Tab 1

Lab Report 3

Prepared by: Heidi Eren

Date: 11/2/25

Executive summary (5 pts):

Describe what you are trying to accomplish/demonstrate. What is the purpose of this test?

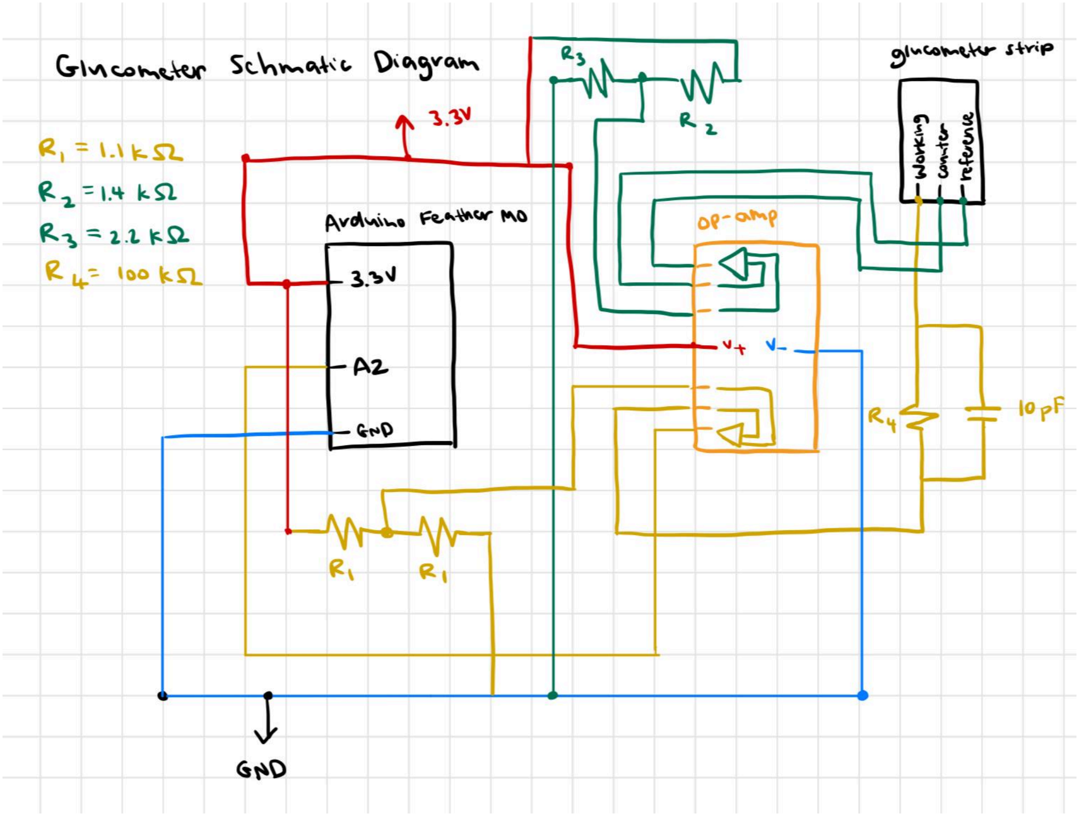
The purpose of this experiment is to demonstrate an understanding of food absorption and glucose-insulin dynamics between normal and diabetic subjects by modelling a system of three nonlinear ordinary differential equations and using a glucometer to measure glucose concentration. By curve fitting glucose concentration and modelling its closed loop system with a PID (Proportional-Integer-Derivative) controller, the dynamics model can be simulated to calculate the glucose intake D(t), glucose absorption Q(t), and blood glucose concentration G(t) over time. By modeling various scenarios with bolus (impulse function) or basal rate added to the meal intake, the optimal PID values can be determined empirically to simulate the closed loop feedback control system of the glucose-insulin dynamics.

