Section 7Performance Comparison Results

December 25, 2025

1 Performance Comparison Results

1.1 Computational Efficiency

Table 7.1: Compute Time Comparison

[TABLE - See Markdown version for details]

Key Finding: All controllers meet hard real-time constraints (¡50 mus budget for 100 mus cycle), as shown in Figure 7.1. **Classical SMC** provides fastest computation (18.5 mus baseline), suitable for resource-constrained embedded systems. STA and Hybrid add 31-45percent overhead but remain well within real-time feasibility (illustrated in Figure 7.1, error bars representing 95percent bootstrap confidence intervals).

Statistical Significance: Welch's t-test shows significant difference between Classical and Adaptive (p¡0.001), confirming computational cost of online adaptation (see Figure 7.1 for mean compute time comparison with confidence intervals).

[FIGURE - See Markdown version]

Figure 7.1: Computational Efficiency Comparison Across SMC Variants. Bar chart displays mean control law compute time for four controllers with 95percent bootstrap confidence intervals (error bars) from 1,000 replicate simulations on Intel i7-9700K (3.6 GHz, single core). **Classical SMC** achieves fastest execution (18.5 ± 2.1 mus baseline), validating simple proportional-derivative sliding surface advantage for resource-constrained embedded systems. **STA-SMC** adds 31percent overhead (24.2 mus) due to continuous fractional power computation ($|\sigma|^{1/2}$) and integral state update, while Hybrid Adaptive STA requires 26.8 mus (+45percent vs Classical) for mode switching logic. **Adaptive SMC** shows highest compute time (31.6 mus, +71percent vs Classical) attributable to online parameter estimation gradient computation and Lyapunov adaptation law evaluation. Red dashed horizontal line indicates hard real-time budget (50 mus for 10 kHz control rate with 100 mus cycle period), demonstrating all variants achieve real-time feasibility with substantial headroom (68-81percent margin). Welch's t-test confirms statistically significant difference between Classical and Adaptive (t=8.47, p¡0.001, Cohen's d=3.52 very large effect), validating computational cost of adaptation. Data supports controller selection guideline: embedded IoT systems with ¡1 MHz processors favor **Classical SMC**; performance-critical applications tolerate STA overhead for transient response gains (Section 7.2).

—

1.2 Transient Response Performance

Table 7.2: Settling Time and Overshoot Comparison

[TABLE - See Markdown version for details]

Key Finding: STA SMC achieves fastest settling (1.82s, 16percent faster than Classical) and lowest overshoot (2.3percent, 60percent better than Classical), as shown in Figure 7.2, validating theoretical finite-time convergence advantage. **Adaptive SMC** trades transient performance (slowest at 2.35s) for robustness to model uncertainty.

Performance Ranking (Settling Time, see Figure 7.2 left panel): - STA SMC: 1.82s (BEST) - Hybrid STA: 1.95s (+7percent vs STA) - **Classical SMC**: 2.15s (+18percent vs STA) - **Adaptive SMC**: 2.35s (+29percent vs STA)

Statistical Validation: Bootstrap 95percent CIs confirm STA significantly outperforms others (non-overlapping intervals, illustrated in Figure 7.2 error bars). Cohen's d = 2.14 (large effect size) for STA vs Classical comparison.

[FIGURE - See Markdown version]

Figure 7.2: Transient Response Performance Comparison. Left panel shows settling time (2percent criterion) across four SMC variants, with **STA-SMC** achieving fastest convergence (1.82s ± 0.15s, 95percent CI), validating finite-time convergence theoretical advantage over **Classical SMC**'s asymptotic stability (2.15s ± 0.18s). Right panel presents overshoot%ages, revealing **STA-SMC**'s superior transient quality (2.3percent ± 0.4percent) compared to Classical (5.8percent ± 0.8percent) and Adaptive (8.2percent ± 1.1percent). Error bars represent 95percent bootstrap confidence intervals from Monte Carlo analysis (n=400 trials). Cohen's d = 2.14 for STA vs Classical comparison indicates large practical significance. Hybrid Adaptive STA achieves intermediate performance (1.95s settling, 3.5percent overshoot), demonstrating tradeoff between adaptation capability and transient speed. Data validates theoretical predictions from Lyapunov analysis in Section 4, with experimental settling times within 8percent of predicted values.

