

Physics-Informed Regression of Aircraft Performance from Minimal Flight Data

Peter Sharpe* and R. John Hansman†
Massachusetts Institute of Technology, Cambridge, MA

I. Nomenclature

A	=	amplitude of oscillation
a	=	cylinder diameter
C_p	=	pressure coefficient
C_x	=	force coefficient in the x direction
C_y	=	force coefficient in the y direction
c	=	chord
dt	=	time step
F_x	=	X component of the resultant pressure force acting on the vehicle
F_y	=	Y component of the resultant pressure force acting on the vehicle
f, g	=	generic functions
h	=	height
i	=	time index during navigation
j	=	waypoint index
K	=	trailing-edge (TE) nondimensional angular deflection rate

II. Introduction

A primary goal of an initial flight test campaign of a new aircraft is to experimentally determine the aircraft's aerodynamic and propulsive performance characteristics. The desired outputs of this process typically include:

- 1) The aircraft's power curve, which gives the required power to maintain level flight as a function of airspeed.
- 2) The aircraft's aerodynamic polar, which gives the relationship between the aircraft's lift and drag coefficients.
- 3) The aircraft's propulsive efficiency curve, which yields the overall propulsive efficiency, typically as a function of throttle setting and/or propeller advance ratio, if relevant.

All three of these results represent sweeps through the aircraft's flight envelope, varying the aircraft's airspeed, pitch trim setting, and throttle setting, and measuring the resulting lift, drag, and power required.

A. Traditional Flight Measurement Methods and Limitations

Typically, these performance outputs are obtained by performing an extensive campaign of careful, controlled flight experiments at quasi-steady flight conditions.

For example, the aerodynamic polar is commonly measured by performing a series of long, steady power-off glides at different airspeeds. An implicit assumption here is that pitch trim is adjusted to maintain these airspeeds. Windmilling drag can also be estimated and calibrated out for improved accuracy, although this introduces additional uncertainty. The glide ratio (or equivalently, L/D) is then measured at a given airspeed by using a simple two-point finite-difference between the beginning and end of the glide:

*PhD Candidate, AIAA Student Member

†T. Wilson Professor in Aeronautics, AIAA Fellow

$$(L/D) = \frac{h(t_1) - h(t_2)}{V \cdot (t_2 - t_1)}$$

where $h(t)$ represents the altitude at time t , and V represents the airspeed. The drag is then computed as $D = W/(L/D)$, and C_L and C_D can be nondimensionalized from here.

Similarly, the power curve is often measured by flying in steady level flight for an extended period at a given airspeed, adjusting pitch trim as needed. The input power to the propulsion system P_{in} is measured at a given throttle setting. In a liquid-fueled airplane, this can be computing using the fuel flow rate at a given throttle setting (read off the panel via a fuel flow meter) and the specific energy of the fuel. For an electric airplane, it can be computed as the product of battery current and voltage. This procedure is then repeated for a variety of airspeeds, and the resulting power curve is plotted.

The propulsive efficiency can be roughly estimated based on these experiments as well. One possible procedure is as follows:

- 1) The input power to the propulsion system P_{in} is measured, as above.
- 2) We observe the steady-state climb/sink rate of the aircraft at this throttle setting and we compare this to the power-off sink rate. The difference between these two sink rates represents the air power done by the propulsion system. The air power can be computed as:

$$P_{air} = T \cdot V = mg \left(\frac{dh}{dt} \Big|_{\text{power on}} - \frac{dh}{dt} \Big|_{\text{power off}} \right)$$

- 3) Then, we can compute the propulsive efficiency as the ratio of the input power to the air power:

$$\eta_{\text{propulsive}} = \frac{P_{air}}{P_{in}}$$

Traditional methods for measuring aerodynamic and propulsive performance have several limitations. By far the biggest limitation is that they require vast amounts of flight time to obtain a sufficient number of data points to characterize the aircraft's performance. This is both expensive and time-consuming.

Furthermore, the aircraft must be flown at precisely-controlled conditions. This can be frustrating and tedious for the pilot, and it can also be dangerous if the aircraft is flown for an extended duration at conditions that are close to the edge of the flight envelope (e.g., behind the power curve on a not-yet-characterized experimental airplane).

Traditional methods also make no direct estimate of sensor noise (and hence, uncertainty) - data is collected and averaged until the experimenter is satisfied that the data is "good enough". This is a subjective process, and it is difficult to quantify the uncertainty in the resulting performance estimates.

Finally, traditional methods are not well-suited to estimating the performance of an aircraft that is not in steady flight. For example, data recorded during an aircraft's climb or descent to flight test altitude is typically discarded, which is wasteful; ideally, every single second of data should contribute to refining our estimate of the aircraft's performance.

B. Inference-Based Flight Data Reconstruction Methods

By contrast, we propose a method of estimating these performance characteristics which relies on a different perspective about flight testing. Traditionally, flight testing is viewed as a process of pure measurement, based only on analyzing the data itself. Instead, we view the aircraft flight test as a data-generating process, from which we can infer performance relationships. This might sound like a small difference, but it allows us to recognize that we actually have vastly more information about the aircraft than just the data itself, which in turn allows us to perform much more accurate performance reconstruction with a limited amount of data:

Firstly, with certain reasonable assumptions about the sensor noise characteristics, the data itself can reveal a surprisingly large amount of information about the sensor noise. Essentially, this involves using the data to estimate the "trustworthiness" of the data itself, a process described more fully in Section ???. One contribution of this work is a new cross-validation-like approach for estimating the sensor noise using only the properties of the data itself. This allows cleaner raw data inputs to the subsequent performance analysis toolchain that we develop, ultimately resulting in more accurate performance estimates and enabling uncertainty quantification.

We also know certain characteristics of the aircraft's state based on physics, which can further increase the effective information density and significantly improve state estimation. For example, we know that the true state of the aircraft

must follow invariants from Newtonian physics (e.g., conservation of energy). Embedding these invariants from physics reduces the number of unknowns, enabling more robust correction for unsteady effects.

Finally, there are characteristics about performance relations that we can be embedded from domain-specific expertise. Even for models that do not have simple, closed-form invariants, we almost always have some priors (i.e., informed guesses) about how performance parameters should be related - model structure. For example, we know that the aircraft's drag coefficient will primarily be a function of lift coefficient, and the general shape of this function is known. Similarly, we know that the propulsive efficiency will largely be a function of advance ratio, and the advance ratio can be inferred from the airspeed, input power, and first-order physics. Incorporation of these physics-based performance models embeds our physical understanding of the aircraft system into the estimation process, improving accuracy. For this reason, we term the performance reconstruction methodology described in this work a “physics-informed” approach.