Tab 2

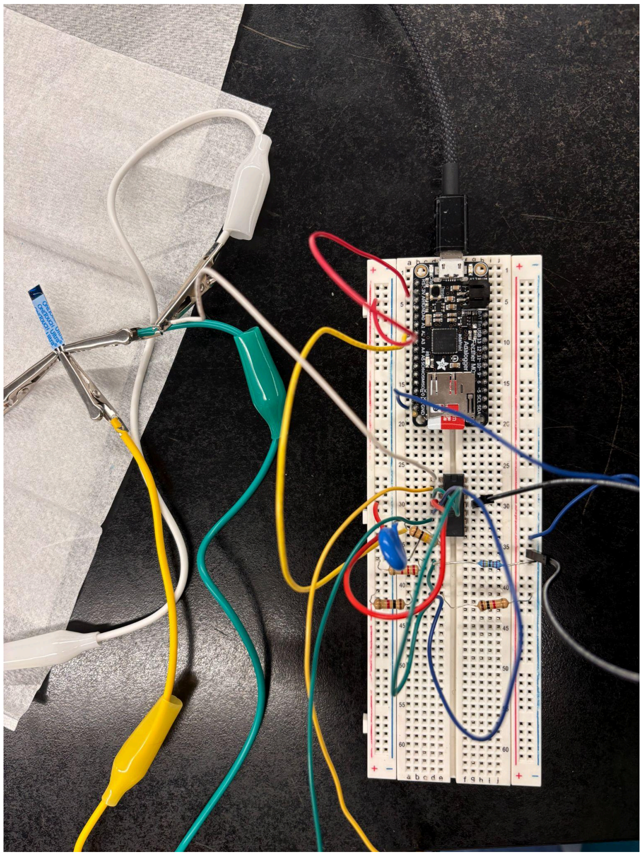
Procedure:

1. Build a circuit to measure glucose concentration. Use the implementation amplifier and capacitors to convert current into voltage.

2. **Sketch a circuit diagram of the circuit (5 pts):**

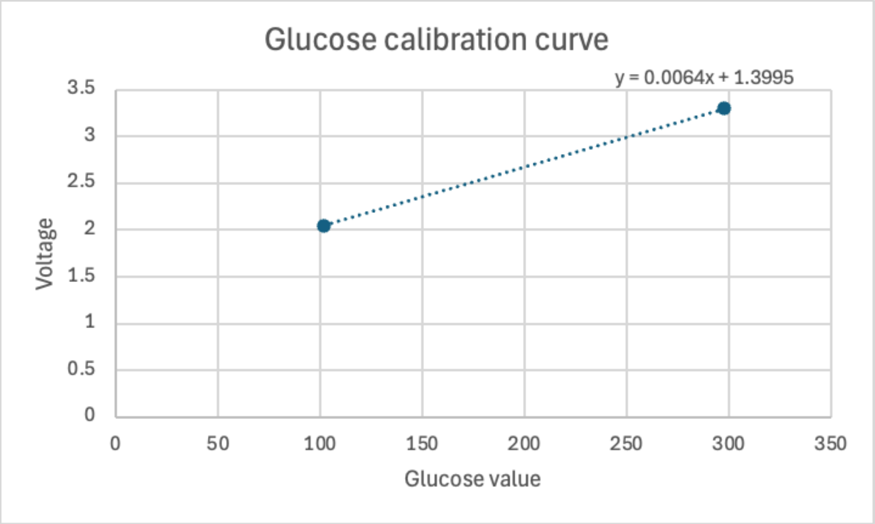


3. **Take a photo of the circuit (3 pts):**



4. Use the Arduino IDE to make two measurements of voltage at two different glucose concentrations: “medium” and “high”. (**You will upload your final code to Canvas**).

5. Using Excel or another program, calculate the linear relationship between voltage and concentration. Add this relationship to the Arduino IDE. (**Upload any files, i.e. .xls, that were used to calculate these coefficients) (4 pts)**.



