# 4.7 Dynamic Identification and Refinement

## Chapter Outline

## Dynamic System Identification, Validation, and Model Refinement

Chapter 4
Vehicle Motion Modeling and Simulation


4.1 Modeling Fundamentals and Kinematics
4.2 Rigid-Body Dynamics and Solvers
4.3 Propulsion, Forces, and Observations
4.4 Trim, Linearization, and Dynamics
4.5 Multicopter Modeling and Simulation
4.6 Measurement and Static Identification
**4.7 Dynamic Identification and Refinement**
4.8 Typical Vehicles and Extensions
4.9 Modeling Tools and Deployment
4.10 API and SDK Modeling Interfaces
4.11 Examples and Integrated Practice
4.12 Exercises and Course Projects


### 6 Units in This Section

**4.7.1 Identification Scope, Excitation, and Data**
Textbook: Sec. 4.2.2.4

**4.7.2 First-Order Dynamic Model Identification**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.3 Frequency Response and Staged Propulsion Models**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.4 Structure Selection and Parameter Updates**
Textbook: Sec. 4.2.2.4

**4.7.5 Independent Validation and Error Interpretation**
Textbook: Sec. 4.2.2.4

**4.7.6 Full-Vehicle Comparison and Flight-Based Refinement**
Textbook: Sec. 4.2.2.4

### 4.7.1 Identification Scope, Excitation, and Data: Open-Loop Plant and Closed-Loop Signal Boundaries

# Input-Output Boundaries in Closed-Loop Records

- **Dynamic identification reverses Section 4.4:** estimate a model and its parameters from recorded inputs and responses, rather than calculate responses from a model.
- First distinguish reference, controller output, actuator command, and torque. Reference → measurement includes feedback; torque → angular velocity differs.

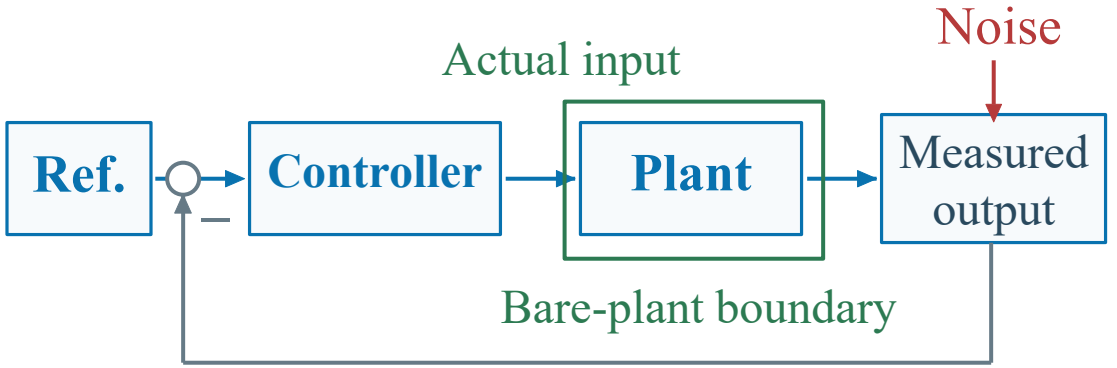



Figure: Input boundary defines the identified system

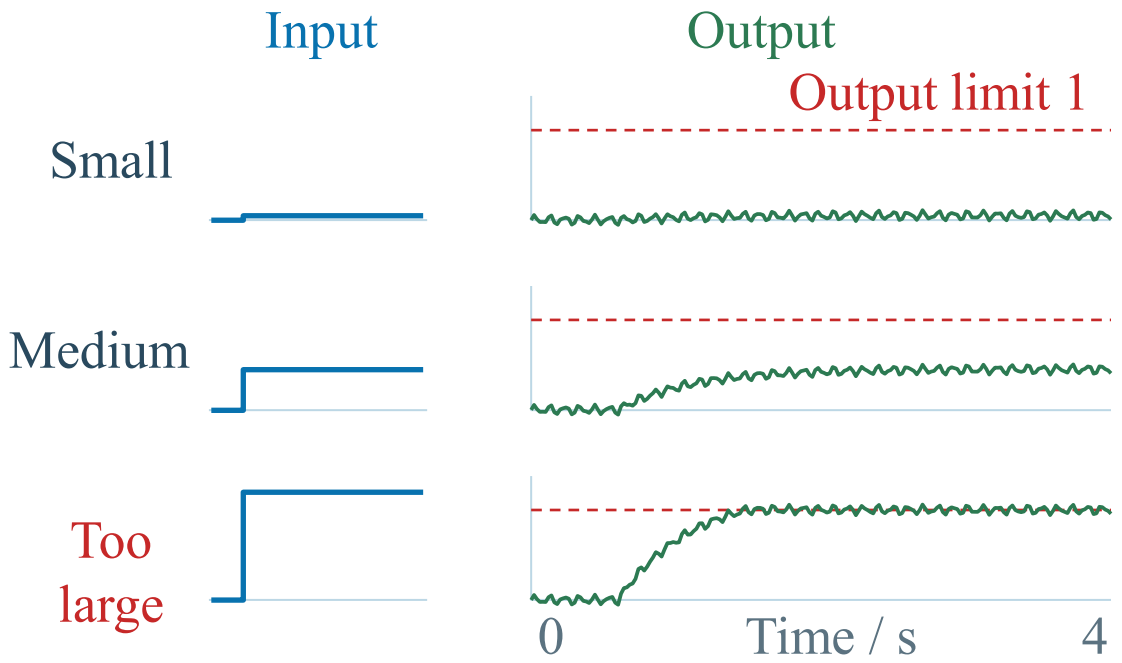
- Feedback can correlate input with noise. Recovering the bare plant requires reliable torque data or actuator calibration and declared filtering. This course does not require disabling the stabilizing loop.

**The identified model covers the entire signal chain between the selected input and output.**

### 4.7.1 Identification Scope, Excitation, and Data: Step Amplitude, Information, and Operating Range

# Step Responses Reveal Gain and Time Scale

- The steady-state change gives gain; the rise and settling reveal the time scale. A small step can disappear in noise. A large step can cross operating regions, saturate the output, or expose nonlinearity.
- Start with an interpretable small input, then repeat with other amplitudes and directions. Reserve complete new trials for validation. The response must be clear and remain in the same operating condition.




Normalized responses: same noise and output limit

**A step must reveal the dynamics while keeping the system in the same operating range.**

## 4.7.1 Identification Scope, Excitation, and Data: Sweep Bandwidth and Response Features

# Frequency Sweeps Reveal Fast and Slow Responses

- A sweep changes a sinusoid from low to high frequency, revealing low-frequency tracking, high-frequency attenuation, and phase lag. Cover the dynamics needed for the model's intended use.
- Each segment needs enough cycles and signal-to-noise ratio. Sweeping too fast adds transients. Dense, fast-sampled records do not guarantee identifiable high-frequency actuator dynamics.

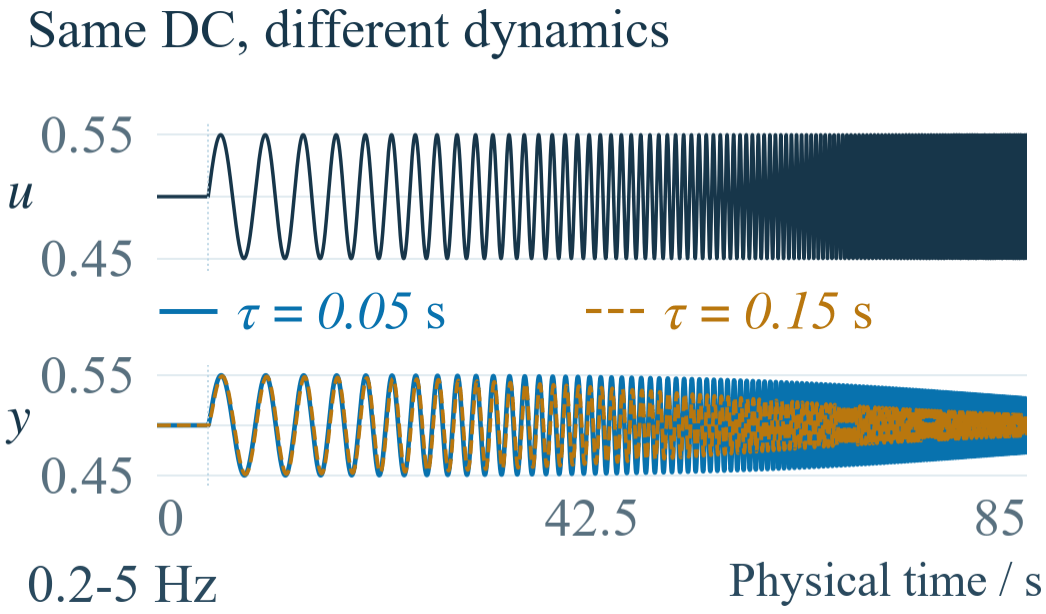



Figure: Same input, different response speeds

**The credible frequency band requires sufficient excitation and a measurable response.**

# 4.7 Dynamic Identification and Refinement

## 4.7.1 Identification Scope, Excitation, and Data: Sampling Rate, Excitation Band, and Evidence

### The Sampling Limit Prevents Aliasing but Does Not Ensure Accurate Identification

- Sampling at 100 Hz gives a 50 Hz Nyquist frequency. If the input contains energy only at 0.2–5 Hz, the model has not thereby been validated to 50 Hz.
- An ideal signal must remain below half the sampling rate to avoid aliasing. In practice, allow margins for noise, filter transition bands, and samples per cycle. Excitation and measurement evidence define the credible band.

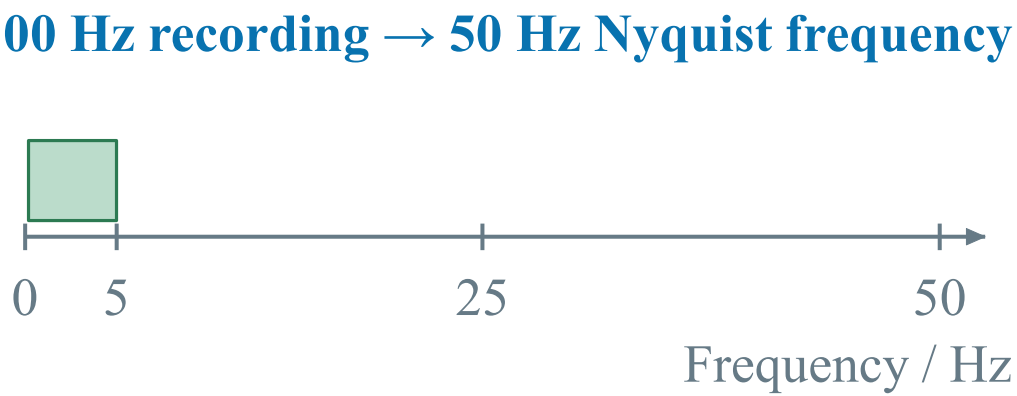



Figure: The sampling range exceeds the excitation range

**Excitation, noise, record length, and the measurement chain also limit the usable band.**

### 4.7.1 Identification Scope, Excitation, and Data: Apparent Delay from Time Misalignment

# Time Misalignment Can Mimic Actuator Delay

- Messages can have different transmission periods, start times, and timestamp units. Establish one time base, resample onto a shared grid, and mark missing intervals.
- Use a zero-order hold from each actual command update; interpolate continuous measurements appropriately. Long-gap interpolation can fabricate dynamics.
- Shifting the input by one 10 ms sample can introduce 36° of phase error at 10 Hz. This apparent delay contaminates estimates of time constant and delay.

$$360^\circ \times 10\,\text{Hz} \times 0.01\,\text{s} = 36^\circ$$

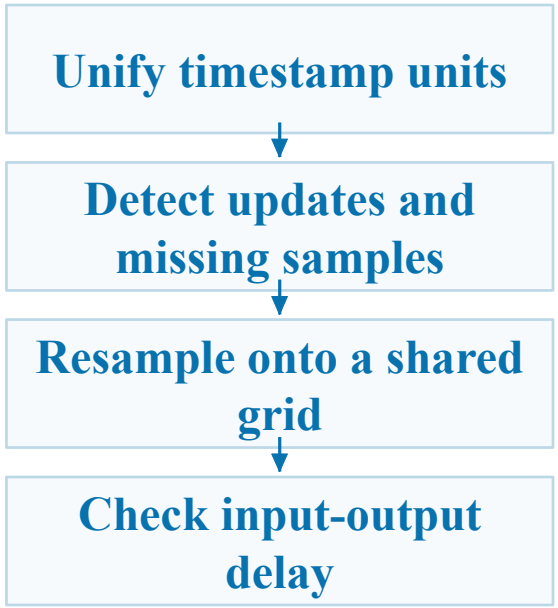


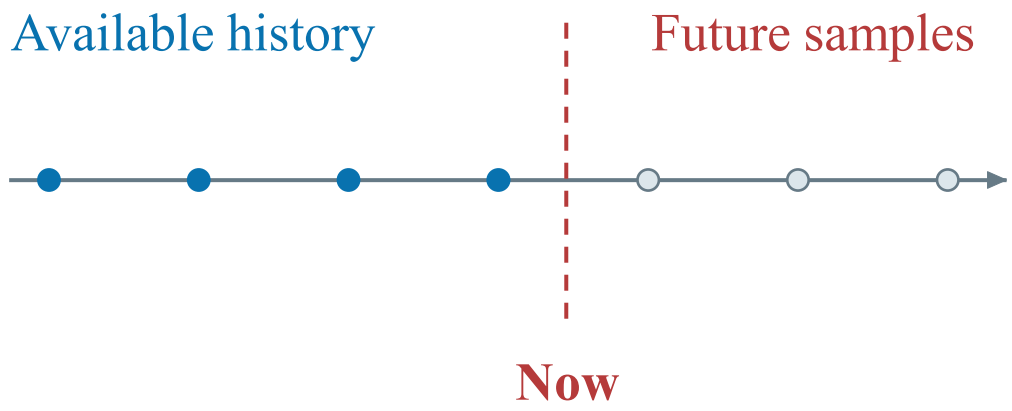

Figure: Timestamps establish sample correspondence

**Pair simultaneous inputs and outputs using timestamps.**

### 4.7.1 Identification Scope, Excitation, and Data: Physical Effects of Filtering and Resampling

## Filtering and Future Samples Change What the Model Knows

- Antialias-filter before downsampling; skipping samples is insufficient. Low-pass output filtering cuts high frequencies and may add lag. The fit may absorb an undocumented measurement chain.
- Offline zero-phase filters can use future samples; real-time models use only present and past values. Specify timestamps, input updates, state advancement, and output sampling. A close training fit does not prove real-time reproducibility.