—

1.3 Chattering Analysis

Table 7.3: Chattering Characteristics

[TABLE - See Markdown version for details]

Key Finding: STA SMC achieves 74percent chattering reduction vs **Classical SMC** (index 2.1 vs 8.2), as shown in Figure 7.3 (left panel), validating continuous control law advantage. **Adaptive SMC** exhibits highest chattering (index 9.7) due to rapid gain changes during online estimation.

FFT Analysis: STA shows dominant low-frequency content (¡10 Hz), while Classical and Adaptive exhibit significant high-frequency components (30-40 Hz) characteristic of boundary layer switching (illustrated in Figure 7.3 right panel).

Practical Implications (based on Figure 7.3 chattering index and frequency content analysis): - STA: Minimal actuator wear, quieter operation, suitable for precision applications (2.1percent high-frequency energy) - Classical: Moderate chattering acceptable for industrial use (12.3percent high-frequency energy) - Adaptive: Higher wear requires robust actuators (15.1percent high-frequency energy)

[FIGURE - See Markdown version]

Figure 7.3: Chattering Characteristics Analysis. Left panel displays chattering index (root-mean-square of control derivative) revealing **STA-SMC**'s 74percent reduction compared to **Classical SMC** (2.1 vs 8.2 N/s), with green annotation highlighting this key finding. **Adaptive SMC** exhibits highest chattering (9.7 N/s) due to rapid gain adjustments during online parameter estimation. Right panel quantifies high-frequency energy content (¿10 Hz band) from FFT power spectrum analysis: **STA-SMC** shows 2.1percent high-frequency energy (dominant content ¡10 Hz), validating continuous control law advantage, while Adaptive exhibits 15.1percent (peak frequency

42 Hz) characteristic of aggressive boundary layer switching. **Classical SMC** demonstrates intermediate behavior (12.3percent high-frequency, 35 Hz peak). Chattering index computed as RMS of —du/dt— over 10s simulation window. Data illustrates fundamental tradeoff: discontinuous control (Classical, Adaptive) achieves robust sliding at cost of high-frequency switching, while continuous super-twisting maintains convergence guarantees with smooth actuation suitable for precision applications requiring minimal actuator wear and acoustic noise.

—

1.4 Energy Efficiency

Table 7.4: Control Energy Consumption

[TABLE - See Markdown version for details]

Key Finding: STA SMC most energy-efficient (11.8J baseline for 10s simulation), as shown in Figure 7.4 (left panel), with continuous control law minimizing wasted effort. **Adaptive SMC** highest energy (13.6J, +15percent vs STA) due to adaptive transients.

Energy Budget Breakdown (**Classical SMC** example, see Figure 7.4 for energy distribution): - Reaching phase (0-0.5s): 6.2J (50percent of total) - Sliding phase (0.5-2.1s): 5.8J (47percent) - Steady-state (¿2.1s): 0.4J (3percent)

Hardware Implications: All controllers ¡15J typical for 10s stabilization, safe for 250W actuators (illustrated in Figure 7.4 right panel for peak power). Battery-powered systems prefer STA (most efficient controller, 11.8J total energy with 8.2W peak power).

[FIGURE - See Markdown version]

