4.6 Measurement and Static Identification

Chapter Outline

Parameter Measurement and Static Model Identification

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

5 Units in This Section

4.6.1 Parameter Sources and Physical Measurement
Textbook: Secs. 4.2.2.1 and 4.2.2.4

4.6.2 Steady-State Data and Units
Textbook: Sec. 4.2.2.4

4.6.3 Least Squares and Thrust Fitting
Textbook: Eq. (4.29), Sec. 4.1.4.4; Sec. 4.2.2.4

4.6.4 Overall and Staged Static Models
Textbook: Sec. 4.2.2.4

4.6.5 Reproduction, Parameter Updates, and Validity
Textbook: Sec. 4.2.2.4

4.6.1 Parameter Sources and Physical Measurement: Apply Equations to a Specific Object

Equal Thrust: Parameters, Initial States, and Inputs

- Equations define the calculation; parameters specify the object. Mass m and inertia J_x describe it. Initial height and velocity define the starting state; thrust over time defines the input.

- With known structure, measure accessible parameters directly and identify the rest from data (gray box). With unknown structure, identify structure and parameters from inputs/outputs (black box). Validate with data not used for tuning.

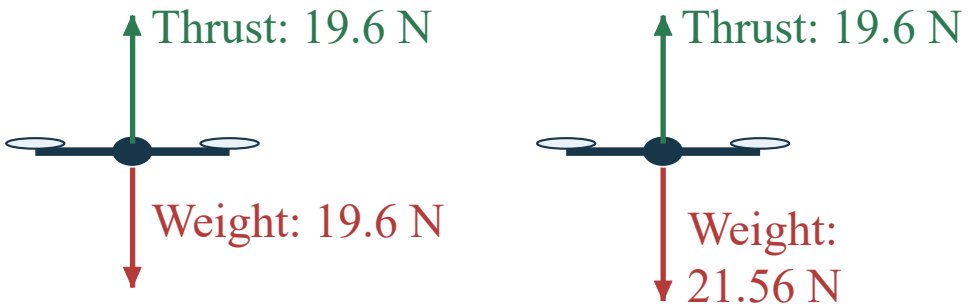


Figure: Equal thrust, different mass and net force

Match parameters to the object and conditions to define what the equations predict.

4.6.1 Parameter Sources and Physical Measurement: White-Box, Gray-Box, and Black-Box Models

Classify Models by Known Structure and Parameters

- **Textbook Sec. 4.2.2.1:** Structure defines variables and relations (e.g., differential equations or transfer functions); parameters give mass, damping, or gain. Compare motors on "System Model Types" in Section 4.1.

White Box	• Known structure and parameters; derive equations from first principles such as mechanics. Simscape and Gazebo provide motor and electronic speed controller (ESC) components. No practical model is fully white box.
Gray Box	• **Known structure:** measure or derive some parameters; identify the rest from experiments. Example: measure multicopter mass and inertia; identify thrust and aerodynamic drag coefficients.
Black Box	• Initially unknown structure and parameters; learn from inputs/outputs, e.g., autoregressive with exogenous input (ARX) and nonlinear ARX (NARX). Fewer physical constraints; greater overfitting risk.

- Our gray-box thrust fit: physics gives $F = c_T \omega^2$; test-stand data estimates c_T.

4.6.1 Parameter Sources and Physical Measurement: Parameter Records

Record Sources to Make the Model Reproducible

- Record values, units, configuration, and method. A propeller, ESC, or supply change can invalidate the mapping for the same motor.
- Hardware-in-the-loop (HITL; HIL in the textbook).

Record	Mass	c_T, N/(rad/s)²	Configuration	Acquisition	Source
Teaching model	2 kg	2×10^{-5}	Assumed rigid quadcopter	Assigned value	Section 4.3
e12 template	1.4 kg	1.105×10^{-5}	e12_ModelTempId quadcopter template	Assigned in Init.m	Section 4.5
FX450 data	N/A	2.657×10^{-5}	Single-rotor setup; 22.3–25.0 V	Fit to test-stand records	e11: Static_data.mat
FX200 case	Not listed separately	Not listed separately	Complete FX200 vehicle in the manual	Historical flight & HITL records	Section 4.7

A usable parameter includes its value, units, source, and configuration.

4.6 Measurement and Static Identification

4.6.1 Parameter Sources and Physical Measurement: Measurement Platforms

From Component Measurements to Vehicle Comparisons



Figure 4.13 Hardware and Software Platform for Unmanned Aerial Vehicle System Identification Based on RflySim

- **(a) Vehicle:** Record battery, propeller, and payload configuration; measure mass, center of mass, and inertia.
- **(b) Thrust test stand:** Fix the propulsion assembly; synchronously record command, rotor speed, thrust, and reaction torque.
- **(c)(d) Wind tunnel:** Control inflow speed and mounting angle; measure aerodynamic forces.
- **(e) HITL:** Connect autopilot hardware to the simulation model and record closed-loop responses.
- **(f) Flight test:** Fly the same task in the real environment and compare with (e). Section 4.7 details (e) and (f).

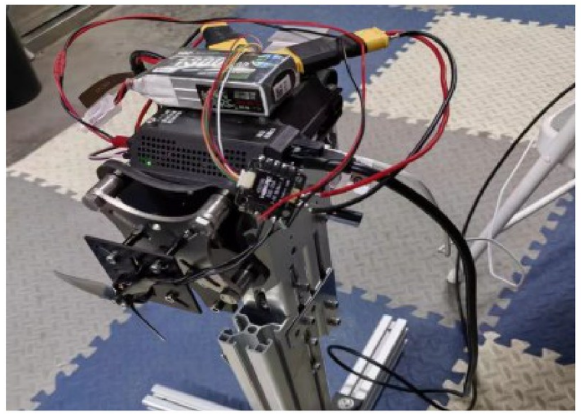
Each experiment controls different conditions; the records jointly constrain model validity.

4.6.1 Parameter Sources and Physical Measurement: Test-Stand Input and Output Boundaries

Measure Command, Rotor Speed, and Thrust



(a) FeisiLab J200 Quadcopter UAV



(b) Motor Thrust Test Rig

Figure 4.13(a)(b) Vehicle and Motor Thrust Test Stand (details)

- In (a), record battery, propeller, and payload. In (b), fix the propulsion assembly and synchronously pair command u, rotor speed ω, and thrust F.
- Fit $F = c_T \omega^2$ from steady-state intervals. Also retain $u \rightarrow \omega$ to distinguish command-to-speed errors from speed-to-thrust errors.

Fix the object and observe intermediate speed to distinguish the motor mapping from rotor thrust.

4.6.1 Parameter Sources and Physical Measurement: Inflow and Environment

A Wind Tunnel Controls Relative Inflow



(c) UAV Wind Tunnel Test Equipment



(d) Wind Tunnel Internal View

Figure 4.13(c)(d) Wind Tunnel and Vehicle Mounting (details)

- Vary inflow speed or mounting angle in (c)(d) and measure forces. Given geometry, computational fluid dynamics (CFD) can provide estimates for wind-tunnel checking.

