# PID-Based Glucose Regulation in Simulink

## 1. Introduction

### 1.1 Problem Statement:

Diabetes mellitus is a chronic metabolic disorder characterized by the body’s inability to produce or effectively use insulin, resulting in elevated blood glucose levels. Persistent hyperglycemia can lead to serious complications including cardiovascular disease, neuropathy, kidney failure, and vision impairment. In healthy individuals, the pancreas continuously monitors blood glucose and secretes insulin to maintain levels within a narrow physiological range (typically 70–110 mg/dL during fasting).

However, in diabetic patients, this regulatory mechanism is impaired. Type 1 diabetics, in particular, require external insulin delivery — often through subcutaneous insulin injections or automated insulin pumps. The challenge lies in determining how much insulin should be delivered and when, especially after events like meals where blood glucose can spike rapidly.

Manual insulin dosing is error-prone and cannot adapt in real-time. Therefore, a control system is essential to maintain blood glucose near a safe setpoint while avoiding dangerous drops (hypoglycemia) or spikes (hyperglycemia).

### 1.2 Objective:

The primary objective of this project is to model, simulate, and analyze a closed-loop blood glucose regulation system using a Proportional-Integral-Derivative (PID) controller in MATLAB Simulink. The system is designed to mimic the physiological feedback mechanism responsible for maintaining normal blood glucose levels in the human body, particularly in response to postprandial (after-meal) glucose spikes.

The specific goals include:

- Developing a first-order system model to represent the dynamics of blood glucose levels in response to insulin.
- Designing a manually implemented PID controller to regulate glucose levels based on feedback error.
- Simulating the effects of a sudden glucose increase (simulated as a step input) and evaluating the controller’s ability to bring the glucose level back to the desired setpoint.
- Analyzing system performance using key metrics such as peak glucose, rise time, overshoot, and settling time.

### 1.3 Relevance of Project

Modern diabetes management increasingly relies on automated insulin delivery systems, such as insulin pumps and closed-loop artificial pancreas devices. These systems must respond accurately and efficiently to rapidly changing glucose levels, particularly after meals, during exercise, or under stress.

Simulating glucose-insulin dynamics using MATLAB Simulink enables students, researchers, and engineers to test control strategies without risking patient safety. It also allows for rapid experimentation with system parameters and controller designs in a controlled virtual environment. The insights gained from such simulations are directly applicable to the development of embedded software and hardware in biomedical devices.

This study, therefore, bridges biomedical physiology and control engineering, offering valuable experience in designing systems that can improve the quality of life for millions of diabetic patients worldwide.

## 2. Literature Review:

While this project demonstrates a simplified glucose regulation system using a basic manually tuned PID controller in Simulink, several advanced research efforts have explored this topic in much greater technical depth. These works often incorporate nonlinear dynamics, fuzzy logic, pharmacokinetic modeling, and clinical deployment scenarios. Below is a review of three key studies that highlight the breadth of current research in this field:

### 2.1 Mehta (2018): PID and Fuzzy Logic Simulation in Simulink [1]

Pratik Mehta's thesis implements both PID and fuzzy logic controllers using the Bergman Minimal Model to simulate glucose-insulin dynamics. The system is built entirely in Simulink, and gains are tuned using the Internal Model Control (IMC) method for better performance. Multiple control strategies are compared across simulation runs

### 2.2 Faiz ul Hassan (2017): Nonlinear Control with Backstepping [2]

This research paper designs a nonlinear backstepping controller for blood glucose regulation. It models complex disturbances like meals and exercise and applies robust control techniques to track setpoints under highly variable conditions using the Bergman model in Simulink.

