

2025-11-01

Quick Reference Command Sheet

DIP-SMC-PSO Project

Print this reference and keep it near your keyboard during the 14-day immersion

CRITICAL: Platform-Specific Commands

Windows (Your Platform):

- Use python NOT python3 (python3 causes exit code 49 on Windows)
- Example: python simulate.py [OK] | python3 simulate.py [ERROR]

Essential Simulation Commands

Basic Simulations

```
lstnumber# Run Classical SMC with plot
lstnumberpython simulate.py --ctrl
    classical_smc --plot
lstnumber
lstnumber# Run Super-Twisting SMC
lstnumberpython simulate.py --ctrl sta_smc --
    plot
lstnumber
lstnumber# Run Adaptive SMC
lstnumberpython simulate.py --ctrl
    adaptive_smc --plot
lstnumber
lstnumber# Run Hybrid Adaptive STA-SMC
lstnumberpython simulate.py --ctrl
    hybrid_adaptive_sta_smc --plot
lstnumber
lstnumber# Run Swing-Up Controller
lstnumberpython simulate.py --ctrl
    swing_up_smc --plot
lstnumber
lstnumber# Run MPC (experimental)
lstnumberpython simulate.py --ctrl mpc --plot
```

Load Optimized Gains

```
lstnumber# Load pre-tuned gains from JSON
lstnumberpython simulate.py --load
    tuned_gains.json --plot
lstnumber
lstnumber# Load your Day 3 optimized gains
lstnumberpython simulate.py --load day3_gains
    .json --plot
```

Configuration Management

```
lstnumber# Print current configuration
lstnumberpython simulate.py --print-config
lstnumber
lstnumber# Use custom config file
lstnumberpython simulate.py --config
    custom_config.yaml --plot
lstnumber
lstnumber# Use your Day 12 capstone config
lstnumberpython simulate.py --config
    capstone_config.yaml --plot
```

PSO Optimization Commands

Tune Controller Gains

```
lstnumber# Optimize Classical SMC gains
lstnumberpython simulate.py --ctrl
    classical_smc --run-pso --save
    gains_classical.json
lstnumber
lstnumber# Optimize with seed for
    reproducibility
lstnumberpython simulate.py --ctrl
    adaptive_smc --run-pso --seed 42 --save
    gains_adaptive.json
lstnumber
lstnumber# Optimize Hybrid controller (Day 6)
lstnumberpython simulate.py --ctrl
    hybrid_adaptive_sta_smc --run-pso --save
    gains_hybrid.json
lstnumber
lstnumber# Quick test run (fewer iterations)
lstnumberpython simulate.py --ctrl sta_smc --
    run-pso --pso-iters 10 --save quick_test
    .json
```

PSO Result Analysis

```
lstnumber# View saved gains
lstnumbercat gains_classical.json
lstnumber
lstnumber# Compare multiple gain sets
lstnumberpython -c "import json; print(json.
    load(open('gains_classical.json')))"
```

Testing Commands

Run Tests

```
lstnumber# Run ALL tests
lstnumberpython -m pytest tests/ -v
lstnumber
lstnumber# Run controller tests only
lstnumberpython -m pytest tests/
    test_controllers/ -v
lstnumber
lstnumber# Run specific controller test
```

```
lstnumberpython -m pytest tests/
    test_controllers/test_classical_smc.py -v
lstnumber
lstnumber# Run integration tests
lstnumberpython -m pytest tests/
    test_integration/ -v
lstnumber
lstnumber# Run benchmarks
lstnumberpython -m pytest tests/
    test_benchmarks/ --benchmark-only
```

Hardware-in-the-Loop (HIL)

Run HIL Simulations (Day 10)

```
lstnumber# Basic HIL run with plot
lstnumberpython simulate.py --run-hil --plot
lstnumber
lstnumber# HIL with custom config
lstnumberpython simulate.py --config
    custom_config.yaml --run-hil
lstnumber
lstnumber# HIL with specific controller
lstnumberpython simulate.py --ctrl
    classical_smc --run-hil --plot
```

