ELEC441
DESIGN PROJECT REPORT
Glucose Control in Type 1 Diabetes

Ben Mattison
38924122

April 7, 2017

1 Background

The purpose of this project is to design a control scheme for use as an "artificial pancreas" to regulate blood glucose levels in a Type-1 diabetic patient. Glucose levels are controlled using insulin, and disturbed with carbohydrate intake at each meal. We consider the following model:

$$Y(s) = \frac{K_M e^{-\tau_{DM} s}}{s(\tau_M s + 1)} U_M(s) + \frac{K_I e^{-\tau_{DI} s}}{s(\tau_I s + 1)} U_I(s)$$

Here, U_M is the carbohydrate disturbance in grams, U_I is the insulin input rate in mU/min and Y is the glucose concentration in mg/dl. The model has the following parameters (nominal):

Table 1: Nominal Model Parameters

K_M	τ_M	τ_{DM}	K_I	τ_I	τ_{DI}
4 mg/dl per g	50 min	15 min	-0.075 mg/dl per mU/min	140 min	25 min

This report outlines multiple designs tested over a simulated 48hr period. The simulated protocol had the following specifications:

- Regular meals were administered at breakfast, lunch and dinner.
- Each design attempted to maintain a blood glucose concentration setpoint of 110 mg/dl.
- A maximum insulin infusion rate of 166 mU/min was allowed, and negative infusion rate was not allowed.
- The effects of a ±25% variation in K_I and meal size were investigated, as well as a ±15 min variation in meal timing. The controllers designed needed to be robust under the given uncertainty.

2 Design Part 1 - Feedback Control

Design a feedback control scheme, that minimizes the episodes of hyperglycemia and particularly of hypoglycemia. Your scheme should be robust to the uncertainty mentioned above.

2.1 Q-design

First, a controller was designed with Q-design for the non-delayed system. The following form for Q was considered:

$$\bar{Q}(s) = \bar{G^i}\bar{F_Q}$$

where $\bar{G^i}$ is the inverse of the non-delayed plant, and

$$\bar{F_Q}(s) = \frac{\omega_N^2}{s^2 + 2\zeta\omega_N s + \omega_N^2}$$

giving

$$\bar{Q}(s) = \frac{\omega_N^2(\tau_I s + 1)s}{K_I(s^2 + 2\zeta\omega_N s + \omega_N^2)}$$

This gives the following controller for non-delayed system:

$$\bar{C}(s) = \frac{\bar{Q}}{1 - \bar{F_Q}} = \frac{\omega_N^2(\tau_I s + 1)}{K_I(s + 2\zeta\omega_N)}$$

which is a not exactly a PID controller, as the integrator in the plant cancels the integral action from the $1 - \bar{F_Q}$. This was not an issue as the final controller block used was the Q transfer function (see Figure 1).

The final controller used was a Smith predictor of the form:

$$C(s) = \frac{\bar{C}}{1 + \bar{G_0}\bar{C}(1 - e^{-\tau_{DI}s})}$$

The block diagram scheme for this design, created in Simulink, is shown in Figure 1

The scheme shown in Figure 1 is a modified Smith predictor for Q-design, where the controller block is replaced by the $\bar{Q}(s)$ transfer function instead. This is a simpler and more elegant solution as suggested in the course notes.

Other things to note about the control scheme:

- The saturation block after the Q controller block limits the insulin input rate between 0 and 166 mU/min.
- The bias blocks of -110 and +110 were implemented as a Simulink bypass to prevent the windup that occurs after an initial setpoint jump at $t = 0$. This allowed the results to purely reflect the disturbance response.

The key parameters for this design are the bandwidth ω_N and the damping ratio ζ. The bandwidth was chosen to be 10 times faster than the fastest plant pole, which was at $\frac{1}{\tau_I} = \frac{1}{140} = 0.0071$ rad/s. So we pick $\omega_N = \frac{1}{14} = 0.071$ rad/s.

This design was first tried with a standard $\zeta = 0.75$ for an underdamped response. The simulation results are shown in Figure 2.

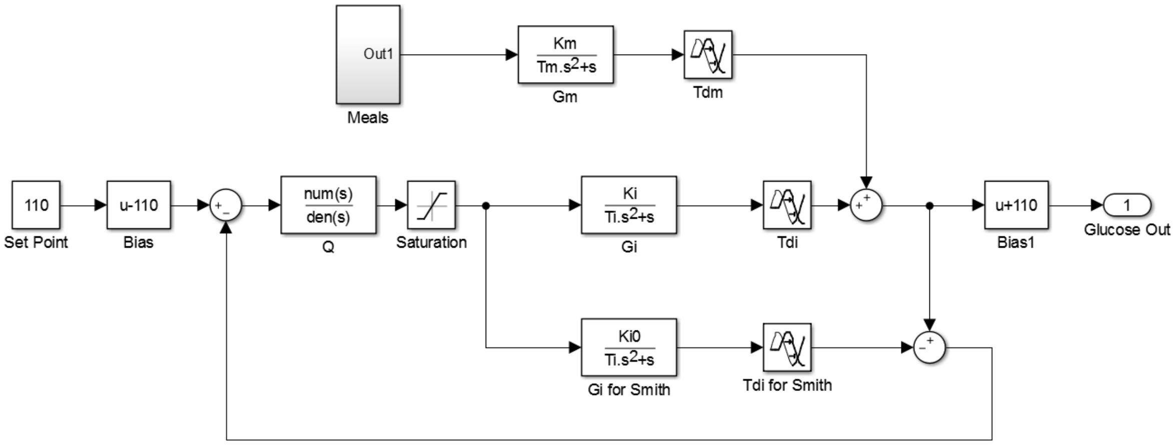

Figure 1: Block diagram used in the modified Q-design Smith predictor feedback control scheme.

