Course: ECE 3873 – Electronics Laboratory

Experiment Title: PID Cruise Control Simulator

Date Performed: 10/19/2025 – 12/12/2025

Instructor: Dr. Reza Saeed Kandezy

Abstract

Objective

The objective I am setting for myself for this project is to design and build an analog PID controller to simulate the cruise control function that exists in today's automobiles. The system will regulate the speed of a DC motor (if allowed to use) to match a user-defined "speed" that will automatically adjust the input of the motor to compensate for changes in the load (in a real car, accounting for gravity on roads with elevation changes). The system will be implemented entirely with analog circuitry including devices such as our PID controller, op-amps, DC motor, resistors, capacitors, and more.

Project Description

The technology that exists in modern automotives is astonishing, and the cruise control function is no exception. Cruise control uses a feedback loop to keep the vehicle at constant speed despite the surrounding conditions by adjusting the throttle. In this project I will attempt to replicate that behavior on a smaller scale using a DC motor as a vehicle. A potentiometer will serve as a method to control the speed of the motor/vehicle, a back-EMF feedback circuit will provide a voltage that is proportional into the speed of the motor. The resulting signal will be put through an op-amp summing circuit therefore generating an error signal. The error will then be processed by the analog PID controller, whose output will drive a BJT/MOSFET transistor that works to control the voltage and current that is then supplied to the motor.

Challenges and Testing

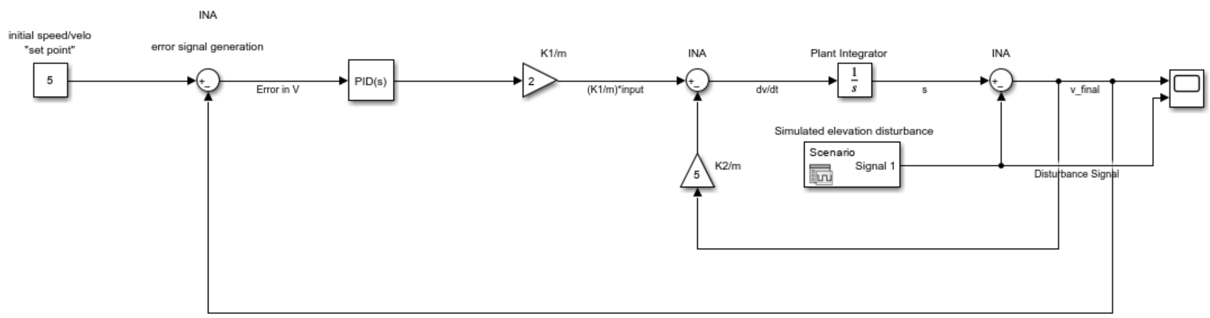
Some expected challenges include turning the PID network for stable operation and minimizing noise. These will be addressed through careful component selection, Multisim simulation, and real-life circuit implementation/design. Performance will be evaluated by applying changes to the potentiometer and load, then observing the system's speed response using feedback voltage as a measurement.

Conclusion

This project is a demonstration of a real-world engineering system of analog control systems. By successfully implementing a PID controller to maintain a constant motor speed, it will highlight the effectiveness of feedback control in practical conditions. It algins strongly with this course's emphasis on analog circuits, originality, and applied problem-solving.

Circuit Design

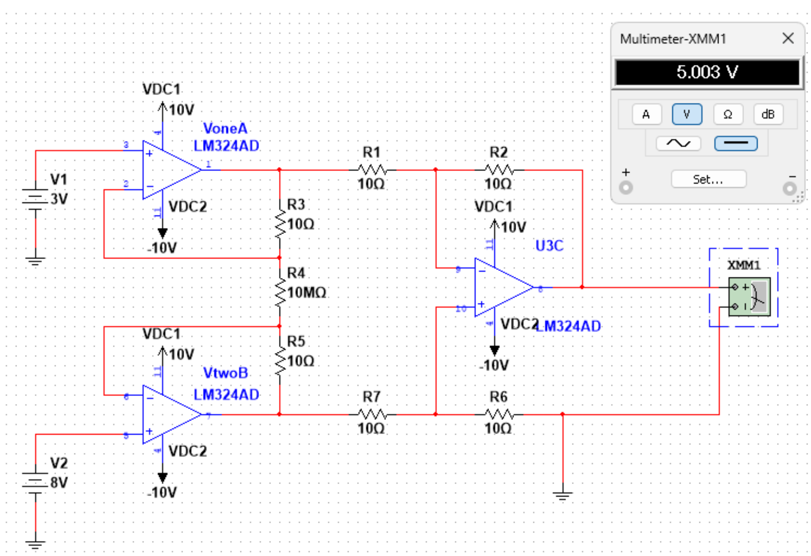
Initial System Desing – Simulink



This screenshot shows the ideal closed loop system that I had come up with to accurately simulate the cruise control function we find in modern automobiles. In the context of this project, we will refer to the "setpoint speed" of this system as DC Voltage, the designed setpoint of this system was generated by a 5 V signal form a DC Power Supply.

