

2025-11-01

E006: Analysis and Visualization Tools

From Gigabytes of Data to Publication Figures

Part 2 · Duration: 15-20 minutes

Beginner-Friendly Visual Study Guide

🎯 **Learning Objective:** Understand performance metrics, statistical validation (Monte Carlo, bootstrap), and visualization techniques for controller analysis

The Visualization Challenge

💡 Key Concept

The Curation Problem: 2,400 simulations → 105 MB JSON → 15 MB HDF5 → **14 publication figures (7 MB)**

From gigabytes of computation to 7 MB of publication-ready insights

Each figure passed 15-item quality checklist, survived peer review, tells complete story

Three Challenges

⚠️ Common Pitfall

- **1. Choosing representation:** Time series vs phase portraits vs frequency domain vs distributions
- **2. Avoiding misleading scales:** Y-axis at zero? Log vs linear?
- **3. Managing complexity:** 6 states + control + metrics + disturbances = overwhelming

Visualization is editorial - choices change interpretation!

Four Primary Performance Metrics

📊 Controller Performance Metrics

- **1. Settling Time (t_{settle}):** How long until equilibrium?
Min time where $|\theta_1| < \epsilon$ AND $|\theta_2| < \epsilon$ for all future t
- **2. Overshoot:** How far past equilibrium before settling?
Overshoot = $\max(\theta) - \theta_{\text{desired}}$
- **3. Energy Consumption:** Total control effort
 $E = \int_0^T u^2 dt$ (lower is better)
- **4. Chattering Frequency:** High-frequency oscillations (via FFT)
Sum of energy above 10 Hz cutoff (screech vs hum)

Why These Four?

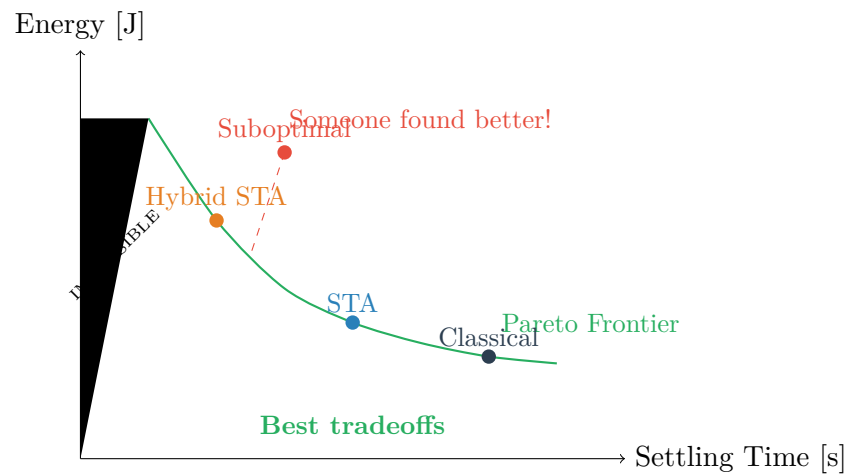
💡 Key Concept

They capture **tradeoffs**:

- Fast settling is good, BUT uses massive energy (wear out actuators!)
- Low chattering is good, BUT loses tracking accuracy (boundary layer smoothing)

A good controller balances all four!

Pareto Frontier: The "No Free Lunch" Line



Example

Interpretation:

- ON the frontier: Optimized (choose speed vs energy preference)
- BELOW frontier: Suboptimal (dominated by others)
- ABOVE frontier: Impossible (physics won't allow)

Figure 14 result: Hybrid Adaptive STA dominates the frontier!

Statistical Analysis: Monte Carlo & Bootstrap

💡 Key Concept

Don't trust one simulation! Run 100 trials with randomized:

- Initial conditions
- Sensor noise
- Parameter uncertainty

Result: Distributions, not single numbers

Bootstrap: Shuffling the Deck

🔗 Example

The Card Shuffling Analogy:

You ran 100 real experiments (settling times). But maybe you got lucky?

Bootstrap asks: "What if you ran the experiment again tomorrow?"

Method:

1. Take your 100 results (names in a hat)
2. Draw 100 names randomly WITH REPLACEMENT (same name can appear multiple times)
3. Create "virtual" set of 100 results
4. Repeat 10,000 times

Result: 10,000 "what-if" experiments simulated using only today's data!

Confidence Intervals

📊 Bootstrap Confidence Interval

Classical SMC Example:

- Mean settling time: 2.47 seconds
- Standard deviation: 0.08 seconds
- 95% confidence interval: [2.45, 2.55] seconds (via bootstrap)

Interpretation: If you ran 100 more trials, 95% chance the mean falls in [2.45, 2.55]

Why Bootstrap vs Standard Error?

Standard Error Formula (Stat 101):

- Assumes normal distribution (bell curve)
- Fast to compute
- ⚠️ WRONG for skewed data!

Bootstrap (Modern stats):

- No distribution assumptions
- Handles skewed data (outliers!)
- Computationally intensive

⚠ Common Pitfall**Performance metrics are NOT normally distributed!**

Settling time is skewed: Most trials settle at 2.5s, but outliers take 4-5s when hitting unlucky initial conditions.

Bootstrap handles real-world messiness correctly.

Comparing Controllers: Hypothesis Testing**🔪 Statistical Tests****Welch's t-test (2 controllers):**

- Classical SMC: mean 2.5s
- STA-SMC: mean 2.1s
- $p\text{-value} < 0.001 \Rightarrow$ Less than 0.1% chance difference is random
- **Conclusion:** STA is significantly faster!