- Use relative air velocity $\mathbf{v}_a = \mathbf{v} - \mathbf{v}_w$. Interpret force data with inflow, attitude, and reference area to estimate aerodynamic coefficients.

Vary inflow and observe force changes to establish aerodynamic relationships.

4.6.1 Parameter Sources and Physical Measurement: Mass, Center of Mass, and Geometry

Mass, Center of Mass, and Dimensions in Dynamics

- Weigh the full test configuration, including battery and payload. Locate the center of mass by support balance or repeated suspension, then express actuator positions relative to it.

- Shifting to a parallel reference axis adds md^2, where d is the perpendicular axis separation. Distinguish arm length, diagonal spacing, and axial projection; measured lengths are not interchangeable.

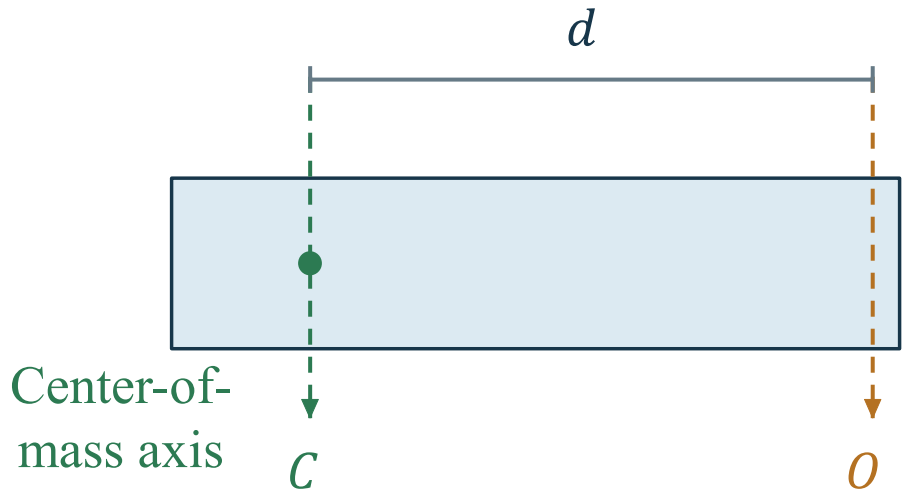


Figure: Shift to a parallel inertia reference axis

$$J_O = J_C + md^2$$

Reference axes and center-of-mass location determine whether measured parameters enter torque equations correctly.

4.6.1 Parameter Sources and Physical Measurement: Bifilar Restoring Torque and Period

Why Can a Bifilar Pendulum Measure Vertical-Axis Inertia?

- Two equal, parallel strings suspend the rigid body symmetrically: spacing d, length L. The test axis is vertical through the center of mass; d is full spacing.
- Small twists raise the center of mass; string tension restores rotation. Equal strings, symmetric mounts, rigid body, small angles assumed. Period T_p: more inertia → longer T_p.

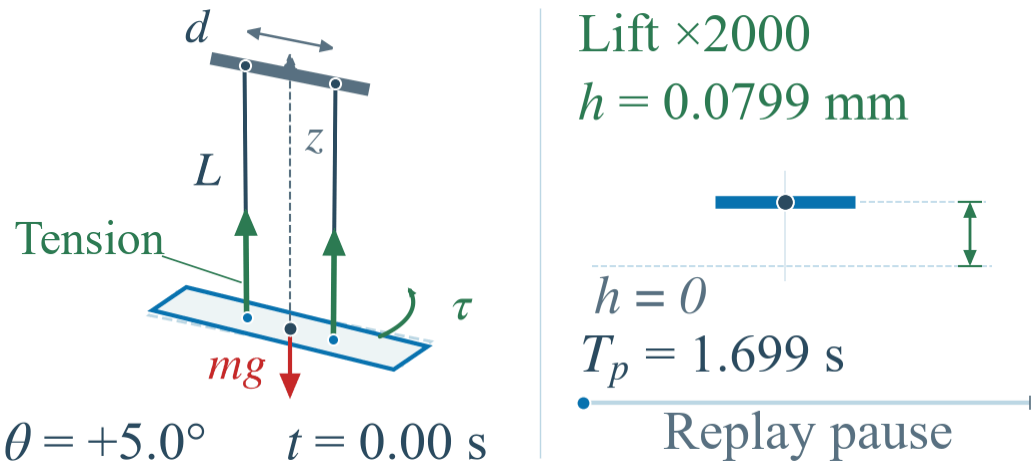


Figure: Bifilar twist, rise, and restoring rotation

$$J\ddot{\theta} + \frac{mgd^2}{4L}\theta = 0, \qquad J = \frac{mgd^2T_p^2}{16\pi^2 L}$$

Twisting raises the center of mass; string tension restores rotation. The period measures rotational inertia.

4.6.1 Parameter Sources and Physical Measurement: Calculate Inertia from Repeated Periods

Calculate Inertia from the Total Time of 30 Periods

- A bifilar test of a vehicle with DJI E310 propulsion (body x axis): $m = 1.4$ kg, $d = 0.23$ m, $L = 0.63$ m, and $g = 9.8$ m/s². Suspend the tested x axis vertically.

1 Timing record

Cumulative times at cycles 10 and 40 are 17.22 s and 68.18 s, spanning 30 periods. The resulting inertia belongs to this E310 vehicle.

2 Mean period

$$T_p = (68.18 - 17.22)/30 = 1.6987\ \text{s}$$

3 Axis inertia

$$J_x = \frac{1.4 \times 9.8 \times 0.23^2 \times 1.6987^2}{16\pi^2 \times 0.63} \approx 0.02105\ \text{kg} \cdot \text{m}^2$$

4.6.1 Parameter Sources and Physical Measurement: Propagation of Measurement Error

Why Does Period Error Grow in the Inertia Result?

- A period measured 1% too high makes inertia $1.01^2 = 1.0201$ times its original value, about 2.01% too high. With all other quantities fixed, inertia is proportional to period squared.
- String spacing is also squared, making attachment-position errors similarly sensitive. More decimal places cannot correct a wrong distance reference or invalid geometry.
- Report the mean and spread of repeated timings. After reducing random timing error, separately address systematic biases such as string length, axis offset, and large oscillation angles.

$$\frac{\delta J}{J} \approx \frac{\delta m}{m} + 2\frac{\delta d}{d} + 2\frac{\delta T_p}{T_p} - \frac{\delta L}{L}$$

Squaring a measured quantity amplifies its relative error.

4.6.1 Parameter Sources and Physical Measurement: Suspension-System Decay

What Can Decaying Oscillation Tell Us?

- First estimate inertia from the ideal bifilar small-angle period. Real amplitudes decay, showing that strings, structure, air, and other elements may dissipate energy.
- To estimate damping from decay, specify the damping model and equation. Viscous damping, friction, and amplitude-dependent resistance produce different decay laws.
- Record the identified axis, object, and suspension setup. See the damping-estimation page for the distinction between setup damping and whole-vehicle aerodynamic damping.