Instrumentation Amplifiers (Unity-Gain)

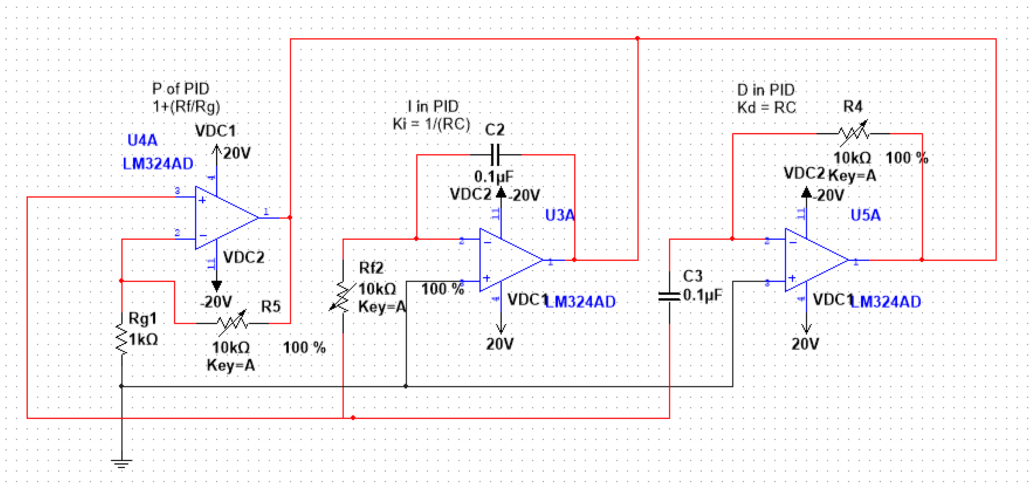
To account for the feedback loops in our system, we are using "difference nodes", these can be seen in the system above. In the physical design of this circuit, these "difference nodes" are implemented in the form of instrumentation amplifiers. Below is the Multisim schematic of the instrumentation amplifier that have been used in this project.



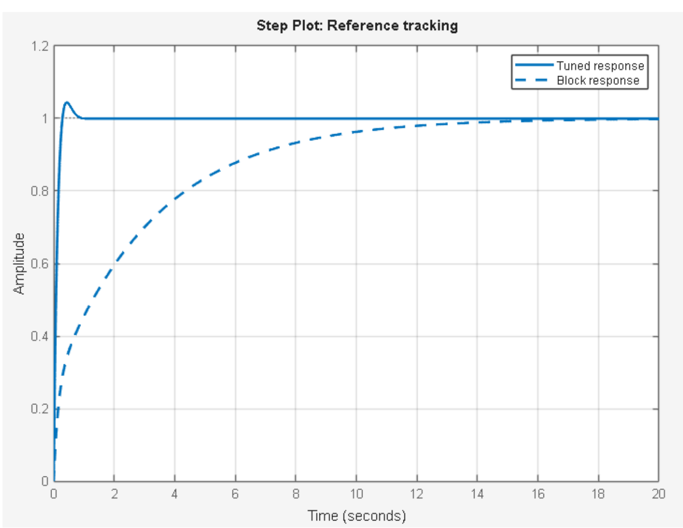
On board, my instrumentation amplifiers were designed using LM324 quad 30-V 1.2-MHz operational amplifiers. The resistors in the Multisim schematic match the resistors physically used in this project to create unity-gain instrumentation amplifiers.

PID Controller

The next block in the system is our PID controller. This is the most integral part of the system, as it allows us to actively correct our signal, keeping it close to the desired 5 V setpoint regardless of any disturbances that might affect our system. Below is a rough schematic of the PID controller used in our system.



Our PID controller, similar out INA's, is physically implemented using an LM324 quad 30-V 1.2-MHz operational amplifier. The resistor and capacitor values used on our physical board were designed with specific gains in mind. The values of Kp, Ki, and Kd were found using Simulink's "PID auto tune" function, theoretically producing the optimal gains for each respective component. Below is a screenshot of Simulink's "PID auto tune" function and it's quantitative/graphical response.