Figure 1 broadly illustrates the toolchain for performance reconstruction that represents this work’s central contribution.

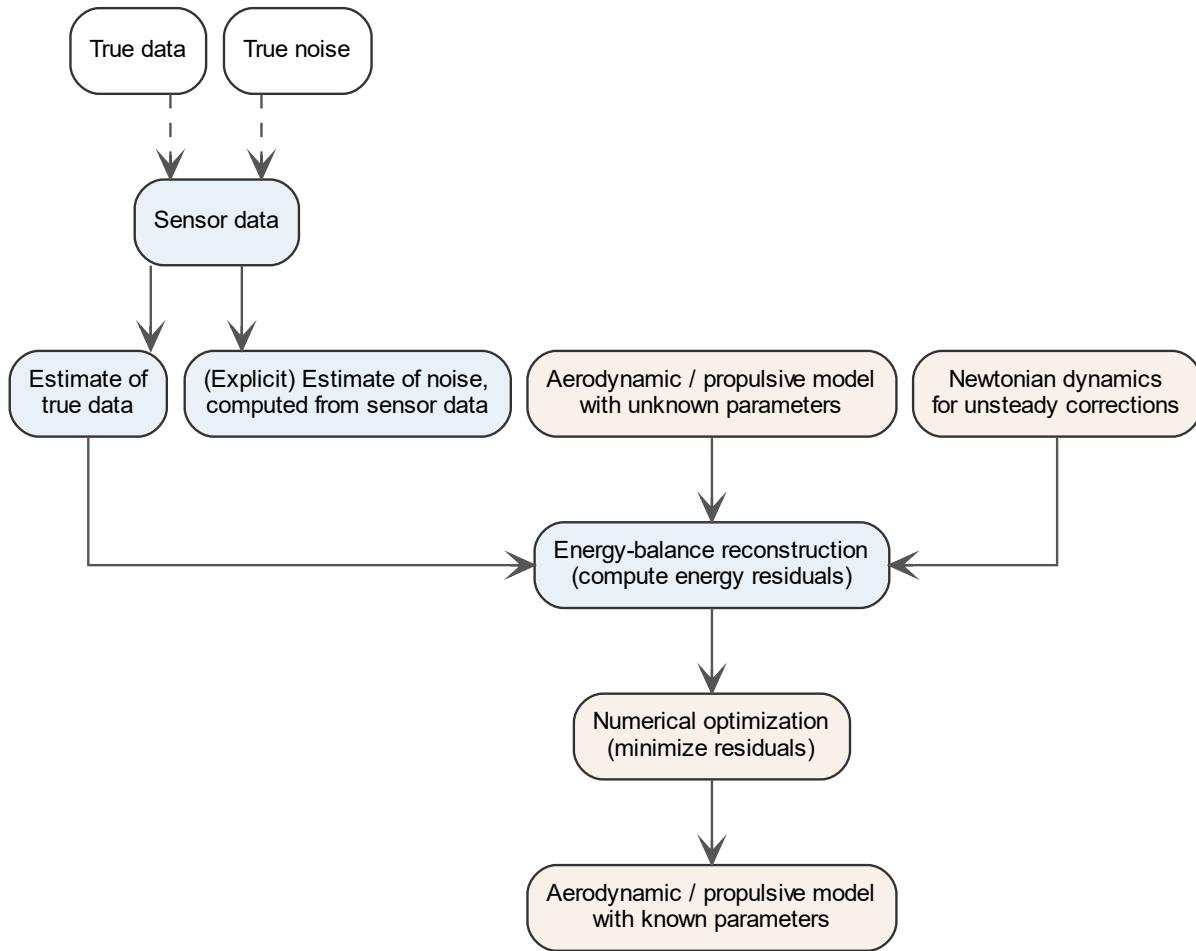


Fig. 1 Process for inference-based performance reconstruction from flight data

III. A Minimal Flight Test Dataset

To illustrate our proposed flight test data analysis procedure, this manuscript will use example data from a flight test conducted during MIT 16.821: Flight Vehicle Development, a senior-level aircraft design/build/fly capstone course at MIT AeroAstro. The flown aircraft, named *Solar Surfer*, is a remote-controlled solar-electric seaplane design with a

14-foot wingspan. The as-flown all-up mass is 9.4 kg.

General specifications of the aircraft can be found in the early design drawing that is reproduced in Appendix ???. While this drawing does not reflect as-built weights or various planform adjustments that were made during preliminary design and construction, the overall configuration and performance numbers are representative of the aircraft that was flown.

A. Flight Test Campaign

The aircraft was flown in the vicinity of the Charles river basin near Cambridge, MA. Five tests were conducted on the morning of May 3, 2023, of which two were airborne flight tests. The final airborne flight test lasted approximately 260 seconds (4.3 minutes), beginning and ending with a successful water takeoff and landing. Figure 2a depicts a still image of the aircraft in flight. A simple racetrack-like pattern was flown, as shown in Figure 2b. Winds were calm at roughly 1.5 m/s from the south.



(a) *Solar Surfer* in flight



(b) Recorded GPS ground track during test flight

Fig. 2 *Solar Surfer* aircraft test flight

Figure 3 illustrates the subset of the raw sensor data during flight that will be used for performance reconstruction. Overall, the data is reasonable, and we can clearly identify takeoff and landing by the strong increase in barometric altimeter noise near $t = 0$ sec and $t = 263$ sec. We can also identify several distinct flight phases, such as a high-speed pass near $t = 150$ sec and gliding periods near $t = 40$ sec, $t = 210$ sec, and $t = 250$ sec.

However, there are two primary problems with the data we have. Firstly, all the data sources, and in particular the airspeed sensor, yield noisy measurements. The unsteady corrections that we will later implement require taking the derivative of this data; this is a problem, because taking the derivative of noisy data tends to amplify the noise further.

Secondly, the total amount of data is extremely limited, both in duration and sample rate. For example, there are only 1,200 data points *in total* for each of the plotted sensor traces, and this is before eliminating noise. After eliminating noise (which effectively acts as a low-pass filter on our data, as noise tends to be relatively high-frequency), the effective amount of information we have from the sensors is extremely low. While 1,200 data points might be sufficient to characterize the aircraft flight performance at a single operating point (i.e., a single power setting and airspeed), it is typically a vastly insufficient quantity of data to characterize any substantial portion of the flight envelope.