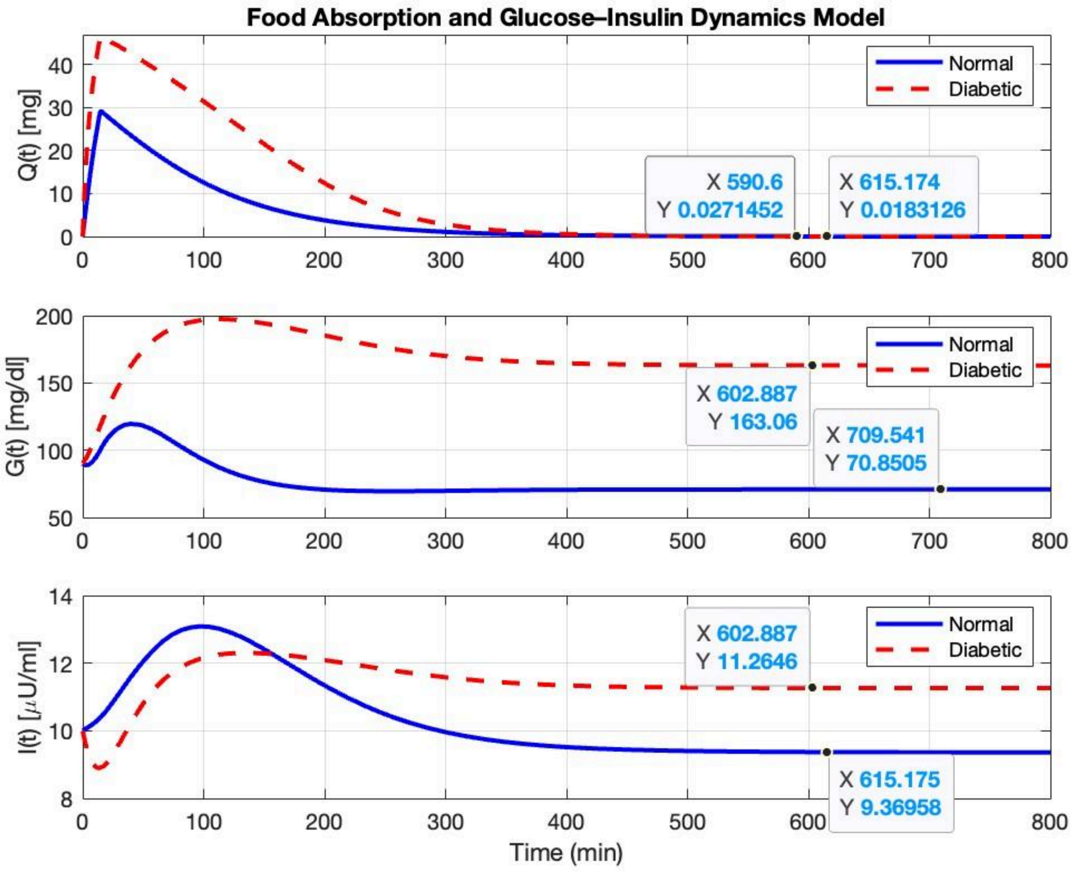
For the remainder - no circuit required

6. In the Arduino IDE, implement the differential equations modeling intestinal glucose, blood glucose, and insulin production.

7. In the Arduino IDE, add the parameters for the “non-diabetic” and “diabetic” models and identify the initial conditions for both (6 pts):

	Q(0)	**G(0)**	**I(0)**
Non-diabetic	0	70.85	9.36
Diabetic	0	163.01	11.26

8. Examine (and submit) the glucose and insulin concentration curves for the “non-diabetic” and “diabetic” models after a “meal” lasting 15 minutes and containing 15g of glucose (*watch your units!*) (**You will submit your code and saved plots on Canvas) (3 pts each)**