Figure 7.4: Control Energy Consumption Analysis. Left panel displays total control energy integrated over 10-second stabilization simulation, revealing **STA-SMC** as most energy-efficient controller (11.8 ± 0.9 J, baseline), with continuous super-twisting control law minimizing wasted actuation effort. Hybrid Adaptive STA achieves second rank (12.3 J, +4percent overhead vs STA) through intelligent mode switching between classical and adaptive strategies. **Classical SMC** requires 12.4 J (+5percent vs STA), while **Adaptive SMC** exhibits highest energy consumption (13.6 J, +15percent vs STA) due to transient oscillations during online parameter estimation phase. Error bars represent 95percent confidence intervals from 400 Monte Carlo trials. Right panel shows peak instantaneous power consumption: STA maintains lowest peak (8.2 W), Classical intermediate (8.7 W), and Adaptive highest (10.3 W) attributable to aggressive gain adaptation transients. Green annotation highlights STA as "Most Efficient" controller for battery-powered applications. Energy budget breakdown (**Classical SMC** example): reaching phase (0-0.5s) consumes 50percent of total (6.2 J), sliding phase (0.5-2.1s) 47percent (5.8 J), steady-state maintenance only 3percent (0.4 J), validating SMC energy concentration during transient convergence. All controllers remain well below 250W actuator thermal limits (¡15 J typical for 10s operation), supporting deployment feasibility. Data validates theoretical prediction: continuous control (STA) reduces control effort variance compared to discontinuous switching (Classical, Adaptive), achieving superior energy efficiency alongside chattering reduction (Figure 7.3).

—

1.5 Overall Performance Ranking

Multi-Objective Assessment:

[TABLE - See Markdown version for details]

1.6 Interpreting Statistical Significance

This section translates statistical metrics into practical meaning, helping practitioners without deep statistics backgrounds understand what the performance comparison results actually tell us.

—

7.6.1 Effect Size Interpretation (Cohen's d)

Cohen's d quantifies how different two groups are in standardized units (standard deviations apart). It measures practical significance, complementing p-values which only indicate statistical significance.

Cohen's d Interpretation Guidelines:

[TABLE - See Markdown version for details]

Numerical Example: Classical vs STA Settling Time (Table 7.2)

Given Data: - **Classical SMC**: mu = 2.15s, $\sigma = 0.18$s - STA SMC: mu = 1.82s, $\sigma = 0.15$s

Cohen's d Calculation:

Practical Interpretation: - Absolute difference: 0.33s (18percent improvement) - Standardized difference: 2.00 standard deviations apart - Overlap: Only 2percent of Classical trials settle faster than median STA trial - Real-world impact: For 1000 stabilization cycles/day: - Daily time savings: 1000 x 0.33s = 330 seconds = 5.5 minutes/day - Annual savings: 5.5 min/day x 365 days = 33.4 hours/year - For time-critical applications (e.g., robotic surgery), 330ms per cycle is highly significant

Is This Difference Meaningful? - For slow processes (10s cycle time): 18percent = 1.8s difference -¿ Marginal (other factors dominate) - For fast processes (100ms cycle time): 18percent = 18ms difference -¿ Critical (affects throughput) - For this DIP system (2s nominal settling): 330ms savings -¿ Significant (enables faster maneuvers)

Effect Size vs Statistical Significance: - p-value ¡0.001: Tells us the difference is unlikely due to chance (statistical significance) - Cohen's d = 2.00: Tells us the difference is large in magnitude (practical significance) - Both must be satisfied for confident recommendation (Section 7.7 decision framework uses both)

—

7.6.2 Confidence Interval Interpretation

Confidence intervals quantify uncertainty in our estimates. A 95percent CI means: "If we repeated the experiment 100 times, 95 of those intervals would contain the true value."

Overlapping vs Non-Overlapping Intervals:

Table 7.6: Confidence Interval Overlap Analysis

[TABLE - See Markdown version for details]

Interpretation Rules: - No overlap: Strong evidence of real difference (high confidence in superiority claim) - Partial overlap: Moderate evidence (difference likely but not certain) - Full overlap: Weak/no evidence (cannot confidently claim difference)

Example Interpretation (Overshoot): - Classical: 5.8percent ± 0.8percent -¿ 95percent CI [5.0, 6.6]percent - STA: 2.3percent ± 0.4percent -¿ 95percent CI [1.9, 2.7]percent - No overlap -¿ Even in worst-case STA trial (2.7percent), still better than best-case Classical (5.0percent) - Conclusion: Can confidently recommend STA for overshoot-critical applications

Example Interpretation (Energy): - Classical: 12.4 ± 1.2J -¿ 95percent CI [11.2, 13.6]J - STA: 11.8 ± 0.9J -¿ 95percent CI [10.9, 12.7]J - Partial overlap [11.2, 12.7]J -¿ Some Classical trials consume less energy than some STA trials - Conclusion: STA's energy advantage (5percent) not statistically significant -¿ Both controllers equivalent for energy-critical applications

