2025-11-01

E002: Control Theory Fundamentals

Mathematics Behind Control Systems

Part 1 · Duration: 30-35 minutes

Beginner-Friendly Visual Study Guide

Repository: `https://github.com/theSadeQ/dip-smc-pso`
Documentation: academic/paper/presentations/podcasts/episodes/

Learning Objective: Understand state-space representation, Lyapunov stability, and the fundamental principles of sliding mode control

What is Control?

Key Concept

Control Problem: Make a system's output track a desired reference, despite:

- Disturbances (external forces, noise)
- Uncertainties (model errors, parameter variations)
- Constraints (actuator limits, safety bounds)

Open-Loop vs. Closed-Loop

Open-Loop (No Feedback):

- Execute predetermined commands
- No correction for errors
- IMPOSSIBLE for DIP (unstable!)

Example

Microwave timer - no temperature feedback, just time

Closed-Loop (Feedback):

- Measure output, compare to reference
- Automatically corrects disturbances
- REQUIRED for DIP stability

Example

Thermostat - measures temperature, adjusts heating

State-Space Representation

General Form

Continuous-Time:

$$\dot{x}(t) = f(x(t), u(t), t) \quad \text{(State dynamics)}$$
$$y(t) = h(x(t), u(t), t) \quad \text{(Output equation)}$$

Where:

- $x(t)$ = state vector (internal system variables)
- $u(t)$ = control input (force on cart)
- $y(t)$ = measured output
- $f(\cdot)$ = dynamics function, $h(\cdot)$ = measurement function

DIP State-Space: The Dashboard

Six State Variables:

enumiCart position (x)

0. enumiFirst pendulum angle (θ_1)

0. enumiSecond pendulum angle (θ_2)

0. enumiCart velocity ($\dot{x}$)

0. enumiFirst pendulum velocity ($\dot{\theta}_1$)

0. enumiSecond pendulum velocity ($\dot{\theta}_2$)

Single Control Input:

- Horizontal force on cart (u)
- Measured in Newtons
- **Underactuated:** 1 input controls 3 outputs

Pro Tip

If you know these 6 numbers at any instant, you know EVERYTHING about the system's state!

How DIP Dynamics Work (Plain English)

Quick Summary

Physics Workflow:

enumi**Mass matrix**: How heavy everything is, how mass is distributed

0. enumi**Coriolis & centrifugal**: How rotation causes forces
0. enumi**Gravity terms**: How gravity pulls pendulums down
0. enumi**Control input**: The push force we apply

Equation says: "Mass $\times$ acceleration + rotation effects + gravity = applied force"
It's just $F = ma$, but for multiple connected bodies!

Lyapunov Stability: The Ball in a Bowl

The Physical Picture

Key Concept

Imagine a marble in a smooth bowl:

0. enumi**Nudge it**: Marble rolls to the side
0. enumi**Gravity pulls down**: Rolls toward bottom
0. enumi**Overshoots**: Has momentum, goes up other side
0. enumi**Friction slows**: Each oscillation loses energy
0. enumi**Settles**: Eventually MUST stop at bottom

Lyapunov's Brilliant Insight

Question: What if we could prove our control system has the same bowl-and-friction property WITHOUT solving differential equations?

Answer: Find an "energy-like" function $V(x)$:

0. At equilibrium: $V = 0$ (bottom of bowl)
- Away from equilibrium: $V > 0$ (higher up bowl)
- Over time: $\dot{V} < 0$ (energy decreases)

Example

SpaceX Rocket: Control engineers design a mathematical "bowl" where vertical upright is the bottom. Control law acts as "friction" dissipating energy. As long as the bowl exists and friction works, the rocket WILL stabilize!

