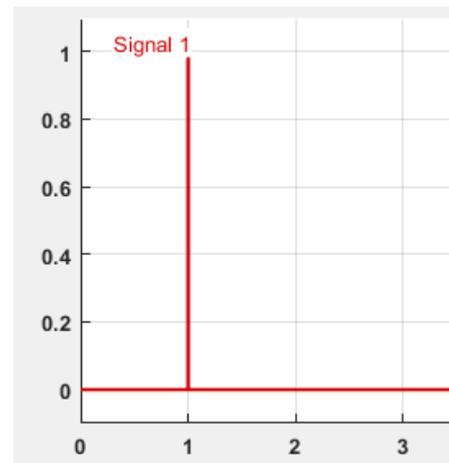
Une personne atteinte de diabète doit calculer la quantité de glucides ingérés pour connaître la quantité d'insuline à s'injecter. Il utilise un ratio qui évolue en fonction du moment de la journée et permet de donner le nombre de dose à s'injecter en fonction du nombre de glucide ingéré.

Le but de cette partie est de modéliser le comportement de la glycémie selon différents facteurs tels que l'alimentation ou l'injection d'insuline. Le taux de glycémie ainsi calculé sera ensuite transmis au calcul du nombre de dose comme donnée du capteur de glycémie.

3.2 Sous Simulink

Afin de réaliser la simulation, nous avons décidé de simplifier le modèle tout en essayant de rester proche du comportement de la glycémie.

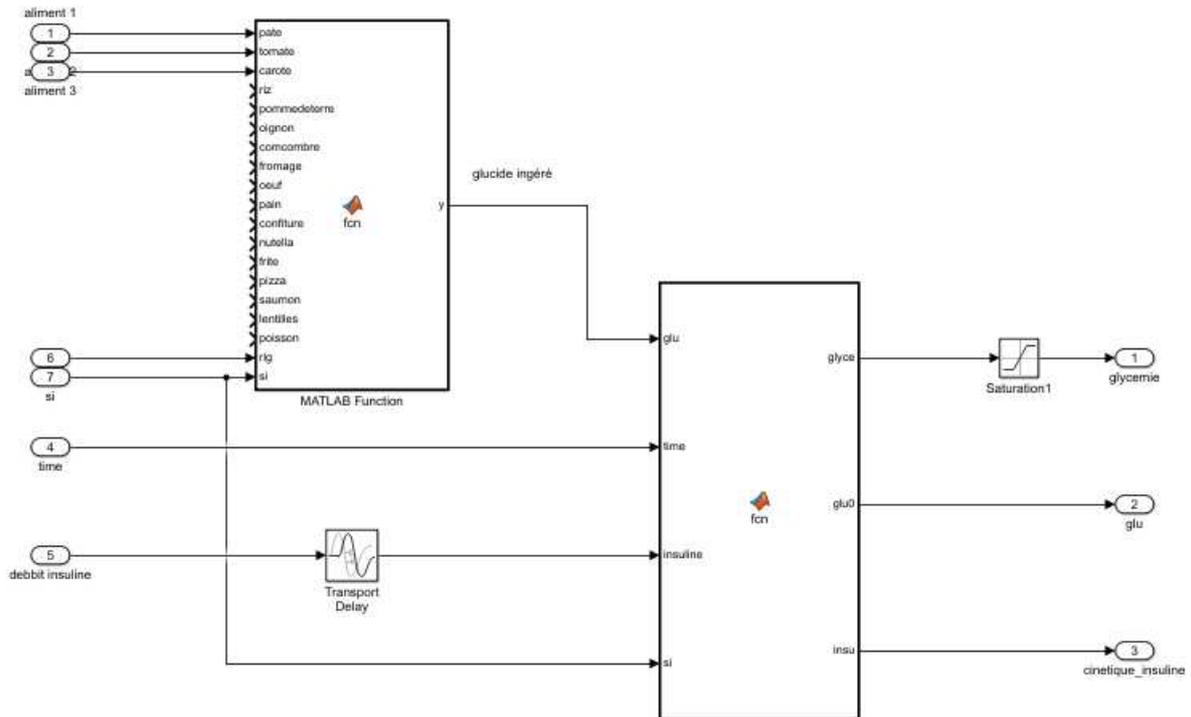
C'est ainsi que nous avons réalisé un système à plusieurs entrées. Chaque entrée correspond à un aliment, en plaçant un dirac d'une valeur de 1 à un moment donné, on simule l'ingestion de l'aliment.



L'émulation calcule à partir de ces entrées le nombre de glucide qui entre dans l'organisme en incrémentant la glycémie d'une unité chaque minute (seconde pour les simulations de test) jusqu'à ce que tous les glucides ingérés soient distribués à la glycémie.

Pour en arriver là, nous avons émis l'hypothèse que tous les aliments intégrés au système ont le même indice glycémique.

Nous avons modélisé cela de la manière suivante :