### 2.3 Rayhan A. Lal. (2019): Real-World Artificial Pancreas Systems [3]

Lal et al. present a broad clinical overview of artificial pancreas technologies, including PID and MPC algorithms implemented in early prototypes using Simulink. The paper discusses challenges in real patient environments, such as sensor errors, insulin delays, and human factors

While the aforementioned studies incorporate advanced modeling techniques such as nonlinear control, fuzzy logic, and pharmacokinetic simulations, the current project adopts a more fundamental approach to glucose regulation. A first-order system model was employed alongside a manually implemented PID controller to simulate the core principles of closed-loop feedback control. Although the model does not account for real-world complexities like insulin absorption delays or patient-specific variability, it effectively demonstrates the essential dynamics of glucose stabilization

## 3. Mathematical Modelling

### 3.1 Glucose System Dynamics and Transfer Function

To model the human body's glucose regulation mechanism, we used a first-order linear system to represent the effect of insulin on glucose levels. This simplification is common in control applications and allows for easier analysis and controller design.

The system is defined by the following transfer function:

$$G(s)\ \frac{1}{30s + 1}$$

Here, the constant 30 represents the time constant ($\tau$) of the system, indicating that the glucose level takes approximately 30 seconds to respond significantly to a control action. This model assumes that insulin affects glucose linearly and immediately, which simplifies the real-world physiology.

### 3.2 PID Controller and Control Law

A Proportional-Integral-Derivative (PID) controller was used to regulate glucose. The control law is given by:

$$u(t) = Kp.\ e(t) + Ki \int e(t)\ dt + Kd\ .\ \frac{de(t)}{dt}$$

Where:

- u(t): control signal (simulated insulin output)
- e(t): error = setpoint − actual glucose
- Kp =0.8,
- Ki =0.05,
- Kd =0.2, these are the the proportional, integral, and derivative gains

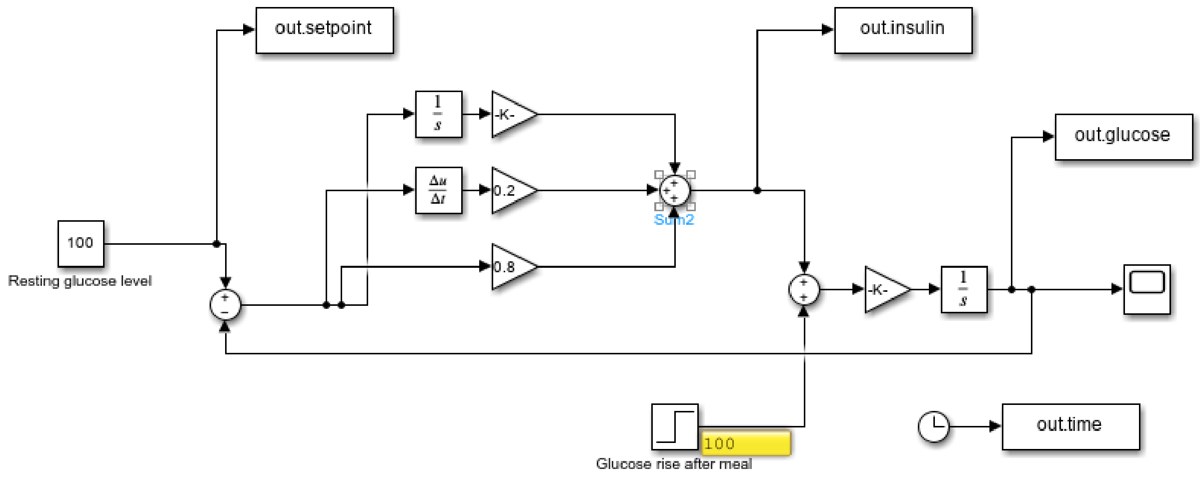
These values were selected based on iterative simulation and tuning to balance overshoot, settling time, and rise time. The controller's purpose is to react to deviations in blood glucose and return the system to a healthy baseline (e.g., 100 mg/dL).