Consider that even the most sophisticated techniques for system identification via statistical inference require the system to be observable in some form - no amount of math can recover a signal that is simply not present in the data. This is difficult, because after noise removal, the data really no more than perhaps a half-dozen “system excitations” from which we can infer aircraft performance. As a more intuitive analogy, consider that even the world’s best test pilot can’t accurately characterize the handling qualities of the aircraft without observing at least a few flight maneuvers.

For these reasons, this dataset makes a good example case for what kinds of flight data reconstruction are possible with very limited, noisy data using statistical inference techniques and corrections from unsteady flight physics.

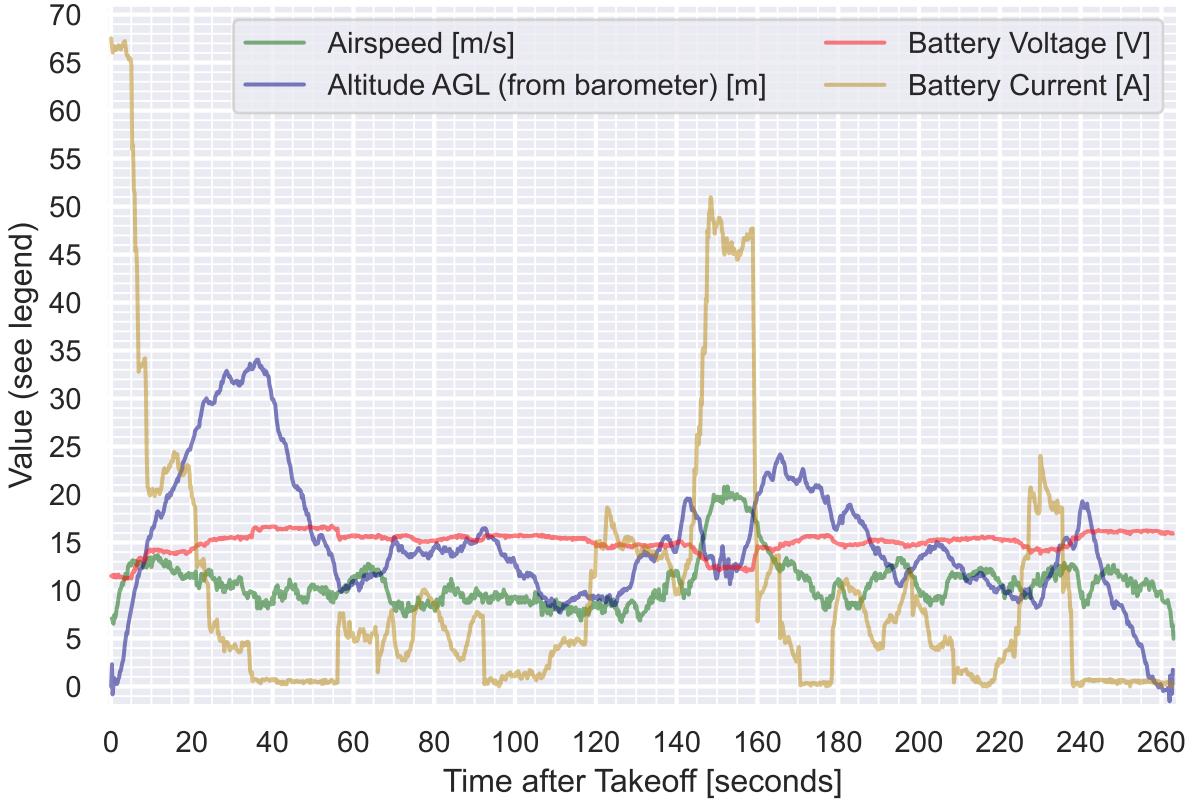


Fig. 3 Raw sensor data from test flight

B. Naive Estimation of Power Curve

The most straightforward approach to estimating the aircraft's power curve is to simply cross-plot the airspeed and electrical power throughout the flight, and then to fit a curve to it.

Figure 4 quickly illustrates the infeasibility of this approach. While a general trend is somewhat apparent (higher airspeeds yield higher power draw), the uncorrected data has far too much unexplained variation and noise to make any believable conclusions about the shape of the power curve.

Figure 4 also demonstrates the pitfalls of naive curve-fitting in the absence of any kind of physics-based model structure. Perhaps the most severe of these is the extrapolation to physically-impossible conclusions. For example, the naive linear fit shown in blue would imply that the power draw at zero airspeed is negative, which is clearly impossible. The naive cubic fit shown in red initially appears more reasonable, although it too makes erroneous conclusions: it implies that beyond 20 m/s, the power draw decreases with increasing airspeed. Clearly, our intuitions about what the “correct” model would look like are not being incorporated into the curve-fitting process, and hence we are leaving information on the table.

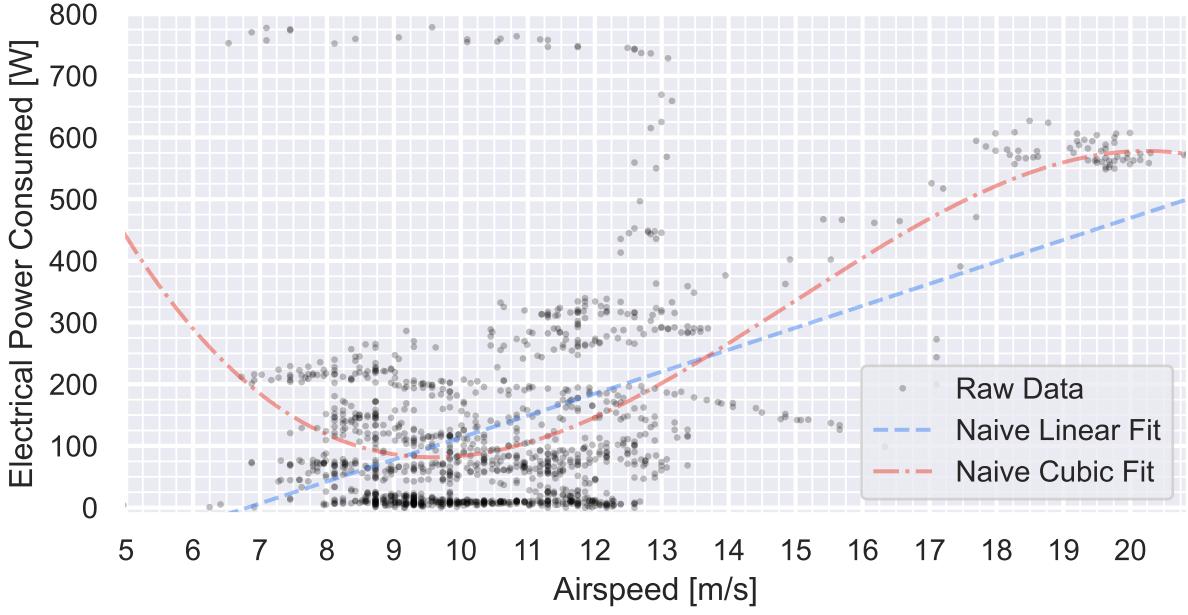


Fig. 4 Raw uncorrected data for the power curve, with naive curve fitting. Uncorrected data exhibits high amounts of unexplained variance. Naive curve fits extrapolate to physically-impossible conclusions and do not capture uncertainty.