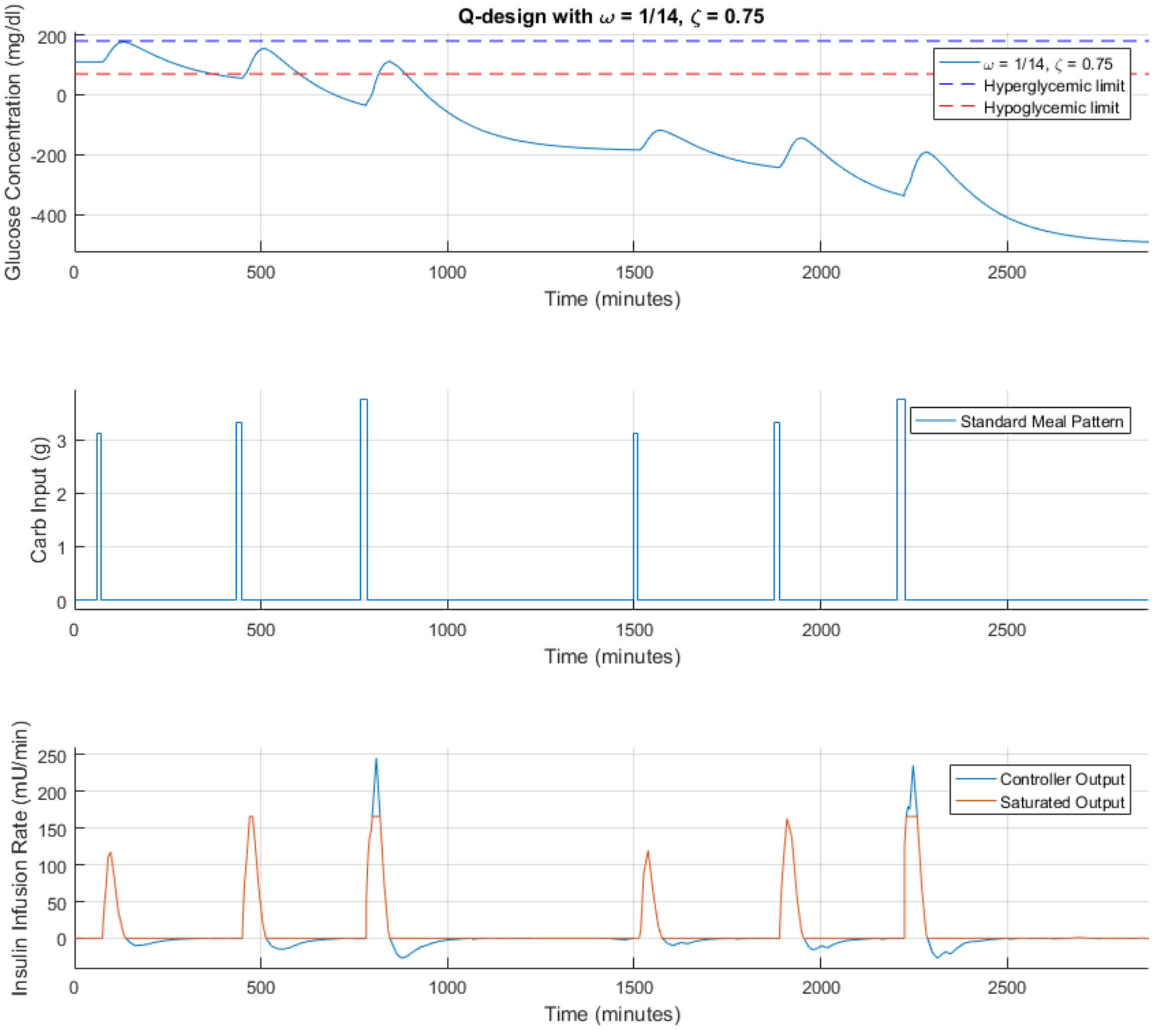

Figure 2: Simulation results for $\omega_N = 1/14$, $\zeta = 0.75$.

Clearly, these parameters produce an unfavourable result. The controller is much too aggressive, and often saturates the insulin infusion rate as seen in the 3rd plot of Figure 2. Especially of note is the fact that the controller's aggressive (ideal) response produces an overshoot which results in a negative (impossible) insulin infusion rate. The culmination of the aggressive response without the ability to have a negative infusion rate leads to an overpowering insulin response hence rendering the patient very hypoglycemic, leading to death.

As we don't want the patient to die from such an aggressive controller, the damping ratio was adjusted so that an overdamped response was produced, while keeping the bandwidth at the same value. After a few quick iterations, the ideal damping ratio was found to be $\zeta = 4.75$. Simulation results for this new parameter are shown in Figure 3.

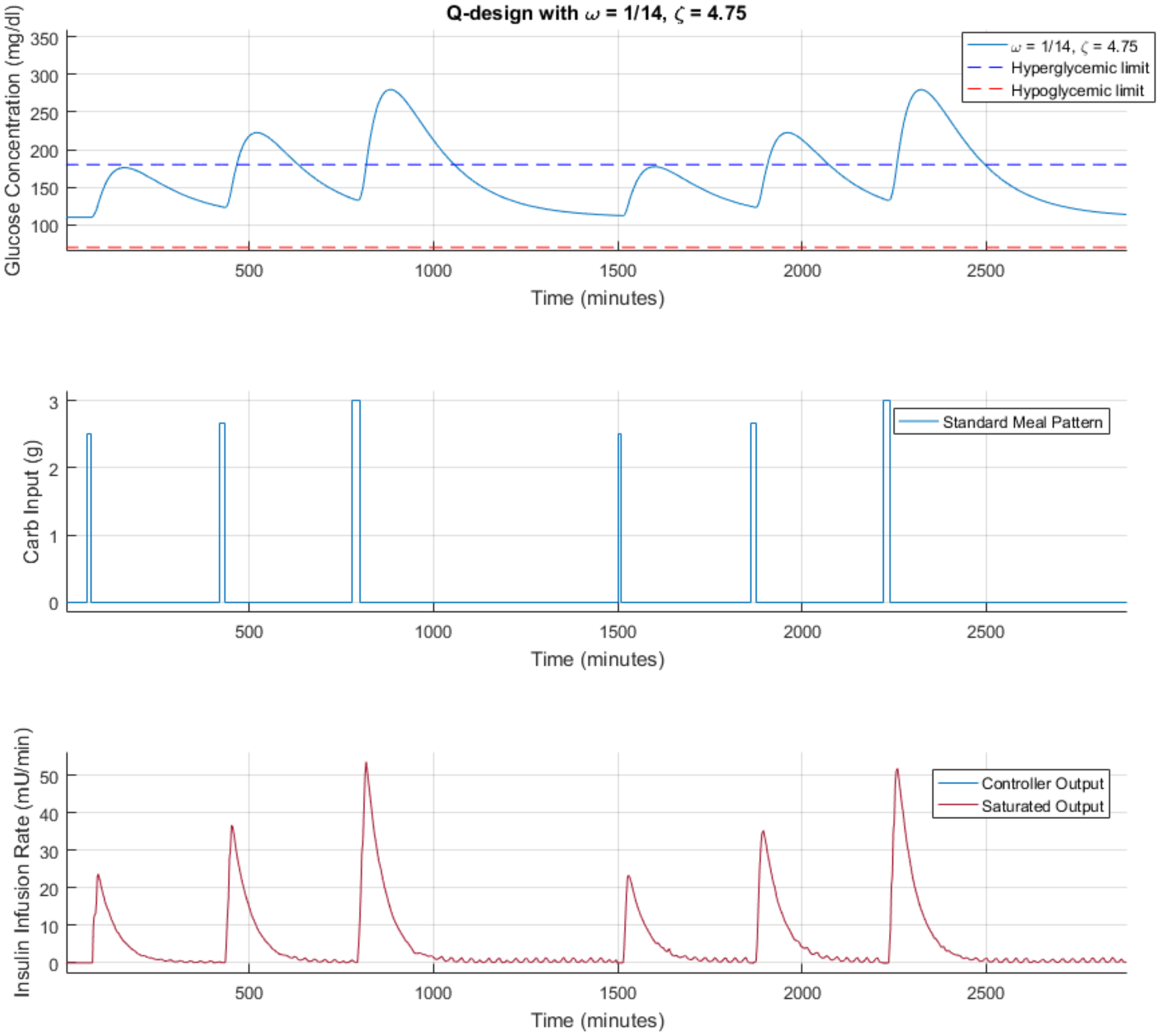

Figure 3: Simulation results for $\omega_N = 1/14$, $\zeta = 4.75$. In the 3rd plot, the controller output and saturated output are indistinguishable as the insulin infusion rate does not saturate.

The results for the overdamped response are much better than the underdamped one. The subject never goes into the hypoglycemic range. The insulin infusion rate never saturates or attempts to administer a negative rate. As the response is much less aggressive, the subject does however go into periods of hyperglycemia following lunch and dinner but the final set point after each day returns to the desired 110 mg/dl, and thus control would not be lost after multiple days. Further, the overall glucose could be brought down by setting a setpoint just above the lower limit of 70 mg/dl, as the glucose concentration never drops below the setpoint in this scheme.

2.2 Robustness Testing

As outlined in Section 1, the control scheme needed to be robust to uncertainty in K_I, meal size and meal timing.

2.2.1 K_I Uncertainty

The results of simulations for each of the extreme values of K_I are shown in Figure 4.