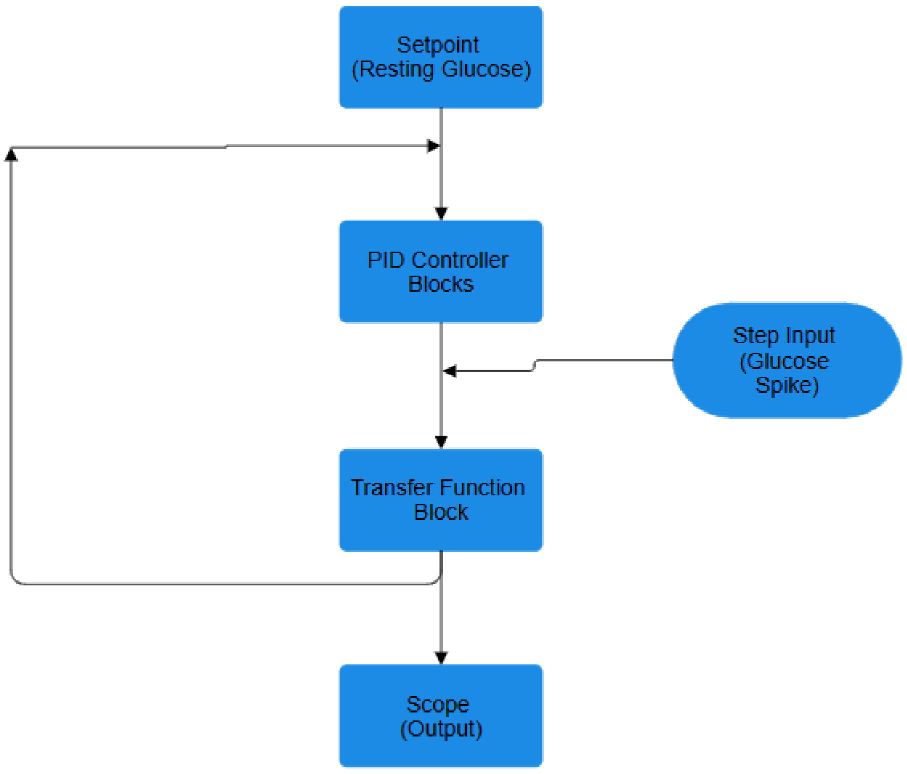
Proportional (Kp):
Generates a control signal proportional to the current error, providing an immediate response.

Integral (Ki):
Accumulates past error over time to eliminate steady-state deviation from the setpoint.

Derivative (Kd):
Predicts future error trends by measuring the rate of change, helping reduce overshoot and oscillation.

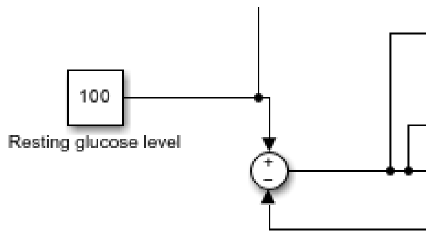
## 4. Simulink Design:





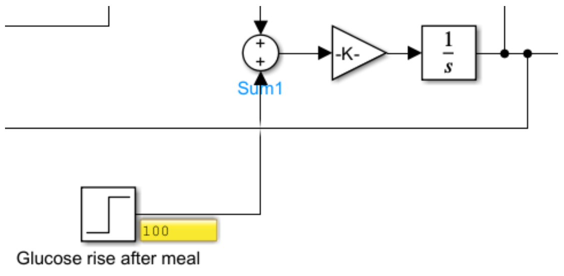
## 4.1 Setpoint, Step Input, and Initial Glucose

To simulate blood glucose regulation, a Constant block was used to define the desired glucose level (setpoint) at 100 mg/dL. This block feeds into the first Sum block, where the difference between actual glucose and the setpoint is calculated to generate the error signal for the PID controller.



A Step block was used to represent a sudden glucose increase, such as what occurs after a meal. The step was configured to apply a value of +100 mg/dL at t = 100 seconds, creating a sharp rise in blood glucose that the PID controller must respond to. This models a typical postprandial glucose spike.