Web Interface

Launch Streamlit App

```
lstnumber# Start web interface (Day 1)
lstnumberstreamlit run streamlit_app.py
lstnumber
lstnumber# Access at: http://localhost:8501
```

Coverage Reports

```
lstnumber# Generate HTML coverage report (Day 8)
lstnumberpython -m pytest tests/ --cov=src --
    cov-report=html
lstnumber
lstnumber# View coverage in browser (Windows)
lstnumberstart htmlcov/index.html
lstnumber
lstnumber# Generate terminal coverage summary
lstnumberpython -m pytest tests/ --cov=src --
    cov-report=term
```

Documentation Commands

Build Sphinx Documentation (Day 13)

```
lstnumber# Build HTML documentation
lstnumbersphinx-build -M html docs docs/
    _build
lstnumber
lstnumber# Build with warnings as errors
lstnumbersphinx-build -M html docs docs/
    _build -W --keep-going
lstnumber
lstnumber# View built docs (Windows)
lstnumberstart docs/_build/html/index.html
```

Verify Documentation Changes

```
lstnumber# Check if file was copied to build
lstnumberpython -c "import os; print(os.path.
    getmtime('docs/_static/style.css'))"
lstnumber
lstnumber# Verify localhost serves new
    content
lstnumbercurl -s "http://localhost:9000/
    _static/style.css" | head -n 20
```

Quick Test During Development

```
lstnumber# Run tests for file you're working
    on
lstnumberpython -m pytest tests/
    test_controllers/test_adaptive_smc.py -v
lstnumber
lstnumber# Run with verbose output for
    debugging
lstnumberpython -m pytest tests/
    test_controllers/ -vv
```

Git Commands for Learning

Basic Workflow (Day 11, Day 14)

```
lstnumber# Check repository status
lstnumbergit status
lstnumber
lstnumber# Create feature branch
lstnumbergit checkout -b feature/my-
    experiment
```

```
lstnumber
lstnumber# Stage changes
lstnumbergit add .
lstnumber
lstnumber# Commit with message
lstnumbergit commit -m "feat: Complete Day 14
    capstone project"
lstnumber
lstnumber# View recent commits
lstnumbergit log --oneline -10
lstnumber
lstnumber# Compare with main branch
lstnumbergit diff main...HEAD
```

Recovery (If You Make Mistakes)

```
lstnumber# Undo uncommitted changes
lstnumbergit checkout -- filename.py
lstnumber
lstnumber# Undo last commit (keep changes)
lstnumbergit reset --soft HEAD~1
lstnumber
lstnumber# View what changed
lstnumbergit diff
```

Project Recovery Commands

Session Recovery (Day 1, After Token Limits)

```
lstnumber# One-command recovery
lstnumberbash .ai_workspace/tools/recovery/
    recover_project.sh
lstnumber
lstnumber# Check roadmap progress
lstnumberpython .ai_workspace/tools/analysis/
    roadmap_tracker.py
lstnumber
lstnumber# View project state
lstnumbercat .ai_workspace/state/
    project_state.json
```

Quick Status Check

```
lstnumber# Verify environment
lstnumberpython --version # Should show 3.9+
lstnumberpip list | grep numpy
lstnumber
lstnumber# Check git remote
lstnumbergit remote -v
lstnumber# Should show: https://github.com/
    theSadeQ/dip-smc-psy.git
```

File Navigation Commands

Find Files

```
lstnumber# Locate controller implementations
lstnumberfind src/controllers -name "*.py" -
    type f
lstnumber
lstnumber# Find all test files
lstnumberfind tests -name "test_*.py"
lstnumber
lstnumber# Find config files
lstnumberfind . -name "*.yaml" -o -name "*.
    json"
```

Quick File Viewing

```
lstnumber# View controller source
lstnumbercat src/controllers/smc/algorithms/
    classical_smc.py
lstnumber
lstnumber# Count lines in file
lstnumberpython -c "print(len(open('src/
    optimizer/pso_optimizer.py').readlines()
))"
lstnumber
lstnumber# View first 50 lines
lstnumberhead -n 50 src/core/
    simulation_runner.py
```