A suspension test measures the inertia and dissipation of the entire experimental system.

4.6.1 Parameter Sources and Physical Measurement: Bifilar Damping: Decay Envelope

Amplitude Decay Rate Is Not a Rotational Damping Coefficient

- **Small-angle equation:** $J\ddot{\theta} + c\dot{\theta} + k\theta = 0$. Viscous rotational damping c has units $\mathrm{N\,m\,s/rad}$, unlike amplitude decay rate. Restoring stiffness k is $k = mgd^2/(4L)$ for a bifilar pendulum.

- **Conversion:** The underdamped envelope $A(t) = A_0 e^{-\lambda t}$ gives $\lambda = c/(2J)$. With $J = 0.02\ \mathrm{kg\,m^2}$ and an envelope multiplied by $e^{-0.1}$ each second, $c = 0.004$.

- **Scope:** Free rotation $J\dot{p} + cp = 0$ instead has decay rate c/J. Measured dissipation includes strings and connectors; it cannot directly represent whole-vehicle aerodynamic damping.

$$c = 2J\lambda = 2 \times 0.02 \times 0.1 = 0.004\ \mathrm{N\,m\,s/rad}$$

Use the decay to establish an equation, then convert decay rate into a damping coefficient with units.

Chapter Outline

Parameter Measurement and Static Model Identification

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

5 Units in This Section

4.6.1 Parameter Sources and Physical Measurement
Textbook: Secs. 4.2.2.1 and 4.2.2.4

4.6.2 Steady-State Data and Units
Textbook: Sec. 4.2.2.4

4.6.3 Least Squares and Thrust Fitting
Textbook: Eq. (4.29), Sec. 4.1.4.4; Sec. 4.2.2.4

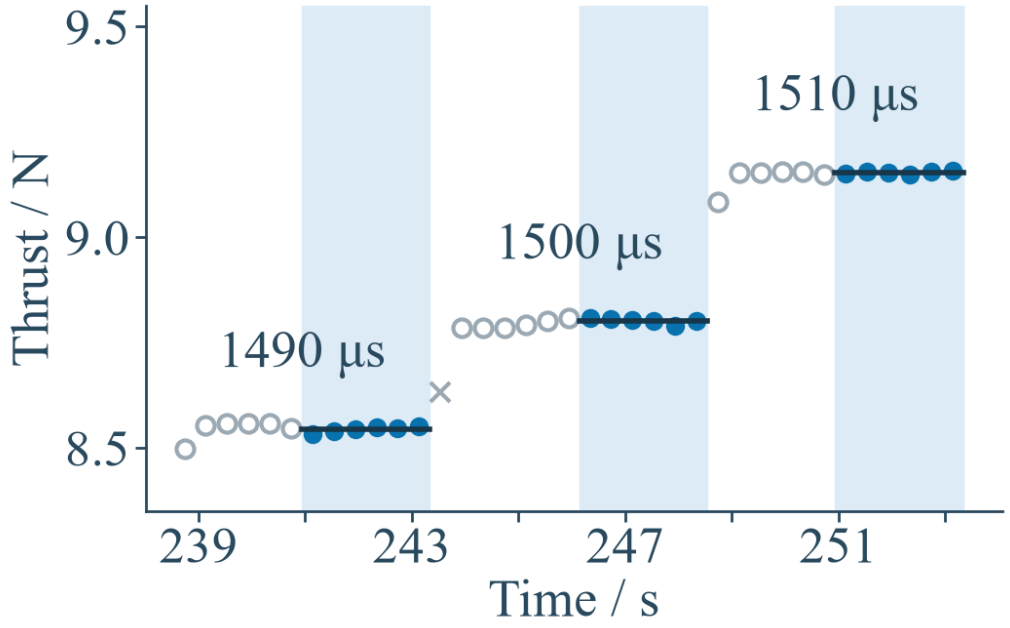
4.6.4 Overall and Staged Static Models
Textbook: Sec. 4.2.2.4

4.6.5 Reproduction, Parameter Updates, and Validity
Textbook: Sec. 4.2.2.4

4.6.2 Steady-State Data and Units: What Makes a Steady-State Sample?

Static Relationships Still Have Measurement Fluctuations

- Hold each command long enough to settle. After a change, hollow points are transients; only the later filled points enter the interval mean (horizontal line). A × marks a transient level with fewer than 6 samples.
- Fixing the test stand removes whole-body displacement only. The thrust sensor can still have zero offset, vibration, and thermal drift; supply voltage can change with load.
- Retain each interval's mean, spread, and time range to trace anomalous points.



Static_data.mat, rows 552–588 (offline plot)

Static describes the steady-state input-output relationship; it does not imply noise-free measurement.

4.6.2 Steady-State Data and Units: Records, Units, and Supply Conditions

Read Field Names and Units Before Selecting Fitting Data

- Static_data.mat: 2385 records, about 0.4 s apart. FKG is a thrust reading in kg; ×9.8 gives N. NRPM × $2\pi/60$ gives rad/s; TNM is already in N · m.

- The neighboring XLSX starts at 16.6 V; the MAT starts at 24.68 V. Times and inputs also differ: these are separate records. Use only the MAT here.

PWMus →	Pulse-width modulation (PWM) / μs
NRPM →	Multiply by $2\pi/60$ → rad/s
FKG →	Multiply by 9.8 → N
TNM, UV →	Original units: N·m, V

Figure: Retain raw fields, then form quantities with units

Unit conversion changes numerical coefficients and must precede regression.

4.6.2 Steady-State Data and Units: From Raw Fields to Calculation Variables

Convert Raw Fields into Numerical Arrays with Units

- **Load and convert:** Set MATLAB's current folder to [installation directory]/RflySimAPIs/4.RflySimModel/2.AdvExps/ e11_PowerSysIdModel/2.StaticThrIntID/ (see its Readme.pdf), then load Static_data.mat. Convert FKG using $g = 9.8\ \mathrm{m/s^2}$ as in the example. Confirm monotonic time before selecting intervals.

```
raw = load('Static_data.mat');
pwm_us = raw.PWMus(:);
omega = raw.NRPM(:)*(2*pi/60);
F_N = raw.FKG(:)*9.8;
Q_Nm = raw.TNM(:);
t_s = raw.TIMEms(:)/1000;
```

Standardize physical quantities and time before any fitting.

4.6.2 Steady-State Data and Units: Extract 91 Steady-State Levels

Extract 91 Steady-State Points from One Stepped Record

- First rising sweep only: Equal consecutive PWM values form intervals. Keep those with ≥6 samples; average their latter half: 91 points at 1050–1950 μs. All later fits/errors use this same recording.

- At 1500 μs, MATLAB rows 571–576 average 5607.85 rpm, 8.80146 N, and 24.235 V. Noise and thermal drift remain on the fixed stand; transient samples are excluded.