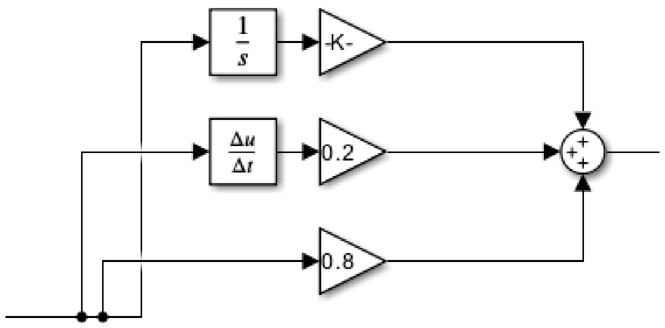
Step function simulating glucose rise after a meal:



### 4.2 PID Controller Implementation:

Instead of using Simulink's built-in PID Controller block, the PID controller in this project was constructed manually using fundamental Simulink blocks. This approach offers full visibility into how each component of the PID algorithm functions, enhancing both understanding and tunability.

The controller was designed to respond to the error signal, calculated as the difference between the setpoint and the actual glucose level. This error is passed into three separate branches representing the proportional, integral, and derivative terms.



#### 4.2.1 Proportional Component:

The proportional branch responds directly to the current magnitude of the error. It was implemented using a Gain block connected directly to the error signal. The block multiplies the real-time error by the proportional gain Kp=0.8.

This component provides immediate correction: the greater the deviation from the setpoint, the larger the control signal. In the simulation, this meant that a sudden rise in glucose (due to the meal step input) resulted in a rapid insulin injection proportional to the size of the spike.

### 4.2.2 Integral Component:

The integral branch addresses accumulated past error over time. It was created by feeding the error into an Integrator block, which sums up the error continuously. The output of this block was then passed through a Gain block set to Ki=0.05.

This term ensures that even small, long-standing deviations are eventually corrected by building up control effort until the error is driven to zero. In terms of glucose regulation, this prevents the system from stabilizing slightly above or below the desired level — it ensures full convergence back to the setpoint.

### 4.2.3 Derivative Component:

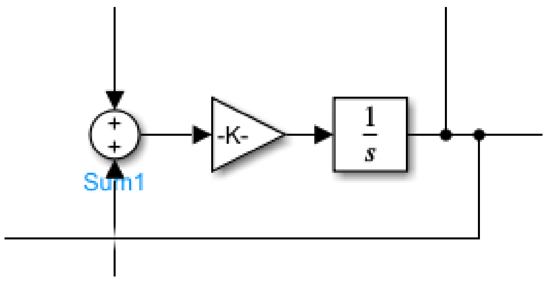
The derivative branch predicts future error behavior based on how fast the error is changing. It was implemented using a Derivative block applied to the error signal, followed by a Gain block with a value of Kd=0.2.

This term acts as a stabilizer: if glucose is rapidly rising, the derivative component responds preemptively and increases insulin output before the spike becomes too large

## 4.3 Transfer Function Modelling

n control systems, a transfer function represents how the output of a system responds to an input over time. For this project, the dynamics of the glucose regulation system were modeled using a simplified first-order transfer function:

$$\frac{1}{30s + 1}$$



This function describes a system that reacts gradually to an input with a time constant of 30 seconds, which approximates how glucose levels might respond to insulin over time. It assumes that the glucose system behaves linearly and that insulin has an immediate effect — a simplification made for clarity and educational focus.

### 4.3.1 Implementation:

Instead of using a built-in Transfer Function block, this model was implemented manually in Simulink using a Gain block and an Integrator block. The Gain block was set to 1/30, and its

output was fed into the Integrator block. This structure effectively mimics the behavior of the above transfer function when placed inside a closed-loop feedback system.
In open-loop terms, this configuration represents $\frac{1}{30s}$ which would continuously rise in response to a step input. However, in the full closed-loop system, the negative feedback and the controller combine to form the intended behavior of $\frac{1}{30s + 1}$ , producing a stable, bounded response that returns to the setpoint after a disturbance.

## 4.4 Feedback Looping

1) The setpoint was defined using a Constant block set to 100 mg/dL, representing the ideal glucose level. This value was sent to a Sum block where it was compared against the actual glucose level (the output of the system), resulting in an error signal.

2) The error signal was fed into the manually constructed PID controller, which included three separate paths: a proportional path with a Gain block (Kp = 0.8), an integral path using an Integrator followed by a Gain block (Ki = 0.05), and a derivative path using a Derivative block followed by a Gain block (Kd = 0.2).

