2025-11-01

E026: Appendix Reference Part 2

DIP-SMC-PSO Educational Series

January 25, 2026

Overview

This episode covers appendix reference part 2 from the DIP-SMC-PSO project.

Part: Appendix
Duration: 15-20 minutes
Source: Comprehensive Presentation Materials

section0 New Controller Variants

Planned Controller Extensions:

- **Terminal SMC (TSMC)**

- Finite-time convergence guarantees - Faster settling than classical SMC - *Reference:* Yu et al. (2005)

- **Integral SMC (ISMC)**

- Eliminate steady-state error - Improved disturbance rejection - *Reference:* Utkin & Shi (1996)

- **Higher-Order SMC (HOSMC)**

- 3rd-order super-twisting - Further chattering reduction - *Reference:* Levant (2007)

- **Neural Network SMC (NN-SMC)**

- Learn unknown dynamics online - Adaptive to model uncertainty - *Reference:* Li et al. (2018)

section0 Advanced PSO Variants

Optimization Algorithm Enhancements:

- **Multi-Objective PSO (MOPSO)**

- Pareto-optimal solutions - Trade-off settling time vs. energy vs. chattering - *Tool:* DEAP library (Python)

- **Adaptive PSO (APSO)**

- Time-varying inertia weight $w(t)$ - Adaptive cognitive/social coefficients - *Reference:* Zhan et al. (2009)

- **Hybrid PSO-GA**

- Combine PSO exploration with GA exploitation - Mutation operator for diversity - *Tool:* Optuna (hyperparameter optimization)

- **Bayesian Optimization**

- Gaussian process surrogate model - Sample-efficient for expensive simulations - *Tool:* scikit-optimize

section0 Real-Time Control Enhancements

Production-Grade Real-Time System:

- **Hard Real-Time Scheduler**

- Preemptive task scheduling - Guaranteed deadline compliance (100 - *Platform:* PREEMPT_RT Linux kernel

- **Predictive Latency Compensation**

- Estimate future state during network delay - Smith predictor for delayed systems - *Reference:* Åström & Wittenmark (1990)

- **Multi-Rate Control**

- Fast inner loop (1 kHz), slow outer loop (100 Hz) - Hierarchical control architecture - *Application:* High-frequency chattering reduction

- **Event-Triggered Control**

- Update control only when error exceeds threshold - Reduce communication bandwidth - *Reference:* Heemels et al. (2012)

section0 Hardware Deployment

Physical Testbed Development:

- **Actuator Selection**

- DC motor with encoder (cart actuation) - Torque: 50 Nm, speed: 3000 RPM - *Vendor:* Maxon Motor, Faulhaber

- **Sensor Suite**

- High-precision encoders (1000 CPR) - IMU for pendulum angles (1 kHz) - Force sensor for control input validation

- **Embedded Controller**

- Raspberry Pi 4 or NVIDIA Jetson Nano - Real-time Linux (PREEMPT_RT patch) - *Interface:* Python + GPIO/SPI/I2C

- **Safety Mechanisms**

- Physical limit switches (cart position) - Emergency stop button (hardware interrupt) - Mechanical dampers (pole protection)

section0 Machine Learning Integration

Data-Driven Control Enhancements:

- **System Identification**

- Learn accurate dynamics from experimental data - Neural network or Gaussian process models - *Tool:* PyTorch, TensorFlow, scikit-learn

- **Reinforcement Learning (RL)**

- Learn optimal control policy via trial-and-error - Compare SMC vs. RL performance - *Tool:* Stable-Baselines3 (PPO, SAC, TD3)

- **Transfer Learning**

- Pre-train on simulation, fine-tune on hardware - Reduce real-world training time - *Technique:* Domain adaptation, sim-to-real

- **Hybrid SMC-RL**

- Use SMC for safety guarantees - Use RL for performance optimization - *Reference:* Cheng et al. (2019)

Resources

- **Repository:** `https://github.com/theSadeQ/dip-smc-pso.git`
- **Documentation:** See `docs/` directory
- **Getting Started:** `docs/guides/getting-started.md`

Educational podcast episode generated from comprehensive presentation materials