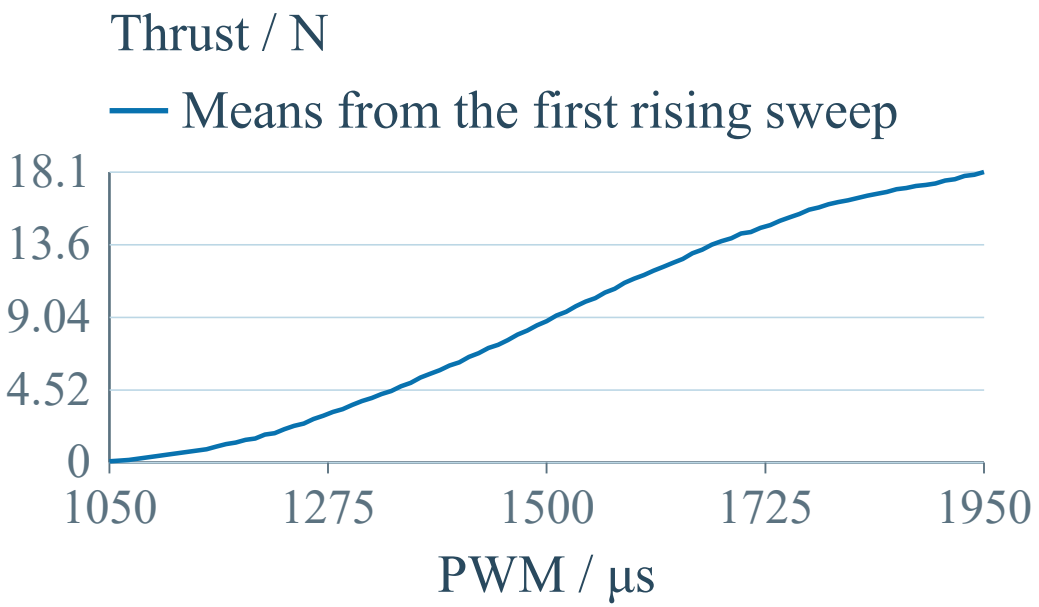


Figure: 91 level means from the recorded MAT data

Processing rules and retained sample intervals jointly determine the fit.

4.6.2 Steady-State Data and Units: Segment Averaging

Segment by PWM; Average Latter Halves into 91 Rows

Use pwm_us and F_N from the raw-field conversion page. Equal consecutive PWM values form intervals. Keep intervals ending by row 1139 (first rising sweep), with ≥6 samples; average their latter half.

```
last = 1139;   % Last row of this record's first rising sweep
st = [1; find(diff(pwm_us) ~= 0) + 1];
en = [st(2:end) - 1; numel(pwm_us)];
rpm = []; thrust_N = [];
for j = find(en <= last & en - st + 1 >= 6)'
    idx = st(j) + floor((en(j) - st(j) + 1)/2) : en(j);
    rpm(end+1,1) = mean(raw.NRPM(idx));
    thrust_N(end+1,1) = mean(F_N(idx));
end
D = table(rpm, thrust_N);   % 91 rows
```

- **st, en:** First and last interval rows
- **Keep:** Last row ≤1139, samples ≥6
- **idx:** Latter-half row indices
- **D:** 91 rows; rpm and thrust_N match the e6 comma-separated values (CSV) file

Chapter Outline

Parameter Measurement and Static Model Identification

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

5 Units in This Section

4.6.1 Parameter Sources and Physical Measurement
Textbook: Secs. 4.2.2.1 and 4.2.2.4

4.6.2 Steady-State Data and Units
Textbook: Sec. 4.2.2.4

4.6.3 Least Squares and Thrust Fitting
Textbook: Eq. (4.29), Sec. 4.1.4.4; Sec. 4.2.2.4

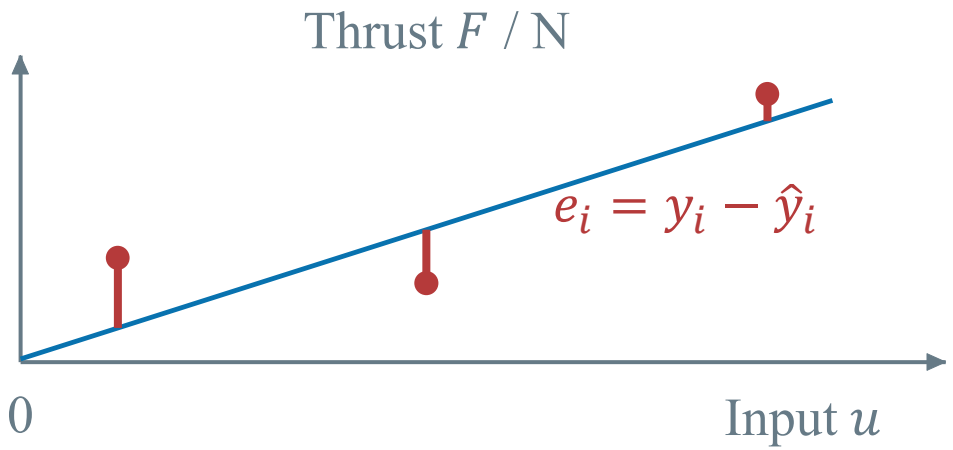
4.6.4 Overall and Staged Static Models
Textbook: Sec. 4.2.2.4

4.6.5 Reproduction, Parameter Updates, and Validity
Textbook: Sec. 4.2.2.4

4.6.3 Least Squares and Thrust Fitting: Least Squares

Least Squares Defines What "Closest to the Data" Means

- For input u_i, predict $\hat{y}_i$. Measured minus predicted is $e_i = y_i - \hat{y}_i$. Signed errors can cancel; squared errors count every deviation.
- Least squares minimizes this sum over parameters within the chosen structure. Outliers or a wrong structure may still bias estimates away from physical values.



Residuals: Vertical distances (teaching schematic)

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}(e_1^2 + \cdots + e_N^2)$$

Choose the model form first, then find parameters that minimize the sum of squared errors.

4.6.3 Least Squares and Thrust Fitting: A Three-Point Line Fit

Fit a Least-Squares Line to Three Teaching Points by Hand

- Teaching points (u, F): (0,0.1), (0.5,2.0), and (1,4.2), with F in N. Fit $F = a + bu$; the means are $\overline{u} = 0.5$ and $\overline{F} = 2.1$.
- The formula gives slope 4.1 N and intercept 0.05 N. Predictions are 0.05, 2.10, and 4.15 N; residuals are 0.05, −0.10, and 0.05 N. The intercept may include sensor offset, rather than real zero-throttle thrust.