Zero-phase filtering may use future samples

Figure: Real-time models cannot use future samples

**Preserve raw data and explain preprocessing effects on frequency and phase.**

### 4.7.1 Identification Scope, Excitation, and Data: Estimation, Selection, and Final Validation Data

## Split by Complete Trials, Then Freeze the Model for New Data

- Estimate on the estimation set; choose structure on the selection set. Freeze before using untouched validation data. Random splits of adjacent samples share nearly identical dynamics across sets.
- Record the system, amplitude, frequency band, load, and environment for new trials. Validate within the identified domain first; report extrapolation to larger amplitudes or new loads separately. Different files do not guarantee independent trials.

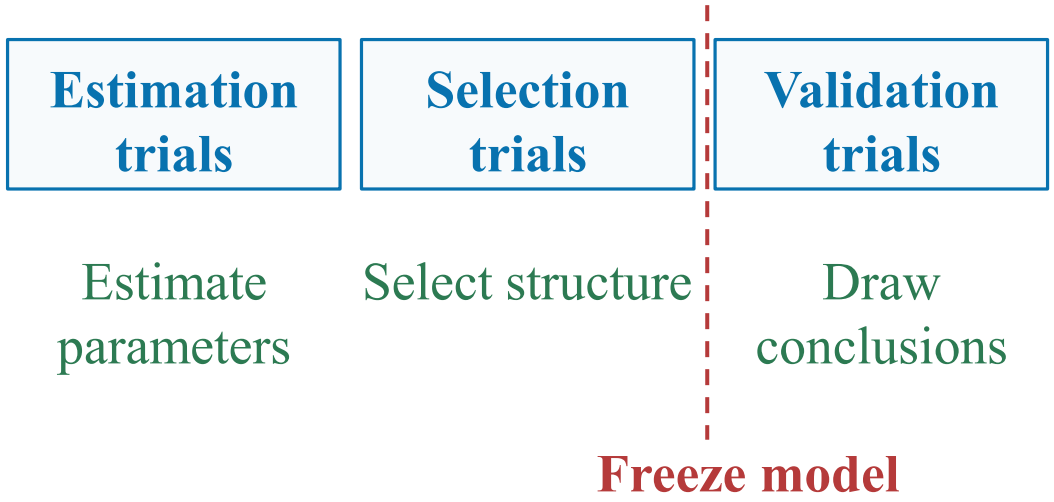



Figure: Freeze parameters before final validation

**Final validation data must remain unused in parameter estimation and structure selection.**

## Chapter Outline

# Dynamic System Identification, Validation, and Model Refinement

Chapter 4
Vehicle Motion Modeling and Simulation

4.1 Modeling Fundamentals and Kinematics
4.2 Rigid-Body Dynamics and Solvers
4.3 Propulsion, Forces, and Observations
4.4 Trim, Linearization, and Dynamics
4.5 Multicopter Modeling and Simulation
4.6 Measurement and Static Identification
**4.7 Dynamic Identification and Refinement**
4.8 Typical Vehicles and Extensions
4.9 Modeling Tools and Deployment
4.10 API and SDK Modeling Interfaces
4.11 Examples and Integrated Practice
4.12 Exercises and Course Projects

## 6 Units in This Section

**4.7.1 Identification Scope, Excitation, and Data**
Textbook: Sec. 4.2.2.4

**4.7.2 First-Order Dynamic Model Identification**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.3 Frequency Response and Staged Propulsion Models**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.4 Structure Selection and Parameter Updates**
Textbook: Sec. 4.2.2.4

**4.7.5 Independent Validation and Error Interpretation**
Textbook: Sec. 4.2.2.4

**4.7.6 Full-Vehicle Comparison and Flight-Based Refinement**
Textbook: Sec. 4.2.2.4

### 4.7.2 First-Order Dynamic Model Identification: Gain, Inertia, and Delay

# Response Features of Static, First-Order, and Delay Elements

- A static map gives the eventual output for an input but cannot explain a slow response. A first-order element adds one dominant memory scale to describe gradual approach to the target.
- A pure delay postpones the first effect of an input. It differs from an asymptotic first-order rise, and both effects may occur in the data.
- Compare responses and errors of these low-order candidates before adding complexity. Higher order cannot repair a wrong signal boundary or an uninformative input.

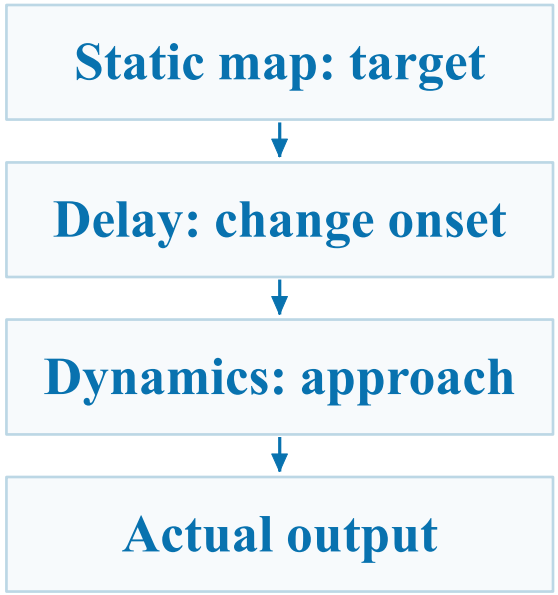



Figure: Different elements explain different response features

**Choose the simplest plausible structure from the observed behavior before estimating its parameters.**

### 4.7.2 First-Order Dynamic Model Identification: Final Value and Characteristic Time in a Step Test

# Gain and Time Constant from One Step

- The teaching actuator input changes from 0 to 0.2 and the output approaches 200 rad/s from 0. The gain is therefore 1000 rad/s per unit input. This example explicitly defines a linear model.
- Let the step occur at $t_0$. After 0.05 s, output reaches 126.424 rad/s, or 63.2% of the steady-state change. A first-order model without delay therefore gives $\tau$=0.05 s.
- For a nonzero initial output, subtract the initial steady value before calculating the fraction. With delay, measure the time constant from the response onset, rather than directly from the command change.

$$K = \frac{200 - 0}{0.2 - 0} = 1000, \qquad y(t_0 + \tau) = 126.424\,\text{rad/s}$$

**The steady-state change gives gain; the time to 63.2% of that change gives the first-order time constant.**

### 4.7.2 First-Order Dynamic Model Identification: Adjacent Samples as a Least Squares Problem

# First-Order Identification Without Numerical Differentiation

- With uniform sampling and a zero-order-held input, $y_{k+1} = ay_k + bu_k$. Each adjacent pair supplies a row $[y_k, u_k]$ and the next output as its target. Use several transitions to estimate $a, b$ together.
- A one-sample offset changes the identified system. Noisy previous outputs also make the regressors noisy, so ordinary least squares is not automatically unbiased. First check indexing with the noise-free teaching estimation set.

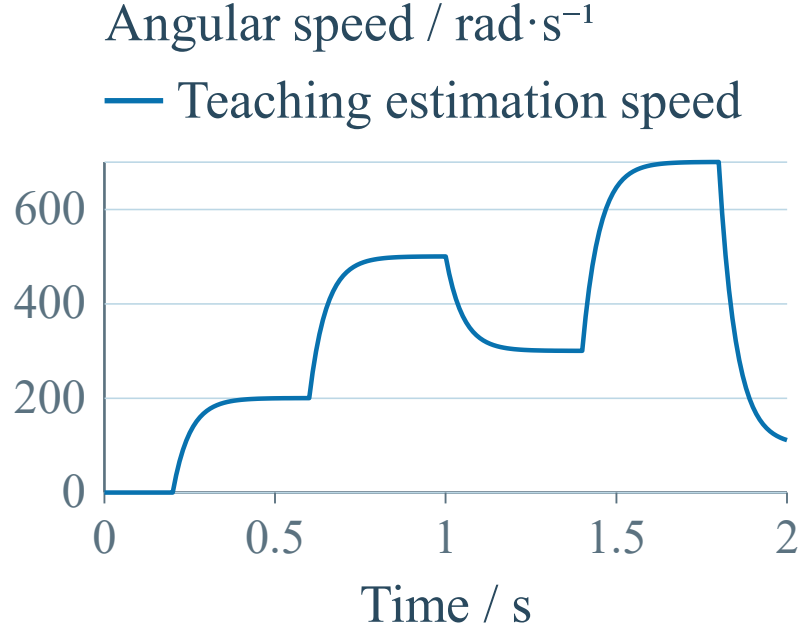
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} y_0 & u_0 \\ y_1 & u_1 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \mathbf{e}$$

**Fitting adjacent-state relationships avoids the noise amplification of explicit differentiation.**

### 4.7.2 First-Order Dynamic Model Identification: A Known First-Order System Checks the Calculation

## Recovering Known Parameters from Complete Teaching Data

- Estimation uses $T_s = 0.01\ s$ and successive holds 0, 0.2, 0.5, 0.3, 0.7, 0.1. An exact discrete model uses teaching gain $K = 1000\ \mathrm{rad/s}$ and $\tau = 0.05\ s$.
- Adjacent-sample least squares gives $a \approx 0.81873075$ and $b \approx 181.26925$. Matching the generation equation checks indexing and solution.
- Two comma-separated-value (CSV) files hold estimation and different-input validation sequences, each with time, normalized input, and angular speed. Both are synthetic.




First-order teaching output for estimation

**A controlled teaching system checks the calculation chain before we interpret real data.**

### 4.7.2 First-Order Dynamic Model Identification: Worked Example: Data to Parameters

# Two Discrete Coefficients Directly from the Teaching CSV

- **Sample indexing:** In 03_Identification, read teaching_estimation.csv. Use simultaneous $y_k, u_k$ to explain $y_{k+1}$. Here $u$ is the normalized command and $y$ is angular speed in rad/s. Expect $a \approx 0.81873075$ and $b \approx 181.26925$.

```
E = readtable('teaching_estimation.csv');
Ts = median(diff(E.time_s));
u = E.normalized_input; y = E.omega_rad_s;
Phi = [y(1:end-1), u(1:end-1)];
ab = Phi \ y(2:end);
a = ab(1); b = ab(2);
```

**Dynamic regression uses the current state and input to explain the next state.**

4.7.2 First-Order Dynamic Model Identification: Recovering Physical Parameters from Discrete Coefficients

# Physical Gain and Time Constant from Discrete Coefficients

- For a stable first-order zero-order-hold (ZOH) model, $a = e^{-T_s/\tau}$, so $\tau = -T_s/\ln a$. Since $b = K(1-a)$, we obtain $K = b/(1-a)$.
- Substituting estimated $a$ and $b$ with $T_s$=0.01 s recovers $\tau$=0.05 s and $K$=1000. Omitting the sampling period gives the wrong time scale.
- For a stable first-order system without delay, $T_s > 0, \tau > 0$ implies $0 < a < 1$. Values outside that range call for checks of delay, alignment, noise, and model structure.

$$\tau = -\frac{T_s}{\ln a} = 0.05\,\text{s}, \qquad K = \frac{b}{1-a} = 1000$$

**Coefficients have a physical interpretation only when the discrete structure and hold assumptions match.**

### 4.7.2 First-Order Dynamic Model Identification: Worked Example: Model to Independent-Input Validation

# Free Simulation with a New Input and Frozen Parameters

- **Independent free simulation:** Keep a and b fixed and recurse with a different staircase input. Start the teaching angular speed at 0. Then use no measured outputs to reset states and do not retune. Compare against teaching measurements with 2 rad/s noise standard deviation on the validation-curve page.

```
V = readtable('teaching_validation.csv');
uv = V.normalized_input;
ys = zeros(height(V),1);
for k = 1:height(V)-1
    ys(k+1) = a*ys(k) + b*uv(k);
end
```

**Independent validation holds the model fixed and tests its predictions with a new input.**

### 4.7.2 First-Order Dynamic Model Identification: Validation Curves Under an Independent Input

# A New Input Sequence Tests the Estimated Model

- Freeze parameters and change the input to 0, 0.6, 0.1, 0.4, 0.2, 0.55. The model uses only the new input and initial state, without measurement-based state correction at each point.
- Validation measurements add independent Gaussian noise with 2 rad/s standard deviation and a fixed seed for reproducibility. Residuals below show no repeated structure.
- **Conclusion:** Root mean square error (RMSE) is 2.148 rad/s, comparable to the added noise, confirming the teaching calculation chain.

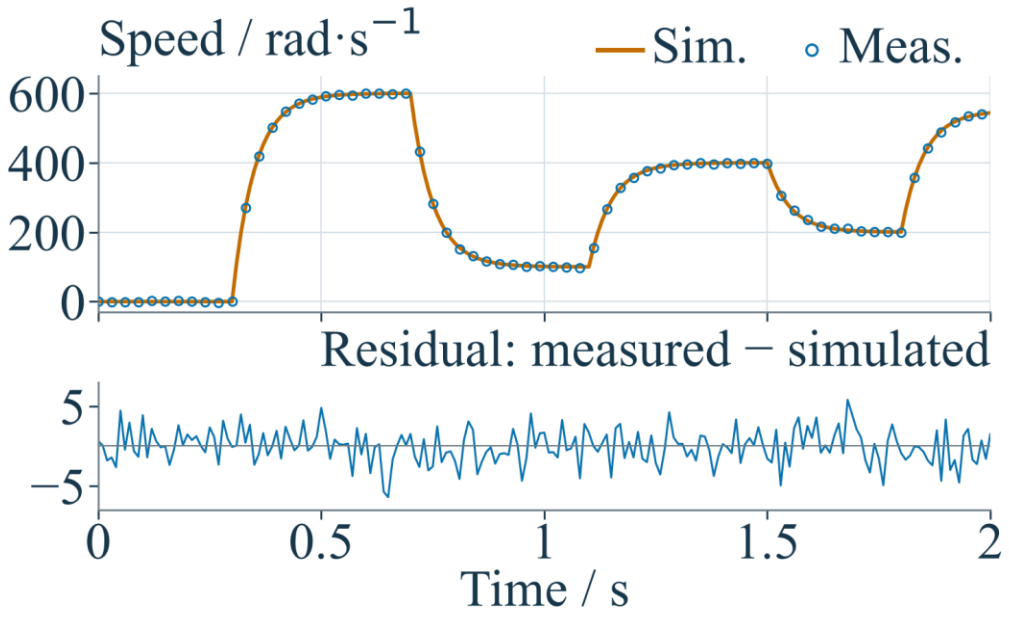



Figure: Teaching measurements, free simulation, and residuals under a new input

**Prediction with a new input can expose parameters and structures that only fit the old sequence.**

### 4.7.2 First-Order Dynamic Model Identification: Two Calculation Routes in the Installed Package

# The Introductory Script: Real and Teaching Results

- **Directory:** [installation directory]/RflySimAPIs/4.RflySimModel/1.BasicExps/e6_ModelingLearningLab/03_Identification/. Read Readme.pdf first. Set this as the MATLAB current folder and run the two lines below.
- **Data and checks:** Dynamics uses the two teaching CSV files here. Statics reads e11_PowerSysIdModel/2.StaticThrIntID/Static_data.mat. If absent, static_available is false and only statics is skipped. Set RFLYSIM_APIS if the root is not found.

```
metrics = identification_beginner;
metrics.static_available
```

- **Expected:** a≈0.81873, b≈181.27 (τ=0.05 s, K=1000); validation RMSE≈2.148 rad/s; FIT≈98.917% (100% means exact agreement). Figures, CSV, and JSON go to outputs.