Controller Parameters

	Tuned	Block
P	3.884	1
I	31.1012	1
D	0.031903	0
N	32.4753	1

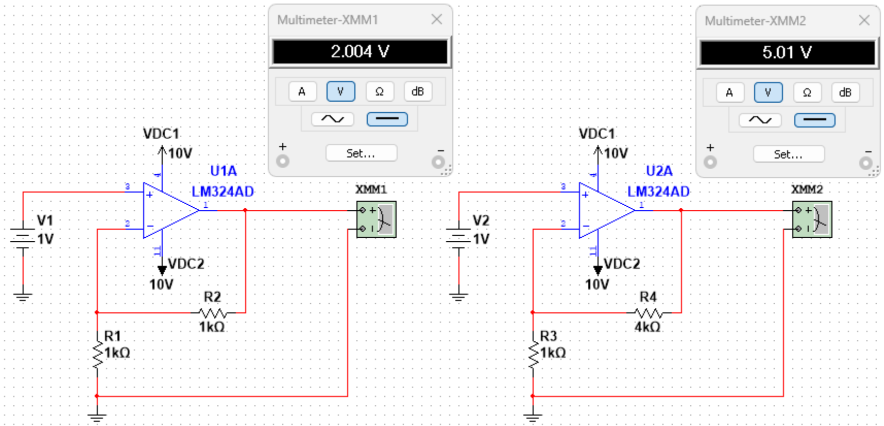
Performance and Robustness

	Tuned	Block
Rise time	0.198 seconds	6.62 seconds
Settling time	0.646 seconds	12.1 seconds
Overshoot	4.38 %	0 %
Peak	1.04	0.999
Gain margin	Inf dB @ NaN rad/s	Inf dB @ NaN rad/s
Phase margin	80.1 deg @ 8.91 rad/s	109 deg @ 0.434 rad/s
Closed-loop stability	Stable	Stable

For simplicity's sake, the values of our PID coefficients were rounded to the nearest integer values. ($K_p \approx 4, K_I \approx 31, K_D \approx 0$).

Buffer Amplifiers

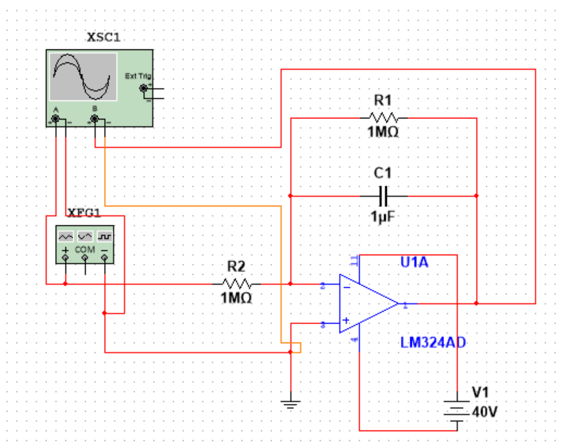
The next components in our system are our buffer amplifier. The purpose of these buffer amplifiers were designed to provide impedance isolation between stages, thus preventing loading effects and unwanted interactions from surrounding components. Below is a screenshot of the circuit schematic that was used when implementing these buffers into our physical system.



These buffer amplifiers are essentially non-inverting amplifiers that were designed using an LM324 quad 30-V 1.2-MHz operational amplifier. Specific values of our resistors were selected to produce a gain of 2, and 5 corresponding to the two buffers in our system. These were the values that were found to produce the best results for our system in Simulink by the process of trial and error.

Op-Amp Integrator (Unity Gain)

The final component found in our system is our plant integrator. The plant integrator was used to model the dynamic behavior of our physical system by integrating the control signal with respect to time, introducing system memory and realistic plant behavior into the system. Below is a screen shot of the schematic that we used when physically implementing our plant integrator.



Like the other components in this system, our op-amp integrator is implemented on an LM324 quad 30-V 1.2-MHz operational amplifier, and the resistor and capacitor values were chosen to produce a unity-gain integrator.

Daily Life Application

This project illustrates the basic control principles applied to several common engineering systems that must be stable and accurate in the presence of disturbances. PID control systems are commonly applied to automobile cruise control, thermostat-based heating, ventilation, and air conditioning (HVAC) systems, motor speed control, drone flight stabilization, and industrial process control. In all the above systems, the controller continuously monitors the difference between the desired setpoint and the output, adjusting the control effort to minimize the error, which leads to predictable systems and the ability to operate smoothly under varying environmental conditions.

The use of buffer amplifiers and a plant integrator in this project closely mirrors real-world implementations. Buffer amplifiers represent signal conditioning stages employed in practical electronics to isolate the sensors, controllers, and actuators, allowing measurements and control signals to not be distorted by loading effects. The plant integrator replicates the performance of actual systems, e.g., vehicle velocity, which depends on the cumulative effects of force applied over time, for the controller to interact with a dynamic system rather than an idealized static gain. Combined, these elements illustrate that abstract control theory translates into physical hardware, and PID-based control systems enable reliable, responsive operation in many technologies humans engage with each day.

