# MANUAL TUNING OF A DISCRETE-TIME PID CONTROLLER

1.0 Learning Outcomes……1
2.0 Hardware Set Up……1
3.0 Practical Investigation via Simulink……..……..2
4.0 Concluding Remarks…….……4
Recommended Reference……,…….5

## 1.0 Learning Outcomes

Upon completing this practical exercise, students should be able to demonstrate the Learning Outcomes presented in Table 1. At the end of the exercise, revisit each Learning Outcome and confirm that it has been achieved.

**Table 1:** *Learning Outcomes for the Practical Exercise*

| ID | Learning Outcome | Achieved |
|---|---|---|
| 1 | Implement and experimentally evaluate proportional control using repeatable closed-loop response measurements. | ☐ |
| 2 | Manually tune proportional and integral gains and investigate whether derivative action improves controller performance. | ☐ |
| 3 | Select, justify and verify controller gains against defined performance requirements and actuator constraints. | ☐ |

## 2.0 Hardware Set-Up

This exercise builds on the hardware and algorithm setup described in the exercise, DC Motor Speed Measurement using an Encoder, which details the operation of a DC motor with an encoder.

# 3.0 Practical Investigation via Simulink

MATLAB and Simulink should be used to complete this practical exercise. Students should build upon the Simulink model developed in DC Motor Speed Control Using a PI Controller, adapting the controller to investigate and manually tune proportional, integral and derivative actions. The investigation compares repeatable closed-loop responses with specified requirements. No mathematical model of the motor is required.

### Establish the Control Requirements

Before tuning, define measurable performance requirements for the DC motor speed control system. Specify the test step, transient response limits, steady-state accuracy, measurement filter and actuator constraints. Use these criteria to evaluate every controller configuration.

**Table 2:** *Performance Requirements for DC Motor Speed Control*

| Requirement | Selected limit / definition |
|---|---|
| Reference speed (RPM) | |
| Rise time (10–90%) | |
| Maximum overshoot (%) | |
| Settling time and tolerance band | |
| Maximum steady-state error (RPM) and averaging interval | |
| Actuator saturation limits | |

### Establish a Repeatable Response Test

1. Verify the reference speed, measured speed scaling and actuator direction. Confirm that the applied actuator command remains within the specified limits.

2. Use the same initial operating condition, reference speed change, measurement filter, sampling period, test duration and mechanical load for each comparison.

3. Configure Simulink to log reference speed $r[k]$, measured speed $y[k]$ and applied actuator command $u[k]$ (after saturation). Capture the transient response and a sufficient steady-state interval for every trial.

4. Check that the controller can be safely stopped and restarted between trials. Initialise controller states, including the integral term, consistently. Keep the anti-windup configuration unchanged across comparisons and avoid prolonged saturation.

Complete Table 3 during tuning. Include a P-only baseline, at least two PI candidates and a PID comparison where safe and appropriate. Record periods

of actuator saturation and unexpected control variation. If needed, assess command activity using the sum of absolute changes in the applied command over an equal recording interval.

**Table 3:** *Experimental Evaluation of Controller Gains*

| | $K_P$ | $K_I$ | $K_D$ | Rise time (s) | Maximum Overshoot (%) | Settling time (s) | Steady state error (SSE) | Actuator Observation |
|---|---|---|---|---|---|---|---|---|
| P-only | | 0 | 0 | | | | | |
| PI-1 | | | | | | | | |
| PI-2 | | | | | | | | |
| PID | | | | | | | | |
| Final | | | | | | | | |

### Investigate Proportional Action

Begin with integral and derivative gains set to zero ($K_\mathrm{i} = 0$ and $K_D = 0$). Select a conservative proportional gain $K_P$ and increase it gradually while the response remains safe. Establish a responsive, well-behaved baseline; zero steady-state error is not necessarily expected with P-only control.

1. Implement proportional-only control and record its response to the defined reference speed change.

2. Adjust $K_P$ in small steps with other test conditions unchanged. Observe rise time, overshoot, settling time, steady-state error and actuator behaviour.

3. Select a suitable proportional baseline and justify the selection using the measured results.

### Introduce Integral Action

Retain the selected proportional gain and introduce a small positive integral gain $K_i$. Integral action can reduce persistent steady-state error, but excessive integral action may increase overshoot, settling time and windup during actuator saturation.

1. Retain the selected proportional gain and investigate at least two different integral gains, changing only $K_I$ initially. Compare the resulting responses against the proportional-only controller.

2. Observe the measured speed and applied actuator command, paying particular attention to overshoot, settling, steady-state error and periods of saturation.

3. Select a suitable combination of proportional and integral gains based on the measured results. Where further improvement is required, refine $K_P$ or

$K_I$, adjusting only one gain at a time and recording the influence on controller performance.

**Investigate Derivative Action**

Derivative action is optional. Starting from the selected PI controller, introduce a conservative derivative gain $K_D$ only where further improvement in transient performance may be possible, such as reducing overshoot or oscillation. Derivative action should preferably be applied to the measured output rather than the tracking error to avoid derivative kick following reference speed changes. Appropriate filtering should also be used to reduce the amplification of measurement noise. The benefits of derivative action should be evaluated against any additional variation in the actuator command.

1. Compare the selected PI baseline with one or more cautious PID trials under identical test conditions.

2. Compare rise time, overshoot, settling time, steady-state error and actuator-command variation.

3. Determine whether a non-zero $K_D$ provides a worthwhile improvement. Retaining PI control is a valid outcome if it meets the requirements with less actuator activity.

**Final Controller Verification**

1. Repeat the selected controller test under the original conditions to confirm repeatability.

2. Test at one additional safe reference speed or load condition to investigate whether acceptable performance is maintained.

3. Compare the measured rise time, overshoot, settling time and steady-state error against the established performance requirements. Verify that the actuator operates within its specified limits and justify the final controller selection using experimental evidence. Where requirements are not satisfied, identify the limitations and suggest appropriate refinements.

## 4.0 Concluding Remarks

This practical exercise introduced the manual tuning and experimental evaluation of a discrete-time PID controller for DC motor speed control. Using MATLAB, Simulink and ACE-Lab hardware, proportional, integral and derivative actions were investigated through repeatable closed-loop experiments.

The results illustrate how controller gains influence rise time, overshoot, settling time and steady-state error. Proportional action establishes a baseline response, integral action can improve steady-state accuracy, and derivative

action may provide further transient improvement where justified. The experiments also highlight the importance of measurement noise, actuator constraints and avoiding excessive controller gains.

By evaluating controller configurations against defined performance requirements, suitable gains can be selected, refined and verified without mathematical modelling of the motor. The exercise reinforces systematic manual tuning and the use of experimental evidence to justify practical engineering decisions.

## Recommended Citation

If you have used this practical exercise in your teaching, learning or research, please cite it using the reference below, adapting the format as required by your preferred referencing style.

> Pickering, J. E. (2026). Manual Tuning of a Discrete-Time PID Controller. ACE-Lab. Teaching Notes: ACE-Lab Practical Exercises, Version 1.0. Zenodo. https://zenodo.org/records/23245132