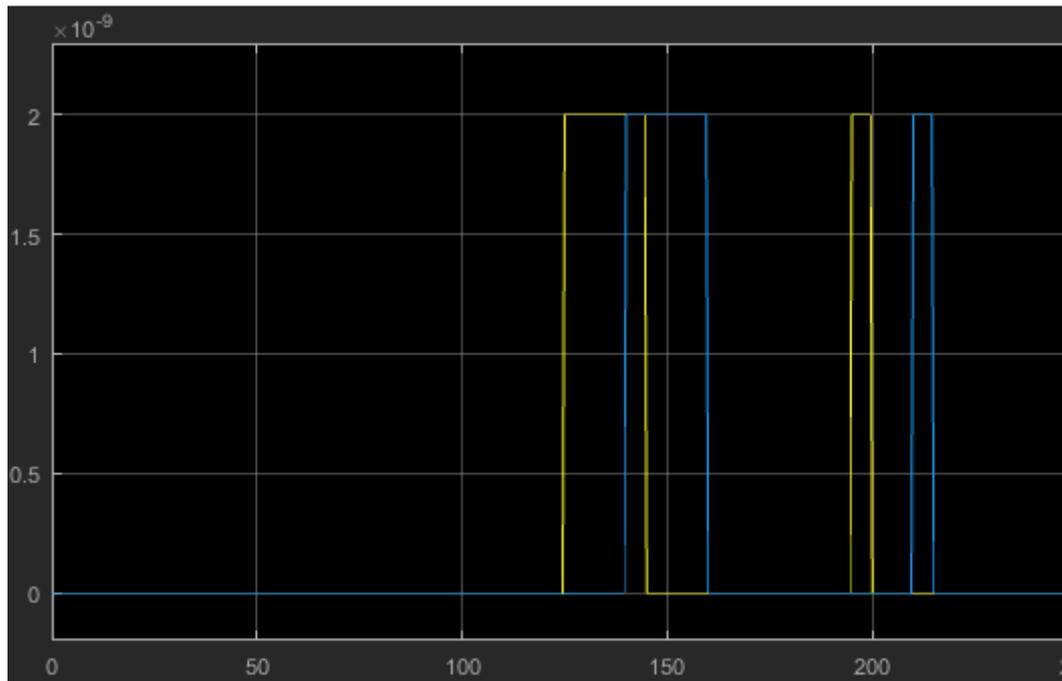


Nous avons programmé un bloc Matlab de manière à choisir les aliments qui entreront en jeu lors de l'émulation. Ce bloc donne à partir des aliments, du ratio glucidique et de la sensibilité à l'insuline du patient, les glucides qui seront assimilés dans le corps patient.

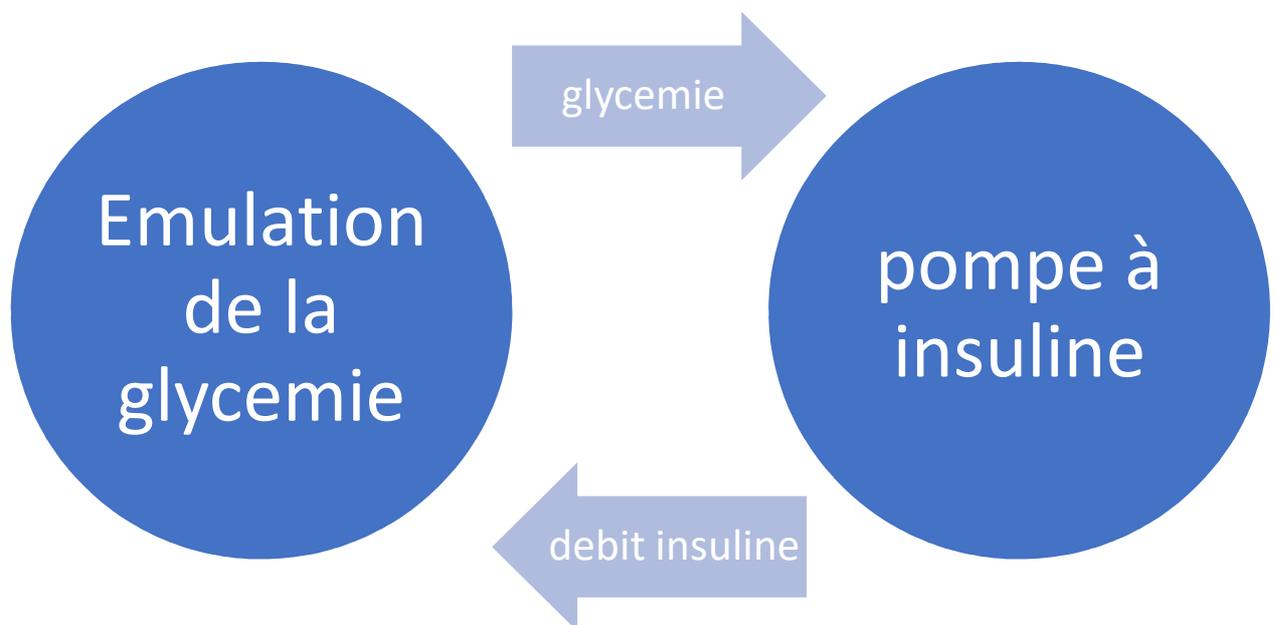
Le second bloc lui prend en entrée le nombre de glucide ingéré, l'horloge, la sensibilité à l'insuline et le débit d'insuline en sortie de la pompe. Le programme décrémente ainsi le nombre de glucide chaque seconde et incrémente en conséquence le taux de glycémie. De plus il consulte le débit d'insuline pour faire diminuer la glycémie en cas d'injection.

De manière à pouvoir intégrer la cinétique de l'insuline (début de la fonction 15 min après l'injection) nous avons ajouté un retard au niveau du débit de l'insuline de manière à ce que cette dernière se mette à faire effet seulement après ce délai.

Débit d'insuline :

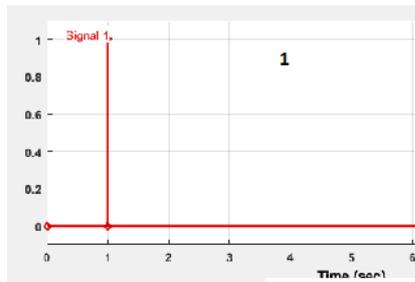


La communication mutuelle entre les deux systèmes se fait ainsi :

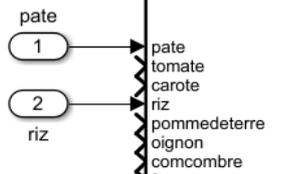
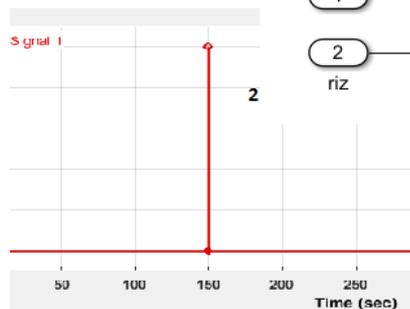


L'émulation donne la glycémie à la pompe qui réagit en conséquence de manière à donner le débit au corps humain et ainsi de suite.