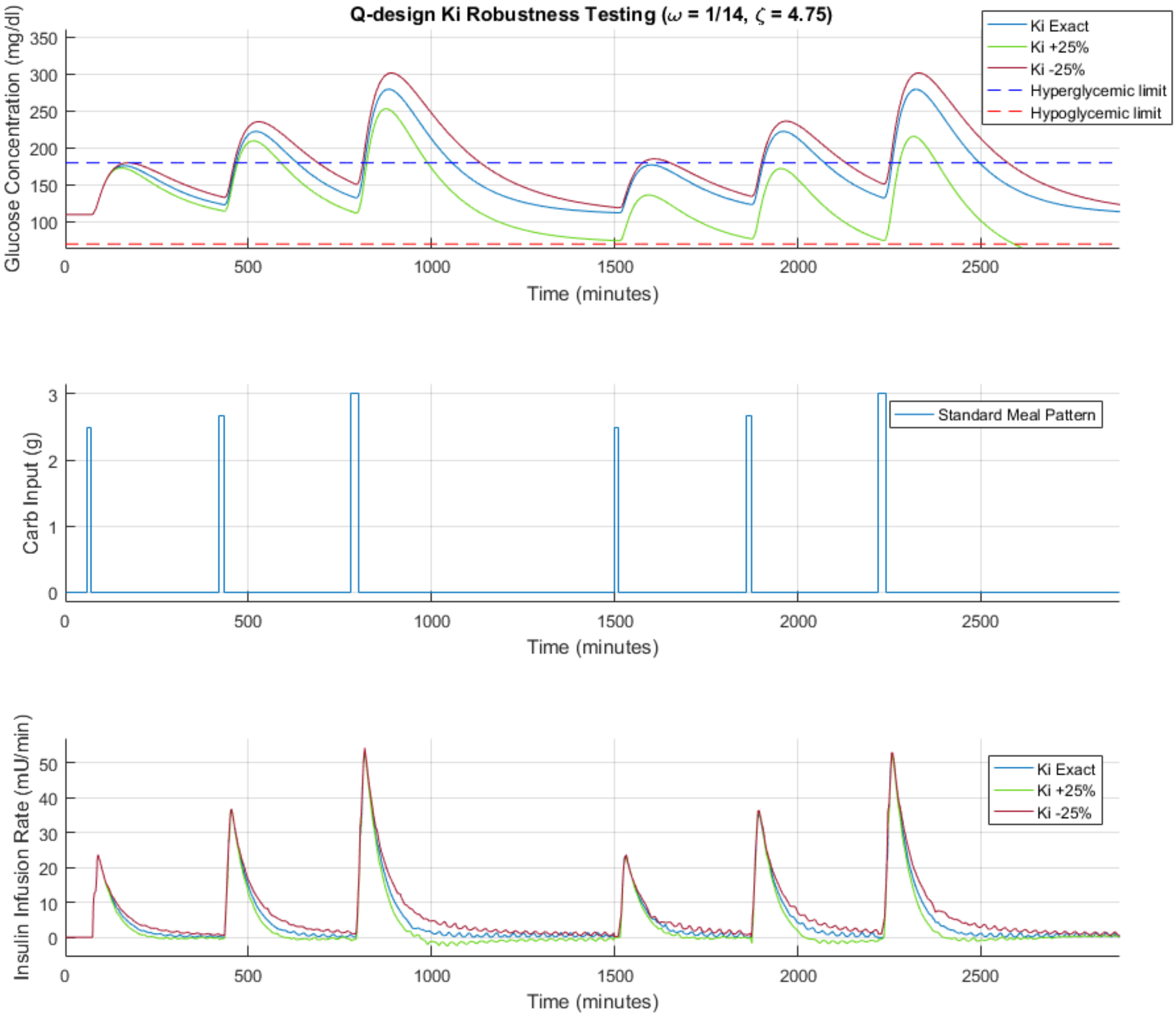

Figure 4: Simulation results for $\omega_N = 1/14$, $\zeta = 4.75$ under $\pm 25\%$ uncertainty in K_I. The insulin infusion rates shown in the 3rd plot are the attempted rates provided by the controller. With the +25% variation in K_I, the negative rate seen was clipped by the saturation block leading to the hypoglycemic episode as seen in the 1st plot.

The scheme proves to be quite robust under this uncertainty in K_I and does not lose control of the glucose concentration level in the patient. On the high end, the episodes of hyperglycemia are only extended minimally and there is only one episode of hypoglycemia at the end of the 48hrs, for the K_I increased by 25%. This hypoglycemia occurs in a similar manner to the original underdamped scheme. The controller overcompensates for its response and attempts to send a negative insulin infusion rate, which is blocked by the actuator saturation block.

2.2.2 Meal Size Uncertainty

The results of simulations for each of the extreme values of meal size are shown in Figure 5.

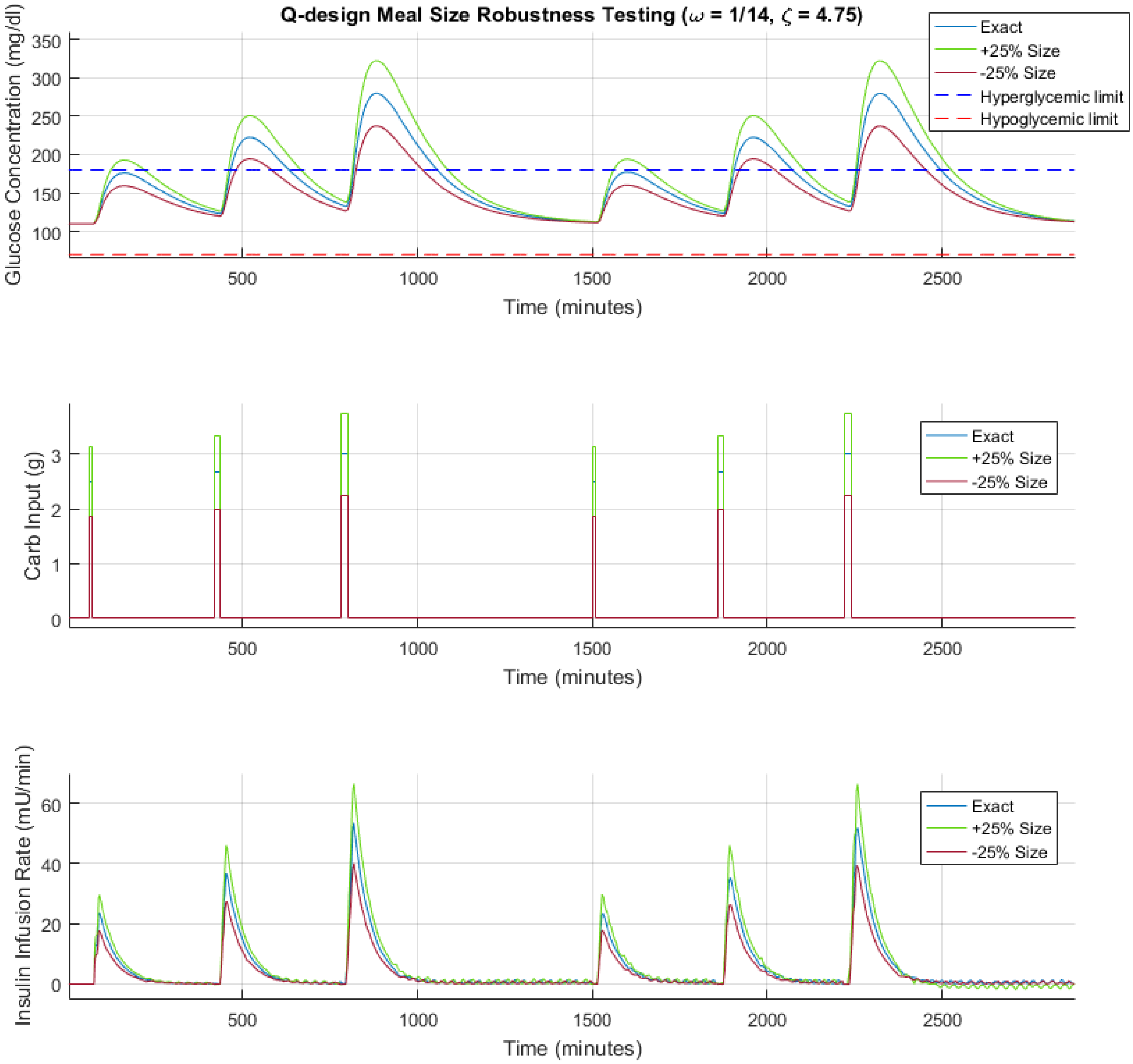

Figure 5: Simulation results for $\omega_N = 1/14$, $\zeta = 4.75$ under ±25% uncertainty in meal size.