**One script can reproduce the algorithm while preserving the evidence boundary between data sources.**

### 4.7.2 First-Order Dynamic Model Identification: Textbook Tools and Data Boundaries

# Identification Tool Routes for the Same Input-Output Data

- Common inputs: Save timestamps, sample period, commands, and measurements. Separate estimation from independent validation; check units, synchronization, and excitation band.
- MATLAB System Identification Toolbox: Estimate transfer functions, state-space models, and other models from time- or frequency-domain input-output data. Compare outputs and residuals on independent data. Installation and a license are required.
- CIFER (Comprehensive Identification from Frequency Responses): The textbook's external tool supports sweeps and coupled-channel analysis. Check version, license, and data format.
- Platform: [installation directory]/RflySimAPIs/4.RflySimModel/2.AdvExps/e13_CopIdModel. Read Readme.pdf and 1.IdAppUsage first. 2.HILFreqId needs an autopilot, logs, and hardware-in-the-loop (HITL; HIL in the textbook). Use that version's actual output records.

**Tools produce candidate models; independent validation and licensing need separate checks.**

## Chapter Outline

# Dynamic System Identification, Validation, and Model Refinement

Chapter 4
Vehicle Motion Modeling and Simulation

4.1 Modeling Fundamentals and Kinematics
4.2 Rigid-Body Dynamics and Solvers
4.3 Propulsion, Forces, and Observations
4.4 Trim, Linearization, and Dynamics
4.5 Multicopter Modeling and Simulation
4.6 Measurement and Static Identification
**4.7 Dynamic Identification and Refinement**
4.8 Typical Vehicles and Extensions
4.9 Modeling Tools and Deployment
4.10 API and SDK Modeling Interfaces
4.11 Examples and Integrated Practice
4.12 Exercises and Course Projects

## 6 Units in This Section

**4.7.1 Identification Scope, Excitation, and Data**
Textbook: Sec. 4.2.2.4

**4.7.2 First-Order Dynamic Model Identification**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.3 Frequency Response and Staged Propulsion Models**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.4 Structure Selection and Parameter Updates**
Textbook: Sec. 4.2.2.4

**4.7.5 Independent Validation and Error Interpretation**
Textbook: Sec. 4.2.2.4

**4.7.6 Full-Vehicle Comparison and Flight-Based Refinement**
Textbook: Sec. 4.2.2.4

### 4.7.3 Frequency Response and Staged Propulsion Models: Static Targets and Dynamic Tracking

## The Static Map Sets the Target; Dynamics Sets the Tracking Speed

- First use steady-state data to map commands to target speed. Then fit the lag from target to actual speed using dynamic records. This prevents the time constant from hiding a steady-state gain error.
- Static and dynamic records must match the motor, propeller, electronic speed controller, and supply conditions. Use FX450 static test-stand records separately from FX310 inflow thrust data in e11/5.DynThrID: the systems differ.
- Deliver a low-order model, parameter table, input-output boundary, and new-data test. The static map sets the target; the dynamic element explains how the output follows it.

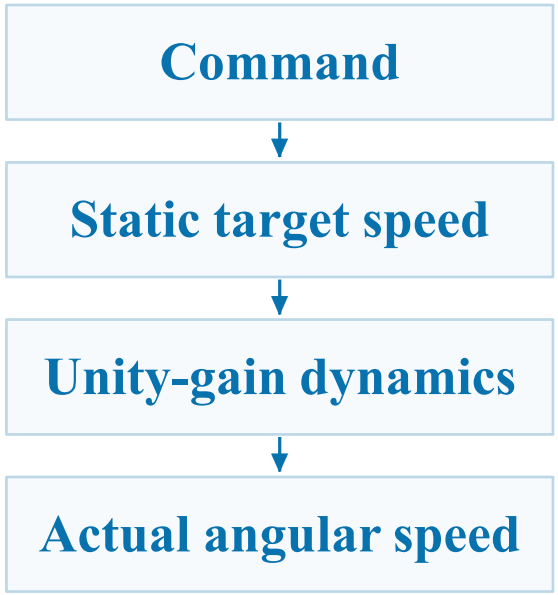



Figure: Separate static mapping from dynamic lag

# Historical Test-Stand Logs to Dynamic Model Candidates

- **Same configuration:** statically map pulse-width modulation (PWM) to target speed or thrust. Fit dynamics from target to measurement, without re-estimating steady gain.
- Preprocessing: Low-pass filter the manual's 10 kHz sweep records, then downsample. Put input and output on the same grid, remove start/end transients, and save the processing rules.
- Estimate frequency response first (see input-output spectra), then try low-order transfer functions. Compare magnitude, phase, and time response over one band; freeze before independent-record validation.
- Case source: Curves and percentages are test-stand results from the Multicopter UAV Model Development Manual. The data/workflow also appear in e11_PowerSysIdModel/3.StaticThrPartID/Readme.pdf.

**Real dynamic identification must preserve system boundaries, processing, and model-selection evidence.**

### 4.7.3 Frequency Response and Staged Propulsion Models: Two Time Grids in the Real MAT File

# A Sweep MAT File Contains Raw and Processed Variables

- **Data location:** MAT is the MATLAB data-file format. This example uses [installation directory]/RflySimAPIs/4.RflySimModel/2.AdvExps/e11_PowerSysIdModel/3.StaticThrPartID/No2_0.2to1.mat
- **Two grids:** 364800 raw records, with TIMEms increments of about 0.1 ms (10 kHz), span about 36.48 s. The 7097 processed records use t1 increments of about 5 ms (200 Hz).
- **Pairing:** The same row number in raw PWM and processed F1 does not mean the same time. Align by time before pairing. See the example's Readme.pdf for configuration, filtering, and segment-selection rules.

**Identify each variable's time grid before pairing input and output.**

### 4.7.3 Frequency Response and Staged Propulsion Models: Reading the Historical Processing Code

## Filtering and Resampling Change the Effective Frequency Band

- **Conditions:** This fragment assumes aligned raw input and output. fs0, fc, and fs1 are the original sample rate, cutoff frequency, and target sample rate, all in Hz. Process both channels alike, then remove end transients. These settings apply only to this stage.

```
fs0 = 10000; fc = 5; fs1 = 100;
uf = lowpass(u_raw,fc,fs0);
yf = lowpass(y_raw,fc,fs0);
u = resample(uf,fs1,fs0);
y = resample(yf,fs1,fs0);
Ts = 1/fs1;
```

**Filtering, sampling, and segment selection jointly define the dynamics a model can explain.**

### 4.7.3 Frequency Response and Staged Propulsion Models: Frequency Response from Input-Output Spectra

# Frequency-Domain Methods Compare Magnitude and Phase

- At one frequency: $U, Y$ are input/output Fourier quantities; $*$ is complex conjugation; $E$ is statistical averaging. Cross-spectrum: $S_{yu} = E[YU^*]$; input autospectrum: $S_{uu} = E[|U|^2]$.

- Average segments; H1 is cross-spectrum divided by input autospectrum. Each frequency needs excitation, reliable input, and output noise uncorrelated with input. Input noise or strong closed-loop correlation blocks bare-plant interpretation.

**Input spectrum $S_{uu}$**

↓

**Shared variation $S_{yu}$**

**Shared variation ÷ input energy**

Figure: Input energy and shared variation by frequency

$$\hat{G}_{H1}(j\Omega) = \frac{S_{yu}(\Omega)}{S_{uu}(\Omega)}$$

**Frequency-domain estimation establishes response evidence before choosing a parametric model.**

### 4.7.3 Frequency Response and Staged Propulsion Models: What Coherence Can Establish

# Coherence Helps Assess Frequency-Response Evidence

- $S_{yy}$ is output autospectrum; coherence $\gamma^2$ ranges from 0 to 1. Low values may reflect noise, nonlinearity, transients, or weak excitation. High values do not prove physical parameters or model boundaries correct.
- Compare frequency response, coherence, input spectrum, and repeat trials. Strong excitation/shared variation occur only in the first region; higher order cannot justify fitting the weakly excited region.

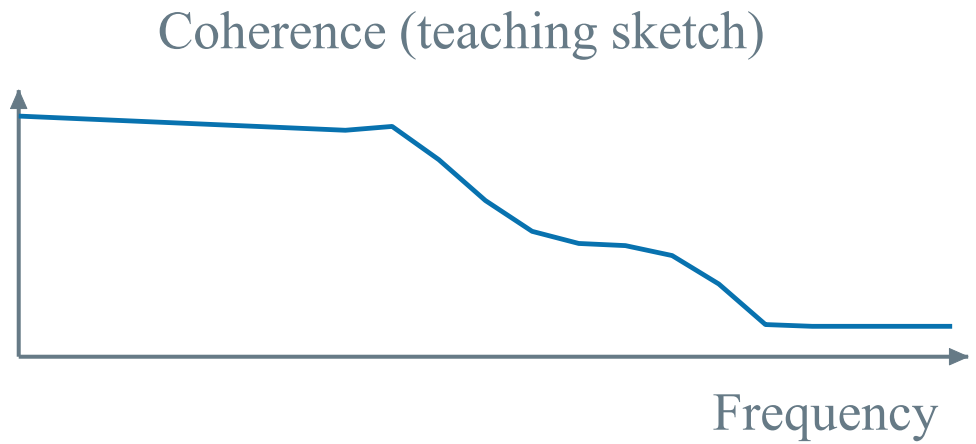



Figure: Teaching illustration of coherence

$$\gamma^2(\Omega) = \frac{|S_{yu}(\Omega)|^2}{S_{uu}(\Omega)S_{yy}(\Omega)}$$

**High coherence indicates strong shared variation; low coherence calls for explanation in that band.**

### 4.7.3 Frequency Response and Staged Propulsion Models: Phase Signatures of Pure Delay and First-Order Lag

# Delay and Time Constant Leave Different Frequency Signatures

- Pure delay $L$ preserves magnitude with phase $-\Omega L$. First-order dynamics changes both. For a cascade, estimate time scale from magnitude, then check extra phase.
- With τ=L=0.05 s, phase at 60 rad/s is about −172° for the delay and −72° for the first-order element. Misalignment can create a similar phase slope. Check timing before attributing it to delay.

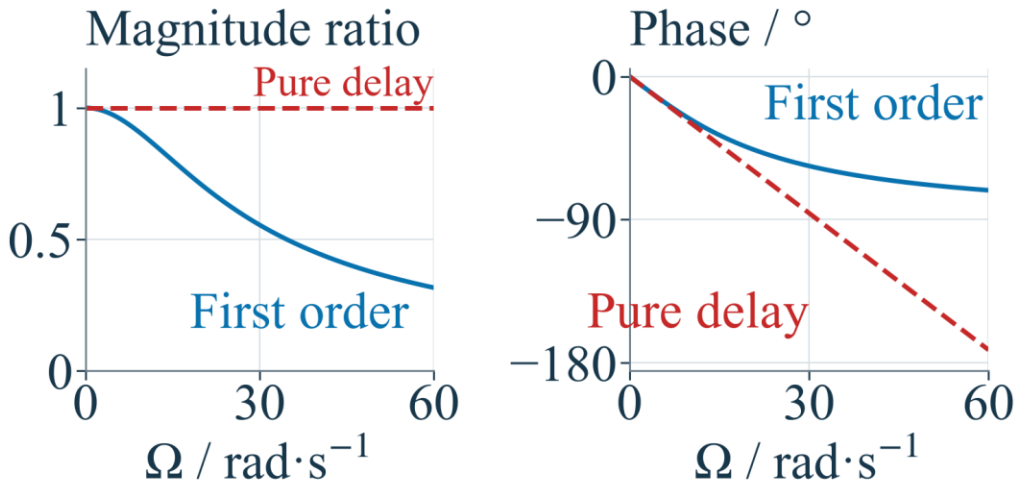



Figure: First-order lag (blue) and pure delay (red)

$$G(s) = Ke^{-Ls}/(\tau s + 1), \qquad \varphi(\Omega) = -\arctan(\Omega\tau) - \Omega L$$

**Delay mainly adds phase lag; first-order dynamics also changes magnitude.**

### 4.7.3 Frequency Response and Staged Propulsion Models: Real Case: Reading Motor Dynamics

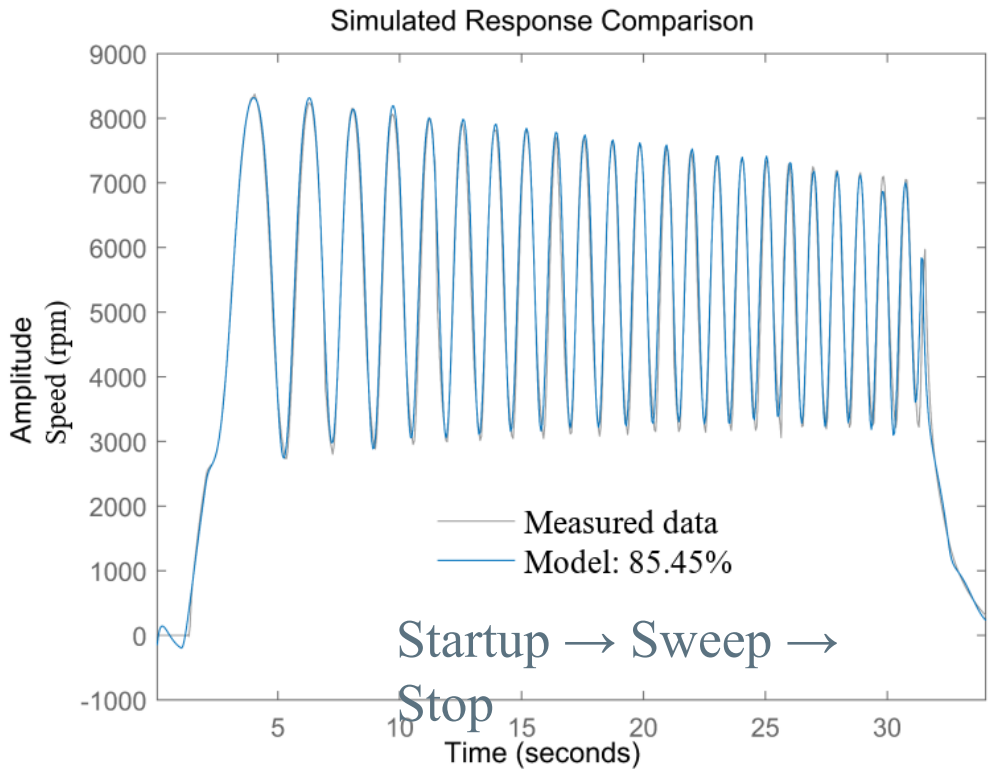
# A Speed Model Must Explain Time Tracking and Frequency Lag

- **Left side of the motor comparison:** Gray is measured angular speed and blue is candidate output, with 85.45% fit. Check startup, continuous sweep, and stopping.
- **Right side:** The horizontal axis is rad/s. Blue is the experimental frequency response; orange is the candidate. Read magnitude attenuation above and phase lag below (prerequisite: Section 4.4).
- **Source boundary:** A second-order formula in another historical source accompanies 82.24%, and cannot be combined with this figure's 85.45% as one model. Save candidate parameters, data interval, and figure together.
- **Fit:** $\mathrm{FIT} = 100\left(1 - \frac{\|y-\hat{y}\|}{\|y-\bar{y}\|}\right)\%$. Exact agreement gives 100%; values may be negative. Denominator: Section 4.7.5.