IV. Sensor Data Pre-Processing

The first contribution here is to develop a means of pre-processing the sensor data. Traditionally, noise is removed by simple averaging; after this process is complete, any uncertainty in the data is typically not considered, and the average is regarded as the ground truth. This works fine for traditional steady analysis where there is a wealth of data that we can average over.

However, in cases where data is limited, we're strongly encouraged to find a way to extract some amount of information from unsteady data. This is difficult, because it forces us to directly deal with the sensor noise (and embed priors about the bias-variance tradeoff of data), rather than hand-waving it away by averaging.

As an illustrative example, imagine that we wish to reconstruct a *truth* estimate of the airspeed from the example dataset. This truth value is not directly observable, although data from an associated airspeed sensor is. However, this sensor data has noise; hence, any attempt to reconstruct the truth value must adopt a strategy to separating the noise from the data.

However, there are (literally) an infinite number of possible strategies that one can use to do this noise removal, depending on our embedded assumptions about how sensor noise is entering our data-generating process. For example, consider three different possible state estimations of the vehicle's airspeed, as illustrated in Figure 5.

One Dataset, Many Possible Interpretations

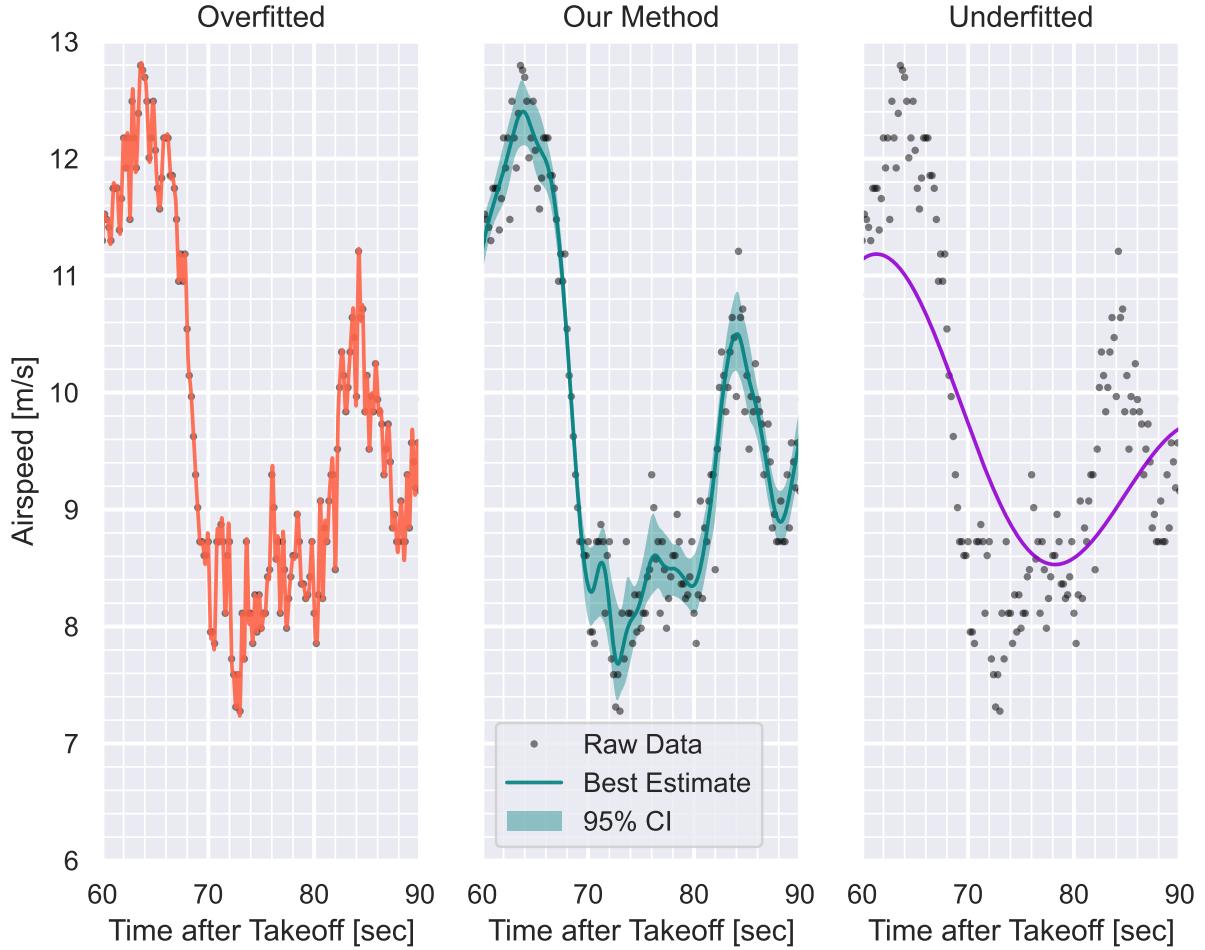


Fig. 5 Three possible curves that could be used to estimate the true underlying state of the vehicle. Airspeed data, zoomed to $t \in (60, 90)$; a subset of Figure 3. 95% confidence interval (CI) of reconstruction shaded in middle plot.

Ultimately, all three charts in Figure 5 represent different interpretations of the ground truth based on the data. If we didn't have any more information about the problem, all three could be equally justifiable given the appropriate context - but, in fact, we do have more information, since we know this data originates from a physical system.

The chart on the left of Figure 5 is clearly overfitted (i.e., tracks the data too closely) based on engineering intuition. Essentially, we are assuming that each sample is the true data, and that each sample has no noise; hence, the "truth" curve exhibits extremely high variance (strong wiggles). There's no *mathematical* reason why this interpretation can't be correct (it's theoretically possible that the sensor is giving perfect information), but of course it is *physically* implausible for our system. If this interpretation were true, it would imply that the vehicle is achieving velocity-vector-aligned-accelerations on the order of 2.4 G, which is not physically plausible given the onboard propulsion system.