Again, the robustness of the scheme is quite good. The scaling of meal size simply scales the response up or down accordingly, especially at its maximum value. A key thing to note is that after a full 24hr cycle the glucose concentration returns to the same setpoint for all 3 values of meal size, meaning that the setpoint does not drift for any uncertainty in meal size.

2.2.3 Meal Timing Uncertainty

The results of simulations for various meal timings, shifted by ±15 mins, are shown in Figure 6. The simulated schedules that were predicted to give the most disruptive behaviour were ones where different meals were shifted closer relative to each other.

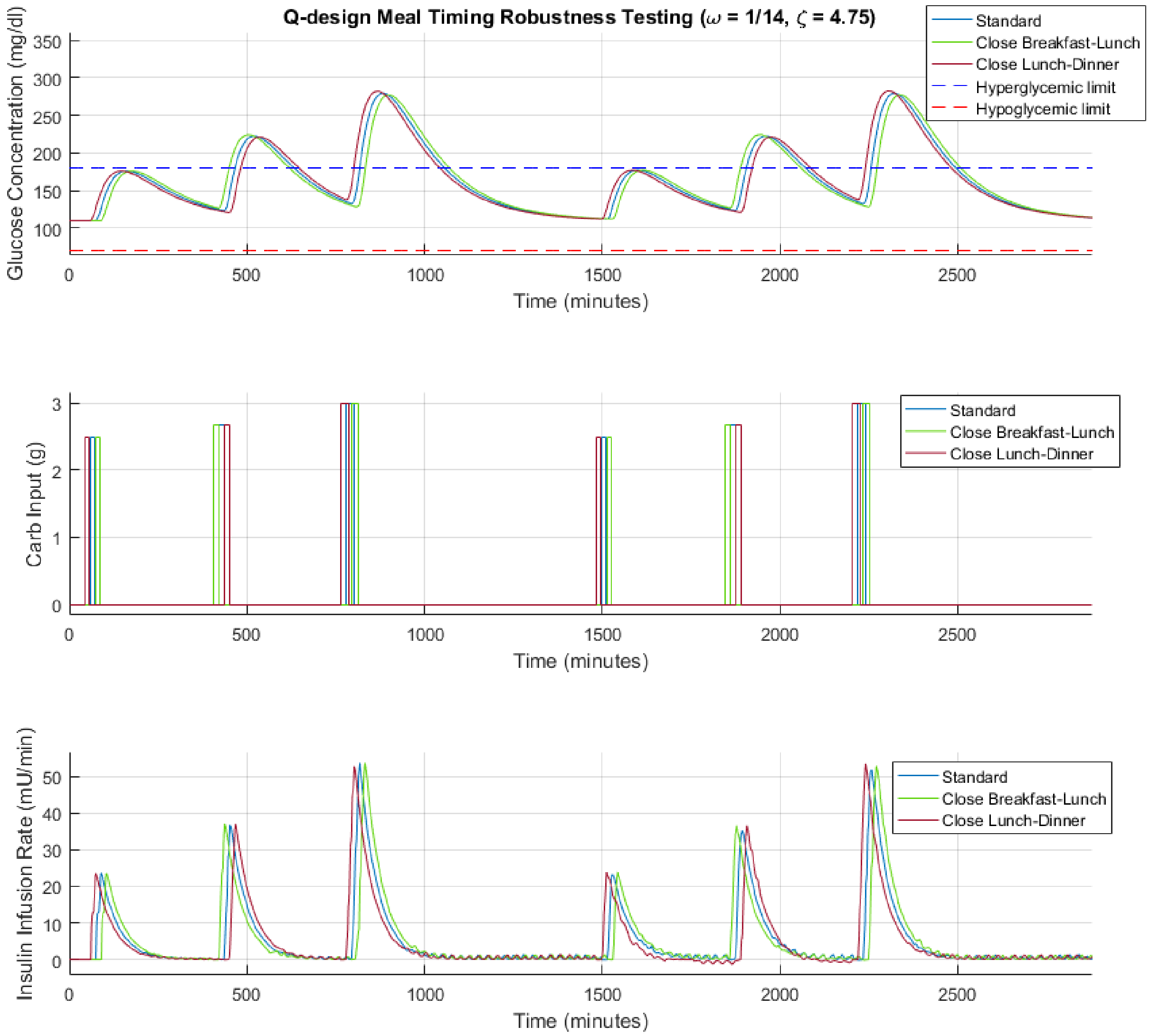

Figure 6: Simulation results for $\omega_N = 1/14$, $\zeta = 4.75$ under various combinations of ±15 min uncertainty in meal timing. "Close Breakfast-Lunch" refers to breakfast being shifted later by 15 mins and lunch being shifted earlier by 15 mins. Similarly, "Close Lunch-Dinner" refers to lunch being shifted later and dinner being shifted earlier.

Overall, the control scheme was very robust to meal timing uncertainty. Even with meals being brought closer together by a combined 30 mins, the system still responded quick enough before the next disturbance and thus was essentially unaffected.

3 Design Part 2 - Feedforward Control

Investigate the use of feedforward control, in which case the subject could announce meals up to 15 min in advance. Once again, your scheme should be robust to the above uncertainty.

3.1 Gain feedforward

To add feedforward control, we consider the most basic scheme (no feedback) as shown in Figure 7.

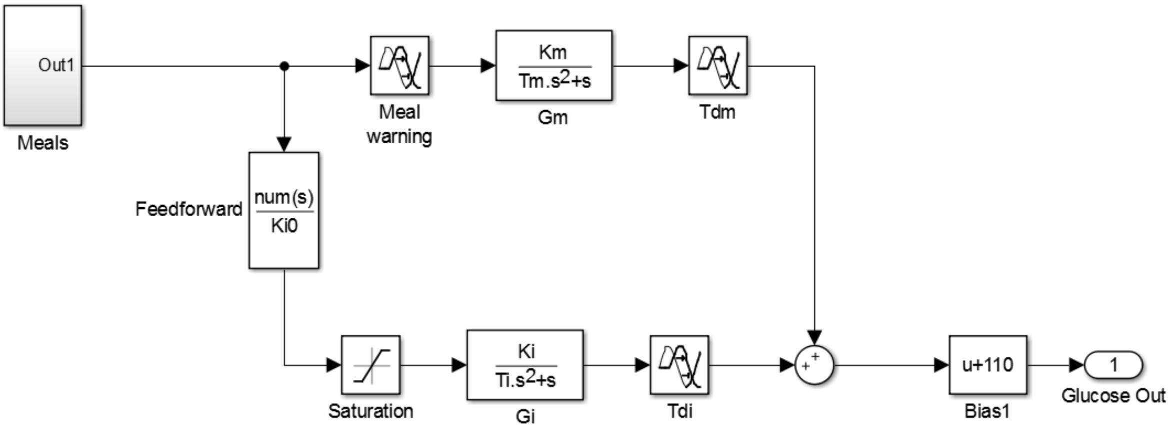

Figure 7: Block diagram for the most simple feedforward scheme, with no feedback control connected.