**Time- and frequency-domain agreement must refer to the same candidate and processing version.**

### 4.7.3 Frequency Response and Staged Propulsion Models: Real Case: Reading Motor Dynamics

# A Motor Candidate in Time, Magnitude, and Phase



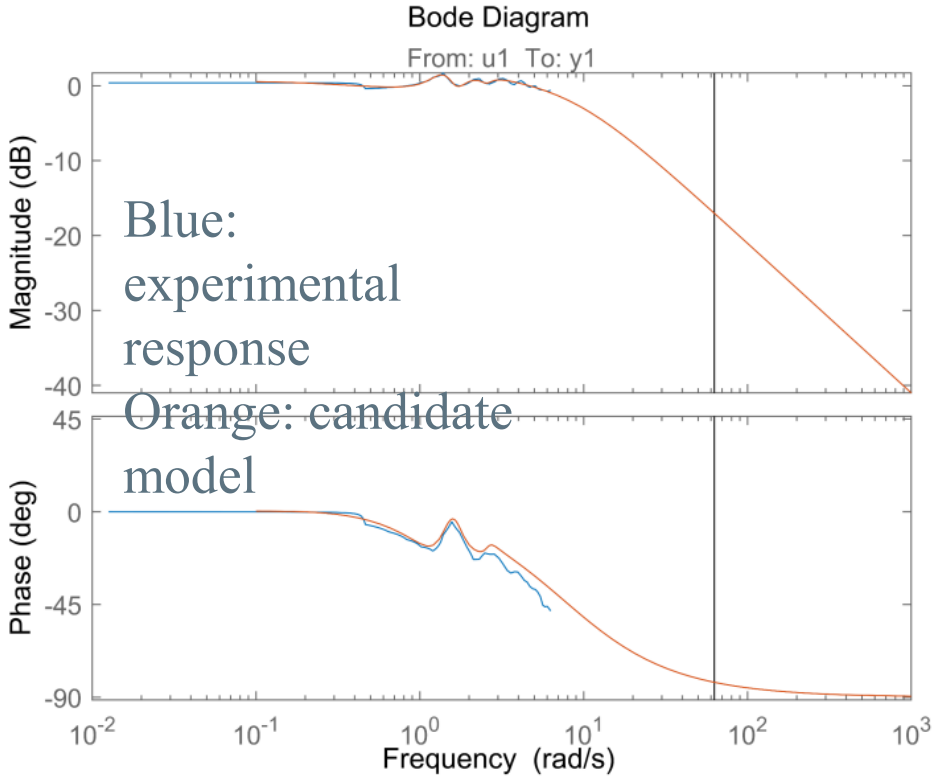




Figure: Time response, magnitude, and phase for the same motor candidate

### 4.7.3 Frequency Response and Staged Propulsion Models: Real Case: Reading Propeller Dynamics

## A Separate Thrust-Dynamics Layer Must Avoid Double Counting

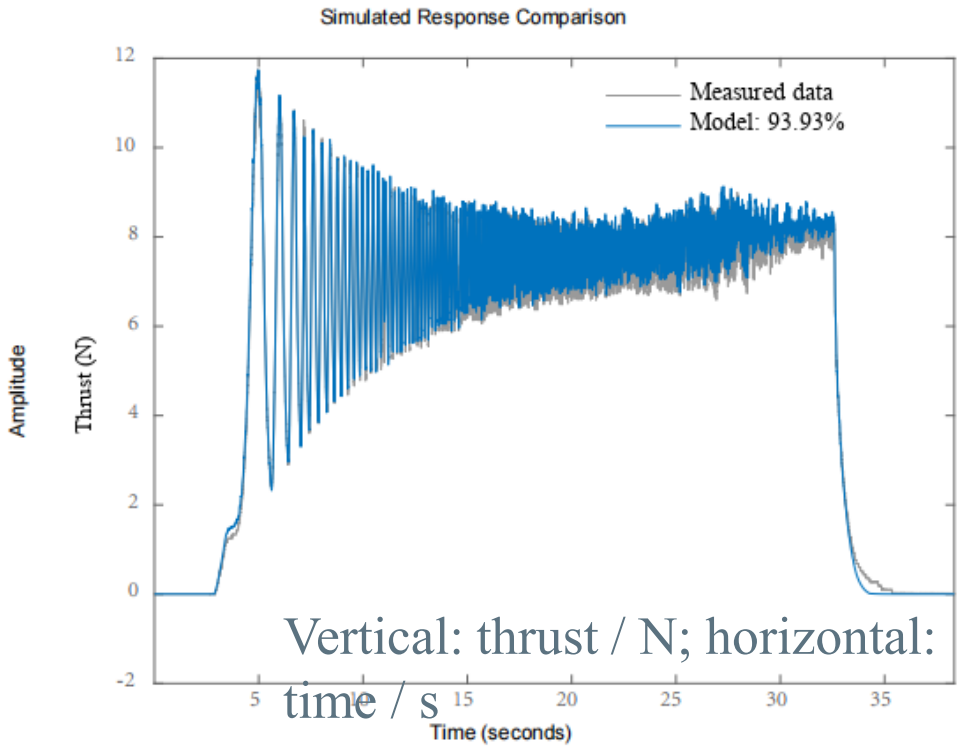
- **Input boundary:** Apply the static curve to actual angular speed to obtain target thrust as this layer's input. Measured thrust is its output. Motor speed lag is already in the preceding layer and must not be added again.
- **Thrust comparison:** The left-hand candidate fit is 93.93%. The right-hand plots show distinct magnitude and phase features near $10^2$ rad/s. Selecting only by low-frequency steady gain would miss these dynamics.
- **Different candidate:** The e11 Readme separately reports 94.59% for a sixth-order model reduced to fourth order. That is another processing version. This figure identifies phenomena the model must explain.

**Clear input-output boundaries prevent staged identification from adding the same dynamics twice.**

# 4.7 Dynamic Identification and Refinement

## 4.7.3 Frequency Response and Staged Propulsion Models: Real Case: Reading Propeller Dynamics

## A Thrust Candidate Must Explain Tracking and Frequency Peaks



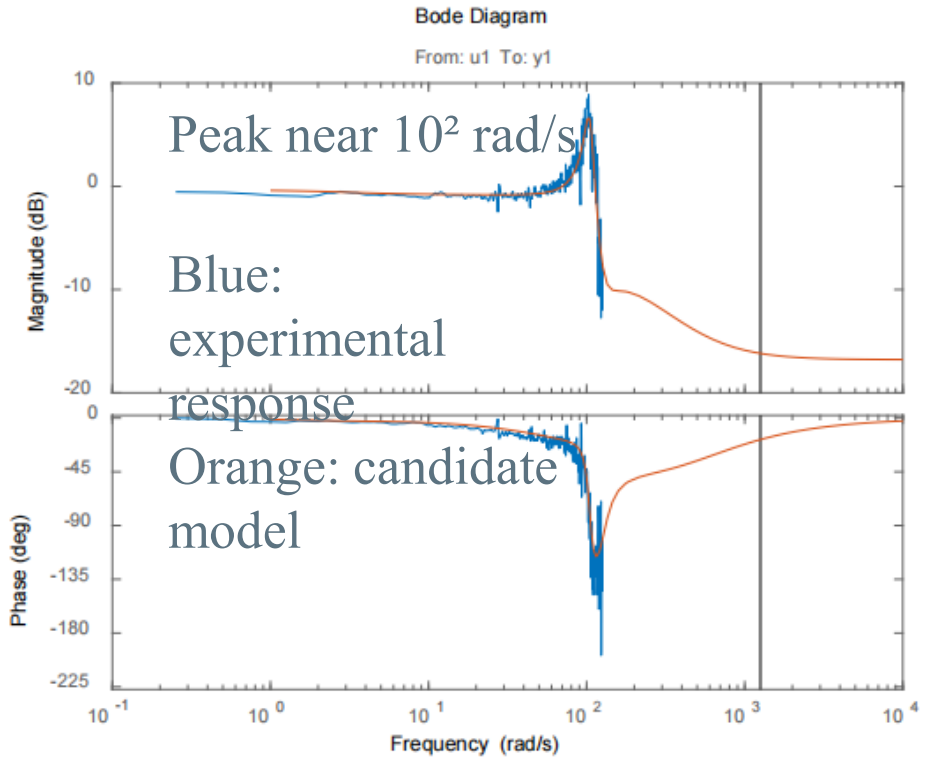




Figure: A close time-domain thrust fit still needs an explanation for the frequency peak

### 4.7.3 Frequency Response and Staged Propulsion Models: Real Case: Correlation and Forward Dynamics

# Poor Reaction-Torque Fits Call for Measurement Checks

- **Original workflow:** Predicting measured reaction torque from its static target fitted poorly. Reversing the fit—from reaction torque to angular speed—gave a historical 68.13% fit.
- **Changed system:** Reversing the fit changes the input-output boundary. Strong correlation does not justify inverting the torque-to-speed model and deploying it as forward speed-to-torque dynamics.
- **Diagnosis:** Check torque-sensor zero, range, noise, and synchronization first, then the static map and dynamic excitation range. Extra poles need evidence of a response mechanism.
- **Refinement:** If the data cannot identify this layer reliably, retain an evidence-based static relationship and state the missing dynamics. This supports later validation better than forcing a high-order black box into the vehicle model.

**Reverse fitting can reveal data relationships, but does not automatically yield causal forward dynamics.**

### 4.7.3 Staged Propulsion: Multiple-Input Multiple-Output Identification

# Two Inputs Require Four Response Channels

- Small hover perturbations: input $u = [\delta\tau_x, \delta\tau_y]^T$; measured body angular rates $y = [p, q]^T$. Neglect other channels only locally, in a linear, unsaturated range.
- Trials: Excite one axis at a time, record p and q synchronously, and repeat. A repeatable response on the other axis above the noise floor suggests coupling, after ruling out common disturbances.
- Model: $Y(j\Omega) = G(j\Omega)U(j\Omega)$, $G \in \mathbb{C}^{2\times 2}$. Four elements map two inputs to two outputs. For a state-space form, $B \in \mathbb{R}^{n\times 2}$, $C \in \mathbb{R}^{2\times n}$.
- Choice: A response matrix shows bands and coupling; transfer functions explain individual channels; shared-state state space supports closed-loop analysis. Single-channel H1 is a special case. Multiple-input multiple-output (MIMO) identification also requires excitation-matrix rank and correlated-noise checks.

**Without cross-response measurements, independence of the two axes remains untested.**

## Chapter Outline

# Dynamic System Identification, Validation, and Model Refinement

Chapter 4
Vehicle Motion Modeling and Simulation
4.1 Modeling Fundamentals and Kinematics
4.2 Rigid-Body Dynamics and Solvers
4.3 Propulsion, Forces, and Observations
4.4 Trim, Linearization, and Dynamics
4.5 Multicopter Modeling and Simulation
4.6 Measurement and Static Identification
**4.7 Dynamic Identification and Refinement**
4.8 Typical Vehicles and Extensions
4.9 Modeling Tools and Deployment
4.10 API and SDK Modeling Interfaces
4.11 Examples and Integrated Practice
4.12 Exercises and Course Projects

## 6 Units in This Section

**4.7.1 Identification Scope, Excitation, and Data**
Textbook: Sec. 4.2.2.4

**4.7.2 First-Order Dynamic Model Identification**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.3 Frequency Response and Staged Propulsion Models**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.4 Structure Selection and Parameter Updates**
Textbook: Sec. 4.2.2.4

**4.7.5 Independent Validation and Error Interpretation**
Textbook: Sec. 4.2.2.4

**4.7.6 Full-Vehicle Comparison and Flight-Based Refinement**
Textbook: Sec. 4.2.2.4

### 4.7.4 Structure Selection and Parameter Updates: Why Physical Parameters May Be Inseparable

# One Roll Curve May Not Separate Three Parameters

- In the roll model $J\dot{p} + dp = K_M u$, $u$ is dimensionless; $J$ has units kg·m², $d$ has units N·m·s/rad, and $K_M$ has units N·m.

- Two $(J, d, K_M)$ sets have identical ratios $K_M/J = 2$ and $d/J = 0.2$, so their input-output responses match. Measure $J$ independently to recover the other two parameters.