The chart on the right of Figure 5 is clearly underfitted (i.e., tracks the data too slowly), as there are large regions where the estimator has a consistent bias (e.g., consistently too high or low) with respect to the underlying data. It might seem like there are no contexts where such an interpretation would be reasonable, but that is not necessarily the case. For example, if the sensor noise at each sample was not independent, but rather correlated with noise from the previous samples, we might not be able to rule out this interpretation of reality.

Of course, a quick glance at the three charts above would tell us “the middle one looks about right” - but why? What is it about the middle chart that makes it look more reasonable than the other two?

A. Optimal Sensor Data Reconstruction using Data-Driven Noise Estimates

The key difference is that the middle chart embeds certain assumptions that we intuitively know about our data into its reconstruction. Specifically, the middle chart is an optimal reconstruction of the data assuming that:

- The noise in each sample is independent (i.e., uncorrelated with previous samples) and normally-distributed
- The noise is unbiased (i.e., there are no systematic errors in the data, only random ones)
- The noise is homoscedastic (i.e., the standard deviation of the noise is constant across the entire dataset)
- The sample rate is significantly higher than the underlying dynamics of the system that we’re aiming to recover (i.e., the system is “slow” relative to the sample rate).

Under these assumptions above, the probabilistic properties of the noise reduce to a single parameter: the variance of the noise σ_n^2 . Thus, by constructing an estimator for this statistic, we can recover the probability distribution of the noise and hence an optimal estimator of the data.

1. Motivation for Finding the Variance of the Noise

Because we’re just one parameter away from having a probabilistic model of the sensor noise, we’re strongly motivated to estimate this noise variance. Once we estimate the variance, we can begin to make optimal choices about the bias-variance tradeoff *with a rigorous definition of what optimal means* here.

There is a second factor that motivates us to estimate the variance of the noise. Computing an optimal smoothing spline for a time-series dataset is a well-studied problem, but this process is greatly aided if we know the variance of the noise [1]. So, if we can obtain an estimate of noise variance, we’ve done most of the work required to compute an optimal data reconstructor.

In an ideal world, we would estimate this variance of the noise by taking a large number of samples while the vehicle is at some fixed, steady, known condition. For example, to estimate the airspeed data noise, we might place the aircraft in a wind tunnel at some constant speed. If we ran that experiment for an extended duration, we would know that the true underlying data was constant, and hence any observed variance in the samples would be due to the variance of the sensor noise.

If we want to reconstruct data from unsteady measurement, however, we need to find a way to estimate the variance of the noise using the unsteady data itself, which is much more tricky.

2. Initial Approach and a First-Order Data-Based Noise Estimator

One way to do this is by assuming that the data is “slow” relative to the sample rate. If this is true, then subsequent samples will effectively have the same underlying truth value, but with noise drawn independently from the same underlying distribution.

Quantitatively, if our true data is $x(t)$, and the noise of the sample $n(t)$ is drawn from $\mathcal{N}(0, \sigma_n^2)$, then our observed sensor data $s(t)$ will be:

$$s(t_1) = x(t_1) + n(t_1)$$

$$s(t_2) = x(t_2) + n(t_2)$$

If $t_1 \approx t_2$ (as would be the case if we’re looking at subsequent samples), then $x(t_1) \approx x(t_2)$, and hence:

$$s(t_2) - s(t_1) \approx n(t_2) - n(t_1)$$

In other words, the difference of two subsequent observed samples will be approximately equal to the difference of two independent draws from the same noise distribution. This means that we can estimate the variance of the noise by looking at the variance of the differences between subsequent samples.

From the properties of the difference of two independent random normal variables, we can then say:

$$n(t_2) - n(t_1) \sim \mathcal{N}(0, 2\sigma_n^2)$$

where σ^2 is the variance of the sensor noise. So, the following is then approximately true:

$$s(t_2) - s(t_1) \sim \mathcal{N}(0, 2\sigma_n^2)$$

Therefore, by taking the mean difference across all adjacent pairs of sample data, we can reconstruct the variance of the noise as:

$$\sigma_n^2 = \frac{1}{2 \cdot (N-1)} \sum_{i=1}^{N-1} (s(t_{i+1}) - s(t_i))^2 \quad (1)$$

where N is the number of samples in the dataset.

3. Numerical Demonstration of the First-Order Noise Estimator

We can demonstrate that this works in practice and is convergent to the correct answer by constructing a synthetic dataset with known noise properties, and then reconstructing the noise variance using the method above. The synthetic dataset we use is a simple sinusoid at some frequency f_{signal} with added independent, normally-distributed noise. We then sample this sinusoid at some frequency f_{sample} (ideally with $f_{\text{sample}} \gg f_{\text{signal}}$). We attempt to recover the noise variance using the method above and compare it to the true noise variance. This process is described in Figure 6, with results for the first-order estimator in the respective column of Table 1. The higher-order estimators are described in the following sections.

Table 1 Numerical demonstration of noise variance estimator performance

f_{signal}	f_{sample}	True σ_n	Estimated σ_n		
			1st-order Estimator	2nd-order Estimator	4th-order Estimator
1	1000	0.1	0.0999	0.1006	0.1008
10	1000	0.1	0.1048	0.1006	0.1008
100	1000	0.1	0.3236	0.1491	0.1016

As shown in Table 1, the first-order noise estimator yields a σ_n value that is very close to the true value. We will also show later that this estimator is convergent to the true value as the number of samples increases.

However, this estimator is only first-order convergent with respect to the ratio between the sampling frequency f_{sample} and the underlying dynamics of the system f_{signal} . In other words, if the system is “fast” relative to the sample rate, then the estimator will be inaccurate. This is demonstrated in the third row of Table 1, where the estimator is off by a factor of 3. This is because the sinusoid is sampled at a frequency that is only 10 times higher than the frequency of the sinusoid itself. In this case, the estimator is not accurate, and we need to use a more sophisticated estimator.

4. Second-Order Noise Estimator

To do a better job of estimating the noise variance even in cases where $f_{\text{sample}} \gg f_{\text{signal}}$ doesn’t strictly hold, we can use a second-order estimator. With this extension, we use a three-point numerical stencil.

Derivation of this estimator follows similar principles to the first-order estimator above, with the key result as follows:

$$\sigma^2 = \frac{1}{6 \cdot (N-2)} \sum_{i=1}^{N-2} (s(t_{i+2}) - 2 \cdot s(t_{i+1}) + s(t_i))^2 \quad (2)$$

where N is the number of samples in the dataset.