Here, the output $Y(s)$ is given by:

$$Y(s) = G_I(s)F(s)U_M(s) + G_M(s)U_M(s)$$

where $G_I(s)$ and $G_M(s)$ are the two plants described in the Section 1 model, and $F(s)$ is the feedforward block.

We can then solve:

$$G_I(s)F(s) + G_M(s) = 0$$

$$F(s) = -\frac{G_M(s)}{G_I(s)} = -\frac{K_M(\tau_I s + 1)}{K_I(\tau_M s + 1)} \approx -\frac{K_M}{K_I}$$

to get the ideal, gain only, feedforward block. Also, as given in the problem description a meal warning block was added. This was chosen to have a time constant of $\tau_{warning} = 10$ mins, to match the constants τ_{DI} and τ_{DM} which are 25 and 15 mins respectively.

This basic design was simulated and the results are shown in Figure 8. The system performs very well, even better than the best modified Smith predictor scheme for Q-design from Section 2.1. The patient only enters the hypoglycemia range for a very short time (approx 120 mins after dinner), and never after breakfast or lunch.

The problem with this scheme is that it is not robust, especially to uncertainty in K_I. This is shown in Figure 9. For both extreme values of K_I, the scheme has no control and the patient goes intensely hyper- or hypoglycemic. As there is no feedback, the simple gain feedforward is administering too much insulin every time (when K_I is bigger than expected) or too little insulin every time (when K_I is smaller than expected) leading to this non-robust response.

To fix this robustness, a feedback scheme was added to the already semi-working feedforward scheme. It is clear the feedforward already provides almost enough (or too much) insulin every time, the proposed feedback scheme should only control on the amount of insulin that the feedforward is missing. To achieve this, an ideal reference signal was created (see Figure 8) and subtracted from the actual output at $Y(s)$. This was fed back

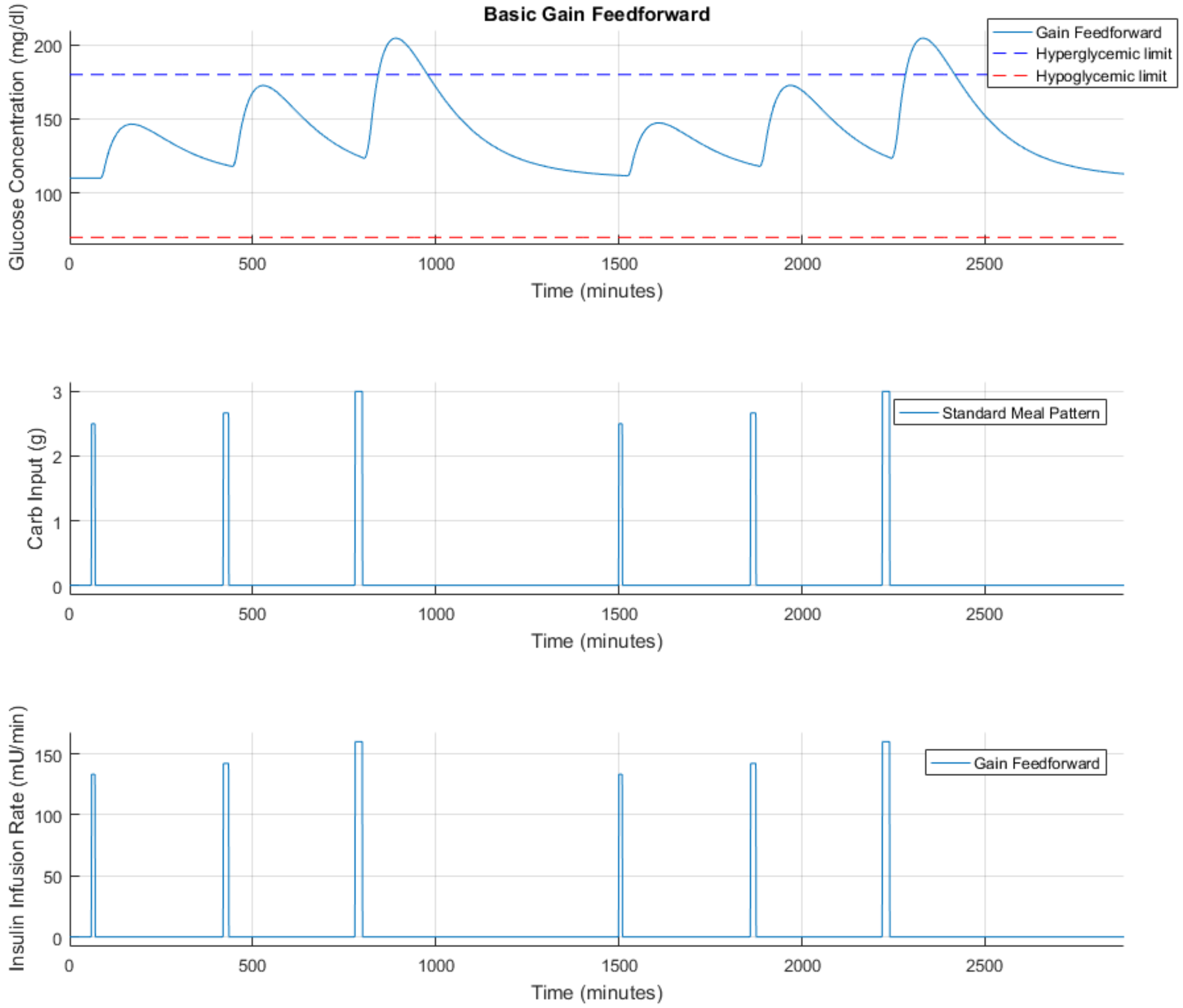

Figure 8: Simulation results for basic gain feedforward block $F = -\frac{K_M}{K_I}$. The scheme performs very well when there is no uncertainty in parameters.

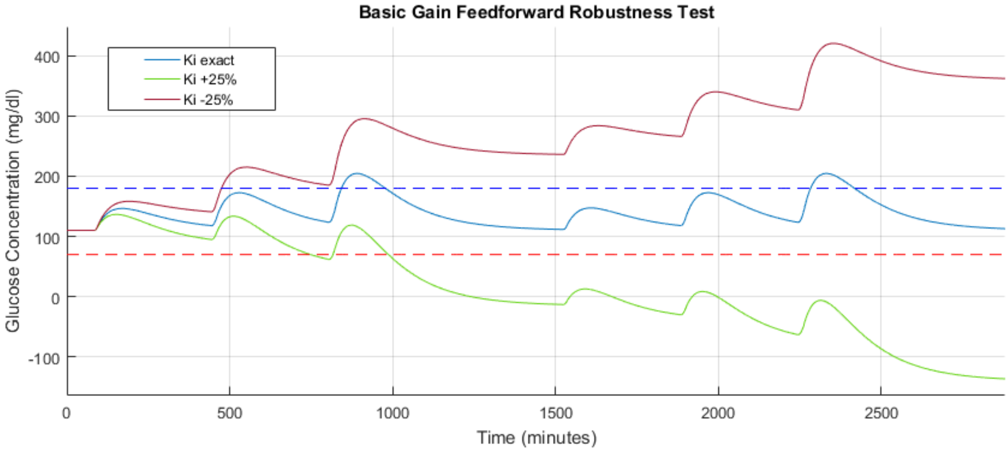

Figure 9: Simulation results for basic gain feedforward block $F = -\frac{K_M}{K_I}$ with variation in the K_I parameter

into a controller which would thus only see the error that the feedforward is missing, and be able to compensate for this. A block diagram of the scheme is shown in Figure 10.