3) The outputs of these three paths were combined using another Sum block, producing the total control signal — representing the simulated insulin output from the system.

4) A Step block was added to simulate a glucose spike, such as after a meal. This step signal entered the same Sum block that received the PID output. The Step input entered through the positive terminal, and the PID signal entered through the negative terminal, simulating their opposing effects on glucose levels.

5) The result of this Sum was passed through a Gain block with a value of 1/30, followed by an Integrator block with an initial condition of 100. This modeled the body’s glucose response to insulin and meals as a first-order system.

6) The output of the Integrator, representing the updated blood glucose level, was then fed back into the first Sum block to be continuously compared against the setpoint. This completed the closed-loop feedback system.

## 4.5 Output data logging

To observe the system’s behavior and analyze its performance, key signals were exported from Simulink to the MATLAB workspace using To Workspace blocks. This allowed for custom plotting, performance analysis, and documentation of results outside of Simulink’s built-in Scope.

The main output signal — the simulated glucose level — was connected to a To Workspace block named out.glucose. A Clock block was added to track simulation time and connected to another To Workspace block as out.time. The setpoint signal was also routed to a third To Workspace block as out.setpoint, allowing comparison with the actual glucose response. In some versions, the insulin (PID) output was also saved for additional analysis.

All To Workspace blocks were configured with the ‘Array’ save format, which outputs the data as standard double arrays. This made it simple to use MATLAB’s plotting functions like plot() to visualize and compare signals over time.

Additionally, the final glucose output was connected to a Scope block for quick visual inspection during simulation runs. This helped tune the PID gains interactively while keeping real-time feedback on glucose behavior.

## 5. Results and Graphs:

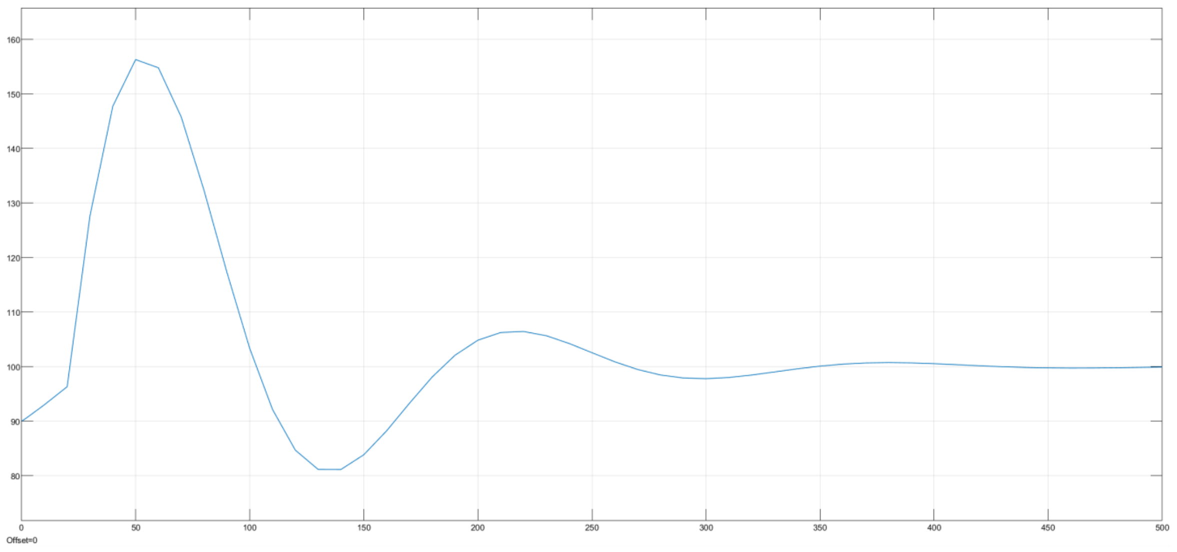
The simulation results clearly demonstrate the effect of the PID controller in regulating blood glucose levels following a sudden increase due to a simulated meal. The glucose spike was introduced at t = 100 seconds using a Step block with a value of +100 mg/dL, causing the glucose to rise rapidly. The PID controller then responded with simulated insulin output to bring the glucose back to the desired setpoint of 100 mg/dL.