—

7.6.3 P-Value Interpretation

What p-value Actually Means: - p¡0.05: ”If controllers were truly identical, ¡5percent chance of observing this difference by random chance alone” - p¡0.001: ”If controllers were truly identical, ¡0.1percent chance of observing this difference” - NOT: ”95percent probability STA is better” (common misconception)

P-Value Thresholds in This Study:

[TABLE - See Markdown version for details]

Multiple Comparisons Correction: - 6 pairwise comparisons (4 controllers choose 2) -¿ Bonferroni correction: $\alpha = 0.05/6 = 0.0083$ - Only comparisons with p¡0.0083 declared statistically significant after correction - Section 7 results: 8/12 comparisons remain significant after correction (robust findings)

—

7.6.4 Sample Size and Variability

Why n=400 Trials for QW-2 Benchmark?

Power analysis (Section 6.3) showed: - To detect 15percent difference in settling time (effect size d=0.5) - With 80percent power (1-beta = 0.80) - At alpha=0.05 significance level - Required: n=100 trials per controller (400 total)

Variability Sources: - Stochastic disturbances: Random sensor noise, friction variations (±10percent settling time) - Numerical integration: RK45 adaptive step size introduces ±2percent variability - PSO optimization: Different random seeds produce slightly different gains (±5percent performance) - Total variability: Captured in confidence intervals (e.g., Classical 2.15 ± 0.18s)

Interpreting Standard Deviations:

[TABLE - See Markdown version for details]

Conclusion: All controllers show consistent performance (CV ¡10percent), validating controller robustness across random disturbances.

—

7.6.5 Practical Significance Decision Matrix

When is a statistical difference practically meaningful?

[TABLE - See Markdown version for details]

Example Application to Section 7 Results:

Scenario: Precision Robotics Application - Thresholds: 5percent settling, 1percent overshoot, 50percent chattering

Classical vs STA Comparison: - Settling: 18percent improvement (STA 1.82s vs Classical 2.15s) -¿ Exceeds 5percent threshold - Overshoot: 60percent reduction (STA 2.3percent vs Classical 5.8percent) -¿ Exceeds 1percent threshold - Chattering: 74percent reduction (STA 2.1 vs Classical 8.2) -¿ Exceeds 50percent threshold

Recommendation: STA SMC strongly recommended for precision robotics (all metrics exceed thresholds)

Scenario: Industrial Automation Application - Thresholds: 10percent settling, 2percent overshoot, 20percent chattering

Classical vs STA Comparison: - Settling: 18percent improvement -¿ Exceeds 10percent threshold - Overshoot: 3.5percent absolute reduction (5.8percent -¿ 2.3percent) -¿ Exceeds 2percent threshold - Chattering: 74percent reduction -¿ Exceeds 20percent threshold

Recommendation: STA SMC recommended for industrial automation (all metrics exceed thresholds, +31percent compute overhead acceptable)

Scenario: Real-Time Embedded System - Thresholds: 15percent settling, 3percent overshoot, chattering not critical

Classical vs STA Comparison: - Settling: 18percent improvement -¿ Exceeds 15percent threshold - Overshoot: 3.5percent reduction -¿ Exceeds 3percent threshold - BUT: Compute time

31percent slower (24.2mus vs 18.5mus) -¿ Tradeoff required

Recommendation: **Classical SMC** preferred if compute budget tight (¡30mus), STA SMC if budget allows (50mus, both feasible)

—

7.6.6 Summary: Statistical Interpretation Checklist

When evaluating controller performance comparisons:

- Check p-value: Is difference statistically significant? (p¡0.05, ideally p¡0.01) - Check Cohen's d: Is difference practically large? (d¿0.5 for medium, d¿0.8 for large) - Check confidence intervals: Do they overlap? (No overlap = strong confidence) - Check application thresholds: Does improvement exceed your application's requirements? - Check tradeoffs: Are there opposing metrics (e.g., faster but more chattering)?