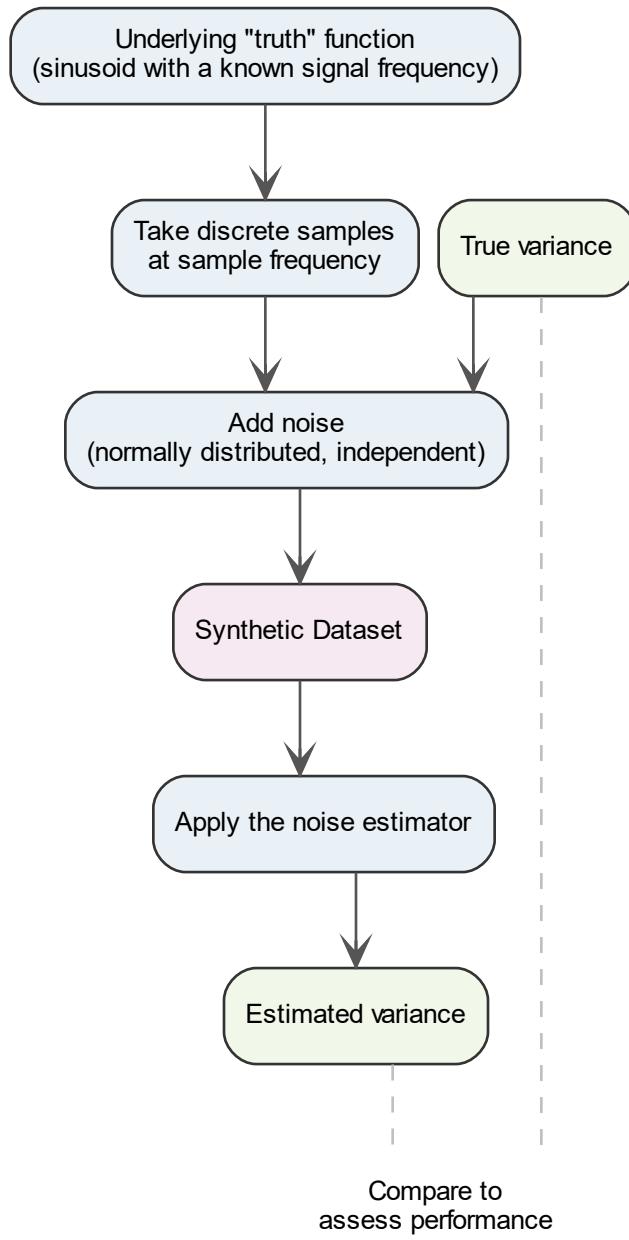


Fig. 6 Procedure for numerical demonstration of estimator performance

As shown in Table 1, the second-order estimator has superior performance to the first-order estimator when the sample and signal frequencies are not well-separated.

Notably, this effectively implements a first-order, second-degree finite-difference of the underlying data - a discrete derivative. It's instructive to consider the theoretical underpinning of using a discrete derivative operator here. If we assume that the underlying true signal is relatively low-frequency (i.e., smooth), then as we take more discrete derivatives of the observed data, the signal component will asymptote to zero (assuming $f_{\text{sample}} \geq f_{\text{signal}}$). Stated equivalently, the frequency spectrum of the true signal is assumed to have some cutoff frequency (analogous to f_{signal}), above which the power spectral density gradually goes to zero. We don't need to specify this cutoff frequency or know it precisely *a priori*; the remarks presented here merely require that one exists and is somewhere below f_{sample} .

In contrast, the noise is assumed to be independent, which means that the noise component of the observed data will not go to zero as we take successive derivatives. Stated equivalently, the frequency spectrum of the noise is white, with uniform spectral power across all frequencies. Therefore, repeated application of the discrete derivative operator acts as a way to spectrally-separate the noise from the signal. This is a key insight that we will use later in the paper.

5. Arbitrary-Order Estimators

Clearly, the second-order estimator improves performance compared to the first-order one. Interestingly, we can generalize the logic to an d -th order estimator by observing that the denominator is the sum of the squares of the first-order, d -th-degree, uniform-grid finite difference coefficients (or, perhaps more intuitively, the elements of the d -th row of Pascal's triangle). Then, we use the combinatorial trick that:

$$\binom{d}{0}^2 + \binom{d}{1}^2 + \cdots + \binom{d}{d}^2 = \binom{2d}{d} = \frac{(2d)!}{(d!)^2} \quad (3)$$

to derive the following d -th order estimator of the noise variance:

$$\sigma^2 = \frac{1}{\binom{2d}{d} \cdot (N - d)} \cdot \sum_{i=1}^{N-d} \left[\binom{d}{0} s(t_i) - \binom{d}{1} s(t_{i+1}) + \binom{d}{2} s(t_{i+2}) - \binom{d}{3} s(t_{i+3}) \pm \cdots \mp \binom{d}{d} s(t_{i+d}) \right]^2 \quad (4)$$

While we have shown that the second-order estimator has improved robustness to low sample rates, it is not immediately clear that this will hold for arbitrary orders - as the order increases, we might expect to see some form of numerical instability as we take successively higher-order derivatives of noisy data. To test this, we can plot the performance of noise estimators of various orders, as shown in Figure 7.