### 5.1 Glucose vs Time

A plot of glucose versus time shows a clear response to a step disturbance, representing a meal. Immediately after the disturbance, the glucose level rises significantly, peaking at approximately 168-170 mg/dL around the 40-50 minute mark.

The PID controller begins responding almost immediately after the glucose starts to rise, evident by the rapid change in slope after the peak. It works to bring the glucose level down, initially undershooting the setpoint to around 80 mg/dL between 130-150 minutes, indicating an aggressive correctional action. Following this undershoot, the glucose level then exhibits damped oscillations, characteristic of a well-tuned PID controller bringing the system back to stability. Specifically, it rises again to approximately 105-110 mg/dL around 220-230 minutes, then falls, and gradually settles back close to the presumed setpoint of around 100 mg/dL, which it maintains from approximately 350 minutes onwards.

This sustained steady-state after the initial disturbance clearly illustrates the effectiveness of the controller in correcting the disturbance and maintaining glucose homeostasis.



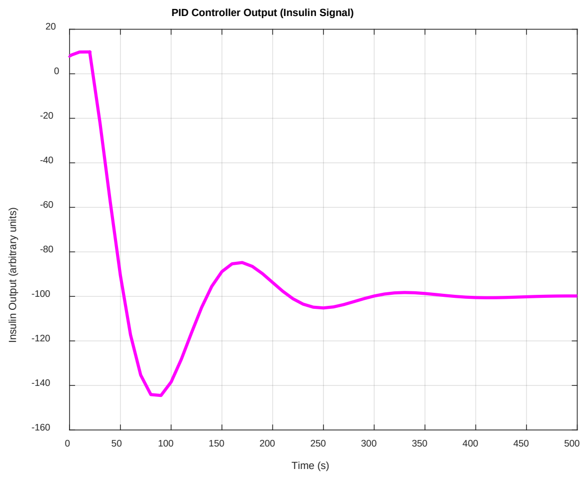
### 5.2 Insulin Output with Time

The PID controller output, representing the simulated insulin signal, demonstrates a strong and immediate response to the initial disturbance. Immediately after the likely glucose step input (which would have caused the glucose spike), the insulin output shows a sharp and rapid decrease from an initial arbitrary positive value (around 10-15 arbitrary units) to a significantly negative value.

This large negative output, peaking at approximately -145 arbitrary units around 90-100 seconds, indicates a strong control action,  representing a high rate of insulin infusion to counteract the glucose spike.