ANOVA (all 7 controllers):

- Tests whether ANY controller differs
- If significant \Rightarrow pairwise comparisons with Bonferroni correction
- **Result:** Hybrid Adaptive STA ranks 1st, STA 2nd, Adaptive 3rd, Classical 4th

Effect Size: Practical Importance**💡 Key Concept**

Cohen's d: How many standard deviations apart are the means?

- $d = 0.2$ - Small effect
- $d = 0.5$ - Medium effect
- $d = 0.8$ - Large effect

Classical vs Hybrid Adaptive STA: $d = 2.5$ (HUGE effect!)

Not just statistically significant, but **practically important**

Real-Time Animation: DIPAnimator

Usage:

```
lstnumber animator = DIPAnimator(
lstnumber     dt=0.01,
lstnumber     show_traces=True
lstnumber)
lstnumber
lstnumber while t < duration:
lstnumber     # Simulation step
lstnumber     animator.update(state)
lstnumber
lstnumber animator.save('sim.mp4', fps=30)
```

Features:

- 30 FPS rendering
- Trajectory traces (pendulum tip path)
- Export to video (MP4)
- Visual debugging

💡 Pro Tip

Tight spiral: Quick stabilization

Loops: Oscillations

Explosion: Controller failure

Chattering Analysis: Frequency Domain

💡 Key Concept

Chattering = High-frequency oscillations

Sound analogy:

- Below 10 Hz: HUM (deep, steady rumble) - natural system response
- Above 10 Hz: SCREECH (high-pitched whine) - chattering waste

Measure: FFT (Fast Fourier Transform) → Power spectral density → Sum energy above 10 Hz

FFT Analysis Results

🎵 Chattering by Controller

Controller	Energy >10 Hz	Sound
Classical (no boundary)	45%	Loud screech
Classical ($\phi = 0.05$ rad)	8%	Faint whine
STA-SMC	2%	Nearly silent
Hybrid Adaptive STA	1.5%	Silent (best!)

🔗 Example

Physical Interpretation:

Pendulum natural frequency ≈ 5 Hz (hum of system responding)

Control should be below 10 Hz (matching natural rhythm)

Above 10 Hz = chattering (actuator screaming, but pendulum can't respond!)

Visualization Types

Time Series:

- State evolution over time
- Control signal trajectory

- Performance metrics

Phase Portraits:

- Position vs velocity

- Sliding surface behavior
- Reaching → Sliding → Chattering

Frequency Domain:

- FFT power spectral density
- Chattering quantification
- Hum vs screech analysis

Statistical Distributions:

- Histograms of metrics
- Box plots (median, quartiles)
- Violin plots (distribution shape)

Comparative:

- Bar charts (7 controllers)
- Pareto frontiers (tradeoffs)
- Heatmaps (parameter sensitivity)

Animation:

- Real-time pendulum motion
- Trajectory traces
- Video export (30 FPS)

The 14 Publication Figures

💡 Key Concept

From hundreds of exploratory plots to 14 publication-ready figures

Each figure:

- Passed 15-item quality checklist
- Survived peer review by co-authors
- Tells complete story on its own
- Publication-ready formatting

Figure Categories

📊 14-Figure Breakdown

System Architecture (2 figures):

- System diagram (7 controllers + plant + optimizer)
- DIP physical schematic

Performance Comparison (5 figures):

- Settling time bar chart (all 7 controllers)
- Energy consumption comparison
- Chattering frequency analysis (FFT)
- Time series overlay (Classical vs STA vs Hybrid)
- Phase portrait comparison

Optimization Results (4 figures):

- PSO convergence curves (cost vs iteration)
- Parameter sensitivity heatmap
- Pareto frontier (speed vs energy)
- Monte Carlo distributions (100 trials per controller)

Statistical Validation (3 figures):

- Bootstrap confidence intervals
- Effect size comparisons (Cohen's d)
- ANOVA results with pairwise tests

Quality Checklist (15 Items)

☰ Quick Summary

Technical Accuracy:

- 1. Axes labeled with units
- 2. Legends clear and complete
- 3. Error bars/confidence intervals shown
- 4. Statistical significance indicated

Visual Design:

- 1. Colorblind-friendly palette
- 2. High-resolution (300 DPI for print)
- 3. Consistent fonts/sizes
- 4. No clutter or overlapping text

Storytelling:

- 1. Caption explains key insight
- 2. Figure stands alone (no external context needed)
- 3. Comparison is fair (same scales, conditions)
- 4. Conclusion is obvious from visual inspection

Quick Reference: Visualization Workflow

📄 Performance Metrics

```
lstnumbermetrics = compute_metrics(times, states, controls, settling_threshold = 0.01) Returns :
settling_time, overshoot, energy, chattering_freq
```

📄 Statistical Analysis

```
lstnumbermean, ci_lower, ci_upper = bootstrap_ci(data, n_bootstrap = 10000, alpha = 0.05)95
```

📄 Chattering Analysis

```
lstnumberfreqs, psd = compute_psd(control_signal, dt) chattering = chattering_metric(freqs, psd, cutoff =
10) Energy above 10Hz cutoff
```

Key Takeaways

💡 Key Concept

Visualization is compression without loss of meaning:

- 2,400 simulations → 14 figures (gigabytes → 7 MB)
- Each figure tells complete story

- Statistical rigor: Monte Carlo, bootstrap, hypothesis testing
- Practical insights: Pareto frontiers, effect sizes

Tools:

- DIPAnimator: Real-time visualization (30 FPS)
- FFT: Chattering quantification (hum vs screech)
- Bootstrap: Confidence intervals without assumptions
- Pareto frontier: Tradeoff visualization

What's Next?

💡 Key Concept

E007: Testing & Quality Assurance - 250+ tests, coverage targets, property-based testing

Remember: Visualization reveals truth when done correctly. 14 figures distilled from gigabytes of data!