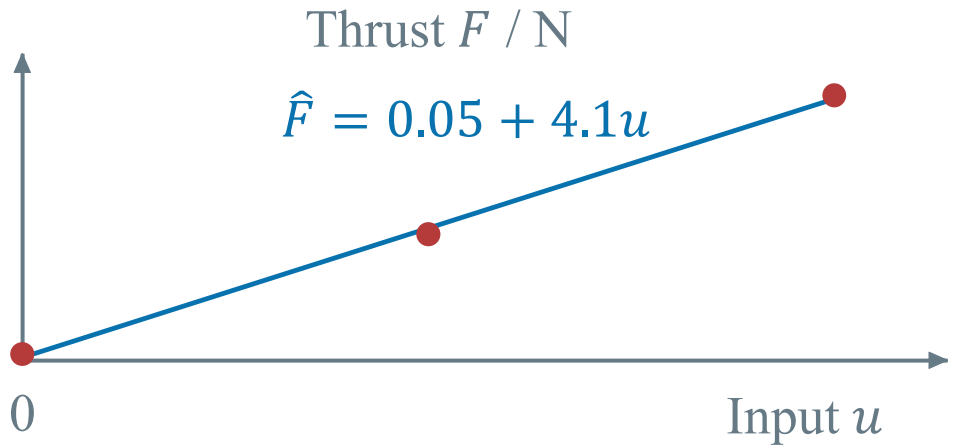


Figure: Three points, fitted line, and residuals

$$b = \frac{\sum_i (u_i - \overline{u})(F_i - \overline{F})}{\sum_i (u_i - \overline{u})^2} = \frac{2.05}{0.5} = 4.1, \quad a = \overline{F} - b\overline{u} = 0.05$$

The slope describes rate of change; the intercept describes zero offset.

4.6.3 Least Squares and Thrust Fitting: Construct the Regression Matrix

Stack Sample Equations into a Matrix to Estimate Parameters

- Stack $F_i = a + bu_i + e_i$ across samples to obtain $\mathbf{F} = X\boldsymbol{\theta} + \mathbf{e}$, with parameters $\boldsymbol{\theta} = [a, b]^T$. Rows represent samples; columns represent parameter contributions.
- Substitute the three points to form $X^{\mathrm{T}}X$ and $X^{\mathrm{T}}F$ below. The solution $a = 0.05$ and $b = 4.1$ matches the three-point fit. With constant input, slope and intercept cannot be separated. Use a linear solver; explicit inversion is unnecessary.

$$X = \begin{bmatrix} 1 & 0 \\ 1 & 0.5 \\ 1 & 1 \end{bmatrix},\ F = \begin{bmatrix} 0.1 \\ 2.0 \\ 4.2 \end{bmatrix},\ X^{\mathrm{T}}X = \begin{bmatrix} 3 & 1.5 \\ 1.5 & 1.25 \end{bmatrix},\ X^{\mathrm{T}}F = \begin{bmatrix} 6.3 \\ 5.2 \end{bmatrix}$$

Each regression-matrix column describes how one estimated parameter acts.

4.6.3 Least Squares and Thrust Fitting: Squared-Speed Thrust: Estimate the Coefficient

One Coefficient for the Squared-Speed Thrust Model

- Let $F = c_{\mathrm{T}}\omega^2$, written as $T = c_{\mathrm{T}}\varpi^2$ in textbook Eq. (4.29). Thrust is nonlinear in speed but linear in unknown c_{T}. Each sample supplies one regressor ω_i^2.
- Differentiate the squared-error sum and set it to zero. The formula below needs only two sums of data and can be calculated directly.
- This structure predicts zero thrust at zero speed. Handle and document sensor offset and low-speed aerodynamic behavior separately before fitting.

$$\hat{c}_{\mathrm{T}} = \frac{\sum_i \omega_i^2 F_i}{\sum_i \omega_i^4}, \quad [c_{\mathrm{T}}] = \mathrm{N/(rad/s)^2}$$

Squared speed as regressor makes thrust linear in its parameter.

4.6.3 Least Squares and Thrust Fitting: Static Thrust: Matrix Solution

Compare Two Static Thrust Models in Five Lines of Code

- **Data and method:** D is the 91-row steady-state table from segment averaging; the e6 CSV rpm and thrust_N columns match it. Backslash solves for coefficients of the matrix columns. Compare root-mean-square error (RMSE) on the same samples and rad/s units.

```
% D: 91-row table from the segment-averaging page
% The e6 CSV rpm and thrust_N columns match D
w = D.rpm*(2*pi/60); F = D.thrust_N;
kT = (w.^2) \ F;
Phi = [w.^2, w]; c = Phi \ F;
rmse1 = sqrt(mean((F-(w.^2)*kT).^2));
rmse2 = sqrt(mean((F-Phi*c).^2));
```

Data: 91 segment-averaged points
Speed: Standardized to rad/s

Solution: Code kT is c_T

Comparison: Both RMSE values are in N

4.6.3 Least Squares and Thrust Fitting: Squared-Speed Coefficient and Residuals

Refit the Squared-Speed Coefficient and Check Residuals

- From these 91 points: $c_T = 2.657299 \times 10^{-5}$ N/(rad/s)^2; training RMSE ≈0.335744 N.

  Samples: The same 91 steady-state points

  Plotted speed range: 84.2–812 rad/s

- If residuals show a trend, check model structure, zero offset, and voltage conditions before adding parameters.
- **Result:** Over 200–600 rad/s, measurements average about 0.4 N below the model; above 600 rad/s they become higher. Residuals vary systematically with speed.

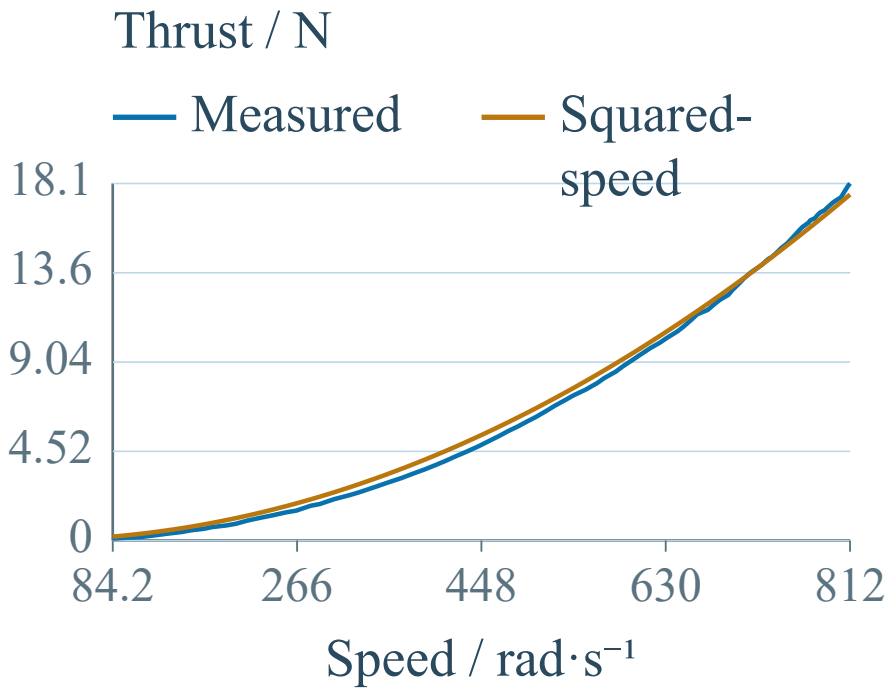


Figure: Recorded static points and squared-speed model

4.6.3 Least Squares and Thrust Fitting: Thrust with an Added Linear Term

Lower Error: Check Low Speed and Extrapolation

- Fit $F = c_2\omega^2 + c_1\omega$ to the same 91 points: training RMSE = 0.103866 N; $c_2 = 3.044977 \times 10^{-5}$ N/(rad/s)^2; $c_1 = -0.002744314$ N/(rad/s).