Data Analysis Commands

View Benchmark Results (Day 9)

```
lstnumber# Navigate to comparative
    experiments
lstnumbercd academic/paper/experiments/
    comparative
lstnumber
lstnumber# List available benchmarks
lstnumberls -lh
lstnumber
lstnumber# View specific results
lstnumbercat MT-5
    _comprehensive_benchmark_summary.txt
```

Quick Data Inspection

```
lstnumber# View PSO optimization logs
lstnumbercat academic/logs/pso/optimization_
    *.log
lstnumber
lstnumber# View benchmark logs
lstnumbercat academic/logs/benchmarks/*.log
```

Common Troubleshooting

Dependency Issues

```
lstnumber# Reinstall dependencies
lstnumberpip install -r requirements.txt --
    upgrade
lstnumber
lstnumber# Verify critical packages
lstnumberpython -c "import numpy; print(numpy
    __version__)"
lstnumberpython -c "import scipy; print(scipy
    __version__)"
```

Configuration Validation

```
lstnumber# Validate config file
lstnumberpython -c "from src.config import
    load_config; load_config('config.yaml')"
```

Clear Caches

```
lstnumber# Clear pytest cache
lstnumberrm -rf .pytest_cache
lstnumber
lstnumber# Clear Python bytecode
lstnumberfind . -type d -name "__pycache__" -
    exec rm -rf {} +
```

Keyboard Shortcuts (Windows)

- Ctrl+C - Stop running simulation or command
- Ctrl+L - Clear terminal screen
- Ctrl+R - Search command history
- Tab - Autocomplete file/directory names
- Up Arrow - Previous command
- Ctrl+Shift+R - Hard refresh browser (Day 13)

Daily Workflow Template

Morning

```
lstnumber# 1. Check git status
lstnumbergit status
lstnumber
lstnumber# 2. Review yesterday's outputs
lstnumberls academic/logs/
lstnumber
lstnumber# 3. Start Streamlit (background)
lstnumberstart streamlit run streamlit_app.py
```

Afternoon

```
lstnumber# 4. Run experiments from checklist
lstnumberpython simulate.py --ctrl <
    CONTROLLER> --plot
lstnumber
lstnumber# 5. Run tests if you modified code
lstnumberpython -m pytest tests/
    test_controllers/ -v
```

Evening

```
lstnumber# 6. Commit your work
lstnumbergit add .
lstnumbergit commit -m "docs: Complete Day X
immersion tasks"
lstnumber
lstnumber# 7. Review progress
lstnumbercat .ai_workspace/edu/
immersion_schedule/progress_tracker.md
```

Emergency Commands

Stuck in Infinite Loop

Press Ctrl+C to interrupt

Simulation Not Plotting

```
lstnumber# Check matplotlib backend
lstnumberpython -c "import matplotlib; print(
matplotlib.get_backend())"
```

Out of Memory

```
lstnumber# Check Python process memory (
Windows)
lstnumbertasklist /fi "imagename eq python.
exe" /fo table
```

Learning Tips

enumi**Copy-paste these commands** - Don't type from scratch during immersion

- 0. enumi**Keep terminal history** - Use Up Arrow to repeat commands
- 0. enumi**Run commands in parallel** - Open multiple terminals (Day 10+)
- 0. enumi**Document failures** - Note which commands error for troubleshooting
- 0. enumi**Bookmark this file** - Refer to it 20+ times per day

Next Steps After Day 14

```
lstnumber# Begin Tutorial 01
lstnumbercat docs/guides/getting-started.md
lstnumber
lstnumber# View available research tasks
lstnumberls .ai_workspace/planning/research/
lstnumber
lstnumber# Explore advanced examples
lstnumberpython simulate.py --help
```

0. Print Date: _____

Keep this reference visible during all 14 days of immersion!