Mathematical Definition

Asymptotic Stability: An equilibrium x^* is asymptotically stable if:

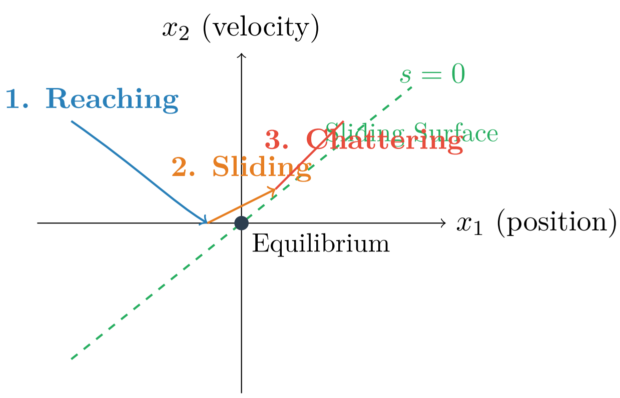
enumiSmall deviations stay small (stable)

0. enumiDeviations actually go to zero over time (asymptotic)

Lyapunov Function: $V(x)$ such that:

0. $V(x^*) = 0$ and $V(x) > 0$ for $x \neq x^*$
- $\dot{V}(x) < 0$ for all $x \neq x^*$

Sliding Mode Control: The Guardrail Analogy



The Mountain Hiking Story

Scenario: Foggy mountain, wind gusts, trying to reach cabin at bottom. Someone built a guardrail path.

Strategy:

enumi**Reaching Phase**: Get TO the guardrail path (don't care about efficiency, just reach it fast)

0. enumi**Sliding Mode**: Follow the path - guardrail geometry guides you safely to bottom

Key Concept

Magic Property: Once on the sliding surface, the system is **insensitive to matched uncertainties**!
When wind hits you (disturbance), you just press harder against the guardrail to compensate. The path geometry naturally rejects disturbances.

Two-Phase Design

Phase 1: Design the Path (Sliding Surface)

Define sliding surface $s(x) = 0$ such that staying on it $\Rightarrow$ exponential convergence to equilibrium
For DIP: Combine angles and velocities in specific relationship

Phase 2: Design the Push (Reaching Law)

Control law that drives system TO surface and keeps it there:

- **Reaching term**: Strong push toward surface (sign function provides direction)
- **Damping term**: Prevents overshoot (slows down as you approach)

The Chattering Problem

What is Chattering?

Common Pitfall

Imagine balancing on a tightrope with instant corrections: lean left, lean right, left, right - infinitely fast.
In practice: Muscles have response time, measurements have noise, sampling is finite.
Result: Rapid oscillations (buzzing sound like cicada or dot-matrix printer)
Bad for: Actuator wear, energy waste, can excite unmodeled dynamics

Boundary Layer Fix

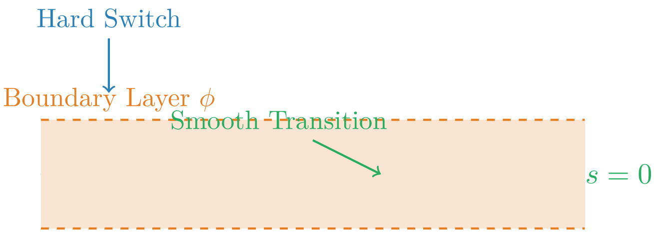
Instead of hard switching, use smooth approximation near surface.

Pro Tip

Trade-off:

- **Wider layer**: Smoother, less chattering, slightly less accurate
- **Narrower layer**: More aggressive, better tracking, more chattering

DIP typical: $\phi = 0.3$ to 0.5



Super-Twisting: The Smooth Operator

Key Concept

Problem with Classical SMC: Sliding surface $s \to 0$, but derivative $\dot{s}$ still has discontinuity $\Rightarrow$ chattering persists
Super-Twisting Solution: Make BOTH s and $\dot{s}$ go to zero simultaneously $\Rightarrow$ truly continuous control

Three Big Advantages

enumi**Dramatically reduced chattering**: Control signal is continuous (no buzzing!)

0. enumi**Finite-time convergence**: Still reaches surface in finite time

0. enumi**Robust to smooth disturbances**: Handles Lipschitz disturbances

Example

Tightrope analogy:
Classical SMC: Oscillate left-right-left-right rapidly
Super-Twisting: Smoothly glide to center and stop - no oscillation!