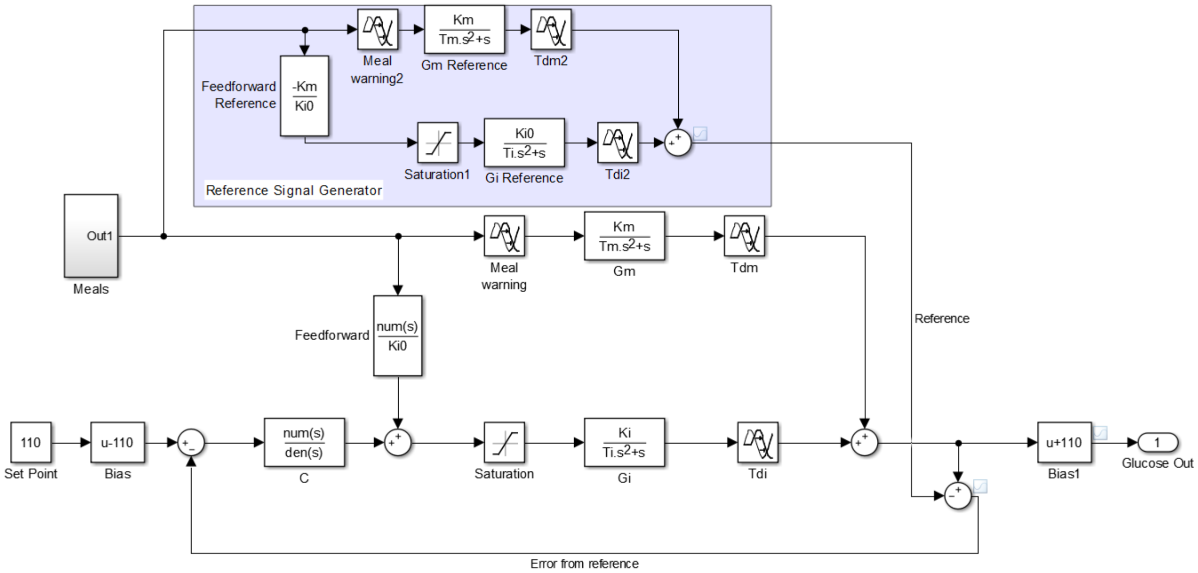

Figure 10: Block diagram for the robust feedforward scheme. Feedback is connected to regulate the error between what the feedforward provides and the expected (ideal) signal is.

A key design change was made to the feedforward block, F(s). The gain was scaled to always under-administer insulin for all potential values of K_I. This meant that the new feedforward block had

$$F = -\frac{K_M}{1.25K_I}$$

as this would administer perfect amount of insulin when K_I was at its highest value, and slightly under-administer when K_I was as expected. The reason for this is because the controller could never provide a negative insulin infusion rate. If the feedforward block was over-administering insulin (see the green characteristic in Figure 9), the controller would not be able to take any away, and the patient would still have an uncontrollable hypoglycemic episode. Thus, to be able to have the controller be effective across all values of K_I, the feedforward block needed to be scaled so there was never an overdose of insulin being provided.

To design the controller, Q-design for the undelayed plant was used similarly to Section 2.1. This gave a controller of the form:

$$\bar{C}(s) = \frac{\bar{Q}}{1 - \bar{F_Q}} = \frac{\omega_N^2(\tau_I s + 1)}{K_I(s + 2\zeta\omega_N)}$$

where $\bar{Q}$ and $\bar{F_Q}$ were as before. The parameters from before were also used, these were:

$$\omega_N = 1/14, \zeta = 4.75$$

3.2 Robustness Testing

Robustness testing for the feedforward control scheme was checked against uncertainty in K_I, meal size and meal timing.

3.2.1 K_I Uncertainty

The results of simulations for each of the extreme values of K_I are shown in Figure 11.

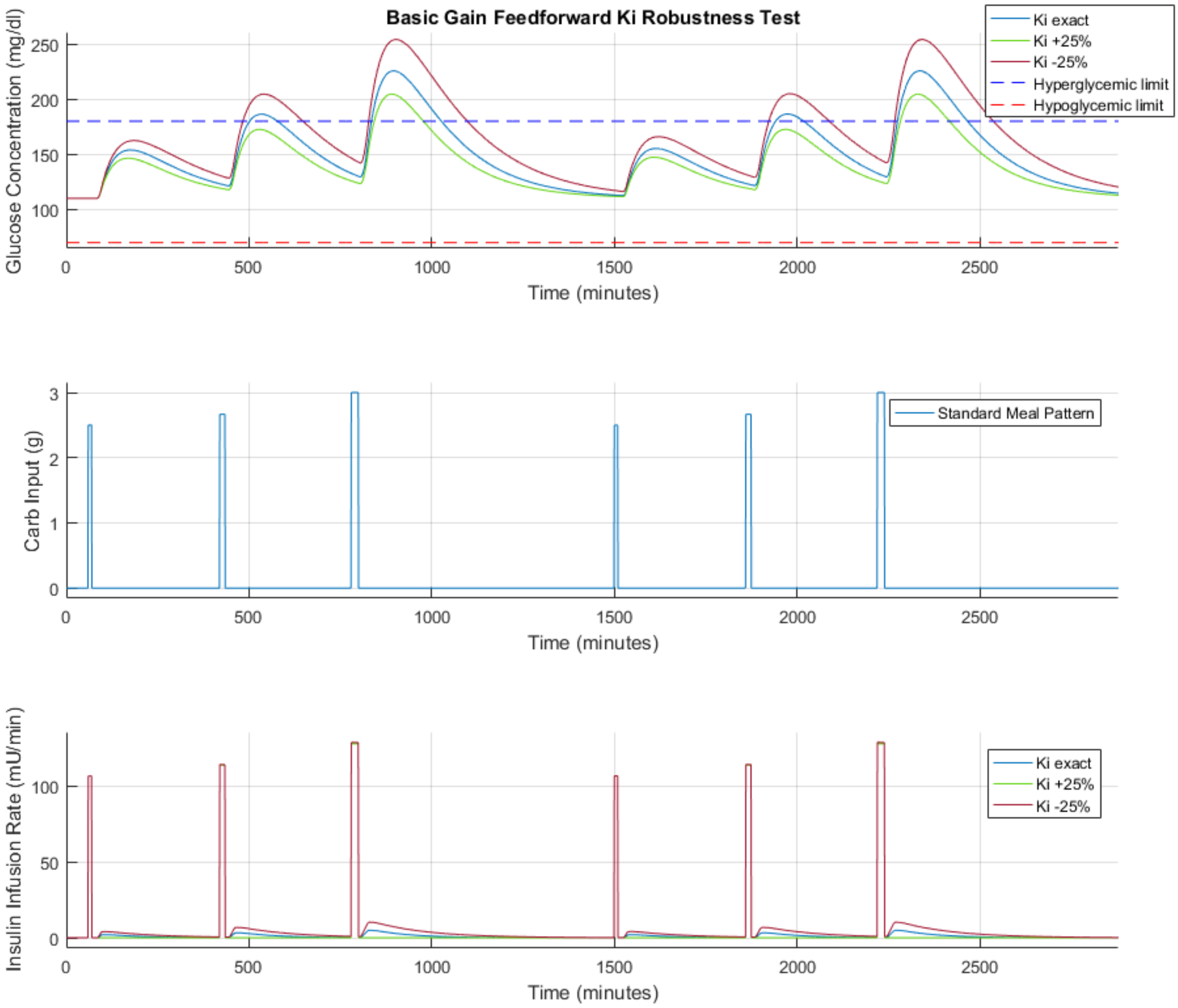

Figure 11: Simulation results for the robust feedforward scheme with ±25% uncertainty in K_I. Note that the best performance comes when K_I is maximized as this is when the feedforward is providing ideal insulin response without any need for the feedback controller to kick in.