- **Result:** Zero crossing $\omega = -c_1/c_2 \approx 90.1$ rad/s. At the lowest 84.2 rad/s, prediction is −0.015 N; measurement is 0.086 N. Below about 90 rad/s, prediction is negative. Define deadband, saturation, and operating limits separately.

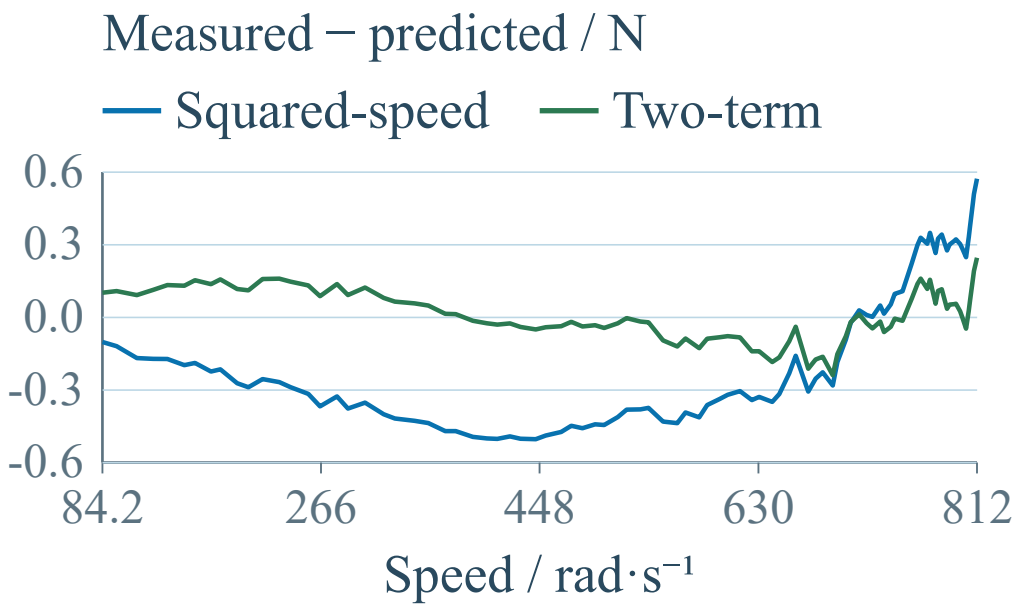


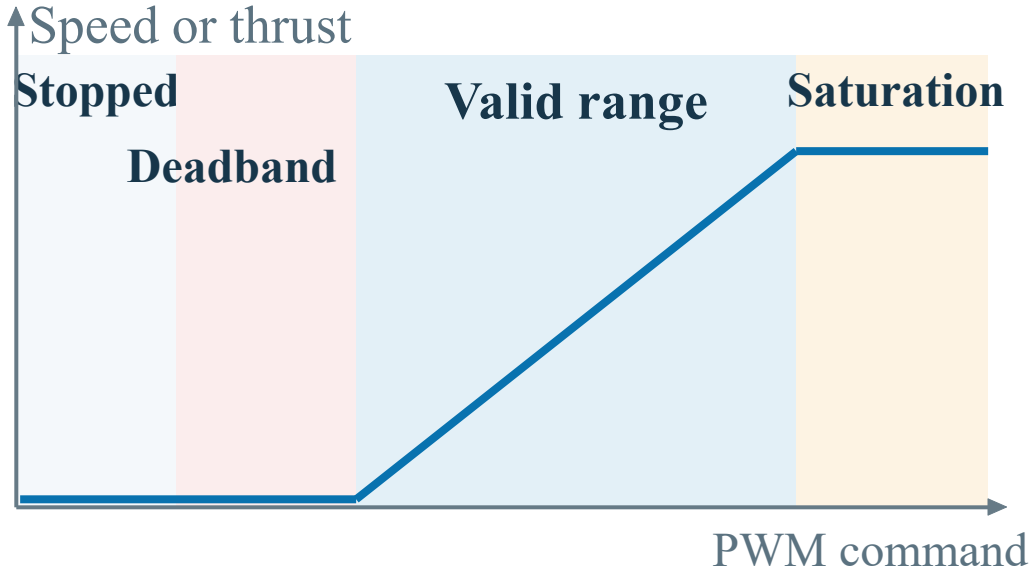
Figure: Two structures on the same recorded points

As complexity grows, check both error reduction and physical behavior.

4.6.3 Least Squares and Thrust Fitting: Deadband, Zero, and Saturation Limits

Define Zero, Range, and Saturation for Static Models

- Stopped, deadband, and saturation are distinct. In the deadband, a changed command may not start the motor; beyond saturation, more command no longer increases expected output.
- Shift and scale the regressor to reduce numerical-scaling problems from high PWM powers. Physical units remain unchanged.
- Compare rising/falling branches for return-sweep behavior. Record voltage, temperature, and hysteresis conditions; these may differ by branch.



Static regions (schematic, not measured)

Data supports the fitted curve within its range; check extrapolation separately.

Chapter Outline

Parameter Measurement and Static Model Identification

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

5 Units in This Section

4.6.1 Parameter Sources and Physical Measurement
Textbook: Secs. 4.2.2.1 and 4.2.2.4

4.6.2 Steady-State Data and Units
Textbook: Sec. 4.2.2.4

4.6.3 Least Squares and Thrust Fitting
Textbook: Eq. (4.29), Sec. 4.1.4.4; Sec. 4.2.2.4

4.6.4 Overall and Staged Static Models
Textbook: Sec. 4.2.2.4

4.6.5 Reproduction, Parameter Updates, and Validity
Textbook: Sec. 4.2.2.4

4.6.4 Overall and Staged Static Models: Overall and Staged Fitting

Where Does the Overall PWM-to-Thrust Model Fit?

- The overall model ("integrated model" in e11) fits PWM directly to thrust for a fixed motor, ESC, propeller, and supply. The staged model ("decomposed model" in e11) retains intermediate rotor speed to help locate errors.
- Errors in the first stage still propagate to thrust. A propeller change affects both rotor force and motor load; recheck both stages.