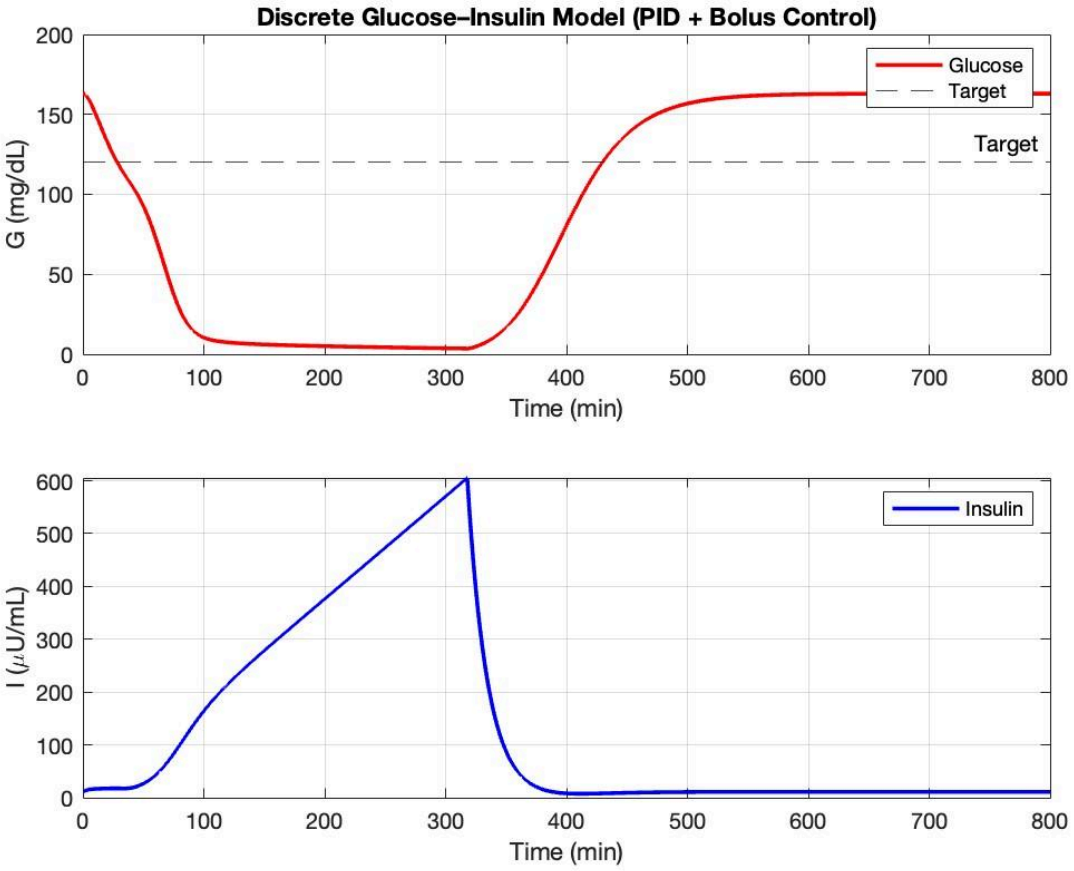


9. Using the “diabetic” model, describe the effects on the concentration of glucose by adding (2 pts each):

1 bolus (15 units) at the start of the meal	Q(t) → 0 mg G(t) → 163 mg/dl I(t) → 11.26 μU/ml There are no significant changes in the concentration of glucose as it increases to 163 mg/dl and stays constant.
1 bolus (15 units) after the meal	Same as the previous scenario; no final changes in the glucose concentration at 163 mg/dl. There is a spike (to 189 mg/dl) and dip after the meal intake but with the bolus input the glucose concentration increases to 163 mg/dl and settles from there.
1 bolus before (15 units) and one “corrective” bolus (15 units) after the meal (Basal rate = continuous low flow of insulin)	The bolus before the meal and the corrective bolus after the meal cause the glucose concentration to spike during this interval to 45 mg/dl then drop and gradually increase to 163 mg/dl, settling from there.
1 bolus before the meal (15 units), plus a “basal” rate (0.5 unit)	The same fluctuations as the previous scenario (but higher spike), where the meal intake and bolus impulse spikes the glucose at t=60 to 38.9 mg/dl and then drops and gradually increases to a constant rate of 86.16 mg/dl.

10. Using the “diabetic” model, implement a PID controller to release insulin in response to changes in glucose. Determine (using trial and error) an optimal value for each of the P, I, D parameters.
    P = 0.3, I = 0.005, D = 0.05

11. **Bonus (5 pts):** Many insulin pumps can only store a limited amount of insulin. Assume your pump can store 50 units of insulin. Determine (using trial and error) the optimal parameters for the above controller subject to that constraint (e.g. once sum(addedInsulin) = 50, Insulin = 0). Describe the process you used to get to these values and **submit the final plots of G and I.**
    P = 0.080, I = 0.0010, D = 0.020
    I tested different combinations of PID gains and ran the closed loop ODE simulation (meal intake + insulin pump controlled by PID + bolus injection + insulin reservoir with a limit on the total insulin used to less than 50 units). I added overshoot and steady state error print statements so that I can gauge the performance of the PID combinations. The figure also helps in visualizing the oscillations or overshoot in the glucose concentration over time.