The scheme is now very robust under this uncertainty in K_I and does not lose control of the glucose concentration level in the patient. On the high end, the episodes of hyperglycemia are only extended minimally and there is still no hypoglycemia episode. The setpoint does not drift, and overall this scheme is much more robust than the simple feedback scheme from part 2.1. The best performance comes when K_I is at its maximum value, as this is the value for which the feedback is providing an ideal insulin response without need for the feedback controller to kick in.

3.2.2 Meal Size Uncertainty

The results of simulations for each of the extreme values of meal size are shown in Figure 12.

The robustness of the scheme is worse with uncertainty in meal size in comparison to K_I. At the highest meal size, the controller is not aggressive enough and the set point

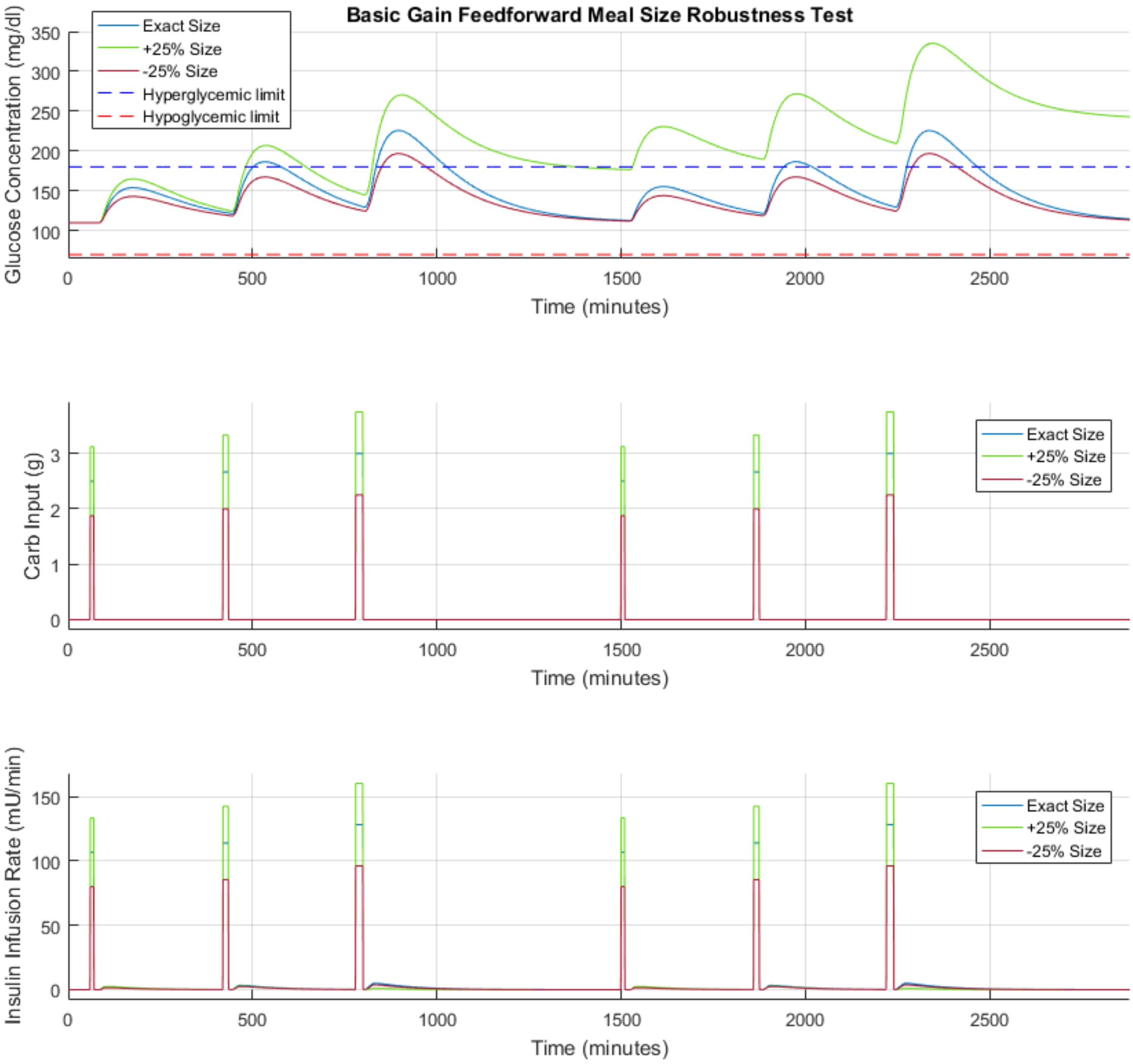

Figure 12: Simulation results for the robust feedforward scheme with ±25% uncertainty in meal size.

does drift leading to the patient ending the 48hrs on a hyperglycemic bender. After the first 24hrs, he begins the next day just inside the hyperglycemic range. For the lower meal sizes it does very well, of course. To solve the hyperglycemic problem, it would not be unreasonable to expect the patient to read his own blood sugar every 24hrs (as opposed to much more frequently), and then administer additional insulin as required. This would be especially true if the patient felt like he had been having abnormally large meals the previous day. The controller does do a decent job and this would only be a daily thing in the morning, which would be already saving the patient lots of time.

3.2.3 Meal Timing Uncertainty

The results of simulations for various meal timings, shifted by ±15 mins, are shown in Figure 13. As before, the simulated schedules that were predicted to give the most disruptive behaviour were ones where different meals were shifted closer relative to each other.

Overall, as with the Smith predictor, the feedforward control scheme is very robust to meal timing uncertainty. Even with meals being brought closer together by a combined