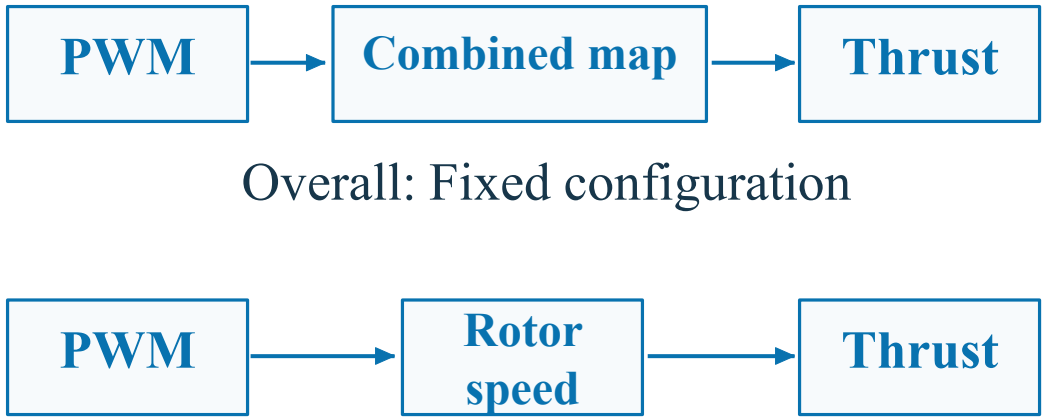


Figure: Fit the whole chain or retain intermediate rotor speed

Overall models use fewer interfaces; staged models retain component meaning.

4.6.4 Overall and Staged Static Models: Motor Static Model

Fit PWM to Steady-State Speed, Then Connect the Propeller

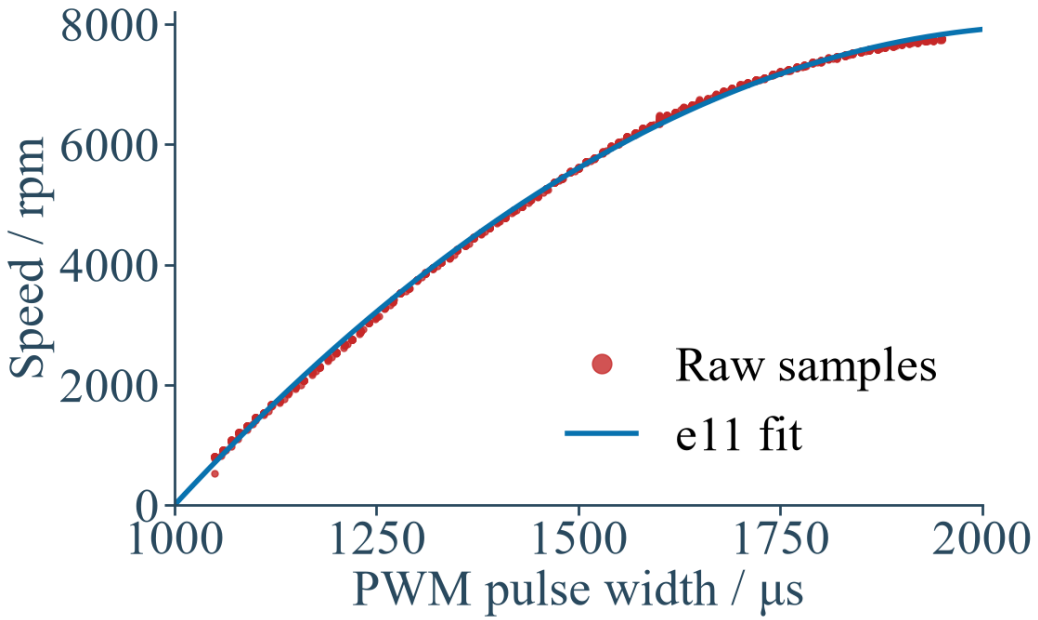


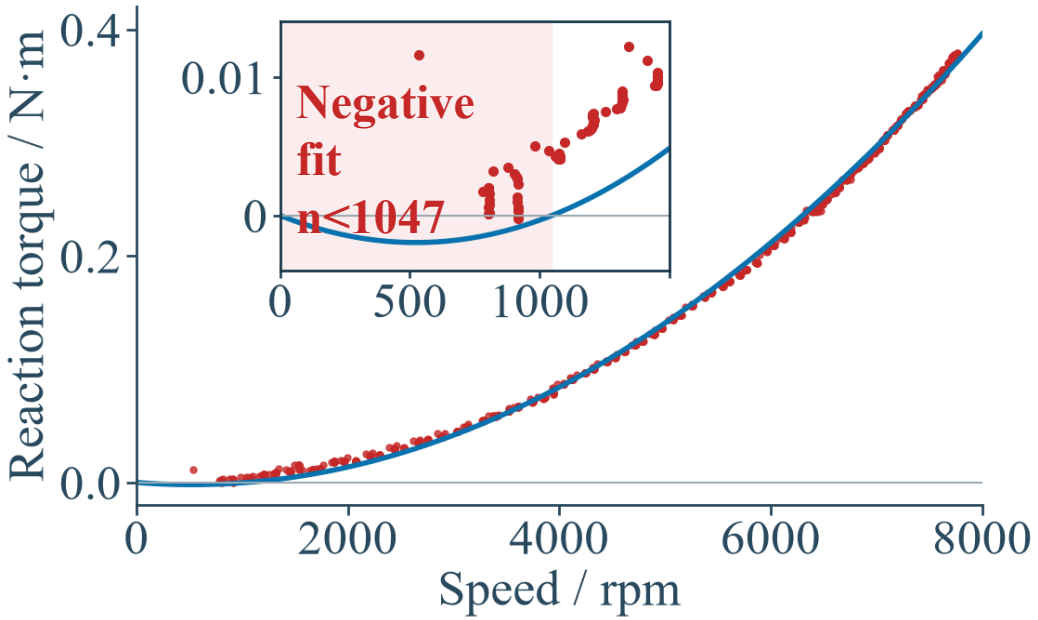
Figure: PWM–speed fit (offline redraw)

- **Read the plot:** Raw samples are red; the e11 fit is blue. Its decreasing slope is hard to represent with one line.
- **Structure:** $\Delta u = u - 1000$ μs; $n = a_1 \Delta u + a_2 \Delta u^2$; $a_1 = 14.50$ rpm/μs; $a_2 = -6.586 \times 10^{-3}$ rpm/μs².
- **Check:** At $u = 1500$ μs, $n \approx 5602$ rpm, about 0.1% below the level's mean of 5607.85 rpm. Zero output at 1000 μs is a structural constraint.

A staged model first determines target speed, making propulsion-chain errors easier to locate.

4.6.4 Overall and Staged Static Models: Propeller Static Model

Identify Reaction Torque Too, with Explicit Speed Units



Speed–reaction-torque fit (offline redraw)

- Thrust causes translation and moment-arm rotation. Identify reaction torque separately for direct yaw.
- **Fit:** e11 uses $Q = \alpha_2 n^2 + \alpha_1 n$, with n in rpm; $\alpha_2 = 7.120 \times 10^{-9}$ N·m/rpm²; $\alpha_1 = -7.457 \times 10^{-6}$ N·m/rpm.
- **Limit:** Below 1047 rpm the fit is negative, reaching about -0.002 N·m in the enlarged red region. Define stopped, valid, and saturated regions separately.