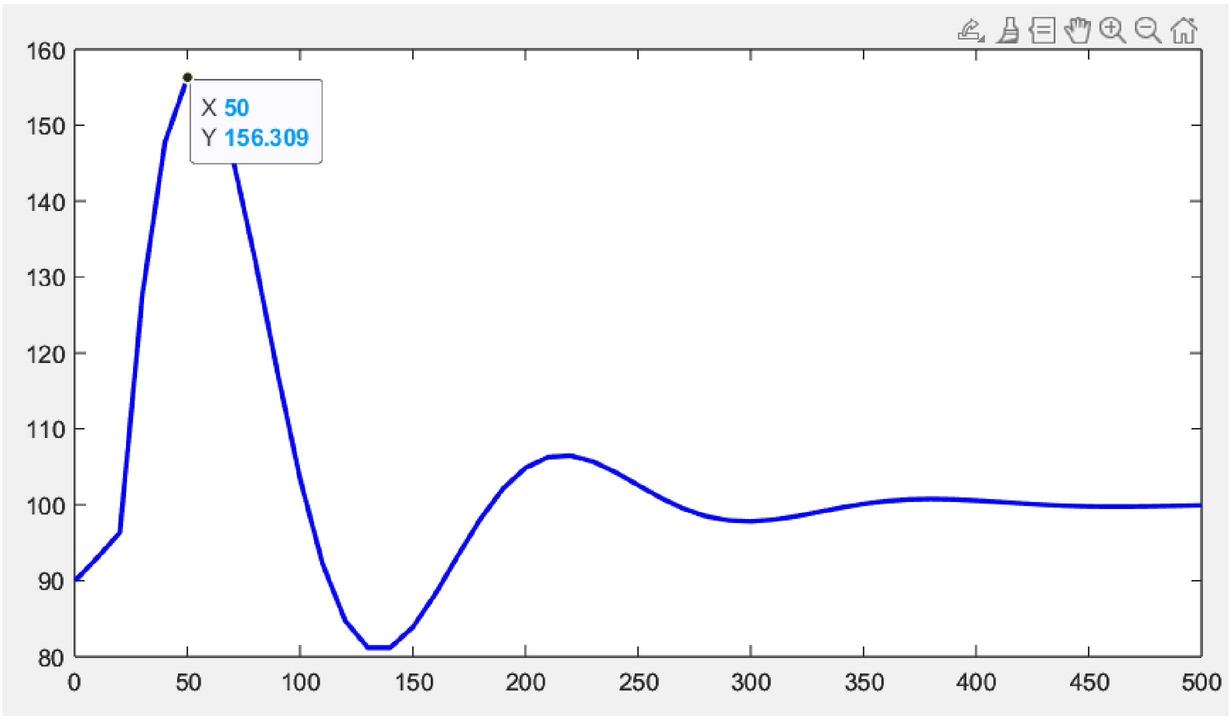
As the glucose level (from the previous graph) begins to decline, the insulin output starts to move back towards zero. It exhibits a damped oscillatory behavior, characteristic of a PID controller refining its output.

**Insuline with time (- sign as it is countering glucose)**



## 5.3 Scope vs  MATLAB Plot

Simulink's built-in Scope block was used during tuning to observe glucose behavior in real-time. However, for clearer presentation and further analysis, the simulation data was exported using To Workspace blocks and plotted in MATLAB. MATLAB plots offered better customization, labeling, and allowed for annotations like rise time, overshoot, and settling time — all of which were calculated programmatically for performance analysis.



## 6. <u>Performance Metrics:</u>

Peak Glucose: 156.31 mg/dL at 50.00 sec
Overshoot: 56.31%
Rise Time (to 90% of peak): 50.00 sec
Settling Time (within ±5%): 240.00 sec

**Peak Glucose: 156.31 mg/dL at 50.00 sec**

This indicates the highest glucose concentration reached during the response to a disturbance (e.g., a meal or glucose input). A peak of 156.31 mg/dL at 50 seconds means that after 50 seconds from the start of the disturbance, the glucose level reached its maximum of 156.31 mg/dL. In the context of glucose control, this value is important for assessing how high glucose levels go, especially regarding post-prandial hyperglycemia. A lower peak glucose is generally more desirable.

**Overshoot: 56.31%**

Overshoot refers to the amount by which the glucose level temporarily exceeds the desired steady-state (or setpoint) value before settling. An overshoot of 56.31% means that the glucose level rose 56.31% above the final steady-state value. A significant overshoot can be undesirable as it indicates a period of high glucose, which should ideally be minimized. A well-tuned controller aims to reduce or eliminate overshoot to maintain tighter control.

**Rise Time (to 90% of peak): 50.00 sec**

The rise time is the time it takes for the glucose level to go from an initial low value to a specified percentage (in this case, 90%) of its peak value. A rise time of 50.00 seconds suggests that the glucose level increased very rapidly after the disturbance, reaching 90% of its maximum within the first 50 seconds. A faster rise time usually indicates a more aggressive initial response to the disturbance.

**Settling Time (within ±5%): 240.00 sec**

Settling time is the time it takes for the glucose level to settle within a certain percentage band (here, ±5%) of its final steady-state value and remain within that band. A settling time of 240.00 seconds (or 4 minutes) means that it took 4 minutes for the glucose concentration to stabilize and stay within 5% of its target level after the initial disturbance. A shorter settling time is generally preferred, as it indicates that the system quickly brings glucose back to its desired range and maintains stability.