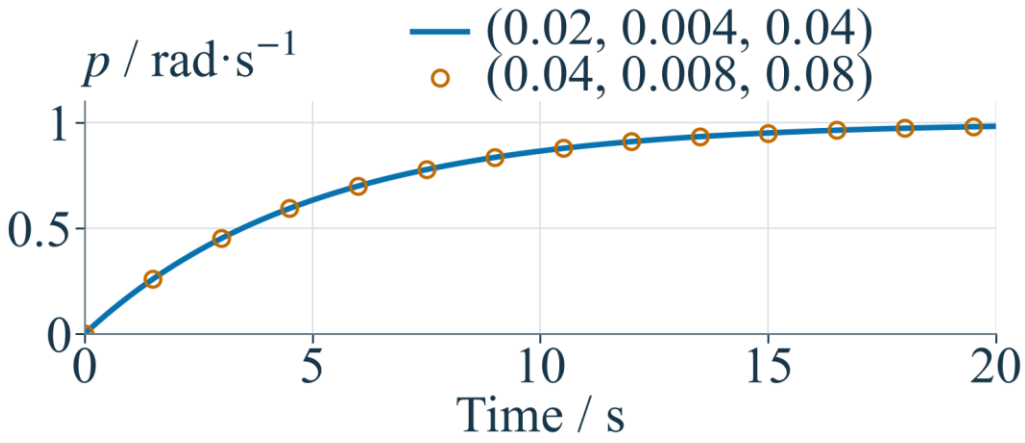



Figure: Two $(J, d, K_M)$ sets give the same $p(t)$ under one step

$$\dot{p} = (K_M/J)u - (d/J)p = 2u - 0.2p$$

**Input-output data may determine only parameter ratios; extra measurements resolve the ambiguity.**

### 4.7.4 Structure Selection and Parameter Updates: Proportional Parameter Scaling

# Proportional Inertia and Torques Can Produce Identical Rotation

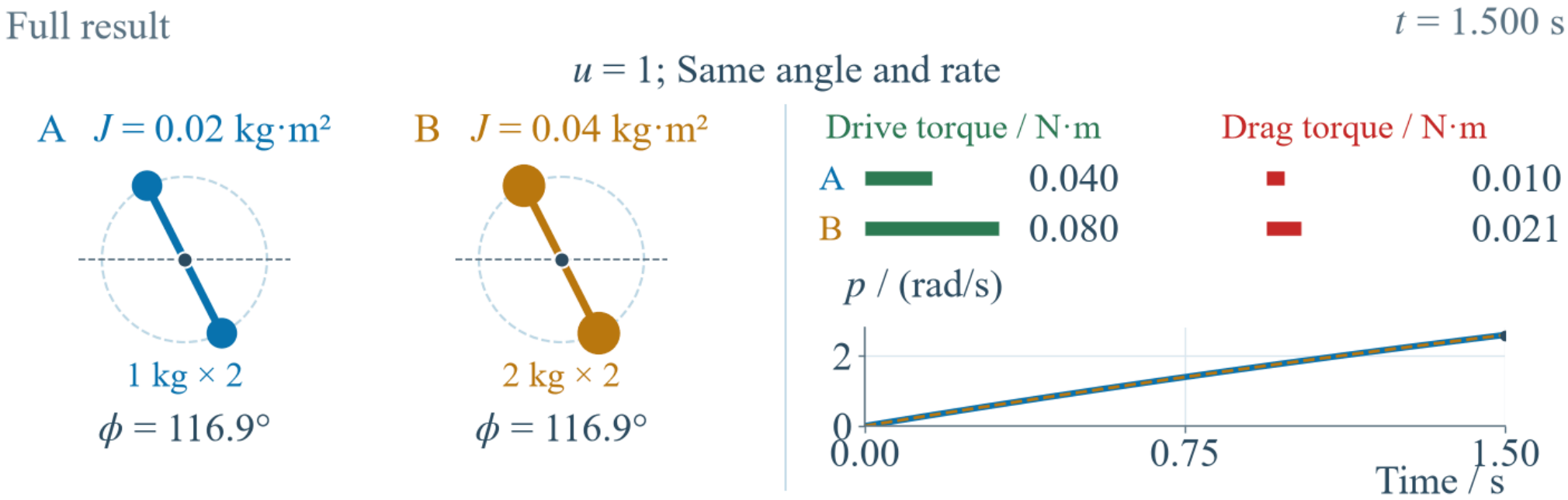



Figure: Fixed-radius two-mass rotors with doubled inertia, driving torque, and resisting torque

**Input-output data may determine only parameter ratios; extra measurements resolve the ambiguity.**

### 4.7.4 Structure Selection and Parameter Updates: Reasonable Structure Before Higher Order

# Independent Data Tests the Value of Higher Order

- Compare errors under the same data conditions, then check physical units, signs, causality, and range. Higher order adds flexibility and extrapolation risk. Keep a simple model when it is adequate.
- **Order choice:** Static validation error is about 108 rad/s; first order reduces it to 2.15. Extra second- and third-order coefficients are near 0 with no further reduction, so choose first order. See the component-to-vehicle acceptance page for staged checks.

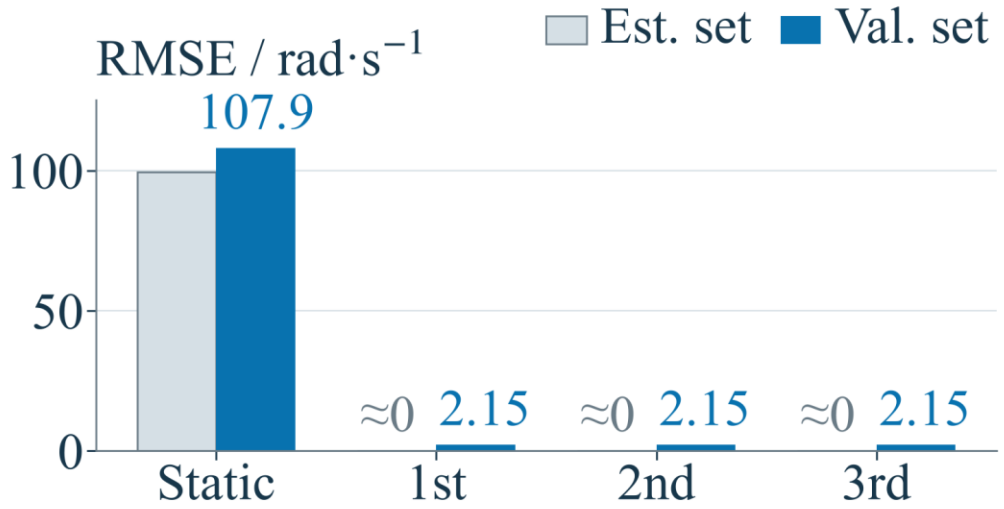



Figure: Free-simulation errors of different orders on the same teaching CSV data

**A smaller error is part of the evidence; physical structure determines whether the model is usable.**

### 4.7.4 Structure Selection and Parameter Updates: A High Fit Can Still Produce an Unstable Motor

# An 82.24% Historical Fit Still Needs Pole Checks

- The manual's second-order motor candidate has 82.24% fit and denominator $s^2 + 8.014s - 9.268$. Its positive real root 1.0253 produces a growing mode.
- A stable motor should reach finite steady speed at fixed input; a positive real pole contradicts this. Preserve original coefficients. Recheck data bounds and identification settings; check poles before updates and log sign changes.

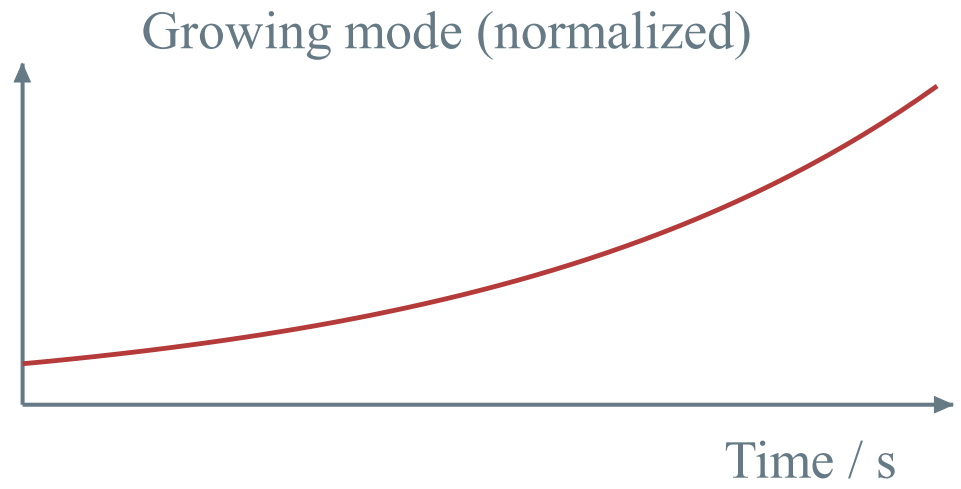



Figure: A growing mode from a positive real root

$$s^2 + 8.014s - 9.268 = 0 \quad \Rightarrow \quad s \approx 1.0253,\ -9.0393$$

**An unstable pole introduces a growing mode; short-record fit cannot replace physical checks.**

### 4.7.4 Structure Selection and Parameter Updates: Motor Convergence and Rigid-Body Integration

# Pole Criteria Depend on the Motion Channel

- A first-order motor settles under fixed input. Undamped angular speed grows with constant torque; position double-integrates constant net force. Do not force rigid-body channels into settling curves.
- The position input is a down-axis net-force increment; an upward total-thrust increment needs a minus sign. Some inherently unstable flight modes rely on control for stability. Judge poles by the system.

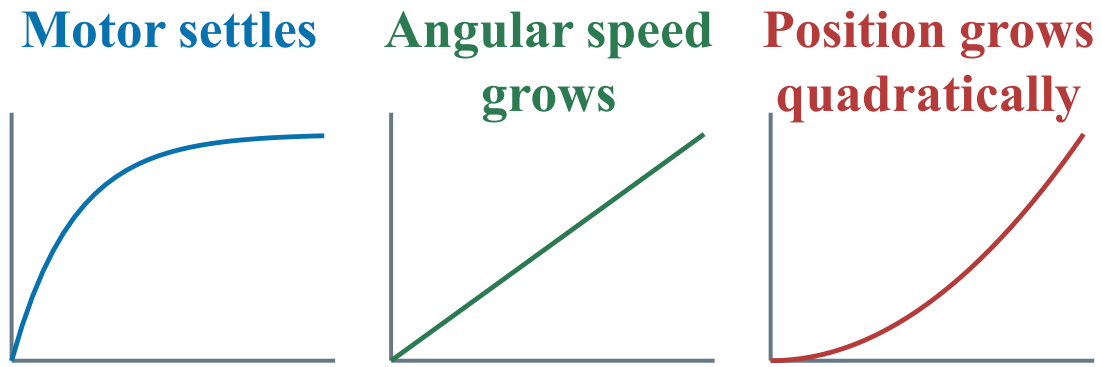



Figure: Different responses under constant inputs

$$G_{\text{motor}} = K/(\tau s + 1), \quad G_{M\to p} = 1/(Js), \quad G_{F_D\to z_D} = 1/(ms^2)$$

**Distinguish actuators, free rigid bodies, and controlled systems before judging stability.**

### 4.7.4 Structure Selection and Parameter Updates: Fit Scores Across Different Signal Boundaries

## Historical Successes and Failures Belong to Their Own Boundaries

- The manual's open-loop frequency-domain case reached at best 43.51% across orders 1–5. A separate FX200 closed-loop pitch-reference-to-pitch model reached 85.49% at third order. Their systems and boundaries differ.
- The latter includes the controller and cannot replace the bare-vehicle torque-to-attitude relation. These values do not fairly rank open- and closed-loop methods or represent all channels and conditions.

| System boundary | Order | Fit |
|---|---|---|
| Open-loop case (frequency domain; no controller) | Best of orders 1–5 | **43.51%** |
| FX200 pitch loop: ref. → actual angle (with controller) | Third order | **85.49%** |

Table: Boundaries, orders, and fit scores from two historical identification cases

**Before comparing fit scores, compare the identified systems and data conditions.**

### 4.7.4 Structure Selection and Parameter Updates: One Replacement with Comparable Conditions

# Recording a Propulsion Submodel Update

- Save the original model and parameter snapshot. State whether the replacement changes a static map, dynamic lag, drag, or another module. Record old and new values, units, and sources.
- Accuracy improvement for a system requires data from that system configuration. Connecting an independent test-stand model to the teaching vehicle must be declared as an assumed component replacement.
- After updating, check hover input, directions, saturation, interfaces, and solver settings. Then compare responses using the same inputs, initial states, and time step.

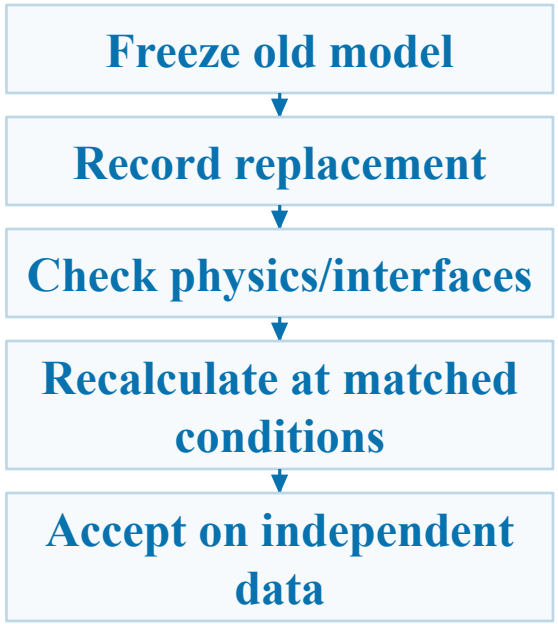



Figure: A comparable model revision

**A parameter update is a controlled model change that must preserve before-and-after comparability.**

### 4.7.4 Structure Selection and Parameter Updates: Acceptance from Component to Full Vehicle

## Static, Dynamic, and Full-Vehicle Checks Build Evidence in Stages

- Static checks cover zero, steady gain, and saturation. Dynamic checks compare rise, delay, and frequency response. They separately assess target value and response speed.
- Full-vehicle checks also cover resultant torque, coordinate transformations, contact conditions, and state integration. A good submodel fit does not automatically establish these connections.
- Claim improved prediction within declared conditions only if errors decrease and the parameters remain physical, interfaces are correct, and independent-condition checks pass.

**Check submodel connections before judging improvement in full-vehicle response.**

## Chapter Outline

# Dynamic System Identification, Validation, and Model Refinement

Chapter 4
Vehicle Motion Modeling and Simulation
4.1 Modeling Fundamentals and Kinematics
4.2 Rigid-Body Dynamics and Solvers
4.3 Propulsion, Forces, and Observations
4.4 Trim, Linearization, and Dynamics
4.5 Multicopter Modeling and Simulation
4.6 Measurement and Static Identification
**4.7 Dynamic Identification and Refinement**
4.8 Typical Vehicles and Extensions
4.9 Modeling Tools and Deployment
4.10 API and SDK Modeling Interfaces
4.11 Examples and Integrated Practice
4.12 Exercises and Course Projects

## 6 Units in This Section

**4.7.1 Identification Scope, Excitation, and Data**
Textbook: Sec. 4.2.2.4

**4.7.2 First-Order Dynamic Model Identification**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.3 Frequency Response and Staged Propulsion Models**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.4 Structure Selection and Parameter Updates**
Textbook: Sec. 4.2.2.4

**4.7.5 Independent Validation and Error Interpretation**
Textbook: Sec. 4.2.2.4

**4.7.6 Full-Vehicle Comparison and Flight-Based Refinement**
Textbook: Sec. 4.2.2.4

### 4.7.5 Independent Validation and Error Interpretation: Free Simulation and One-Step Prediction

# Why Simulation and One-Step Prediction Give Different Errors

- Free simulation uses only the initial state and the input sequence. It advances its own state, allowing errors to accumulate. This tests reproduction of a complete trajectory.
- One-step prediction continually uses measured history. Match prediction horizon, initial state, and data interval when comparing results. A one-step fit cannot replace long-duration free-simulation validation.