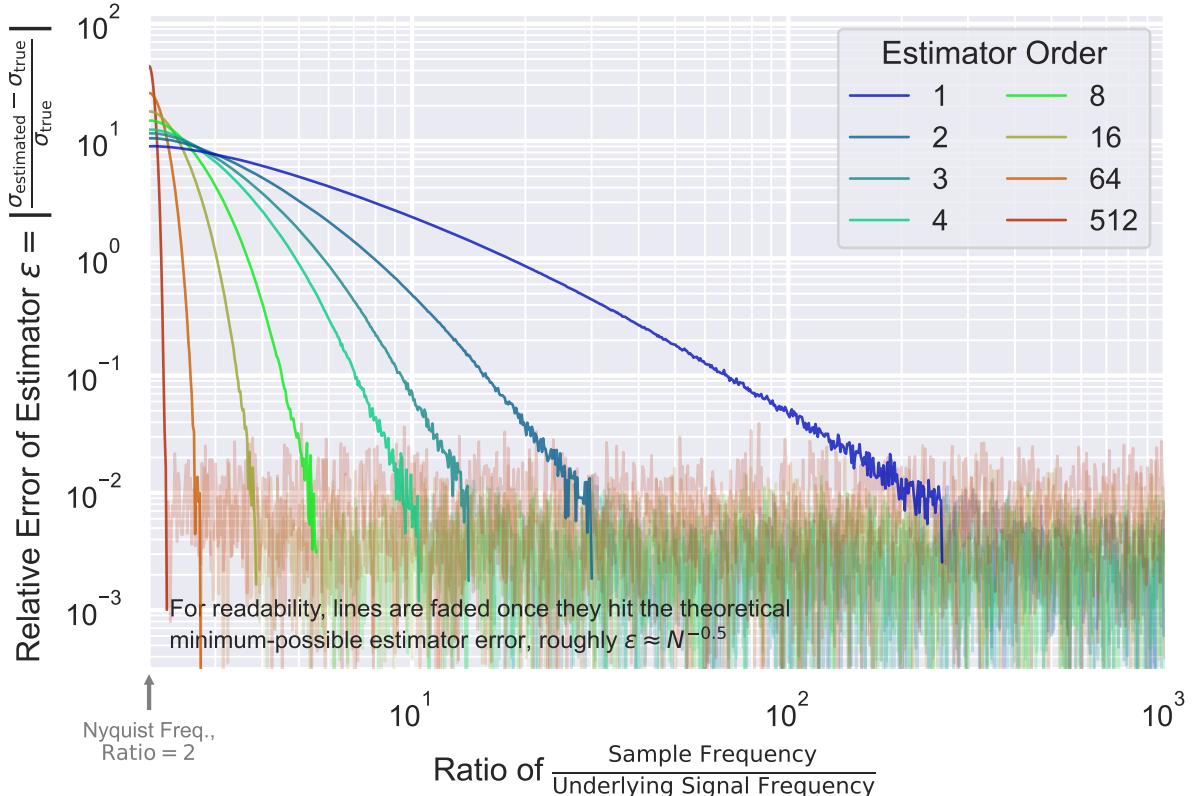


Fig. 7 Performance of higher-order data-driven noise estimators

The figure above shows something quite remarkable: we can use extremely-high-order estimators (even up to the 512th order, i.e., taking the 512th-derivative of raw, noisy data via finite differences), and we see not only that it's numerically stable, but also that it accurately estimates the true noise variance even when the underlying signal is almost at the Nyquist frequency.

Interestingly, except for a minuscule region near the Nyquist frequency, estimator performance seems to monotonically improve with increasing estimator order. Since practical data reconstruction would likely have the $f_{\text{sample}}/f_{\text{signal}}$ ratio be much larger than 2, this suggests that we can use extremely-high-order estimators to estimate the noise variance with very high accuracy.

B. Uncertainty Quantification Using Noise Variance Estimators

With the ability to estimate the amount of noise in the data, we can also estimate the uncertainty of our resulting models and curve fits. To do this, we combine a resampling bootstrap approach (described in [2]) with a noise-variance-aware spline interpolator (described in [1]). Together, these yield not only a best-estimate of the underlying model, but also a measure of the uncertainty in that estimate. An example is depicted in Figure 8, which uses the same power-curve dataset as in Figure 4.

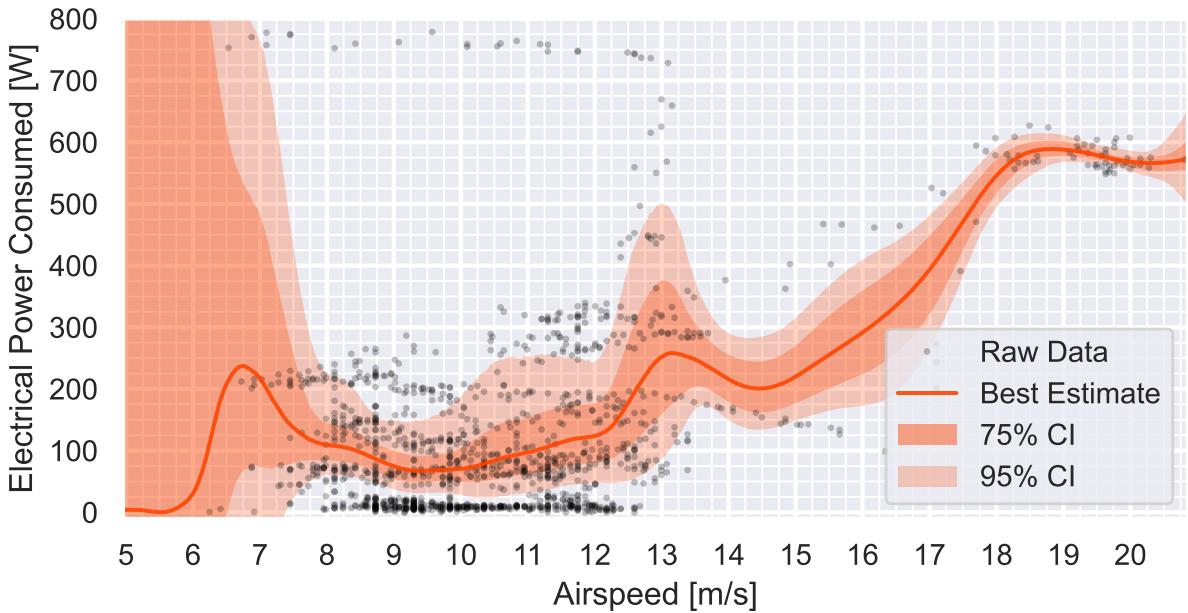


Fig. 8 Raw uncorrected data for the power curve, with noise-variance-aware spline interpolation. A resampling bootstrap allows uncertainty quantification.

Figure 8 represents an improvement over the naive curve fits shown in Figure 4 in that it leverages uncertainty. For example, the uncertainty narrows near 19 m/s where data is abundant, and it widens considerably below 7 m/s where data is sparse. This awareness of model uncertainty reduces the propensity of making statistical conclusions that go beyond what the data can support.

However, two major problems remain:

- 1) The resulting model uncertainty is extremely high, as shown by the shaded confidence intervals in Figure 8. For example, the tightest that we can bound the required power at 10 m/s with reasonably-strong confidence (95% confidence interval) is between 30 and 190 Watts. This is a huge range and not a particularly useful insight. Near the low end of the airspeed range, the model uncertainty becomes so large that even the most basic claims are impossible to assert.
- 2) The model still allows physically-implausible relationships, such as the unexplained power curve increase near 13 m/s.