The Fractional Power Trick

Control Law

Two components:

0. enumi**Integral term**: Gradually builds force from accumulated error
0. enumi**Proportional term with** $\sqrt{|s|}$: Square root provides automatic gain scheduling

Why square root?

- Far from surface (large s): $\sqrt{s}$ still significant $\Rightarrow$ strong control
- Close to surface (small s): $\sqrt{s}$ makes error even smaller $\Rightarrow$ gentle control

Result: ABS brakes - smooth, continuous corrections without harsh on-off behavior

Code is Remarkably Simple

Quick Summary

enumiCompute proportional term: $K_2 \cdot \text{sign}(s) \cdot \sqrt{|s|}$

0. enumiUpdate integral term: Accumulate signed error over time, scaled by K_1
0. enumiSum them: $u = u_{\text{integral}} + u_{\text{proportional}}$

No if-statements, no hard switches - just smooth mathematical functions!

Adaptive SMC: The Smart Learner

Problem with Fixed Gains

Common Pitfall

Delivery truck example:

0. Empty truck: Gains too stiff
- Loaded truck: Gains too soft

Must tune for worst-case ⇒ wasting energy during normal operation

Adaptive Solution

Key Concept

Controller adjusts its own gains in real-time:

- Large error: "Increase gain!"
- Small error: "Ease off gains"

Dead zone: Don't adapt to noise (only genuine large errors)

Lyapunov-Based Adaptation

Theoretical Justification

Construct Lyapunov function including:

- Sliding surface error
- Gain error (difference between current and ideal)

Adaptation law designed so combined "energy" always decreases ⇒ mathematically proven stability
Beautiful part: Even without knowing ideal gain, math drives gains toward stable value automatically!

Implementation Features

Quick Summary

enumi**Dead Zone**: If $|s| <$ threshold, don't adapt (ignore noise)

0. enumi**Gain Leak**: Slowly decrease gains in dead zone (prevents ratcheting)

0. enumi**Bounded Adaptation**: Enforce min/max limits (don't go to zero or infinity)

Update rule: Large error ⇒ increase gains proportionally. Small error ⇒ gently leak gains down. Always stay within bounds.

Robustness Properties

Matched Uncertainties

Key Concept

Disturbances in control channel:

$$\dot{x} = f(x) + (B_0 + \Delta B)u + Bd$$

SMC Property: Complete rejection once on sliding surface!
Why: On $s = 0$, disturbance d cancels out in $\dot{s} = 0$ equation

Unmatched Uncertainties

Common Pitfall

Disturbances NOT in control channel:

$$\dot{x} = f(x) + d_{\text{unmatched}} + Bu$$

SMC **cannot** perfectly reject these, but can attenuate them

Quick Reference: Key Equations

Sliding Surface

$$s = C \cdot e + \dot{e}$$

Where $e = x - x_{\text{desired}}$ (tracking error), C = convergence rate

Classical SMC Control Law

$$u = u_{\text{eq}} + u_{\text{sw}}$$

u_{eq} = equivalent control (feedforward), u_{sw} = switching control (feedback)

Super-Twisting Control Law

$$u = -K_1 \cdot \text{sign}(s) \cdot |s|^{1/2} + u_{\text{int}}$$
$$\dot{u}_{\text{int}} = -K_2 \cdot \text{sign}(s)$$

Adaptive Gain Update

$$\dot{K} = \begin{cases} \gamma \cdot |s| & \text{if } |s| > \epsilon \\ -\lambda K & \text{if } |s| \leq \epsilon \end{cases}$$

γ = adaptation rate, λ = leak rate, ϵ = dead zone

Controller Comparison

Performance Trade-offs

Controller	Pros	Cons
Classical SMC	Simple, robust, proven	Chattering
STA-SMC	Smooth, low chattering	More complex tuning
Adaptive SMC	Self-tuning, efficient	Slower transients

What's Next?

Key Concept

E003: Plant Models & Dynamics - Lagrangian mechanics, Coriolis forces, complete DIP physics
E004: PSO Optimization - How to automatically tune all these gains
Remember: Theory is beautiful, but PSO gives us the practical gains that make it work on real hardware!