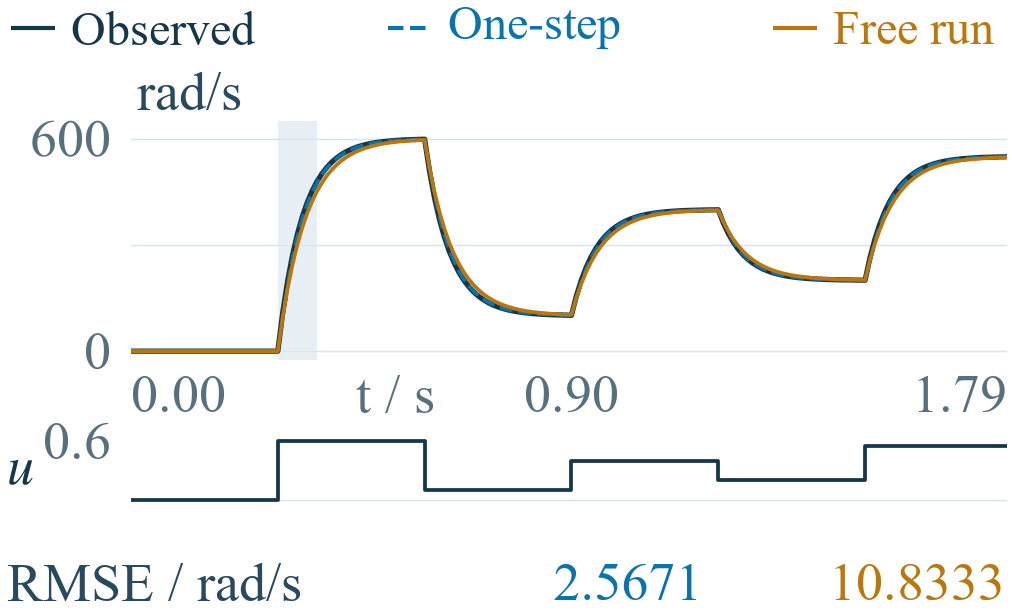



Figure: Two recursion starts, one frozen model

**Simulation accumulates its own errors; prediction can use measured history to inform the next step.**

### 4.7.5 Independent Validation and Error Interpretation: RMSE Calculation and Units

# Root Mean Square Error Retains the Original Units

- For simultaneous values, square $e_k = y_k - \hat{y}_k$, average, then take the square root. RMSE keeps the original units, and positive and negative errors cannot cancel.
- Teaching altitude errors of 0.1, −0.2, and 0.1 m have mean 0 but RMSE 0.1414 m. Report units for each channel to compare with the task's allowed deviation.

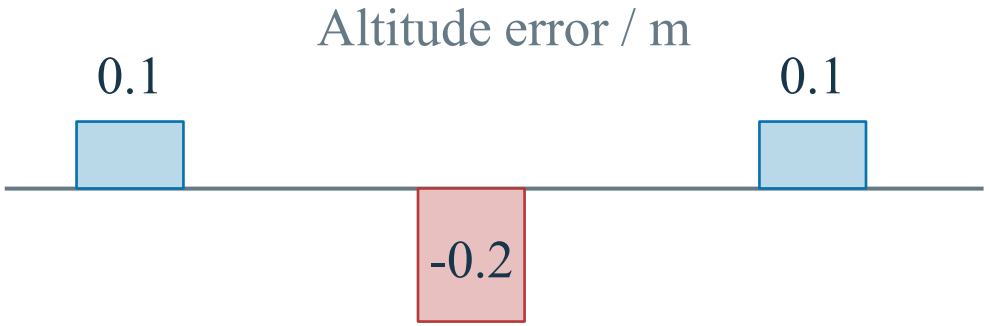



Figure: Three altitude errors

$$\mathrm{RMSE} = \sqrt{\frac{e_1^2 + \cdots + e_N^2}{N}}, \qquad \sqrt{\frac{0.1^2 + 0.2^2 + 0.1^2}{3}} = 0.1414\,\mathrm{m}$$

**An error with units relates directly to the task's permitted deviation.**

### 4.7.5 Independent Validation and Error Interpretation: FIT Denominator and Degenerate Cases

# The FIT Denominator Defines the Comparison

- Normalized fit compares model error with the error from predicting the whole record's mean at all times. A worse-than-mean prediction can give negative FIT. A strictly constant output makes its denominator zero.
- The teaching new-input validation gives 98.917% by the formula below and RMSE 2.148 rad/s. Report both with the sequence, units, and free-simulation definition. One percentage cannot specify the model's valid range.

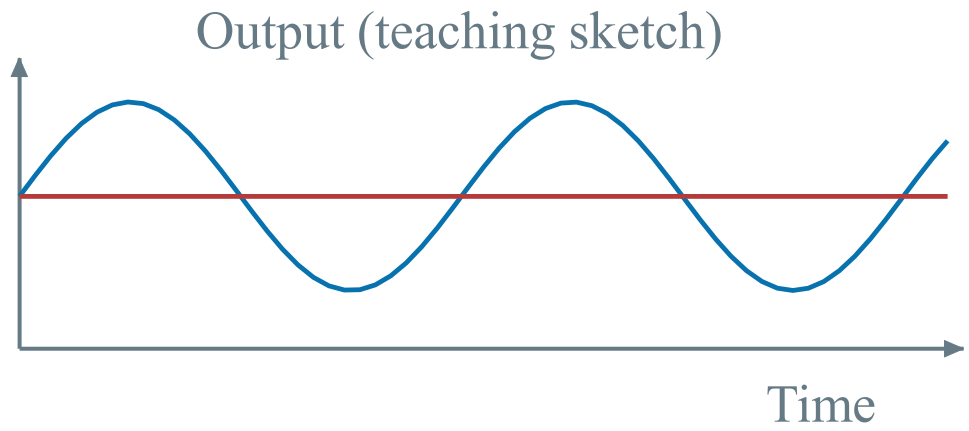



Figure: Red is the mean-value prediction

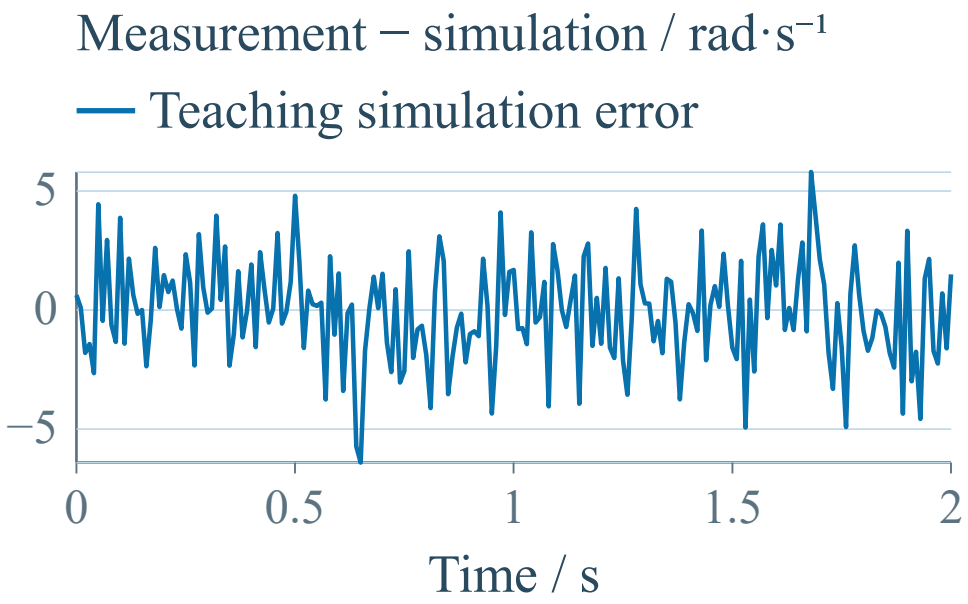
$$\mathrm{FIT} = 100\left(1 - \frac{\| \mathbf{y} - \hat{\mathbf{y}} \|_2}{\| \mathbf{y} - \bar{y}\mathbf{1} \|_2}\right)$$

**Normalized fit evaluates error relative to output variation and may lose meaning on constant segments.**

### 4.7.5 Independent Validation and Error Interpretation: Missing Dynamics from Error Patterns

# Error Patterns Reveal More Than One Overall Score

- Persistent bias suggests zero or gain error. Peaks after every switch suggest alignment, delay, or missing dynamics. Error that grows systematically with amplitude may indicate nonlinearity; periodic patterns may indicate unexplained excitation response.
- Here the error is prescribed independent random noise with no repeated structure. Real-data causes also require the input, residual definition, and independent trials.




Teaching validation error: measurement − free run

**Bias, delay, and omitted dynamics leave different time structures.**

### 4.7.5 Independent Validation and Error Interpretation: Residual Correlation and Closed-Loop Noise

# Residuals Reveal Predictable Information the Model Misses

- Identification tools often define residuals as one-step prediction errors, rather than always measurement minus free simulation. State the predictor and noise model before discussing white-noise behavior.
- Correlation with past residuals may reveal predictable dynamics; correlation with past inputs may reveal a missing input relation. Closed-loop noise affects later inputs, so correlations at all positive and negative lags cannot mechanically count as failures.

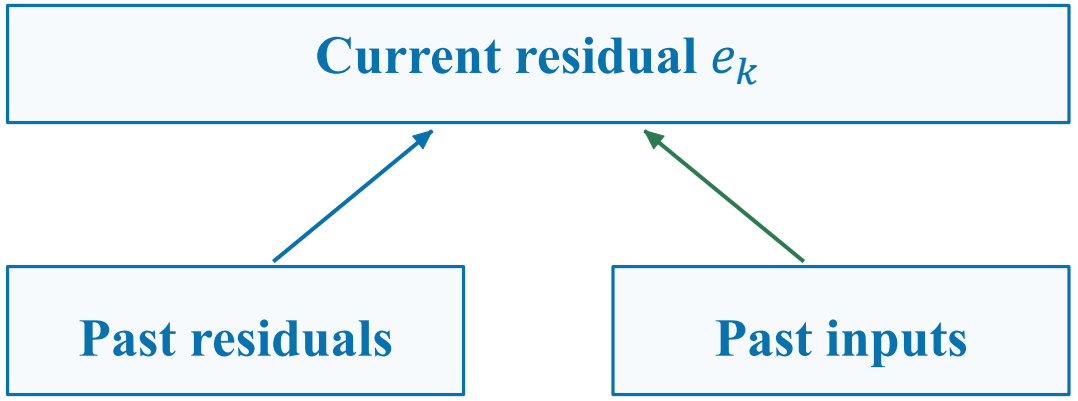
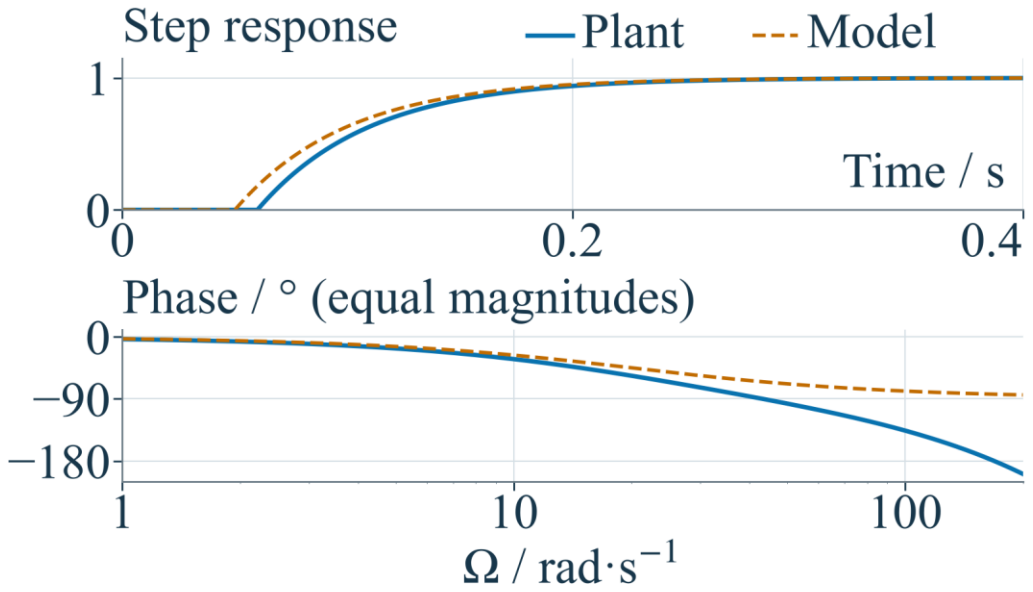



Figure: Define error, then compare it with past information

**The residual definition and feedback conditions govern how correlations should be interpreted.**

### 4.7.5 Independent Validation and Error Interpretation: Time Trajectories and Frequency Magnitude/Phase

# Trajectories and Frequency Responses Test Different Errors




Teaching sketch: +10 ms barely changes the step, but adds ≈57° lag at 100 rad/s.

- Match inputs, initial states, and measurement definitions. Time-domain checks compare steady offset, rise, overshoot, delay, and RMSE; frequency-domain checks compare magnitude, phase, and credible band.
- Similar trajectories can hide large phase errors in one band. Low-frequency agreement cannot establish suitability for fast maneuvers or high-frequency control. Accept against the intended task.

**Time-domain checks examine motion; frequency-domain checks examine response across time scales.**

## Chapter Outline

# Dynamic System Identification, Validation, and Model Refinement

Chapter 4
Vehicle Motion Modeling and Simulation

4.1 Modeling Fundamentals and Kinematics
4.2 Rigid-Body Dynamics and Solvers
4.3 Propulsion, Forces, and Observations
4.4 Trim, Linearization, and Dynamics
4.5 Multicopter Modeling and Simulation
4.6 Measurement and Static Identification
**4.7 Dynamic Identification and Refinement**
4.8 Typical Vehicles and Extensions
4.9 Modeling Tools and Deployment
4.10 API and SDK Modeling Interfaces
4.11 Examples and Integrated Practice
4.12 Exercises and Course Projects

## 6 Units in This Section

**4.7.1 Identification Scope, Excitation, and Data**
Textbook: Sec. 4.2.2.4

**4.7.2 First-Order Dynamic Model Identification**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.3 Frequency Response and Staged Propulsion Models**
Textbook: Secs. 4.1.4.3, 4.2.2.4

**4.7.4 Structure Selection and Parameter Updates**
Textbook: Sec. 4.2.2.4

**4.7.5 Independent Validation and Error Interpretation**
Textbook: Sec. 4.2.2.4

**4.7.6 Full-Vehicle Comparison and Flight-Based Refinement**
Textbook: Sec. 4.2.2.4

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Textbook Platform

## In-the-Loop and Flight Tests Need Motion and Input Comparisons




(e) Quadcopter HIL Simulation Test




(f) Quadcopter Real Flight Test

- Panel (e): a physical autopilot with a simulated plant; (f): real flight. Before comparison, record configuration, load, voltage, wind, and controller settings.
- Align motion states and also compare control inputs such as PWM. This unit's FX200 curves are historical manual results; a new configuration needs new validation.