Example Full Analysis: STA vs Classical for Precision Robotics

- p-value: p¡0.001 (highly significant for settling, overshoot, chattering) - Cohen's d: 2.00 settling, 1.08 overshoot, 3.52 chattering (all large/very large) - Confidence intervals: No overlap for overshoot and chattering (strong confidence) - Application thresholds: All metrics exceed precision robotics thresholds - Tradeoffs: +31percent compute time (24.2mus vs 18.5mus) -¿ Acceptable for precision app, still ¡50mus budget

Final Recommendation: STA SMC strongly recommended for precision robotics (robust statistical evidence, large practical effects, acceptable tradeoffs)

1.7 Controller Selection Decision Framework

This section provides practical guidelines for choosing the optimal SMC variant based on application requirements, converting research results into actionable controller selection.

—

7.7.1 Decision Tree for Controller Selection

START: What is your primary constraint?

Quick Selection Heuristic: - Budget ¡30mus? -¿ **Classical SMC** - Chattering critical? -¿ STA SMC - Parameters unknown? -¿ Adaptive or Hybrid - Otherwise? -¿ STA SMC (default best choice)

—

7.7.2 Application-Specific Recommendations

Table 7.7: Controller Recommendations by Application Domain

[TABLE - See Markdown version for details]

Application Category Guidelines:

Category 1: Resource-Constrained Embedded (**Classical SMC**) - Characteristics: ¡1 MHz CPU, ¡16 KB RAM, cost-sensitive - Examples: Industrial PLCs, Arduino automation, legacy systems - Justification: 18.5mus compute time enables deployment on low-end hardware

Category 2: Precision / Low-Noise (STA SMC) - Characteristics: High accuracy required, sensitive to vibration/noise - Examples: Medical devices, optical systems, laboratory equipment - Justification: 74percent chattering reduction (2.1 index) critical for precision

Category 3: Parameter Uncertainty (Adaptive / Hybrid) - Characteristics: Unknown or time-varying parameters (mass, inertia, friction) - Examples: Cranes, material handling, multi-mission robots - Justification: 16percent parameter tolerance (Section 8) handles uncertainty

Category 4: General-Purpose (STA SMC) - Characteristics: Modern hardware (¿10 MHz), balanced requirements - Examples: Drones, mobile robots, electric vehicles - Justification: Best overall performance (Rank 1, 9.0/10 score)

—

7.7.3 Performance Trade-off Matrix

Table 7.8: Weighted Performance Scoring

[TABLE - See Markdown version for details]

How to Use This Matrix:

- Adjust weights based on your application priorities - Recalculate weighted score: Score = (Weight x Rating) - Select controller with highest weighted score

Example 1: Real-Time Embedded Application (Compute Critical) - Adjusted weights: Compute 50percent, Transient 20percent, Chattering 15percent, Energy 10percent, Robustness 5percent - **Classical SMC**: 0.50x10 + 0.20x6 + 0.15x5 + 0.10x7 + 0.05x6 = 8.6/10 (BEST) - STA SMC: 0.50x7 + 0.20x10 + 0.15x10 + 0.10x10 + 0.05x6 = 7.8/10 - Recommendation: **Classical SMC** (compute constraint dominates)

Example 2: Battery-Powered Precision Robot (Energy + Chattering Critical) - Adjusted weights: Compute 10percent, Transient 20percent, Chattering 35percent, Energy 30percent, Robustness 5percent - STA SMC: 0.10x7 + 0.20x10 + 0.35x10 + 0.30x10 + 0.05x6 = 9.5/10 (BEST) - **Classical SMC**: 0.10x10 + 0.20x6 + 0.35x5 + 0.30x7 + 0.05x6 = 6.5/10 - Recommendation: STA SMC (energy + chattering dominate)

Example 3: Unknown Payload Application (Robustness Critical) - Adjusted weights: Compute 15percent, Transient 20percent, Chattering 15percent, Energy 10percent, Robustness 40percent - **Adaptive SMC**: 0.15x5 + 0.20x4 + 0.15x3 + 0.10x4 + 0.40x10 = 6.4/10 (BEST) - Hybrid STA: 0.15x8 + 0.20x8 + 0.15x7 + 0.10x8 + 0.40x9 = 7.9/10 (BETTER!) - Recommendation: Hybrid Adaptive STA (robustness + acceptable other metrics)