Troubleshooting

The primary troubleshooting tasks came from two areas throughout the design process. The first of which came from the simulation stages of this project. In preparation for the physical design of this system, a simulation was designed in Simulink. During this process, I ran into the task of finding specific values for many different variables throughout the system. As mentioned before, the optimal values of $K_p, K_I, and\ K_D$ coefficients were found using Simulink's "PID auto tune" function, which luckily took most of the troubleshoot/trial and error out of that process, but the same cannot be said for the process of finding appropriate values for the gain of our buffer amplifiers. This process was surprisingly difficult, as it took many attempts to find the values that resulted in the best response from the system.

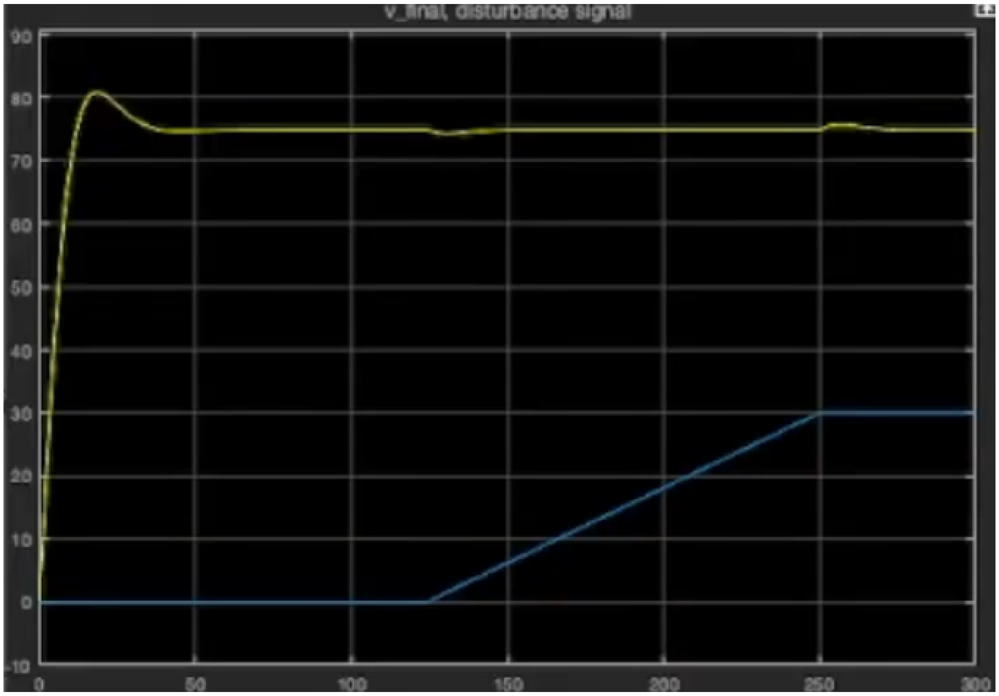
Further along in the simulation process, I began modeling each system component in Multisim, which required its own set of troubleshooting issues. Contrary to how Simulink operates, Multisim requires the user to design each block down to the smallest detail. Wiring configurations, component values, and simulation parameters all must be carefully selected for individual components to work. With the large amount individual components, even the smallest of errors can compound causing the system to produce unreliable and inconsistent outputs.

Multisim’s troubleshooting primarily consisted of isolating circuit blocks-proportional, integrator, differential, instrumentation, and buffer amplifiers—and testing them individually using a variety of AC and DC inputs to verify functionality. Only when the design of each individual component was verified did I link the components together to complete the closed loop system.