Thrust and reaction torque govern translation and rotation separately; coefficients must match the input units.

4.6.4 Overall and Staged Static Models: From rpm Fits to rad/s Code

When Input Units Change, Convert Every Coefficient

- **Original definition:** In e11's reaction-torque model $Q = \alpha_2 n^2 + \alpha_1 n$, n is in rpm. For a rad/s program input, first write $n = (60/2\pi)\omega$.
- **Substitute term by term:** In $Q = \beta_2 \omega^2 + \beta_1 \omega$, the quadratic term contains two conversion factors and the linear term one. Coefficient values change accordingly.
- **Equivalence check:** Factors $60/2\pi \approx 9.549$ and $(60/2\pi)^2 \approx 91.19$. At $n = 6000$ rpm ($\omega = 628.3$ rad/s), both forms give $Q \approx 0.2116$ N · m.

$$\beta_2 = \alpha_2 \left(\frac{60}{2\pi}\right)^2, \qquad \beta_1 = \alpha_1 \frac{60}{2\pi}$$

Convert coefficients according to each term's power to retain the same physical model after changing input units.

4.6.4 Overall and Staged Static Models: Whole-Curve and Component-Curve Tradeoffs

One Test-Stand Record Can Produce Two Model Deliverables

Aspect	Overall (u: PWM pulse width)	Staged (n: rotor speed)
Form	$F = f_F(u), Q = f_Q(u)$	$n = f_n(u)$, then $F = g_F(n)$ and $Q = g_Q(n)$
Interfaces	PWM input only; fewest interfaces. Rotor speed is diagnostic.	One extra intermediate variable: rotor speed. Measure it synchronously.
Error location	Internal errors are hard to assign to motor or propeller.	Locate each stage's error: command → speed; speed → load
Conditions	Fixed motor, ESC, propeller, and voltage combination. Report supply and range.	Study speed, propeller changes, or staged dynamics; report supply and range.
Risks	Retest the whole mapping after changing any component.	Error: $\delta F/F \approx 2\,\delta\omega/\omega$; first-stage speed +5% gives about +10% thrust.

Choose model decomposition according to the physical stages to explain or replace.

Chapter Outline

Parameter Measurement and Static Model Identification

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

5 Units in This Section

4.6.1 Parameter Sources and Physical Measurement
Textbook: Secs. 4.2.2.1 and 4.2.2.4

4.6.2 Steady-State Data and Units
Textbook: Sec. 4.2.2.4

4.6.3 Least Squares and Thrust Fitting
Textbook: Eq. (4.29), Sec. 4.1.4.4; Sec. 4.2.2.4

4.6.4 Overall and Staged Static Models
Textbook: Sec. 4.2.2.4

4.6.5 Reproduction, Parameter Updates, and Validity
Textbook: Sec. 4.2.2.4

4.6.5 Reproduction, Parameter Updates, and Validity: Reproduce Static Fitting

Reproduce the Recorded Static Fit from the Installation Folder

- **Folder:** [installation directory]/RflySimAPIs/4.RflySimModel/1.BasicExps/e6_ModelingLearningLab/03_Identification/
- **Steps:** Read its Readme.pdf, set MATLAB's current folder here, then run the commands below. Computation is offline; results go to outputs.

```
metrics = identification_beginner();
metrics.static_available
```

- **Checks:** static_available is true; static_steady_candidates.csv has 91 rows; model RMSE values are about 0.336 N and 0.104 N; $c_T \approx 2.657 \times 10^{-5}\ \mathrm{N/(rad/s)^2}$.
- If false, check that 2.AdvExps/e11_PowerSysIdModel/2.StaticThrIntID/ is installed, or pass the RflySimAPIs folder to identification_beginner(api_root).

A reproducible static-parameter table includes the data range, processing rules, and fitting error.

4.6.5 Reproduction, Parameter Updates, and Validity: Update Parameters and Retrim

Update Parameters, Then Recalculate Hover

- **Only for the 2 kg teaching vehicle (hypothetical):** replace nominal $c_T = 2 \times 10^{-5}$ (Sec. 4.3) with measured 2.657299×10^{-5} N/(rad/s)²; all other parameters stay fixed.
- **Retrim:** Keep $g = 9.8\,\mathrm{m/s^2}$; each rotor needs 4.9 N, requiring 429.416 rad/s. Old hover speed 494.975 rad/s no longer applies.
- **Validity:** 429.4 rad/s is within the 91-point range 84.2–812 rad/s: interpolation. Outside it, measure further. Keep the old model; compare at equal initial states, inputs, and step size.

$$\omega_0 = \sqrt{\frac{mg}{4c_T}} \approx 429.416\ \mathrm{rad/s}$$

Updating static coefficients changes the operating point. Recheck units, interfaces, and trim together.

Classroom Exercise

Refit the Second Sweep to Check Supply Effects

1 Task	Same file, second sweep: After stopping, command steps down 1950 → 1050 µs (MATLAB rows 1204–2342). Apply the same steady-state extraction rule.
2 Fit	Fit $F = c_T\omega^2$ and $F = c_2\omega^2 + c_1\omega$; calculate both RMSE values. Compare c_T and speed at 1500 µs with the first rising sweep.
3 Reference values	91 points: $c_T \approx 2.662 \times 10^{-5}$ N/(rad/s)² (rising sweep: 2.657×10^{-5}). RMSE values: 0.223 N and 0.099 N. At 1500 µs: $\omega \approx 549.9$ rad/s, 7.79 N, 22.76 V (rising: 587.3 rad/s, 8.80 N, 24.24 V).
4 Judgment	c_T differs by about 0.2% between sweeps: the speed-to-thrust relationship is stable. At the same PWM, speed falls with voltage. Record supply conditions for PWM-to-speed parameters; this demonstrates the value of a staged model.

Section Summary

Record Sources and Ranges; Convert by Input Powers

Method	Record each parameter's value, units, configuration, and method. Use the first rising sweep, averaging the latter half of intervals with ≥6 samples. Fit squared-speed and two-term thrust models; choose overall or staged form for the task. Convert coefficients by input powers; retrim after updates.
Key Judgments	$c_T = 2.657 \times 10^{-5}$ N/(rad/s)²; training RMSE: 0.336 N and 0.104 N. The two-term zero crossing is near 90 rad/s. Updated hover speed 429.4 rad/s is within 84.2–812 rad/s: interpolation.
Common Errors	Using half the bifilar string spacing gives a factor-of-four inertia error; omitting ×9.8 for FKG; failing to convert rpm coefficients by powers for rad/s; treating training RMSE as independent validation; using the negative-linear-term model below its zero crossing.