—

7.7.4 Deployment Decision Flowchart

—

7.7.5 Common Deployment Scenarios

Scenario 1: Migrating from PID to SMC - Starting point: Existing PID controller (adequate but not optimal) - Recommendation: **Classical SMC** (easiest transition, similar compute budget) - Upgrade path: Classical -¿ STA (when hardware upgraded) -¿ Hybrid (if parameters vary) - Risk mitigation: Validate Classical first, then optimize with STA if performance gap exists

Scenario 2: New Design with Modern Hardware - Starting point: Greenfield project, ARM Cortex-M4+ processor (¿100 MHz) - Recommendation: STA SMC (best overall, hardware supports 24.2mus easily) - Alternative: Hybrid if robustness to parameter uncertainty needed - Cost: No penalty (modern MCUs handle STA overhead trivially)

Scenario 3: Retrofitting Legacy System - Starting point: Existing embedded controller, cannot change hardware - Recommendation: Measure compute budget first (critical constraint) - If budget ¿30mus: STA SMC (performance improvement) - If budget ¡30mus: **Classical SMC** (only feasible option) - Risk: May not have headroom for STA -¿ Classical safer choice

Scenario 4: High-Volume Production (1000s of units) - Starting point: Cost-sensitive, need cheapest MCU meeting specs - Recommendation: **Classical SMC** (enables lowest-cost hardware) - Cost savings: Can use $1 - 2MCU(8 - bit, 16MHz)insteadof$5-10 MCU (32-bit, 100 MHz) - Tradeoff: Accept moderate chattering (8.2 index) for 50-75percent BOM cost reduction

Scenario 5: Research Platform / Testbed - Starting point: Flexible system for algorithm comparison - Recommendation: Implement all 4 controllers (factory pattern, Section 3) - Benefit: Can switch controllers via configuration file, compare empirically - Use: Establish baseline (Classical) -¿ validate STA advantage -¿ test Adaptive if needed

—

7.7.6 Controller Selection Checklist

Before deploying to production, verify:

Technical Validation: - [ ] Compute time measured on target hardware (not development PC) - [ ] Real-time deadline met with 50percent+ margin (safety factor for worst-case) - [ ] Settling time meets application requirement (e.g., ¡2.0s for this DIP) - [ ] Overshoot acceptable for safe operation (e.g., cart stays on track) - [ ] Chattering tested with actual actuator (acoustic noise, wear) - [ ] Energy consumption within power budget (battery life, thermal limits)

Robustness Validation (Section 8 tests): - [ ] Controller tested with ±10percent parameter variations - [ ] Disturbance rejection validated (friction, sensor noise, external forces) - [ ] Numerical stability confirmed (1000+ trials, no NaN/overflow) - [ ] Worst-case performance acceptable (95th%ile settling time)

Implementation Validation: - [ ] Gains optimized via PSO (Section 5) or manual tuning (Section 3.9) - [ ] Boundary layer epsilon tuned for chattering-precision tradeoff - [ ] Integration tolerance appropriate (atol=10$^-6, rtol = 10^-3, Section 6.1) - []Reproducibility verified(seed = 42, bitwise identical results, Section 6.6)$

Deployment Readiness: - [ ] Pre-flight validation protocol passed (Section 6.8, all 5 tests) - [ ] Documentation complete (controller type, gains, parameters) - [ ] Monitoring configured (latency, deadline misses, performance metrics) - [ ] Fallback strategy defined (switch to Classical if STA fails, safe stop mode)

Recommendation Confidence Levels:

[TABLE - See Markdown version for details]

—

7.7.7 Summary: Controller Selection Decision Guide

Quick Decision Table:

[TABLE - See Markdown version for details]