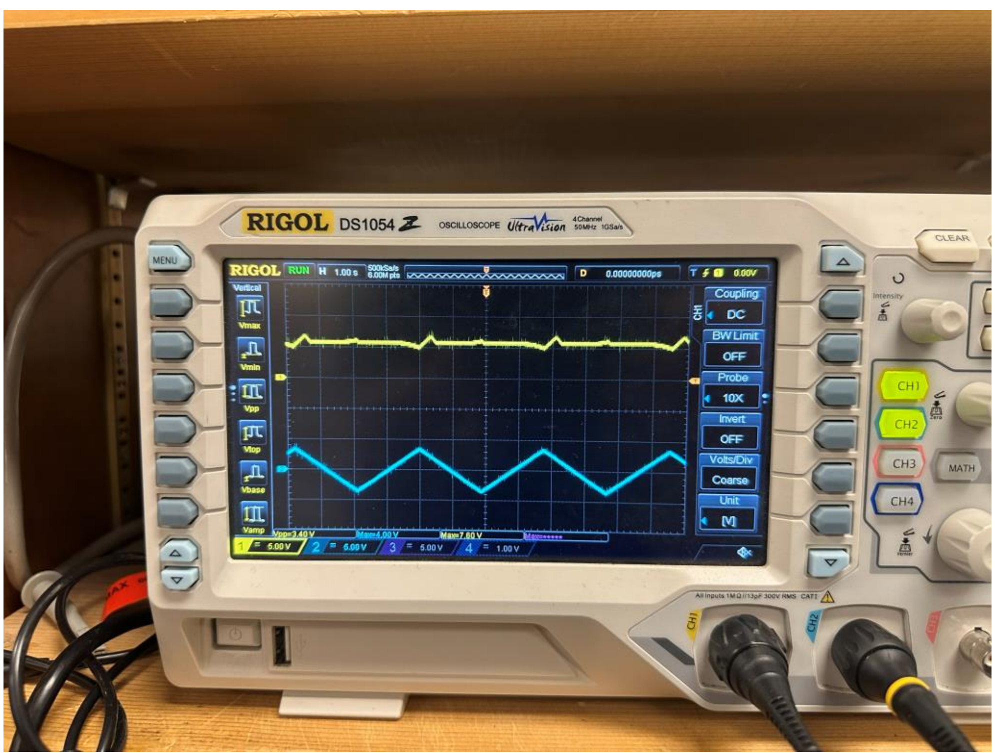
Once a Multisim model had been fully verified, I began to transition to the physical/real implementation of the system. This, unsurprisingly, presented its own plethora of troubleshooting issues. Unlike the circuits simulated in Multisim, practical circuit components are non-ideal and often behave unpredictably, this created a small level of error throughout the entire system which continued to be an issue until the conclusion of this experiment. Along with the unpredictability of practical components, I also ran into specific wiring issues with different op-amp configurations. Ensuring a reliable connection at every point throughout a complex circuit proved difficult. Physical troubleshooting was done by probing signals at critical nodes throughout the whole system, checking power rails, and verifying functionality in each subsystem separately before connecting the full closed-loop circuit. This stage-based approach helped ensure that the physical behavior of the circuit closely corresponded with simulations and helped mitigate the issues discussed above.