Figure 4.13(e)(f): Hardware-in-the-loop and physical flight tests (details)

**Full-vehicle comparisons test the connected plant, interfaces, sensors, and controller.**

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Joint Checks of States and Actuator Inputs

# Similar Trajectories Can Require Different Control Effort

- **Align the task:** Under the same velocity command, align time using shared events such as command switches. Record load, voltage, wind, control parameters, and measurement source.
- **Quantify together:** Compare velocity peaks and settling time, plus each PWM channel's mean and peak over the same stage. Keep each quantity's units.
- **Locate differences:** Small state errors with persistent input bias call for checks of controller settings, supply, and thrust mapping, followed by a new segment under matched conditions.

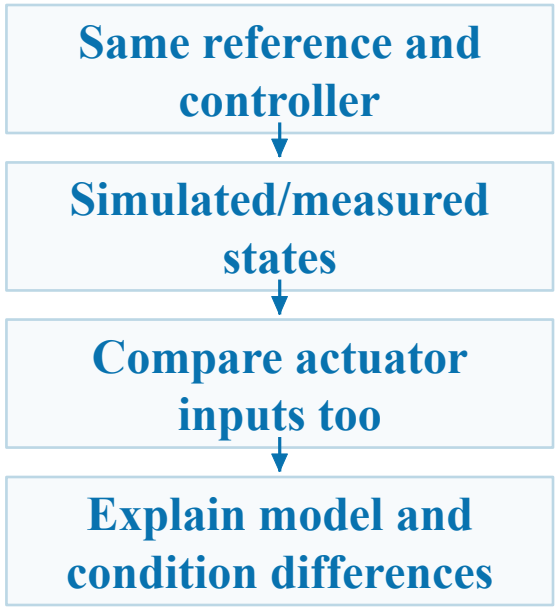



Figure: Joint comparison of states and inputs

**Strong control can mask plant differences; validate states and inputs together.**

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Full-Vehicle Case: FX200 Step Validation

# FX200 Velocity Steps on One Shared Time Axis

- **Reading:** Blue is the command, orange physical flight, and dashed green HITL. Compare direction, peaks, transients, and steady values over the same interval, without shifting individual curves.
- **Observation:** Both responses track steps of about ±0.5 m/s with similar main peaks and settling. Small flight fluctuations still differ from the model.

- **Next:** Examine the control effort used to produce this velocity curve on the propulsion-model diagnosis page.

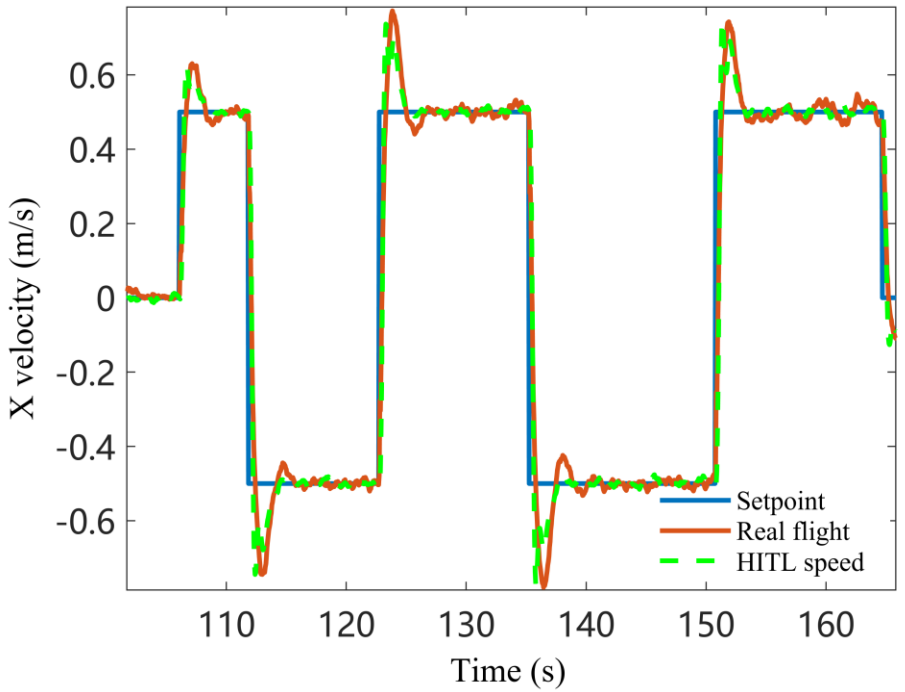


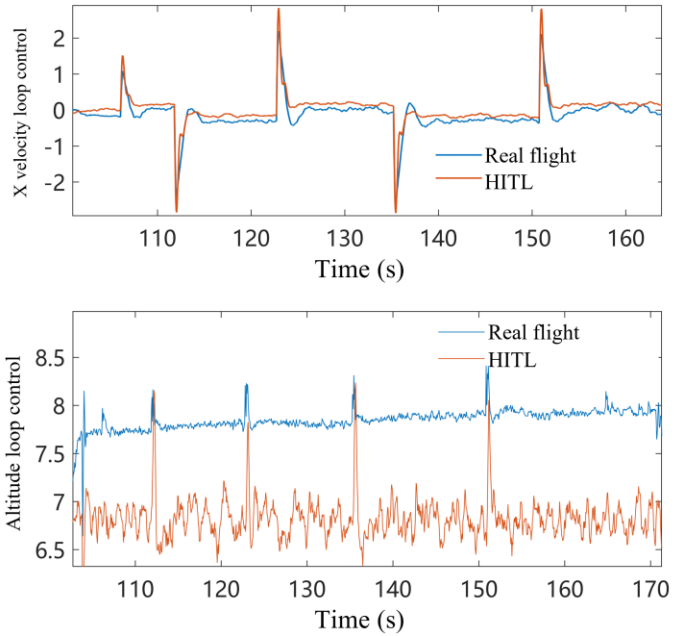

Figure: Historical FX200 velocity command, physical flight, and HITL step responses

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Propulsion Diagnosis

# Close Velocities Still Hide Systematic Control Differences

- **Lateral control:** Physical-flight and HITL transient peaks above are broadly similar, but plateau bias remains. See the enlarged velocity-loop control plot.

- **Altitude:** Below, physical flight is about 7.8 and HITL about 6.8, with stronger HITL fluctuations. The axis says only control signal, with no physical unit. The difference cannot directly mean 1 N.
- **Diagnosis:** Check this signal's definition, mass, voltage, and thrust calibration, then wind, ground effect, and log processing. This figure alone cannot uniquely identify a faulty coefficient.




Overview (manual Figure 94): Velocity loop above, altitude loop below.
The next two pages enlarge their control signals.

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Propulsion Diagnosis

# Velocity-Loop Control: Similar Peaks, Biased Plateaus

Similar transient peaks

Plateau bias suggests propulsion-model differences

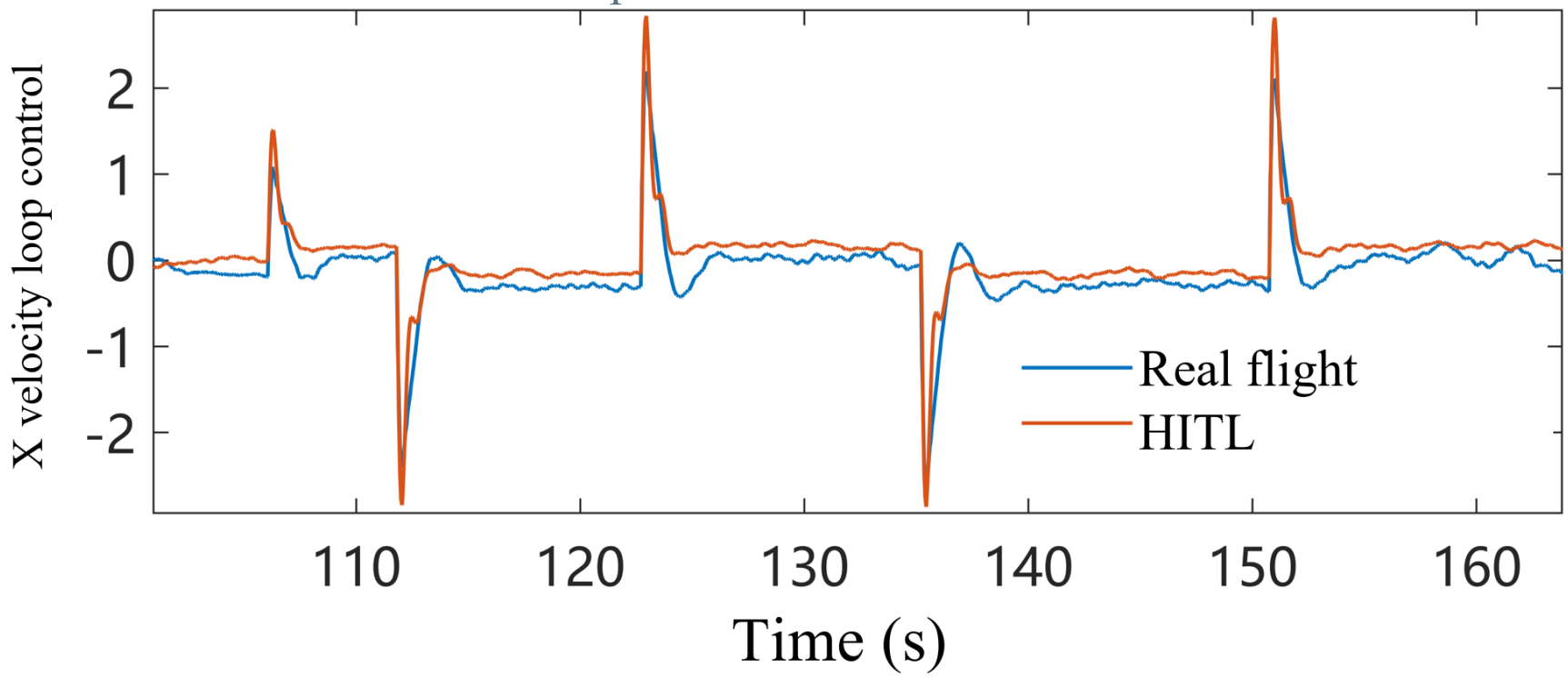
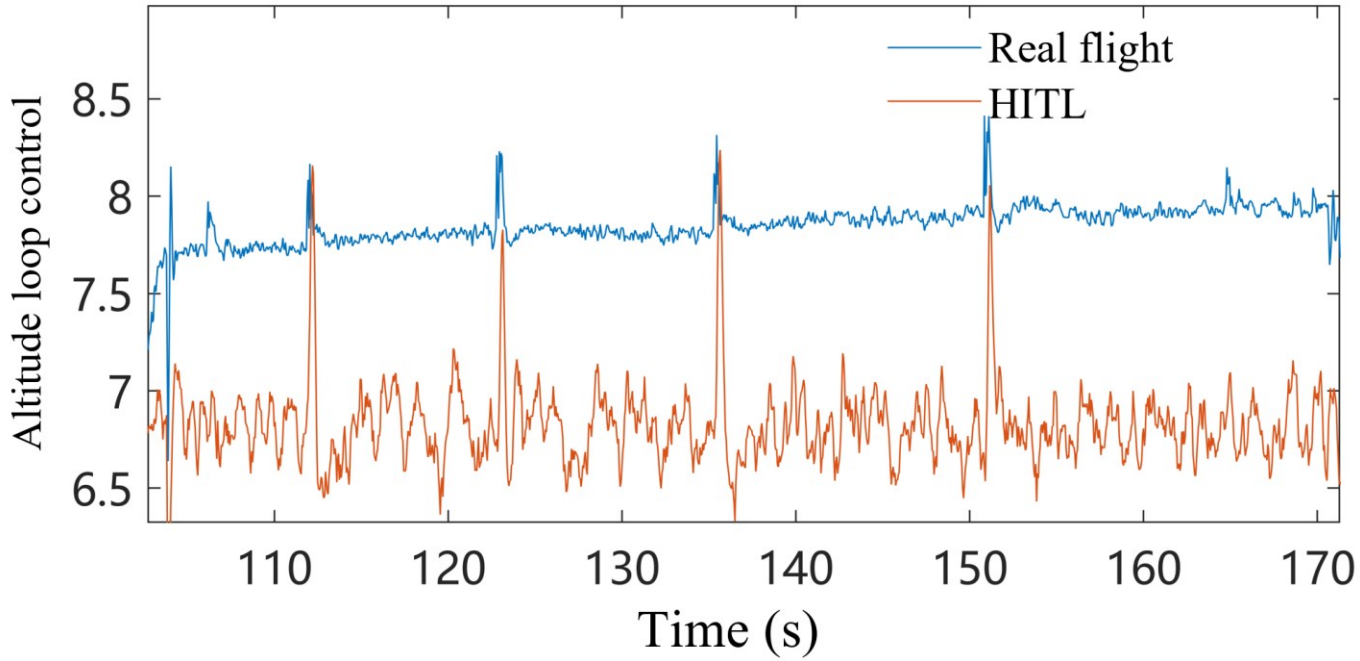



Figure: FX200 velocity-loop control comparison (detail of manual Figure 94, top)

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Propulsion Diagnosis

# Altitude-Loop Control: A Persistent Mean Difference




Physical flight ≈7.8

HITL ≈6.8; stronger fluctuations

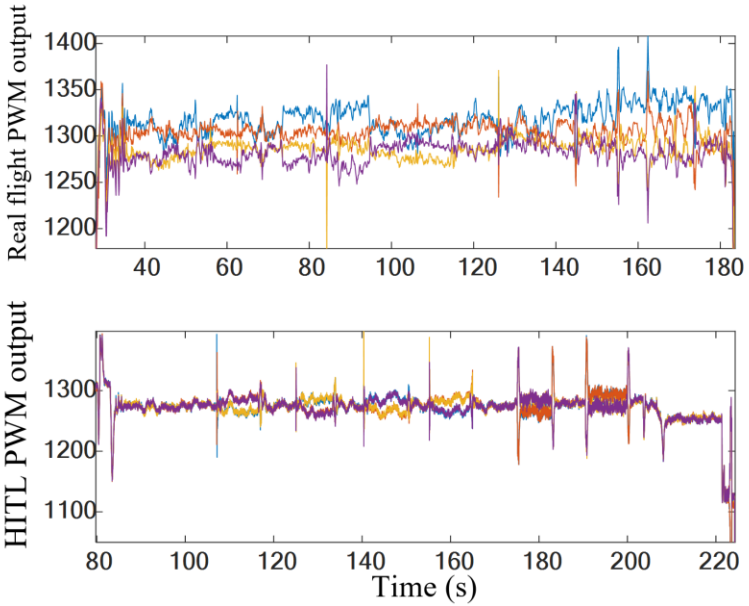
Figure: FX200 altitude-loop control comparison (detail of manual Figure 94, bottom)