Decision Confidence: - High: Strong statistical evidence (p¡0.01, d¿0.8, CI no overlap) + clear application match - Medium: Moderate evidence (p¡0.05, d¿0.5) or tradeoffs require consideration - Low: Marginal differences (p 0.05, d¡0.5) or conflicting metrics -¿ need extended testing

When in Doubt: - Start with STA SMC (best overall, Rank 1) - If compute budget issues -¿ fallback to **Classical SMC** - If parameter uncertainty issues -¿ upgrade to Hybrid Adaptive STA - Validate choice with pre-flight protocol (Section 6.8)

1.8 Theoretical Predictions vs Experimental Results

This section validates theoretical analysis (Sections 3-4) by comparing predicted performance to experimental measurements, confirming model accuracy and explaining expected deviations.

—

7.8.1 Validation Comparison

Table 7.9: Theoretical Predictions vs Experimental Results

[TABLE - See Markdown version for details]

Overall Validation Assessment: - 15/17 metrics validate theoretical predictions (88percent accuracy) - All settling time predictions accurate within 10percent - All overshoot predictions accurate within ranges - Chattering qualitative predictions confirmed quantitatively - Robustness predictions slightly conservative (theoretical bounds pessimistic by 10-20percent)

—

7.8.2 Sources of Deviation

Why Experimental Results Differ from Theory:

- Theoretical Bounds Are Conservative (Intentionally) - Lyapunov analysis uses worst-case assumptions: - Maximum disturbance: d = 1.5 N (actual disturbances 0.3-0.8 N, Section 6.5) - Minimum control gain: Lower bounds for stability (actual PSO-tuned gains higher) - Parameter uncertainty: ±20percent assumed (actual system ±5percent variation) - Result: Theoretical settling time ≥ experimental (safety margin built-in) - Example: STA predicted ¡2.0s, actual 1.82s (theory guarantees upper bound, not tight estimate)

- Numerical Integration Effects - RK45 adaptive time-stepping smoother than continuous-time model: - Discontinuous sign(sigma) function approximated by steep sigmoid in discrete time - Adaptive step size reduces numerical noise - Integration tolerance atol=10$^{-}6enforcessmoothness - Result : Experimentalchatteringslightlylowerthantheoreticaldiscontinuousmodel - Example :$ **Classical SMC**$chattering8.2(experiment)vs"moderate"(theory)- > quantificationrevealsnumericalsmoothin$

- Boundary Layer Smoothing - Practical implementation uses boundary layer epsilon=0.02: - Theory: Discontinuous control u = K·sign(sigma) - Practice: Continuous approximation u = K·sat(sigma/epsilon) (Section 3.2) - Smoothing reduces chattering at cost of sliding precision - Result: Experimental chattering 60-70percent lower than pure discontinuous control - Trade-off validated: Section 7.3 shows acceptable chattering (8.2 index) while maintaining performance

- PSO Optimization vs Generic Gains - Theoretical analysis uses generic gain values: - Example: K=15, lambda=5 (representative values, Section 3) - No optimization, worst-case parameter assumptions - Experimental setup uses PSO-tuned gains (Section 5): - **Classical SMC**: [5.2, 3.1, 10.5, 8.3, 1.5, 0.91] (optimized for this DIP system) - Multi-objective cost minimizes settling, overshoot, chattering simultaneously - Result: Experimental performance better than theoretical generic gains - Example: Classical settling 2.15s (PSO-tuned) vs 2.2s predicted (generic gains) -¿ 2.3percent improvement

- Monte Carlo Averaging - Experimental results average 400 trials (Section 6.3): - Random disturbances, sensor noise, numerical variations - Outliers (instability, integration failures) excluded - Mean performance better than worst-case single trial - Theoretical analysis considers worst-case single scenario: - Maximum disturbance, worst parameter combination - No averaging, conservative single-shot prediction - Result: Experimental mean = 5-10percent better than theoretical worst-case

—

7.8.3 Validation Interpretation

What Close Agreement Tells Us:

- Model Accuracy Confirmed - DIP dynamics model (Section 2) captures real system behavior - Simplifications (massless links, frictionless joints) acceptable approximations - Numerical values (masses, lengths, inertia) representative of actual hardware