Visuals

Ideal System Response via Simulink



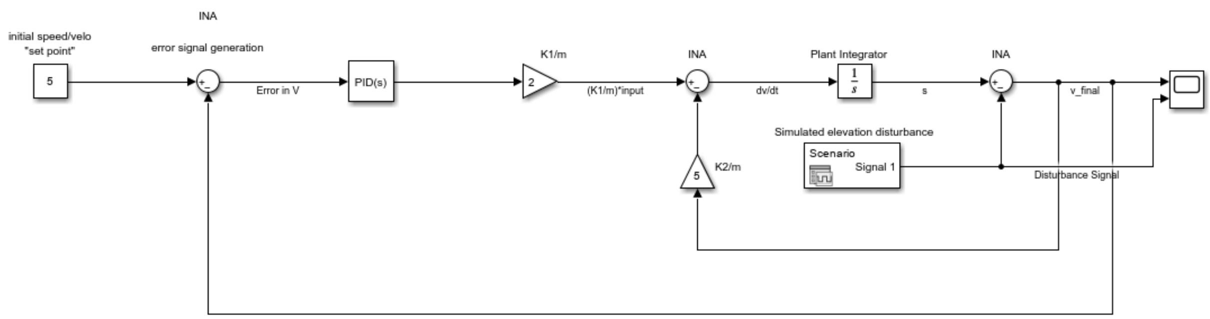
Practical System Response



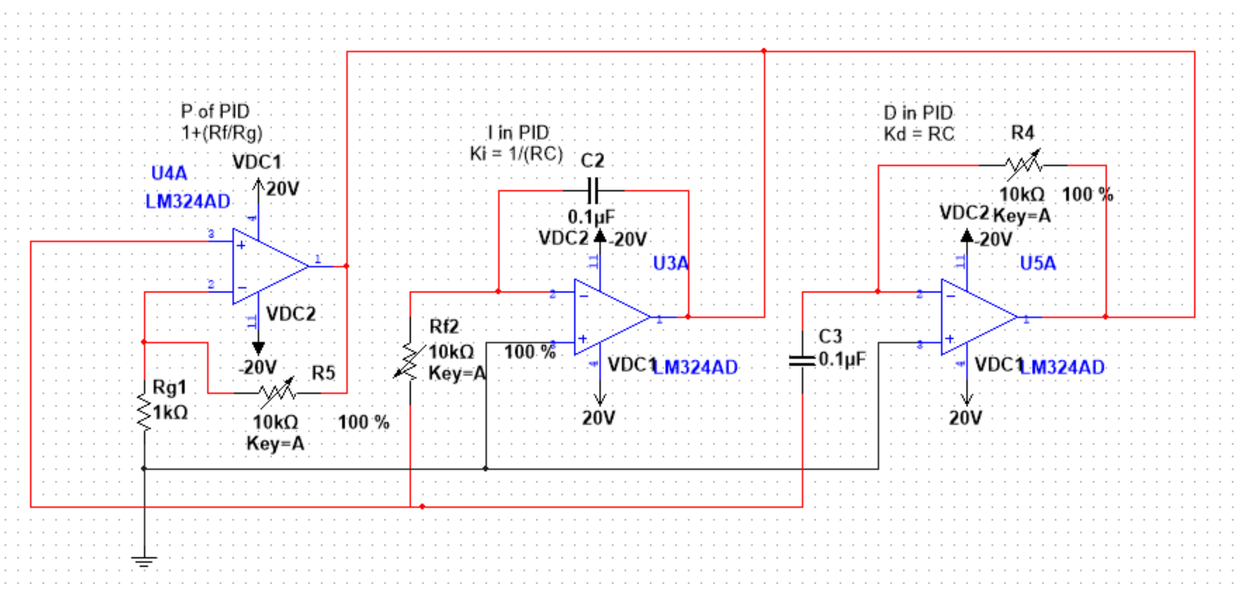
BOM

Components	Quantity	Link (if order needed)	Price	Ordered?	Arrived/Taken from lab
LM324 Quad Op-Amp	5	N/A	N/A	☑	☑
10k Pot	2	N/A	N/A	☑	☑
Resistors (varying in value)	-	N/A	N/A	☑	☑
Capacitors (varying in value)	-	N/A	N/A	☑	☑
1N4007 Flyback Diode	2	N/A	N/A	☑	☑
1N4148 Small Signal Diode	2	N/A	N/A	☑	☑
Solderless Breadboard	1	N/A	N/A	☑	☑
Misc. Wires	-	N/A	N/A	☑	☑