Discrete Glucose–Insulin Model (PID + Bolus Control)
G (mg/dL)
200
150
100
50
0
Glucose
Target
Target
0
100
200
300
400
500
600
700
800
Time (min)
I (μU/mL)
600
500
400
300
200
100
0
Insulin
0
100
200
300
400
500
600
700
800
Time (min)

Tab 3

Notes and recommendations:

1. Describe the response of the glucose strip to the application of the glucose solution. How do we measure a chemical process? (3 pts)

The glucose strip converts the chemical reaction into an electrical signal by measuring and quantifying how quickly the reaction occurred. In the lab, the higher glucose means more electrons flowing which results in a higher current. That current is then converted to a voltage by the op-amplifier, which then is read as an analog voltage in Arduino. A chemical process (enzymatic oxidation of glucose) is measured by coupling it to a physical change that is quantified, known as a transduction. In this case, the transduction mechanism was an electrochemical process where the charge transfer is shown through a change in voltage (proportional to the current).

2. Compare and contrast the “non-diabetic” and “diabetic” models including: steady state values, maximum values, change from baseline, rate of change, etc. (5 pts)

The non-diabetic models and diabetic models had different peak glucose (overshoot) at 120.5 mg/dL and 198 mg/dL respectively. There was a much higher and faster spike in the diabetic model. The steady state for the nondiabetic was around 70.9 mg/dL while the steady state for the diabetic model was around 163 mg/dL. The baseline after the meal intake took around 97 min for the nondiabetic while the diabetic model did not return to within 5 mg/dL of the first 90 min of the window. The rate of change dG/dt for the non diabetic model peaked at 1.70 mg/dL/min which was lower than the diabetic slope at around 2.23 mg/dL/min.
The diabetic and non-diabetic models are similar in the sense that they have the same overall shape or pattern after a meal. There is a rapid glucose increase after meal intake, as the insulin release is proportional to the glucose concentration, the glucose concentration decreases as the insulin increases. As a result, all systems (D, G, I) reached a steady state over time for both models.

3. How did your insulin bolus(es) affect blood sugar in the “diabetic” model? Did the closed loop controller improve the “pump” performance? If so, how? (5 pts)

The insulin bolus affected the blood sugar in the model at each of the scenarios. When the bolus was inputted at the start of the meal, the glucose peak decreased yet reached the target glucose faster. This means the insulin increase in the plasma resulted in less glucose being absorbed in the blood. The bolus after the meal conveyed minimal differences, which may be due to the glucose already being in the system before the insulin rises. The bolus at the start and corrective bolus had a lower glucose concentration and fast return to the baseline glucose concentration. The first pulse stopped the spike in glucose and the second pulse cleared the residual glucose. The bolus at the start and basal rate had a smooth stable curve which conveys how the basal rate can maintain clearance while the bolus handles the load. The bolus in general helped the system reach normal glucose concentration behavior that the open loop system was not able to do. Yes, the closed loop controller helped improve the pump performance since the P provided a rapid bolus-like response (by correcting the proportion of glucose error), the I adjusted the basal rate for long term offset (by integrating the residual rate over time), and the D minimized overshoot and oscillation (by adding predictive damping).

4. Describe the process you used to “tune” the PID controller. What had the largest impact? What kind of performance were you able to achieve? (5 pts)

In order to tune the PID controller, not only did I have to establish a baseline (15 g over 15 min), but had to ensure the steady state and overshoot was within optimal range. The PID values can be tuned by increasing the P (Kp) without oscillations, adding I (Ki) to minimize steady state error, and adding D (Kd) to reduce overshooting. I also included a threshold parameter that ensured that the derivative would not reach a negative value. The largest impacts were changing the Kp which dominated the peak of the graph and Kd which changed the damping (overshoot). The performance I achieved was a smooth curve where the peak of the glucose concentration was within range (over target) and setting time and steady state error to a constant rate. The oscillations were minimal and the trends of the figure aligned with the input scenarios.

5. Comment on the use of physiological control loops in medical devices (generally). What are the risks? What are the benefits? (5 pts)

The common use of using physiological control loops in medical devices is to help increase efficiency, reduce costs, and minimize errors during development. They help in creating a real time feedback system that adjusts based on the patient or physiological variables that enhance the care for the patient.

In general, there are risks associated with the variability in the patient response or external environment, noise in sensors, calibration requirements, and automation bias with these systems. For open loop control systems, although they are easy to implement and robust against noise, they can be difficult to adjust to external perturbations and quite inflexible. For closed loop control systems, they can easily maintain homeostasis against external noise with reduced cognitive workload and improved response time. However, they are often complex to implement and latency/error can result in high error rates.