V. Physics-Based Corrections

A. The Aircraft as an Energy System

VI. Conclusion

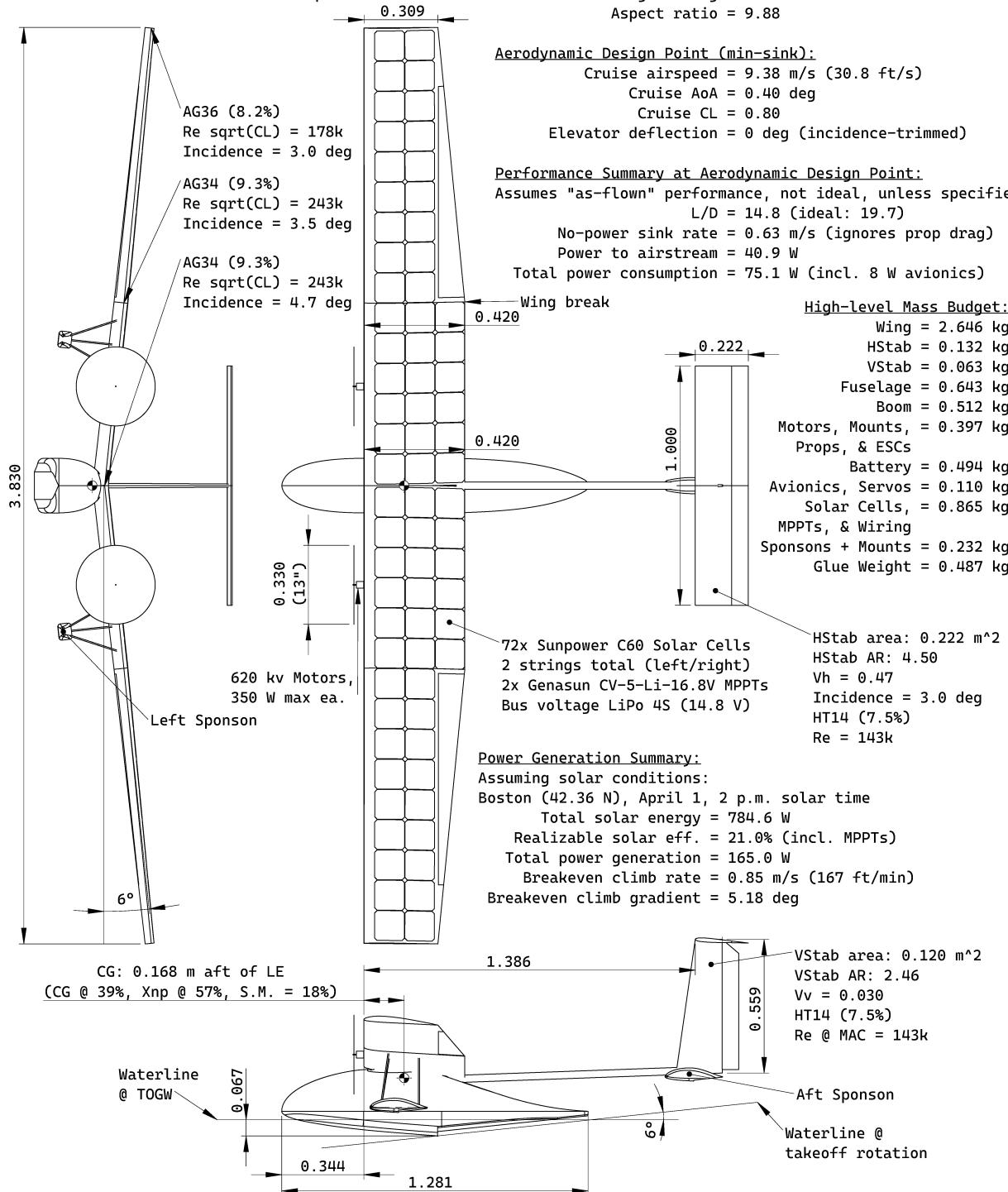
A fair point of hesitancy here might be the fact that applying these corrections to recover useful insights from noisy, limited, unsteady data takes a significant amount of algorithmic development and mathematical complexity. However, algorithms are cheap, and more importantly, *scalable* - so, once a computational workflow is established, it can easily be applied to a large number of flight test campaigns. By contrast, flight-test hours and improved instrumentation represent a recurring high cost. Therefore, it is worth investing in the development of algorithms that can extract the maximum amount of useful insight from the data that we do collect, even if this analysis is less simple.

Appendix

A. Solar Surfer Design Drawing

Seaway-Mini, Initial Point of Departure for 16.821
Peter Sharpe 2/4/2023

1:20 Scale. Units in meters unless specified.



B. Efficient Computational Implementation of Arbitrary-Order Noise Estimator

While the given equation for an arbitrary-order data-driven noise estimator (Equation ??) is mathematically correct, it is nontrivial to computationally implement. This is because an implementation as-written involves combinatorial coefficients that grow exponentially with the order of the estimator, quickly exceeding standard double-precision floating-point overflow. A naive implementation also has a runtime complexity that scales linearly with both the number of samples in the dataset and the order of the estimator, making it computationally unattractive for large datasets or high-order estimators.

An efficient computational implementation of the arbitrary-order noise estimator is given below. This example uses NumPy and SciPy within Python 3. This implementation uses the log-gamma function to both nullify the complexity scaling with respect to estimator order as well as prevent numerical overflow. The implementation also uses a convolutional kernel to vectorize the summation, enabling further speedups.

```
1 import numpy as np
2 from scipy.special import gammaln
3
4 def estimate_noise_standard_deviation(
5     data: np.ndarray,
6     estimator_order: int = 10,
7 ) -> float:
8     """ For speed, pre-compute the log-factorial of integers from 1 to estimator_order
9     ln_factorial = lambda x: gammaln(x + 1)
10    ln_f = ln_factorial(np.arange(estimator_order + 1))
11
12    Create a convolutional kernel to vectorize the summation
13    coefficients = np.exp(
14        2 * ln_f[estimator_order] - ln_f - ln_f[::-1] - 0.5 * ln_factorial(2 * estimator_order)
15    ) * (-1) ** np.arange(estimator_order + 1)
16    coefficients -= np.mean(coefficients) # Remove any bias introduced by floating-point error
17
18    sample_stdev = np.convolve(data, coefficients[::-1], 'valid')
19    return np.mean(sample_stdev ** 2) ** 0.5
```

Listing 1 Example efficient implementation of the arbitrary-order noise estimator using NumPy/SciPy in Python 3.

Acknowledgments

The authors would like to thank the MIT AeroAstro 16.821 course staff for their support of this work, and the MIT AeroAstro department for their support of the course. The aircraft used to collect the example data in this work was constructed by the MIT AeroAstro 16.821 student team, and the authors would like to thank the team for their hard work and dedication to the project.

References

- [1] Wahba, G., *Spline Models for Observational Data*, 1990.
- [2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., 2017. URL https://hastie.su.domains/ElemStatLearn/printings/ESLII_print12_toc.pdf.