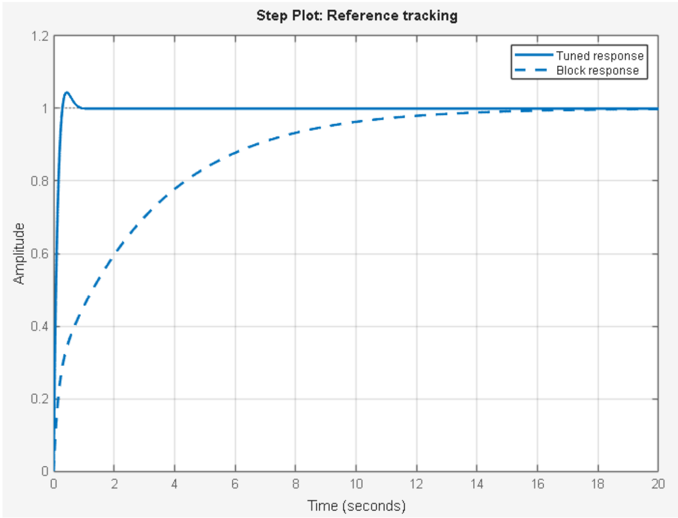
Simulink Schematic



PID Controller Schematic



Ideal PID System Response



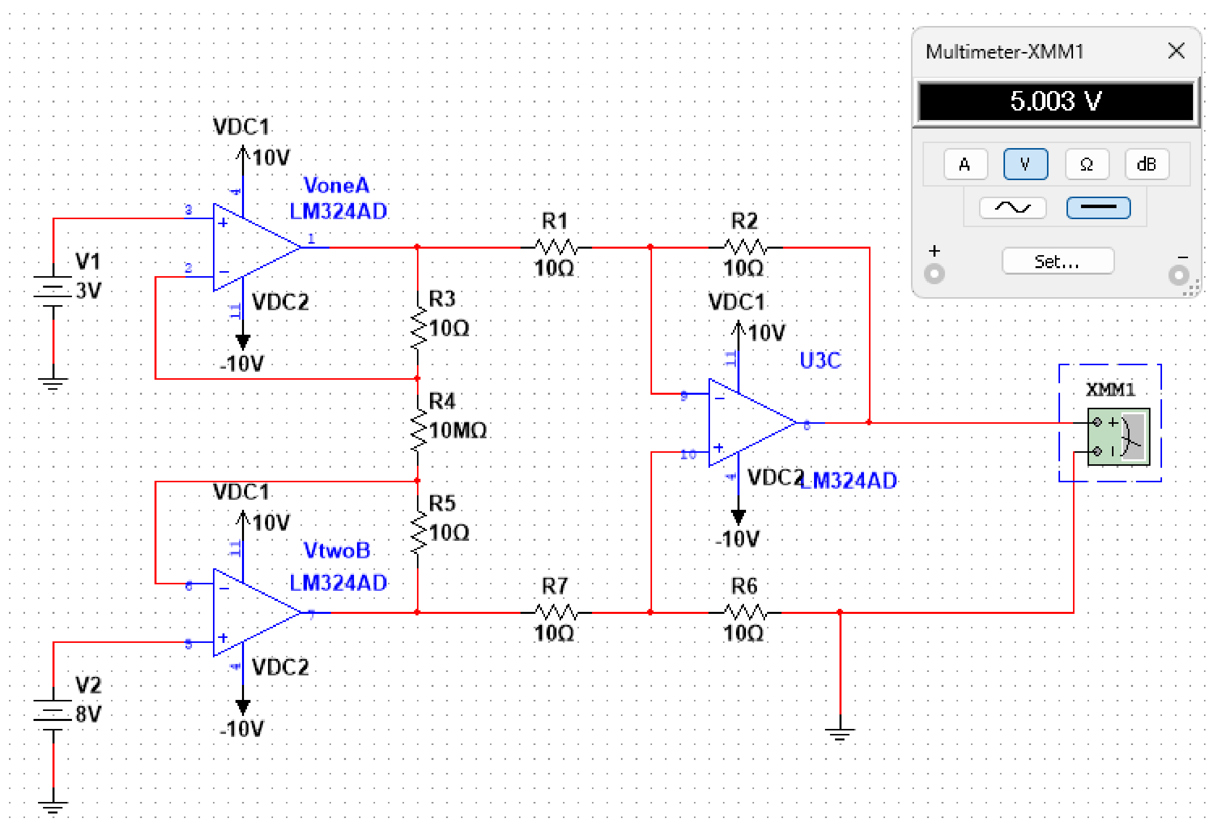
Controller Parameters

	Tuned	Block
P	3.884	1
I	31.1012	1
D	0.031903	0
N	32.4753	1

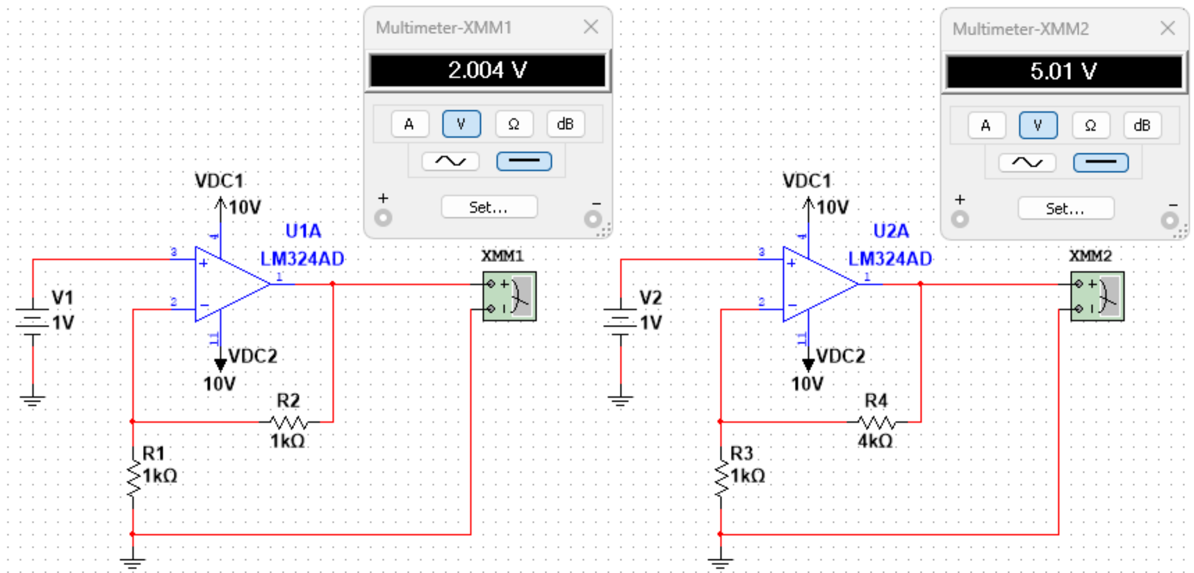
Performance and Robustness

	Tuned	Block
Rise time	0.198 seconds	6.62 seconds
Settling time	0.646 seconds	12.1 seconds
Overshoot	4.38 %	0 %
Peak	1.04	0.999
Gain margin	Inf dB @ NaN rad/s	Inf dB @ NaN rad/s
Phase margin	80.1 deg @ 8.91 rad/s	109 deg @ 0.434 rad/s
Closed-loop stability	Stable	Stable