- Lyapunov Analysis Valid - Stability proofs (Section 4) hold in discrete-time implementation - Convergence rate predictions accurate (lambda-dependent exponential decay observed) - Finite-time convergence confirmed for STA (1.82s ¡ 2.0s theoretical bound)

- Controller Implementation Correct - Discretization (dt=0.01s, Euler integration for control law) preserves stability - Boundary layer approximation (epsilon=0.02) adequate for chattering reduction - PSO optimization (Section 5) improves performance beyond generic theoretical gains

What Deviations Tell Us:

- Conservative Theoretical Bounds (Expected) - Robustness predictions 10-20percent pessimistic -¿ provides safety margin in practice - Example: **Adaptive SMC** tolerates 16percent parameter error (predicted 20percent) -¿ still robust, just not quite as generous as theory suggested

- Practical Smoothing Benefits - Boundary layer (epsilon=0.02) reduces chattering significantly (8.2 vs theoretical infinite frequency) - Numerical integration (RK45) inherently smooths discontinuous control - Trade-off validated: Slight sliding precision loss (2percent overshoot increase) for 70percent chattering reduction

- Optimization Value - PSO-tuned gains outperform generic theoretical values by 2-10percent - Multi-objective cost function balances competing metrics effectively - Validates PSO methodology

(Section 5) for practical deployment

—

7.8.4 Confidence in Theoretical Framework

Metrics of Theoretical Framework Quality:

[TABLE - See Markdown version for details]

Overall Confidence: High (theory validated by experiment, deviations explainable and expected)

—

7.8.5 Implications for Future Work

What Validated Theory Enables:

- Extrapolation to Untested Scenarios - Theory validated for this DIP system -¿ likely valid for similar underactuated systems - Can predict performance of: - Different DIP geometries (vary link lengths, masses) - Higher-order systems (triple inverted pendulum) - Different disturbance levels (d = 0.5-3.0 N) - Caution: Extrapolation assumes model structure similar (linear actuator, rigid links)

- Controller Tuning Shortcuts - PSO-tuned gains outperform theory by 2-10percent -¿ validates optimization necessity - But theoretical gain bounds (Section 3.9) provide good starting point (within 15percent of optimal) - Recommendation: Start with theoretical gains, fine-tune with PSO if performance critical

- Deployment Confidence - Close theory-experiment agreement -¿ can trust simulations for preliminary design - Reduces need for extensive hardware prototyping - Workflow: Simulate -¿ Validate theory -¿ Deploy with confidence

What Deviations Suggest for Improvement:

- Tighter Robustness Bounds - Theoretical ±20percent conservative -¿ could refine Lyapunov analysis with tighter assumptions - **Adaptive SMC** actual tolerance ±16percent -¿ suggests adaptation law could be more aggressive - Future work: Revisit Lyapunov conditions, explore faster adaptation (higher gamma gain)

- Chattering Quantification - Theory predicts ”moderate/low/high” (qualitative) -¿ experiment quantifies (8.2, 2.1, 9.7 indices) - Future work: Develop analytical chattering index formula from boundary layer theory - Would enable chattering prediction without simulation

- Boundary Layer Optimization - Current epsilon=0.02 reduces chattering 70percent with acceptable precision loss - Future work: Formalize epsilon selection (currently empirical, Section 3.9) - Trade-off curve: chattering vs sliding precision for optimal epsilon choice

—

7.8.6 Summary: Theory-Experiment Validation

Validation Scorecard:

[TABLE - See Markdown version for details]

Bottom Line: - Theoretical framework validated by experimental results (88percent accuracy) - Deviations expected and explainable (conservative bounds, practical smoothing, optimization) - High confidence in using theory for controller design, simulation, and deployment - Minor opportunities for theory refinement (tighter robustness bounds, chattering quantification)

Recommendation for Practitioners: - Use theoretical predictions for preliminary design (settling time, overshoot ranges) - Apply PSO optimization for 2-10percent performance improvement beyond theory - Validate on hardware before production deployment (theory accurate but not perfect) - Trust simulation results for rapid prototyping (close theory-experiment agreement)

—