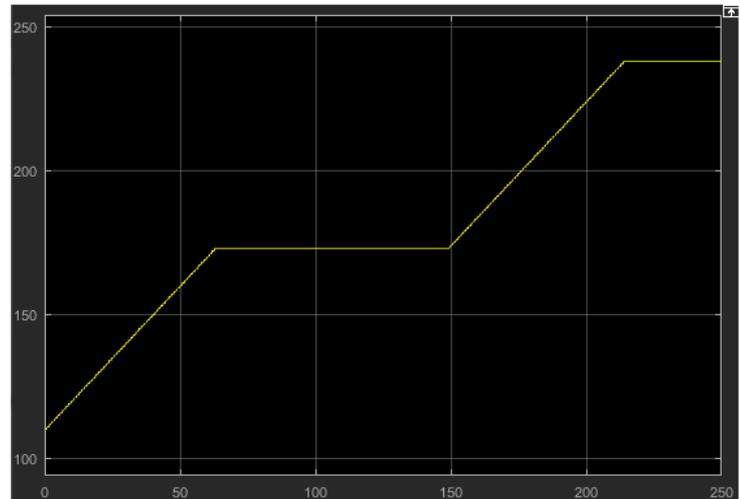
Nous avons simulé le comportement de cette partie sans considérer d'injection d'insuline afin de faciliter les calculs de l'ordinateur et nous avons récupéré les courbes suivantes :



aliment ingéré



glycemie



On observe clairement à $t=1$ l'augmentation de 60 unités de glycémie et à $t=150$ l'augmentation de 65 unités.

Rappel : pour calculer l'augmentation de glycémie, on récupère le nombre de glucide ingéré que l'on multiplie avec la sensibilité à l'insuline et le ratio glycémique.

Nous avons également mis en place cette émulation avec un dispositif Arduino que nous expliquerons par la suite.

3.3 Sous Arduino

Notre dispositif sous Arduino communique par liaisons série, une reliée à l'ordinateur pour indiquer l'absorption des aliments (perturbations de la glycémie) et une seconde qui communiquera avec la pompe lorsqu'elle sera réalisée (envoi des données glycémiques et réception du débit d'insuline). De cette façon, on tape à l'ordinateur les aliments ingérés et le système réagit en conséquence.

Nous avons tout d'abord intégré un tableau de valeur équivalent aux glucides des aliments que l'on peut ingérer lors de la simulation.

```
glucide[19]={0 , 25 , 26 , 17 , 3.6 , 8 , 3.6 , 1.3 , 10 , 1.1 , 50 , 60 , 58 , 30 , 26 , 20.5 , 16.5 , 4,0};
```

Au démarrage de l'application, nous affichons les aliments et un chiffre correspondant via la liaison série de l'Arduino avec l'ordinateur, l'utilisateur n'a plus qu'à taper le numéro de l'aliment ingéré et le système l'assimilera en considérant un ajout du nombre de glucide correspondant dans l'estomac.

```
COM4
debut de l'emulation de la glycemie, vous commencerez avec une glycemie dans la norme c'est a dire de 110
vous pouvez faire en sorte d'augmenter cette derniere en tapant le chiffre associer a d'un des aliments suivant
1-pate 2-riz 3-pommedeterre 4-tomate 5-oignon 6-comcombre 7-fromage 8-carote 9-oeuf 10-pain 11-confiture 12-nutella
13-frite 14-pizza 15-saumon 16-lentilles 17-poisson 18-fini de manger
par exemple si vous tapez pate, ne systeme reagira comme si ces derniere ont ete mange par le patient
ne pas oublier destipuler la quantite ingere
De plus si vous tapez 100 alors le systeme reagira comme si le patient avait realiser une seance de sport
pour commencer veuillez entre rapport glucidique( ex 1:8 )du patient que vous souhaitez emuler en tappant le premier chiffre
```

Ensuite, chaque seconde (ou minute selon le besoin de simulation) la glycémie s'incrémentera ou se décrémentera.

flag_augmentation 0	flag_augmentation 0	flag_augmentation 130
flag_regressif 0	flag_regressif 0	flag_regressif 0
evol_glycemie 0	evol_glycemie 58.0000000000	evol_glycemie 145
glycemie 109	glycemie 104	glycemie 121
	glycemie 110	

De manière à pouvoir simuler, nous avons diminué les temps de simulation en considérant les minutes comme étant des secondes.

Sinon il aurait fallu attendre plus d'une heure pour pouvoir voir le système assimiler l'ingestion d'un repas, ce qui n'est pas pratique à simuler. De manière à passer de seconde à minute, il suffit de modifier cette ligne : `if(currentMillis - previousMillis >= 1000) {` et mettre beaucoup plus de temps(10000000) .

Le système indique également si on est en hyper ou hypoglycemie en émetant des alertes sur le port serie .

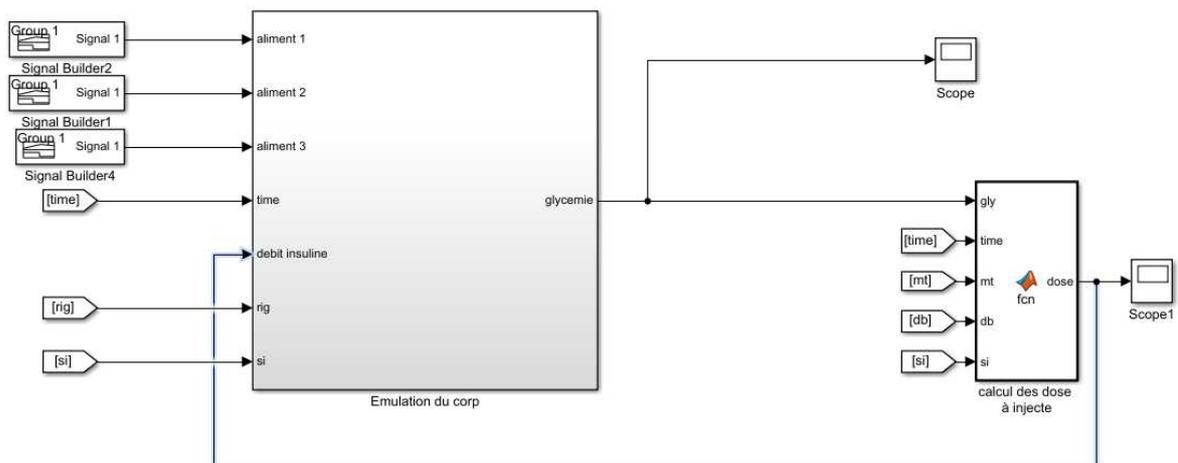
```
flag_augmentation
56
flag_regressif
0
evol_glycemie
145
glycemie
176
-----
ALERTE
HYPERGLYCEMIE
-----
```

4. Modèle complet avec Matlab/Simulink

4.1 Calcul des doses à injecter

De manière à faire le lien entre la pompe et l'émulation il est nécessaire de calculer le débit d'insuline à injecter pour corriger la glycémie en conséquence.

C'est pour cela que nous avons réalisé un module qui donne la quantité d'insuline à injecter selon la glycémie, la sensibilité à l'insuline, le débit basal et le temps.



De manière à ne pas injecter trop d'insuline, l'insuline met beaucoup de temps à faire effet, nous enregistrons dans une mémoire les quantités injectées et nous les prenons en compte dans le calcul des injections.

```
if (time > t)
    if gly > 120
        n=0;
        % memoire de ce qui a ete injecte l
        if nm-1<=0
            n=n+mem(10+nm-1);
        else
            n=n+mem(nm-1);
```

Si la glycémie est supérieure à 120, il faut injecter de l'insuline pour la faire diminuer. Plus la glycémie est élevée, plus il faut injecter d'insuline. Pour prendre en compte le fait qu'une dose ait pu être injectée précédemment, nous regardons la quantité d'insuline injectée lors des 25 dernières secondes (minutes en temps réel) et nous soustrayons cette quantité à la commande.

```

if gly < 130
    if 1e-8 * ins / si > n
        nd = nd + 1e-8 * ins / si - n ;%injection
        mem(nm)= 1e-8 * ins / si - n;%memoire
    else
        mem(nm)=0;%on injecte rien et on met en r
    end
else
    if gly < 140
        if 1e-8 * ins*2 / si > n
            nd = nd + 1e-8 * ins*2 / si - n ;
            mem(nm)= 1e-8 * ins*2 / si - n;
        else
            mem(nm)=0;
        end
    end
end

```

Nous gérerons également les injections basales. Définie dans l'initialisation en unité d'insuline /h, cette valeur est donnée par le médecin et est différente pour chaque personne.

Le but étant d'injecter une quantité fixe répartie sur une heure. Pour simplifier l'injection, nous souhaitons l'injecter en plusieurs fois plutôt qu'en continu. C'est pourquoi qu'à chaque mt (temps en minute entre chaque injection) nous injectons une fraction de la quantité basale à injecter en une heure.

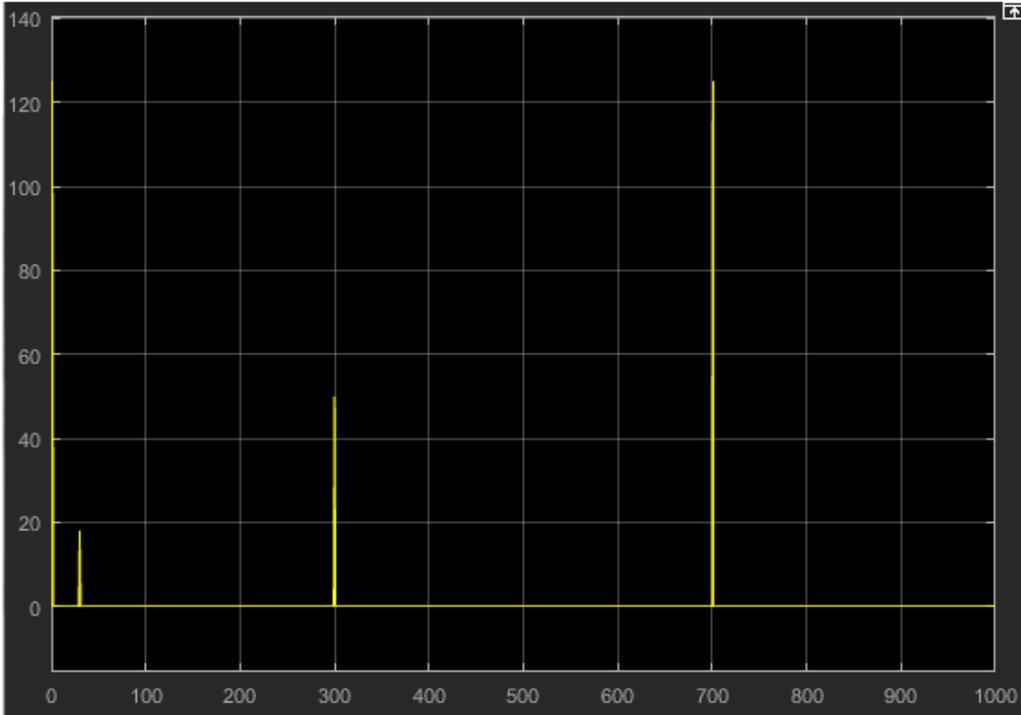
```

% injection insuline basale
if time >= t2
    if gly > 90
        nd=nd+ 1e-8 * (db / (60/mt));
    else
        if gly > 70
            nd=nd+ (1e-8 * (db / (60/mt)))*0.5;
        end
    end
    t2=t2+mt;%*60;
end

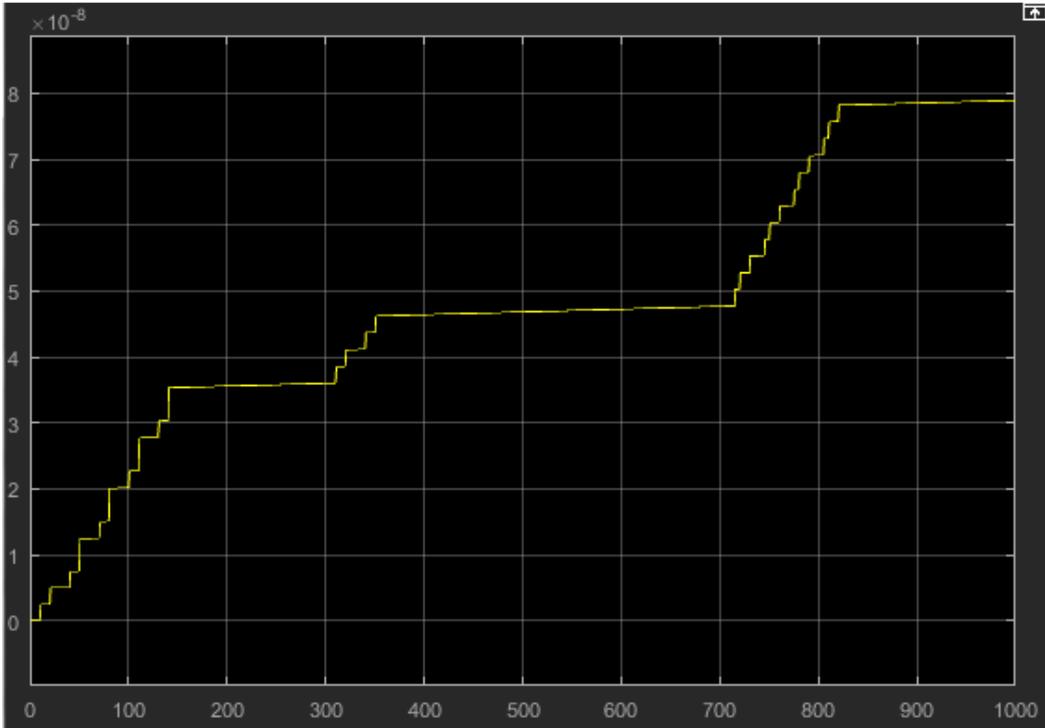
```

Afin de valider la commande des injections, nous avons lié l'émulateur de glycémie à cette partie. En donnant une consigne composée de 3 repas, nous observons une ingestion de glucose à des temps distincts.