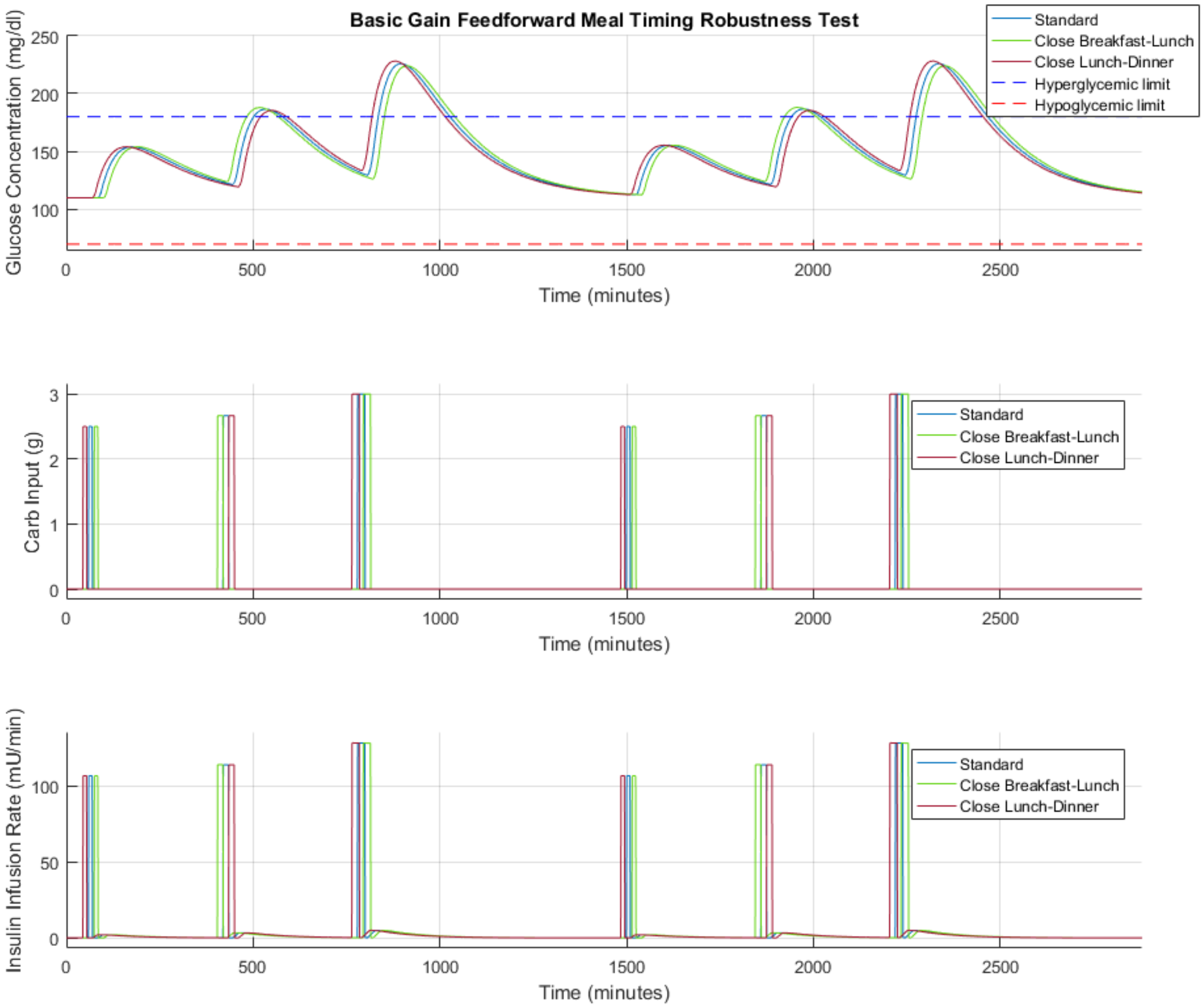

Figure 13: Simulation results for the robust feedforward scheme under various combinations of ±15 min uncertainty in meal timing. "Close Breakfast-Lunch" refers to breakfast being shifted later by 15 mins and lunch being shifted earlier by 15 mins. Similarly, "Close Lunch-Dinner" refers to lunch being shifted later and dinner being shifted earlier.

30 mins, the system still responded quick enough before the next disturbance and thus was essentially unaffected.

4 Conclusion

This report outlines multiple control scheme designs that could be used as an "artificial pancreas" to regulate blood glucose levels in a Type-1 diabetic patient. The control schemes center around the administration of insulin to the body, while the glucose levels are being disturbed by carbohydrate intakes during 3 meals per day. The designs were simulated in Simulink and tested for robustness across a wide uncertainty range of plant gain, K_I, as well as disturbance size and timing. The two primary control schemes used were a modified Smith predictor for Q-design (Section 2), and a modified gain feedforward scheme with reference error feedback control (Section 3).

Both schemes ended up performing very well. Under ideal conditions (no parameter uncertainty) both schemes kept the patient at healthy blood glucose concentrations and never entered a hypoglycemic range, while minimizing the hyperglycemic episodes to approximately 120 mins at most. Further, they were both extremely robust to meal timing uncertainty.

The modified Smith predictor was characterized by relatively short hyperglycemic episodes after lunch and dinner (but not breakfast). This scheme did encounter some robustness issues however with uncertainty in K_I. With the largest value of K_I, the controller was slightly too aggressive and ended up finishing the full 48hrs with the patient in a hypoglycemic episode.

The modified feedforward design improved on the first scheme. Under ideal conditions, the patient had a hyperglycemic episode after dinner only, and the maximum blood glucose concentration level was lower than for the previous scheme. Further, it was very robust to uncertainty in K_I. Where it did have issues was when dealing with uncertainty in meal size. With the biggest meal sizes, the controller was not aggressive enough and the patient ended up in a hyperglycemic range after 24hrs.

Overall, the feedforward scheme ended up producing better results than the modified Smith predictor feedback scheme. In designing both schemes, there was a constant balance between designing a controller that was too aggressive (saturating the insulin infusion rate and not being able to compensate with a negative rate, thus leading the patient to hypoglycemia) and too passive (not administering enough and having a drifting set point towards hyperglycemia). This project hopefully provides a starting point for anyone wishing to attempt to create an "artificial pancreas" in the near future.

A Appendix - MATLAB Code

The following MATLAB scripts were primarily used in setting up the simulations, which were run in Simulink and thus not able to be copied as an appendix directly. For access to all MATLAB/Simulink files visit `https://github.com/benmattison/ELEC441`.

```
%% DiabetesVariables.m
% This script defines the parameters of the model. The values are read from
% the workspace and used in Simulink for all simulations. This script is
% useful for for modifying the uncertainties on various parameters to test
% for robustness.

Ki_uncertainty = 0;
mealsize_uncertainty = 0;

Km = 4;
Tm = 50;
Tdm = 15;
Ki = -0.075*(1+Ki_uncertainty);
Ti = 140;
Tdi = 25;
break_amt = 25/10*(1+mealsize_uncertainty);
lunch_amt = 40/15*(1+mealsize_uncertainty);
dinner_amt = 60/20*(1+mealsize_uncertainty);

```

```
break_timing = 0;
lunch_timing = 0;
dinner_timing = 0;
```

```
%% Qdesign.m
% This script generates a Q-designed controller based on input
% parameters Wn and zeta. The data from this script is then read directly
% into Simulink blocks for use in simulations.

%% Design Variables
Ki0 = -0.075; % need to design for no uncertainty

syms x

%% Best Params
% Wn = 0.09;
% zeta = 6;

Wn = 1/14;
zeta = 4.75;

%% Extra integrator param
alpha = Wn^2*(2*zeta+1)/Wn^3;

%% no intergrator
num = expand(Wn^2*(Ti*x+1)*x);
den = expand(Ki0*(x^2+2*Wn*zeta*x+Wn^2));
numc = [0 double(coeffs(num))];
denc = double(coeffs(den));

%% with integrator
inum = expand(Wn^3*(Ti*x+1)*x*(alpha*x+1));
iden = expand(Ki0*(x^2+2*Wn*zeta*x+Wn^2)*(x+Wn));
inumc = [0 double(coeffs(inum))];
idenc = double(coeffs(iden));

%% controller
Cnum = expand(Wn^2*(Ti*x+1));
Cden = expand(Ki0*(x+2*zeta*Wn));
Cnumc = double(coeffs(Cnum));
Cdenc = double(coeffs(Cden));
```

```
%% plotSetup.m
% This script modifies the figure output from "Simulation Data Inspector"
% in Simulink. The data inspector has an option "Send to Figure" which is
% used before running this script.
```

```

figure(1);
subplot(3,1,1);
title('Basic Gain Feedforward Meal Timing Robustness Test');
xlabel('Time (minutes)');
ylabel('Glucose Concentration (mg/dl)');
hold on;
time = linspace(0,48*60,1000);
hyper = linspace(180,180,1000);
hypo = linspace(70,70,1000);
plot(time,hyper,'b--');
plot(time,hypo,'r--');
ylim([65 250]);
legend('Standard','Close Breakfast-Lunch','Close Lunch-Dinner','Hyperglycemic limit','Hypoglycemic limit');

subplot(3,1,2);
xlabel('Time (minutes)');
ylabel('Carb Input (g)');
legend('Standard','Close Breakfast-Lunch','Close Lunch-Dinner');

subplot(3,1,3);
xlabel('Time (minutes)');
ylabel('Insulin Infusion Rate (mU/min)');
legend('Standard','Close Breakfast-Lunch','Close Lunch-Dinner');
```