**Investigate the sustained altitude-control difference; establish units before interpreting it as a force.**

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Full-Vehicle Case: Further Input-Side Evidence

## Four PWM Channels Suggest Center-of-Mass and Mounting Checks

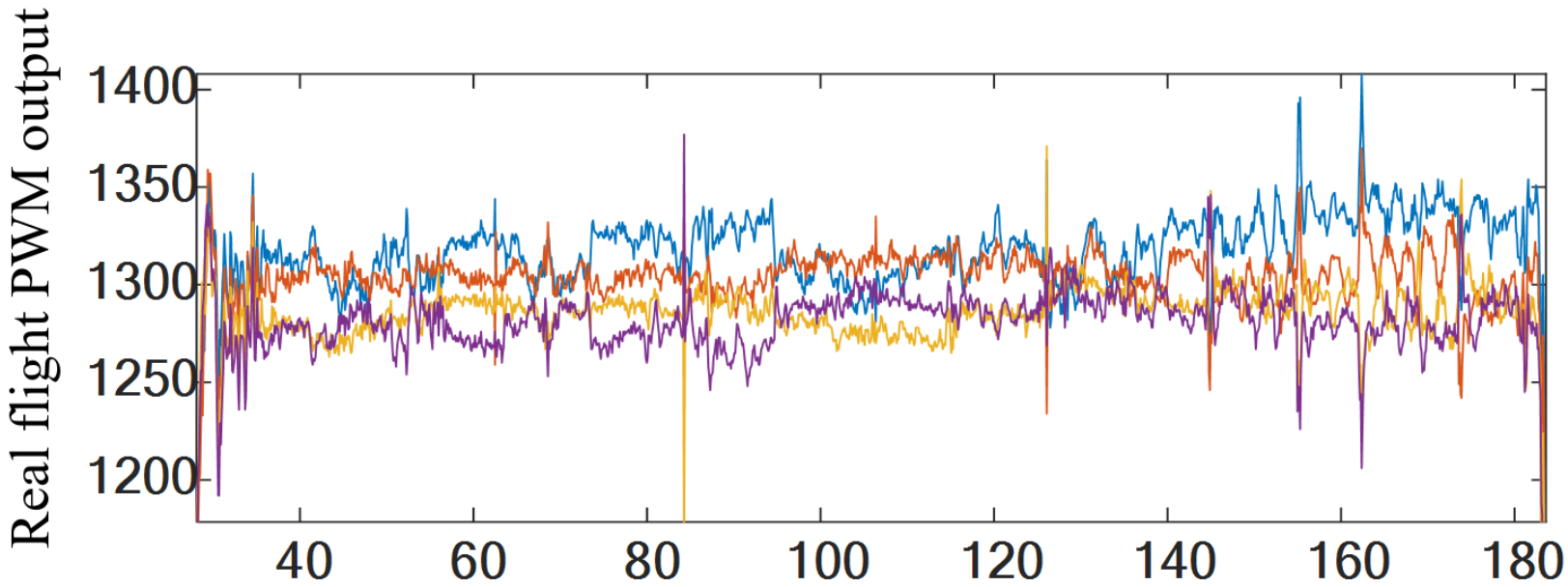
- **Reading:** Physical flight is above and HITL below. All four PWM channels lie near 1300, but flight channel means are more dispersed; model channels cluster more closely.
- **Physical interpretation:** Unequal thrust may counter center-of-mass offset or mounting error. Ideal symmetry yields closer channel commands. Geometry measurements and calibration must support this explanation.
- **Comparison boundary:** The time ranges differ. Compare distributions and magnitudes first. Before pointwise subtraction, align a shared reference event and check commands, initial states, and control configuration.




Overview (manual Figure 99): Physical flight above, HITL below. The next two pages enlarge the four PWM channels.

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Full-Vehicle Case: Further Input-Side Evidence

# Flight PWM Channels Have Different Long-Term Means



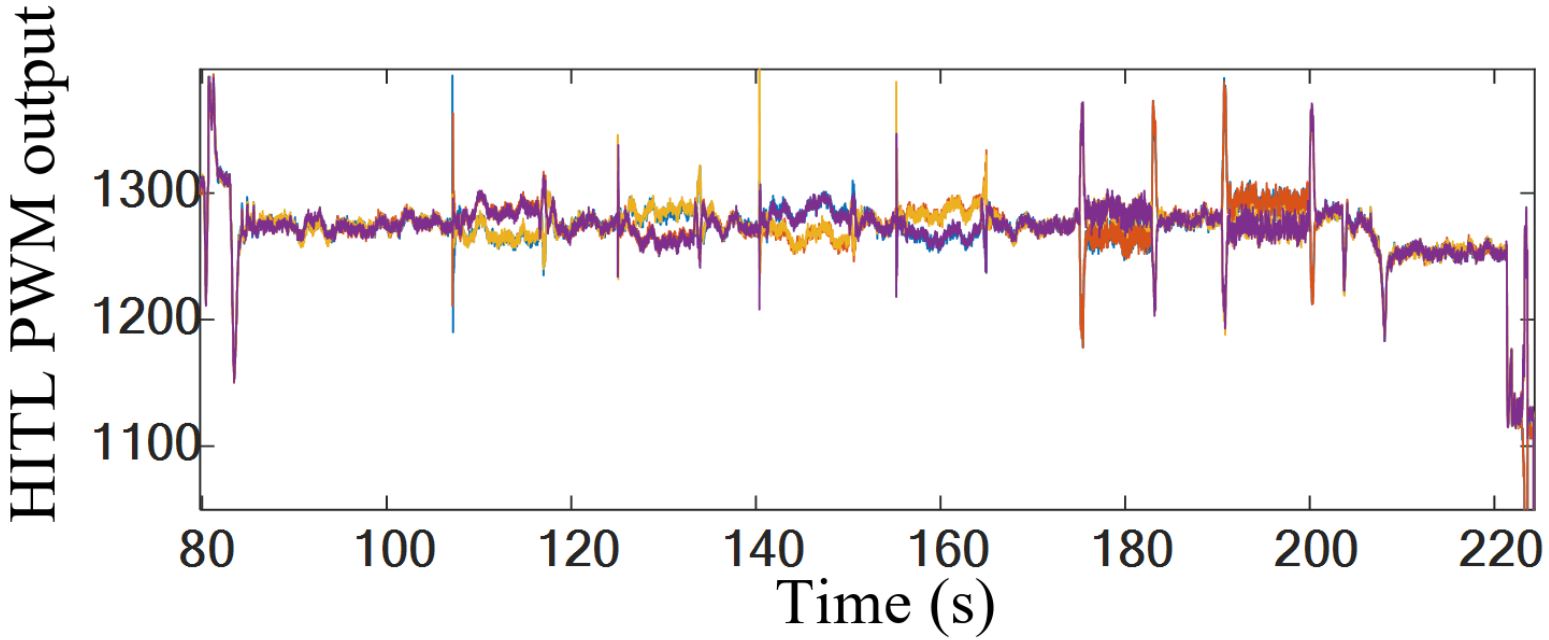

Separated long-term channel means

Figure: Four FX200 physical-flight PWM channels (detail of manual Figure 99, top)

**The four inputs differ; geometry, mounting, and propulsion differences need independent measurements.**

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Full-Vehicle Case: Further Input-Side Evidence

# HITL PWM Commands Cluster in the Symmetric Model




Closer channel levels;
time starts near 80 s

Figure: Four FX200 HITL PWM channels (detail of manual Figure 99, bottom)

**The records span different times. Compare distributions first, then align shared events.**

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Drag from Attitude and Airspeed

# Steady Level Flight Constrains Drag Through Force Balance

- **Conditions:** Steady forward flight; no sideslip; heading along motion; nearly constant speed and altitude. Forward body-tilt magnitude: $\alpha$. Neglect other vertical aerodynamic forces: $T\cos\alpha = mg$.

- **Derivation:** Horizontal balance gives $D = T\sin\alpha$, hence $D = mg\tan\alpha$. With forward airspeed $v_a > 0$, fit candidates such as $D = c_1 v_a$ or $D = c_2 v_a|v_a|$ over several steady points.
- **Teaching example:** For $m = 2\,\text{kg}$, $g = 9.8\,\text{m/s}^2$, and $\alpha = 5°$, $D \approx 1.715\,\text{N}$. At true airspeed 5 m/s, $c_1 = D/v_a \approx 0.343\,\text{N s/m}$.
- **Real-data boundary:** Wind is present in the manual's flight data. Subtracting mean ground speed does not remove wind. Measure or estimate relative airspeed. One speed point also cannot distinguish linear from quadratic drag.

**Steady force balance constrains drag; airspeed and operating conditions determine its physical meaning.**

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Flight Refinement: Updating and Revalidating

# Flight Trim Conditions Constrain Static Thrust Refinement

- **Usable segments:** Average steady altitude, velocity, and attitude. Log voltage, load, ground effect, and inflow. Inertial forces prevent direct use of steady balance during acceleration.
- **Linear operating region:** Over a narrow range, use per-propeller $T_i = k_u u_i + c_0$, where $u_i$ is normalized command and $k_u$, $c_0$ are slope and intercept. Sum all four thrusts.
- **Build samples:** Each steady segment contributes one mean-value row below; $\phi$ and $\theta$ are roll and pitch. Stack segments into a matrix to estimate $k_u$ and $c_0$ jointly.

$$\begin{bmatrix} \sum_{i=1}^{4} u_i & 4 \end{bmatrix} \begin{bmatrix} k_u \\ c_0 \end{bmatrix} \approx \frac{mg}{\cos\phi \cos\theta}$$

**Test component parameters in full-vehicle equilibrium, then use new conditions to decide whether to retain the update.**

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Identifiability and Acceptance

## New Flight Segments Decide Whether a Refinement Is Retained

- **Limited information:** Similar throttles give similar regression rows, so slope and intercept cannot be separated reliably. Constrain parameters with test-stand data or add comparable operating points.
- **Freeze model:** Refine only on preselected estimation segments, then freeze parameters. Compare velocity, attitude, and actuator inputs on flight segments unused for refinement.
- **Control confounding:** Record controller, load, voltage, wind, and ground effect. If model parameters and control configuration change together, improvement cannot be attributed to propulsion refinement alone.
- **Case conclusion:** The manual illustrates equilibrium-based refinement with joint state and control comparisons. Its project constants apply only to the original vehicle. Recalibrate and accept on independent segments before transfer.

**Separate refinement from independent acceptance; test improvement under new conditions.**

### 4.7.6 Full-Vehicle Comparison and Flight-Based Refinement: Model Delivery and Validation Conditions

## Stage Experiment A2: Reproducible Models and Validation Boundaries

- **Choose one basic route:** Refit e11 statics, or fit teaching-CSV dynamics and validate with a new input. Extension: Explain a historical plot's signal boundary, processing conditions, and conclusion.
- Submit/check parameters, units, estimation rules, frozen model, independent input, errors, and valid range; match the figure. A2 is Chapter 4's second stage experiment. Sequence: Section 4.3, "Stage Experiment A1: Actuators and Force/Torque Synthesis."

**Real statics (e11 test stand)**

91 settings: $c_T \approx 2.657 \times 10^{-5}\ \mathrm{N/(rad/s)^2}$

**Teaching dynamics CSV**

a≈0.81873, b≈181.27, FIT≈98.917%

Figure: Self-check values for the two routes

**A complete identification result includes the model, data, errors, and applicable conditions.**

Classroom Exercises

## New-Data Exercises: Parameter Conversion, Coefficients, and Errors

- **Exercise 1 (parameter conversion):** Motor data sampled at $T_s = 0.02$ s gives $a \approx 0.81873$ and $b \approx 362.54$. Find $\tau$ and $K$, then compare with the teaching system in 4.7.2.
- **Exercise 2 (coefficient check):** Another record at $T_s = 0.01$ s gives $a \approx 1.02$. What free response does it imply? Which parts should be checked first?
- **Exercise 3 (error metrics):** Three validation measurements are $y$=0, 10, 20 rad/s; free simulation gives $\hat{y}$=1, 9, 21 rad/s. Find RMSE and FIT.
- **Exercise 4 (system boundary):** An altitude-hold model maps altitude setpoint to actual altitude,
  with 92% fit. Can it replace total-thrust-to-vertical-acceleration dynamics? Which two signals should be recorded?

Reference Answers

# Reference Answers: Parameter Conversion, Coefficients, and Errors

- **Exercise 1:** $\tau = -T_s/\ln a = 0.1$ s, $K = b/(1-a) \approx 2000$ rad/s. Although $a$ equals the 4.7.2 value, doubling the sampling period doubles $\tau$. Looking only at $a$ would incorrectly suggest equal response speed.
- **Exercise 2:** $a > 1$ corresponds to a positive real pole and a growing free response, contrary to a stable motor. First check alignment and delay, closed-loop correlation, noise, and structure. The τ conversion does not apply to the assumed stable motor here.
- **Exercise 3:** Errors are −1, 1, −1, so RMSE=1 rad/s. Error norm $\sqrt{3} \approx 1.732$ and output-deviation norm $\sqrt{200} \approx 14.142$ give FIT≈87.75%.
- **Exercise 4:** No. The input is altitude setpoint, so the model includes the altitude controller and feedback. The 92% reflects closed-loop tracking only. Record motor speeds (convert to total thrust) and vertical acceleration, then identify separately under matched conditions.

Section Summary

## Boundary, Excitation, Estimation, Freezing, Validation, and Refinement

- **Method:** Define input-output boundaries and conditions; design steps/sweeps and align timestamps. Estimate by adjacent-sample least squares or frequency response. Freeze, then freely simulate on data untouched by decisions. Compare full-vehicle states and controls; revalidate on new segments.
- **Key judgments:** A stable first-order model without delay has $0 < a < 1$. Positive real poles give growing modes. Do not force fits in low-coherence bands. FIT normalizes by output variation about its mean and may be negative.
- **Common errors:** Treating a reference-to-output closed loop as the bare plant; mistaking misalignment for delay; substituting one-step prediction for free simulation; ranking fits across different boundaries; calling training error independent validation.
- **Next:** Section 4.8 applies modeling and validation to other typical vehicles. Stage Experiment A4 in Section 4.12 completes one model change with independent validation.

**Deliver the model and its limits: freeze, validate independently, then decide on refinement.**