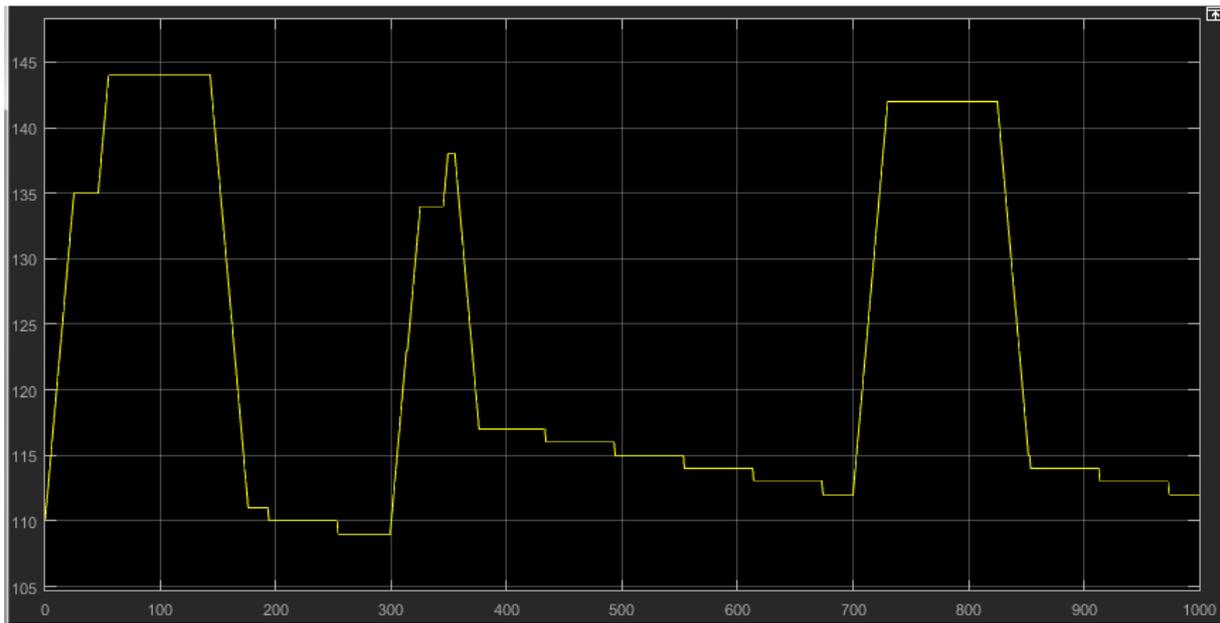
Consigne d'ingestion des aliments :



Quantité d'insuline injectée pour réguler le taux de glycémie :



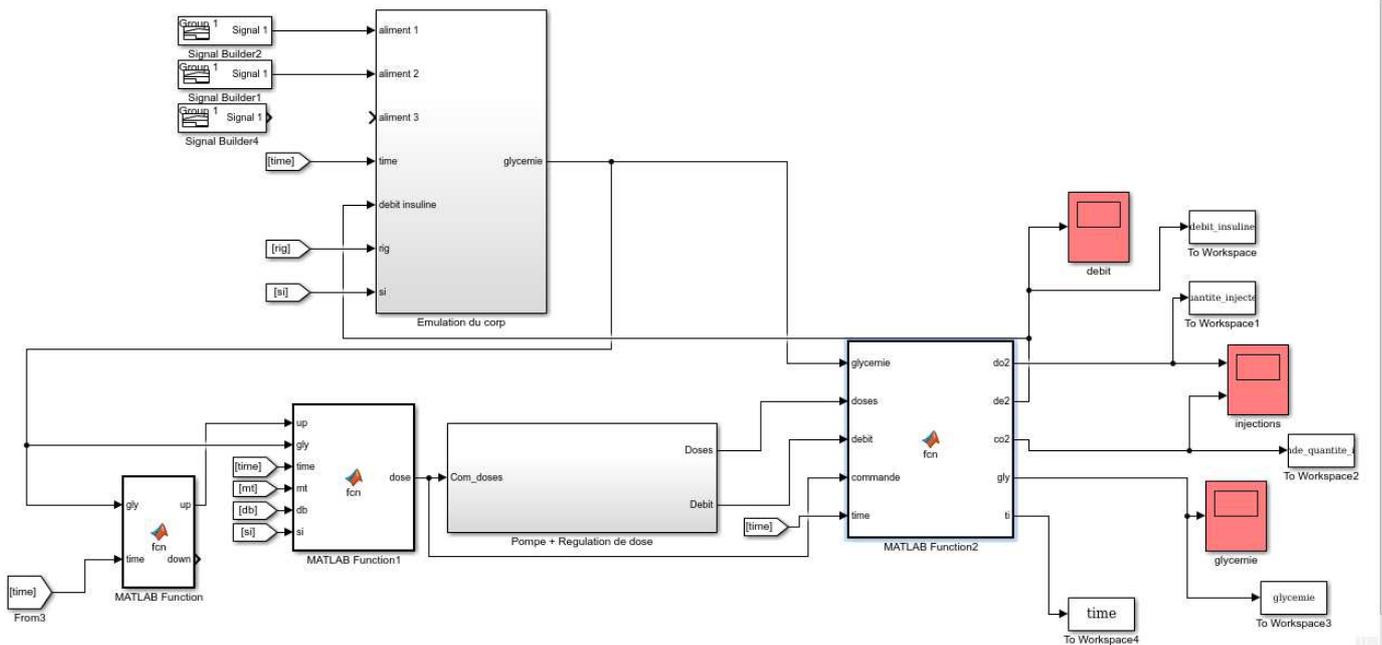
Glycémie durant la simulation :



Nous observons bien les pics de glycémie au moment des repas qui sont largement diminués par l'insuline injectée en sortie du régulateur. La glycémie est bien régulée entre 100 et 120 mg/dl.