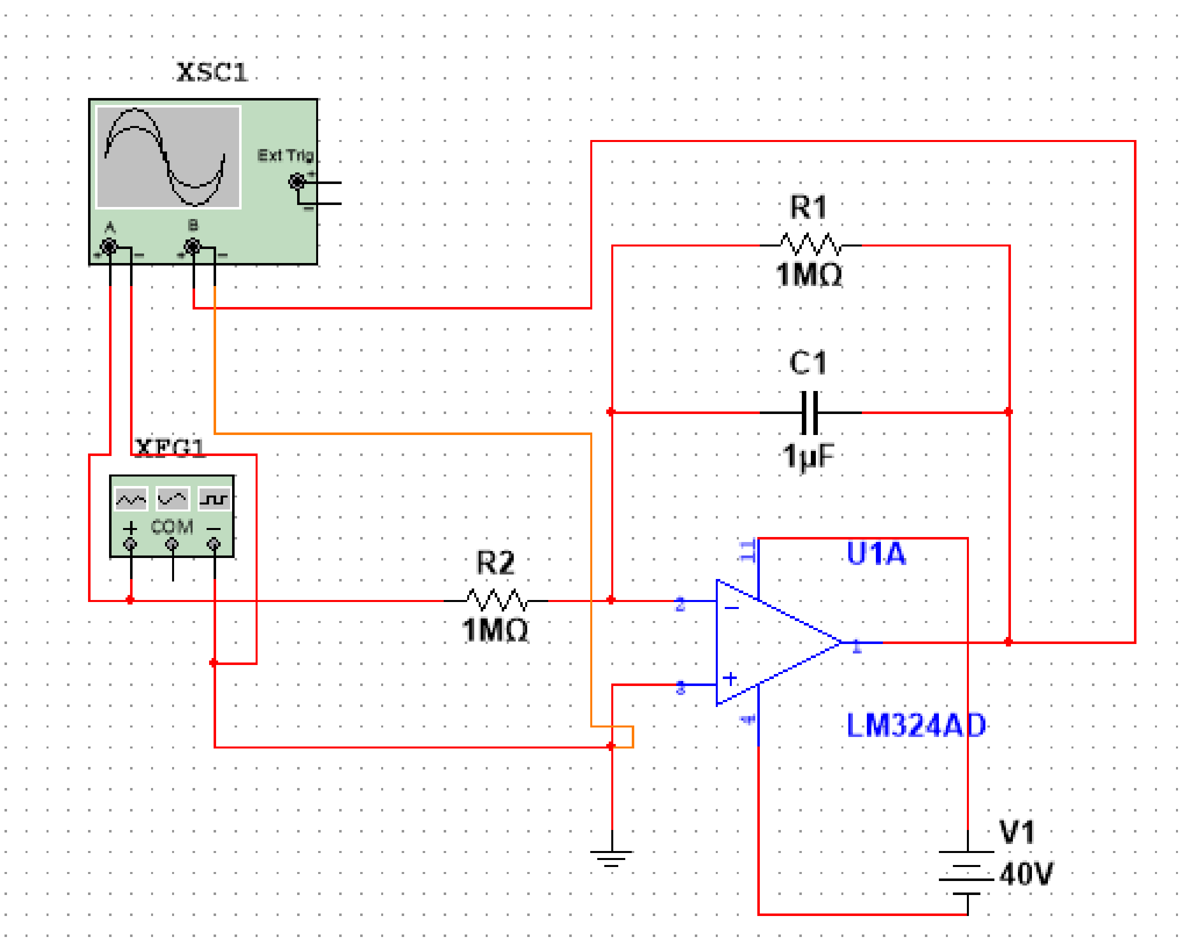
Instrumentation Amplifier Schematic



Buffer Amplifier Schematic



Plant Integrator Schematic (Unity-Gain)



References

[1] Pratik Rana, “Cruise Control of a Vehicle in Simulink,” YouTube, Apr. 03, 2023. Available: https://www.youtube.com/watch?v=cP7XbaOf8iA. [Accessed: Dec. 13, 2025]

[2] “The PID Controller — Part 1,” Nuts and Volts Magazine. Available: https://www.nutsvolts.com/magazine/article/the_pid_controller_part_1

[3] Electronics Tutorials, “Op-amp Integrator - The Operational Amplifier Integrator,” Basic Electronics Tutorials, Feb. 04, 2019. Available: https://www.electronics-tutorials.ws/opamp/opamp_6.html

[4] Electronics Tutorials, “Non-inverting Operational Amplifier - The Non-inverting Op-amp,” Basic Electronics Tutorials, Feb. 2019. Available: https://www.electronics-tutorials.ws/opamp/opamp_3.html

[5] Texas Instruments, LMx24, LMx24x, LMx24xx, LM2902, LM2902x, LM2902xx, LM2902xxx Quadruple Operational Amplifiers, SLOS066AE, August 1975 – Revised September 2025

[6] University of Michigan, “Control Tutorials for MATLAB and Simulink - Introduction: PID Controller Design,” *ctms.engin.umich.edu*. Available: https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID