CARIMO

Licensed from IIT Bombay

Control theory, but you can actually spin it.

CCLL, the CARI Control Learning Lab, puts a real DC motor on your bench and three control paradigms in your students' hands. Tune it, break it, watch it recover.



PID
Classical control
Tune gains from a single step test

MPC
Model predictive control
Plans ahead and respects hard limits

Fuzzy
Intelligent control
Plain-English rules, no plant model needed

3 in 1
PID, MPC and Fuzzy on one motor

Live data
speed and PWM streamed to your PC

Real motor
a spinning plant, not a simulation

USB
plug into any Windows PC

Hardware + **Python-based experiments** + **MATLAB-based experiments**

= One complete learning experience.

For students
Learn control systems hands-on

For educators
A flexible teaching platform

For researchers and industry
Real-world control applications

Experiment. Learn. Analyze. Innovate.

www.carimotech.com

CARI Control Learning Lab
DC Motor Control Learning Kit

Three pillars. One motor.

CCLL
CARI Control Learning Lab

CCLL, the CARI Control Learning Lab: one motor, three ways to control it, with Python and MATLAB experiments for each.

PID

Classical control

Plant ID
one step test identifies the motor

2-DOF
setpoint weighting tames overshoot



Push, erase, anticipate.

PID asks one question: how far is the motor from target? The kit identifies the plant from a single step test, then derives the gains by formula.

P: push now · I: erase the gap · D: anticipate · 2-DOF weighting

Hardware experiments

1 Open-loop step
2 PI setpoint tracking
3 Disk disturbance rejection
4 Setpoint weighting (2-DOF)
5 Anti-windup
6 Sinusoidal tracking
7 PI vs PID, derivative-N sweep

Python-based experiments · MATLAB-based experiments

MPC

Model predictive control

Predictive
forecasts the motor and plans ahead

Limits
hard constraints handled natively



Measure, predict, optimise, apply.

It forecasts the motor and picks the gentlest moves that hit the target within every hard limit.

Hardware experiments

1 Aggressive vs smooth tuning
2 Constraint handling
3 Disturbance observer
4 Model mismatch (A / B / C)
5 Sine vs step response
6 Control horizon sweep
7 Software disturbance rejection
8 Staircase tracking
9 MV-rate limit sweep
10 Move-suppression sweep
11 1 Hz sinusoidal tracking
12 Prediction horizon sweep
13 Output weight sweep

Python-based experiments · MATLAB-based experiments

Fuzzy

Intelligent control

Rule-based
plain-English rules, no plant model

Open
edit the rules, never a black box



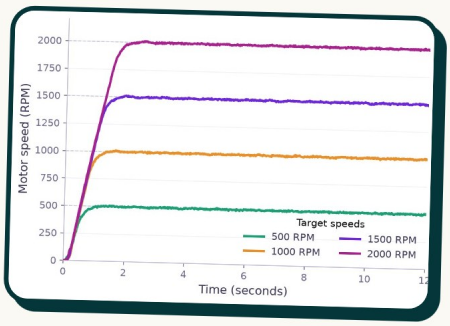
Four plain-English rules.

"If too slow, push harder. If overshooting, ease off." No transfer functions, no model. The rule builder is open, never a black box.

No-math entry. Control hardware before deriving an equation.

Write your own rules. Membership functions and defuzzification.

Industry-relevant. The skills employers ask for.



Real data: the fuzzy controller locks onto target speeds on a live kit.

Hardware experiments

Python-based experiments · MATLAB-based experiments

Hardware, Python and MATLAB experiments, code and guides: all included.

www.carimotech.com

Datasheet

CCLL, CARI Control Learning Lab

Spec by spec. Nothing hidden.

CCLL is a real plant on a bench. A permanent-magnet motor with encoder connects to a Windows PC over USB, and hot-swap load disks inject real disturbances.



Hardware specifications

Motor	Permanent-magnet brushed DC (PMDC)	**Encoder**	Quadrature, ±8 RPM accuracy
Max speed	~3200 RPM at full supply	**Controller**	32-bit ARM Cortex-M4F, 84 MHz, on-chip PWM
Motor driver	Integrated H-bridge, current-regulated, up to 3.5 A peak; over-current and over-temperature protection	**Power regulation**	Regulated on-board supply
Load	Hot-swap magnetic load disks of varying magnet count; no-load and loaded	**Power**	12 V DC adapter; 230 V AC mains input
Connectivity	Windows PC over USB 2.0	**Enclosure**	Durable ABS, tabletop, portable
Control modes	Open-loop and closed-loop speed; speed and position in the Python edition	**Monitoring**	Live RPM and PWM; timestamped CSV logging

Control software, all included

Package	Method	Needs MATLAB?	Status
PI / PID, Python	Classical	No, standalone Windows app	Included
PI / PID, Simulink	Classical	Yes: Simulink	Included
MPC, Python	Predictive	No, desktop app	Included
MPC, Simulink	Predictive	Yes: Simulink and MPC Toolbox	Included
Fuzzy Logic, Python	Rule-based	No, desktop app	Included
Fuzzy Logic, Simulink	Rule-based	Yes: Simulink and Fuzzy Logic Toolbox	Included

Every edition ships with the kit. MATLAB-based experiments run on your own MATLAB and Simulink installation and come with editable Simulink models.

Control hardware
Motor, encoder, driver and load disks

Python and MATLAB code
Python app and Simulink models

Documentation and guides
Step-by-step experiments and reports

Experiments and projects
PID, MPC and Fuzzy on real hardware

In the box

- Control hardware with encoder
- 12 V adapter and USB cable
- Documentation and guides
- User manual
- Hot-swap magnetic load disks
- Python and MATLAB code
- Experiments and projects
- 1-year hardware warranty

What you need

Windows 10 or 11 with a free USB 2.0 port. MATLAB-based experiments also need MATLAB R2021a or later with Simulink, plus the MPC and Fuzzy Logic Toolboxes.

CARIMO

www.carimotech.com
contact@carimo.tech +91 90823 97027

Developed at IIT Bombay under Remote-Triggered Labs (MHRD / NMEICT). DPIIT and MSME recognised. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. Specifications subject to change without notice.

Outcomes

CCLL, CARI Control Learning Lab

What students take away.

Control theory stops being abstract when students can touch the plant. Every experiment ends with something they measured themselves.

Control on real hardware

Tune PI/PID, MPC and Fuzzy Logic on a real motor and see the response live, not just in a simulation.

Disturbance rejection

Swap magnetic load disks and watch the controller fight real disturbances.

Real data, real analysis

Log timestamped CSV data and compare controllers on measured results.

Python or MATLAB, their choice

Standalone Python desktop apps, or Simulink models, all included.

By the end of the course, students can

- ✓ Model and identify a real DC motor
- ✓ Design and tune a speed controller
- ✓ Compare classical, predictive and rule-based control
- ✓ Judge a controller by measured data

One kit. A whole semester of labs.

A tabletop plant, a USB cable and ready-made experiments. Students start measuring in the first session.



1. Run
Spin the motor and log live RPM.

2. Disturb
Swap a load disk.

3. Control
Tune PID, MPC or Fuzzy.

4. Compare
Judge results from CSV data.

Learn control by doing it.

www.carimotech.com
contact@carimo.tech +91 90823 97027

Developed at IIT Bombay under Remote-Triggered Labs (MHRD / NMEICT). DPIIT and MSME recognised. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. Specifications subject to change without notice.