4.2 Validation du modèle total

Nous avons ensuite pris soin de mettre en commun les deux parties afin de valider le fonctionnement de la pompe et l'algorithme de calcul des doses. Pour cela, nous avons importé les deux modélisations expliquées précédemment ainsi que la commande de la glycémie et avons récupéré le schéma Simulink suivant :



Nous avons ensuite réalisé des simulations du modèle total afin de valider ce dernier.

Pour cela, nous avons donné à l'émulation de la glycémie la consigne d'ingestion d'aliment suivante :

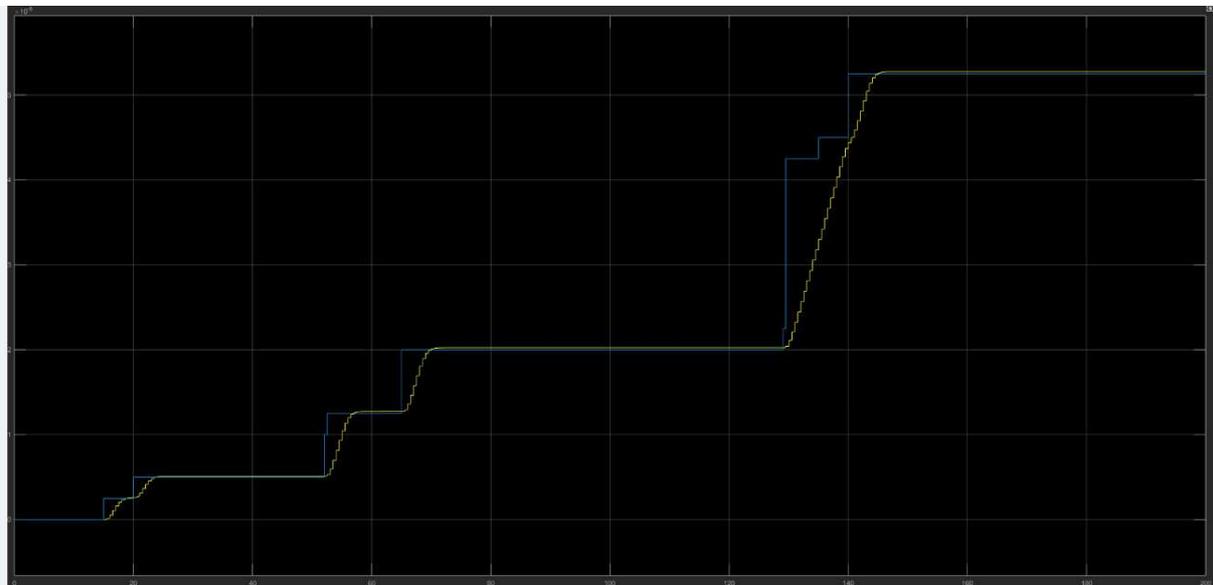
A $t = 1$ s l'individu ingérera un aliment à 25g de glucide pour 100g

A $t = 30$ s l'individu ingérera un aliment à 3.5 g de glucide pour 100g

A $t = 300$ s l'individu ingérera un aliment à 10 g de glucide pour 100g

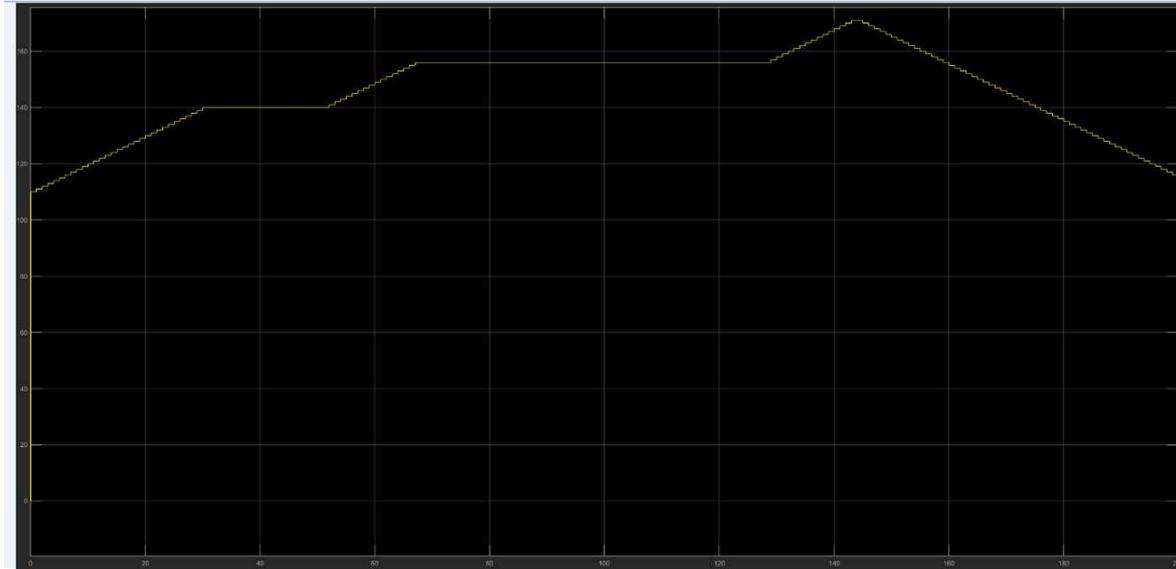
Nous avons ensuite lancé une simulation de 200s et avons observé les courbes de l'injection et de glycémie.

Quantité d'insuline injectée :



Bleu : commande de quantité / Jaune : quantité injectée en sortie de pompe

Glycémie :



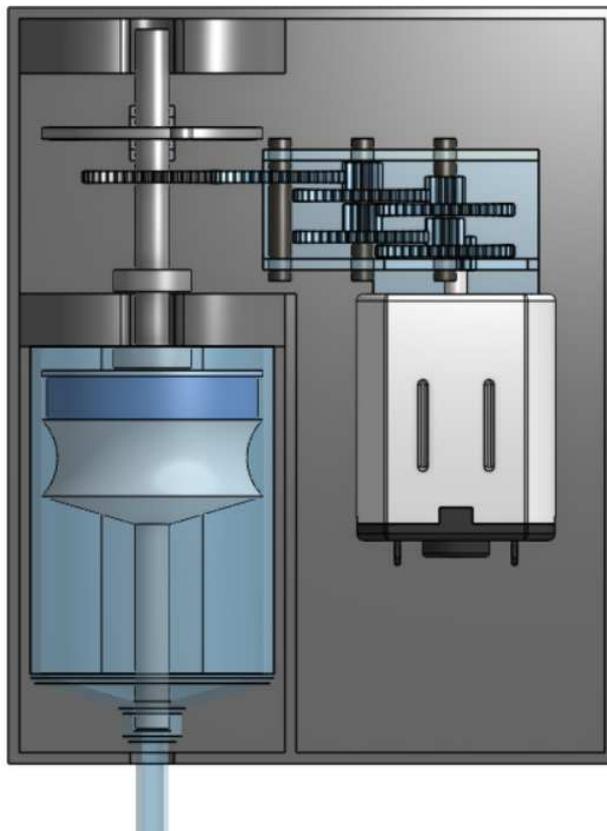
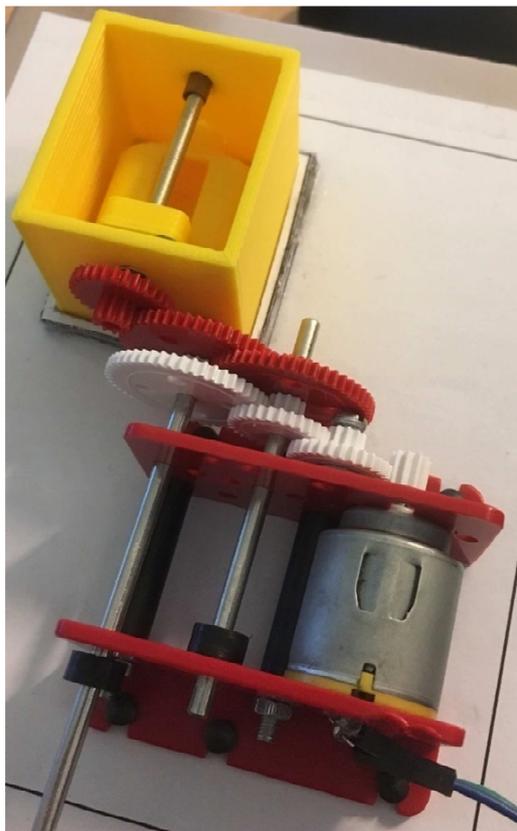
On peut remarquer que les injections se font lorsque la glycémie augmente, un temps d'injection d'une dose est d'environ 5 seconds.

5. Prototype

5.1 Le modèle 3D

La pompe a été modélisée pour tenir 5 jours sans recharge.

Entre la première modélisation et la nouvelle, il y a quelques modifications :

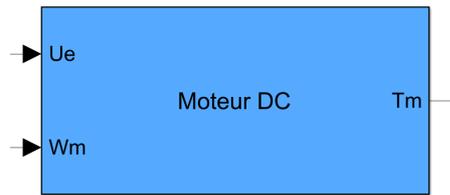


La nouvelle pompe ressemble plus à une seringue, nous avons décidé d'utiliser un piston existant (celui d'une seringue 50cc). La vis se trouve au centre du piston et la sortie est donc excentrée. Pour cette nouvelle pompe, nous avons créé notre propre réducteur avec quatre roues 40/10 ce qui crée une réduction de 1024. Nous avons également créé des cales pour la vis, cela manquait lors des tests réalisés sur la première pompe, la vis se déplaçait linéairement et quittait ses maintiens.

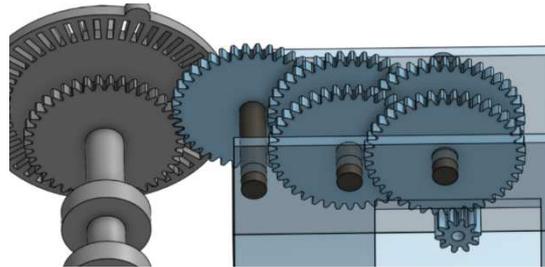
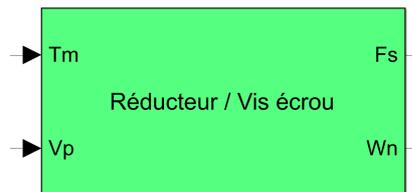
Attention, le moteur dessiné n'est qu'un exemple, le moteur n'a pas encore été choisi.

Sur cette modélisation 3D, on retrouve chaque sous parties du modèle analytique :

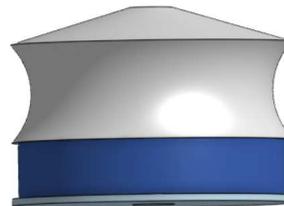
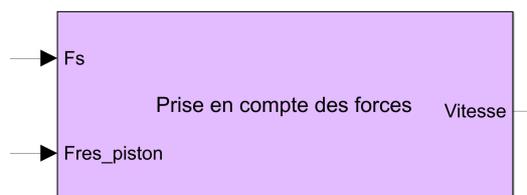
Moteur :



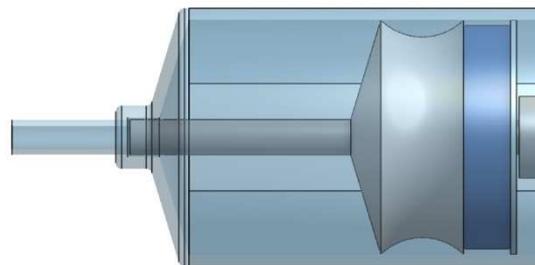
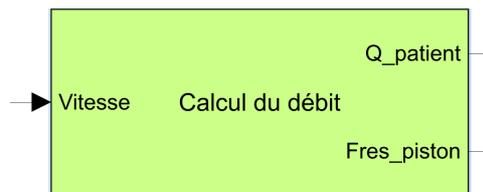
Réducteur / Vis écrou :



Prise en compte de la masse et des forces résistantes :



Partie hydraulique :

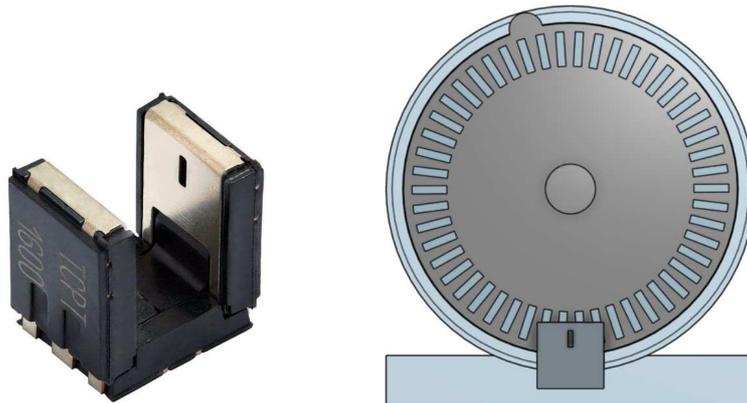


5.2 Mesures et capteurs

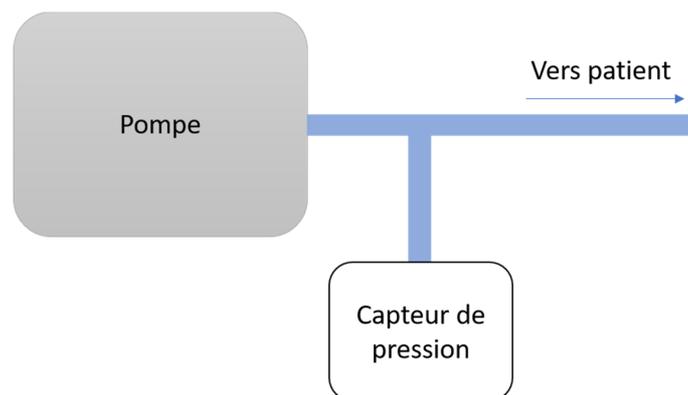
Nous avons besoin de différentes valeurs pour utiliser la pompe. Pour son fonctionnement classique, il faut mesurer la quantité d'insuline injectée car c'est ce que l'on souhaite réguler. Pour une question de sécurité, il faut savoir si le cathéter est bouché. Pour l'interface avec le patient, il faut connaître la charge de la pile et la quantité d'insuline restante dans la pompe.

Pour mesurer la quantité injectée il y a différents moyens. Il est possible de placer un capteur de débit en sortie de la pompe et intégrer la valeur (comme en simulation) mais comme nous l'avons découvert dans la première moitié du projet, un capteur aussi précis est trop cher pour être utilisé dans une pompe à insuline. Mais étant donné que nous avons réalisé le modèle analytique du système, il est possible de récupérer la quantité injectée à partir d'autres valeurs telle que la vitesse de rotation de la vis.

C'est pour cela que nous avons pensé à utiliser un commutateur optique et une roue à trou pour mesurer la vitesse de rotation. Plus la roue à de trou, plus nous serons précis.



En utilisant un capteur de débit, il aurait été possible de vérifier si le cathéter n'était pas bouché mais comme cela n'était pas possible, nous avons réfléchi à une autre idée. Nous avons pensé utiliser un capteur de pression. On place un T en sortie de la pompe, le T sera connecté sur le capteur de pression et sur le cathéter. De cette façon, si le cathéter se bouche, la pression sur le capteur augmentera.



Ces quelques capteurs peuvent être utilisés pour notre application mais ce ne sont pas les seules solutions possibles. Il faudrait peut-être d'autres capteurs pour mesurer d'autres grandeurs.

6. Les problèmes rencontrés

Temps de simulation :

Les simulations Matlab utilisent énormément de RAM, ce qui a souvent rendu nos ordinateurs lents. Lors de la mise en commun et de la validation du système total, nos ordinateurs n'étaient plus capables de lancer une simulation.

Ces problèmes sont dus au temps d'échantillonnage très petit ($10e-6$) qu'il faut utiliser avec le système pompe (les simulations bloquent avec un temps plus grand). Nous avons donc cherché un ordinateur avec 16GO de RAM avec Matlab.

Nous en avons trouvé au P2 mais nous n'avons pu réaliser qu'une simulation car la simulation de 200s a duré près de 10h et nous n'avons pas eu d'accès à cette salle par la suite. Nous avons décelé des erreurs lors de cette simulation, nous les avons normalement corrigées mais il n'a pas été possible de valider ces modifications.

Conclusion

Ce projet nous a permis d'acquérir de nouvelles compétences multiphysiques théoriques grâce à la modélisation, les simulations, la commande et le dimensionnement du système. Nous avons pu utiliser différents types de logiciels tels que Matlab/Simulink pour la modélisation et la simulation ou Onshape pour réaliser la pompe en 3D.

Ce projet diffère des projets que l'on avait réalisés jusqu'à présent car il a un but précis et important pour les diabétiques. Nous ne nous sommes pas seulement fixés sur le travail technique à réaliser mais nous avons également appris de nombreuses choses dans le domaine médical et biologique car le pancréas artificiel est un système bionique.

Un long travail reste à faire pour que notre système soit commercialisé mais nous sommes heureuses d'avoir pu y participer.

Bibliographie

Etat de l'art :

<https://www.federationdesdiabetiques.org/federation/actualites/une-etude-clinique-pour-le-pancreas-artificiel-diabeloop-associe-a-la-pompe>

<https://www.pourquoidocteur.fr/Articles/Question-d-actu/17799-Diabete-le-Etats-Unis-donnent-leur-feu-vert-au-pancreas-artificiel>

<https://www.pourquoidocteur.fr/Articles/Question-d-actu/23400-Le-pancreas-artificiel-commercialise-France-2018>

<https://www.anses.fr/fr/system/files/NUT-Ra-Glucides.pdf>

<http://sante.lefigaro.fr/article/le-pancreas-artificiel-est-il-proche-/>

<https://www.cellnovo.com/fr/pump>

<http://animas.ca/fr/animas-vibe>

<http://www.dlife.com/files/Timeline/>

<http://diabsante.com/index.php/la-pompe-a-insuline/les-differentes-pompes-a-insuline>

Modélisation analytique de la pompe :

Raideur de la vis : https://fr.wikipedia.org/wiki/Raideur_%28m%C3%A9canique%29

Réducteur : <http://relatifetincongru.free.fr/node55.html>

Vis Ecou : http://pedagogie.ac-limoges.fr/sti_si/accueil/FichesConnaissances/Sequence3SSi/co/S3B22_Association_modele_composant_14.html