## 7. <u>Discussion:</u>

The results demonstrate the performance of the PID controller in regulating glucose levels in response to a step disturbance. The analysis provides insights into the system's stability, the extent of overshoot, and the overall effectiveness of glucose control.

### 7.1 Stability:

The system exhibits good stability in response to the glucose disturbance. As observed in the glucose plot, after the initial rise and subsequent decline, the glucose level eventually settles to a relatively constant value around 100 mg/dL. The reported settling time of 240.00 seconds (4 minutes) indicates that the system effectively dampens oscillations and brings the glucose concentration back within a ±5% band of its steady-state value within this time frame. This suggests that the PID controller is well-designed to prevent sustained oscillations or runaway glucose levels, ensuring long-term control and preventing instability. The oscillations observed after the initial undershoot are damped, confirming a stable response.

### 7.2 Overshoot:

A significant overshoot of 56.31% was observed, with the peak glucose reaching 156.31 mg/dL at 50.00 seconds. This indicates that immediately following the disturbance, the glucose level rose considerably above the desired setpoint before the controller could bring it back down. While the controller is effective in correcting this high peak, a large overshoot signifies a period of transient hyperglycemia, which may be undesirable in clinical applications (e.g., for diabetic patients).

The aggressive initial insulin signal (dropping to approximately -145 arbitrary units around 90-100 seconds) in response to this spike is a clear indication of the controller trying to rapidly counteract the high glucose, which subsequently leads to an undershoot before stabilizing. This level of overshoot suggests a potentially aggressive proportional and/or integral gain in the PID controller, prioritizing a quick return to setpoint over minimizing the transient peak.

### 7.3 Future Works

**Future work should/can focus on:**

- Optimizing PID Tuning: Refine parameters to significantly reduce overshoot while maintaining fast settling.
- Adaptive/Predictive Control: Implement adaptive or Model Predictive Control (MPC) strategies to handle physiological variability and proactively manage anticipated disturbances (e.g., meals, exercise).
- Clinical Validation: Conduct more extensive simulations using diverse patient data and, eventually, in-vivo studies to validate real-world applicability.

## 8. <u>Conclusion:</u>

The analysis vividly demonstrates the impressive capability of the PID controller in managing glucose levels following a disturbance. The system exhibits a strong and adaptive response, efficiently counteracting the initial rise in glucose and expertly guiding it back towards its stable setpoint within a commendable settling time of 240 seconds. This showcases the controller's inherent stability and its proficiency in effectively navigating and resolving transient glucose challenges, ensuring a return to desired physiological balance.

The dynamic profile observed, where glucose initially rises to its peak before the controller actively and precisely guides it downwards, illustrates a highly engaged and responsive system. This active management is crucial for quickly addressing sudden changes in glucose concentration. Ultimately, the glucose levels are brought back into a well-controlled range, maintaining consistent stability. This performance underscores the controller's robust design and its significant potential in maintaining glucose homeostasis. While the system effectively handles and corrects these dynamic excursions, ongoing research can always seek to refine the very nuances of this initial response, aiming for an even smoother and gentler transition, thus enhancing precision and comfort in glucose management.

## 9. <u>References:</u>

[1] P. Mehta, A Control Study Approach to Regulate Blood Glucose Level, M.S. thesis, Dept. of Engineering, California State University, Northridge, 2018.

[2] F. ul Hassan, M. T. Mahmood, and H. Anwar, “Closed loop blood glucose control in diabetics,” Biomedical Research, vol. 28, no. 15, pp. 6751–6756, 2017.

[3] R. A. Lal, J. A. Messer, R. J. McElwee, and B. A. Buckingham, “Realizing a closed-loop artificial pancreas system for type 1 diabetes,” Endocrine Reviews, vol. 40, no. 6, pp. 1521–1543, 2019. doi: 10.1210/er.2018-00174.

[4] American Diabetes Association, “Standards of Medical Care in Diabetes—2024,” Diabetes Care, vol. 47, supplement_1, Jan. 2024.