

# Physics-Informed Regression and Uncertainty Quantification of Aircraft Performance from Minimal Flight Data

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

This work presents a system-identification-like process for estimating an aircraft's aerodynamic and propulsive performance characteristics from minimal flight data. The method is based on a physics-informed regression approach, which uses an unsteady physics model of the aircraft's dynamics to constrain the regression problem. The method is demonstrated on a flight test dataset collected from a small electric aircraft. The results show that the method is able to estimate the aircraft's power curve, aerodynamic polar, and propulsive efficiency curve with surprisingly high accuracy given limited data. The method is able to estimate the uncertainty in these estimates.

Another contribution is a new method for estimating the statistical properties of a signal's noise component from the signal itself. Among other assumptions, this method relies on assuming that the true signal and the noise are spectrally-separated. Noise is isolated by repeated numerical differentiation, which is shown to be stable in numerical experiment. An example implementation of this noise variance estimator is given and shown to be convergent to the true noise variance in numerical experiment. Higher-order implementations of this estimator are also presented and shown to isolate the noise accurately even when the underlying true signal is quite close to the Nyquist frequency.

## I. Nomenclature

AGL	= above ground level
$C_D$	= drag coefficient
$C_L$	= lift coefficient
$c_J$	= a quantity proportional to the propeller advance ratio $J$
$c_n$	= a quantity proportional to the propeller rotational rate $n$
$D$	= drag force
$d$	= order of noise estimator
$f_{\text{sample}}$	= frequency of sensor measurements
$f_{\text{signal}}$	= highest-relevant frequency of true signal
$g$	= gravitational acceleration
$h(t)$	= altitude at time $t$
$J$	= propeller advance ratio
$L$	= lift force
$m$	= instantaneous aircraft mass
$\eta_{\text{propulsive}}$	= propulsive efficiency
$N$	= number of samples in dataset
$n(t)$	= noise component of measurement at time $t$
$n$	= propeller rotational rate
$P_{\text{loss}}$	= power lost from the aircraft at time $t$ ; primarily drag, but also other system losses
$P_{\text{in}}$	= input power to propulsion system
$P_{\text{thrust}}$	= useful power at airstream, $T \cdot U$
$\sigma_n$	= standard deviation of noise component

---

\*PhD Candidate, AIAA Student Member

†T. Wilson Professor in Aeronautics, AIAA Fellow

$s(t)$	= true signal component of measurement at time $t$
$\theta_i$	= various unknown parameters of the aircraft's performance curves
$T$	= thrust force
$t$	= time
$U(t)$	= true airspeed at time $t$
$V(t)$	= battery voltage at time $t$
$x(t)$	= measured data (signal + noise) of a generic sensor at time $t$

## II. Introduction

A primary goal of a new aircraft's initial flight test campaign 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  $C_L$ - $C_D$  aerodynamic polar (referred to throughout this work as simply the *aerodynamic polar*). This gives the relationship between the aircraft's lift and drag coefficients as angle of attack is varied, representing the locus of possible aerodynamic operating points.
- 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, thus isolating the effect of propulsive power. (If the aircraft is propeller-driven, propeller windmilling drag can also be estimated and calibrated out for improved accuracy, although this introduces significant additional uncertainty.) An implicit assumption here is that pitch trim is adjusted to maintain these airspeeds. 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:

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

where  $h(t)$  represents the altitude at time  $t$ ,  $U$  represents the airspeed, and  $t_1$  and  $t_2$  are the beginning- and end-times of the glide. The drag is then computed as  $D = W/(L/D)$ , and  $C_L$  and  $C_D$  can be nondimensionalized from here.

For aerodynamically-efficient aircraft with high  $L/D$  ratios (such as sailplanes), another common strategy for estimating aerodynamic polars during flight testing is to fly these power-off glides along with a wingman aircraft. This wingman aircraft flies the same power-off glide as the test aircraft, at the same airspeed, with the same starting altitude, and in the same vicinity. The wingman aircraft is assumed to have a known aerodynamic polar, and hence the difference in altitude between the two aircraft at the end of a glide segment can be used to estimate the test aircraft's aerodynamic polar. This approach calibrates out the effect of local airmass motion (e.g., thermals) on the measured glide ratio, which can be significant for aerodynamically-efficient aircraft.

More extreme strategies have also been implemented to measure the aerodynamic performance curve. For example, the glide ratio of the P-51 fighter aircraft was allegedly measured by removing the propeller, towing it aloft, and gliding it down. This removes the error associated with estimating propeller windmilling drag, but this is obviously an onerous, expensive, and potentially less-safe procedure to include in a flight test campaign.

Full-scale wind tunnel testing is another possible method for estimating the aerodynamic polar of the as-built aircraft, although this is similarly expensive and time-consuming, rendering it infeasible for most aircraft development programs. Scale models in wind tunnels as well as CFD simulation can also provide useful aerodynamic estimates, but these results can differ significantly from the aerodynamics of real as-built aircraft. For example, Hoerner [1] gives a pathological

example of the Messerschmitt Bf 109 fighter aircraft, where roughly *half of the drag of the as-built aircraft* is attributable to components that are typically de-featured in modern computational analysis\*. This Bf 109 case is somewhat extreme, but even for modern aircraft, the drag of the as-built aircraft can differ significantly from computational estimates or small-scale wind tunnel testing. This motivates flight-test-based aerodynamic polar characterization.

Like the aerodynamic polar, the power curve is also often measured by flying in steady flight segments at various airspeeds. For each airspeed, level flight is held with power and pitch trim adjusted as needed for stabilized cruise. The required input power to the propulsion system  $P_{in}$  is then measured. In a conventionally-fueled airplane, this can be computed 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, or indirectly from the engine RPM depending on the powerplant data available. For an electric airplane, it can be computed as the product of battery current and voltage, which are readily available. 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 thrust power done by the propulsion system. The thrust power (or “air power”) can be computed as:

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

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

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

These traditional methods for measuring aerodynamic and propulsive performance have several limitations. The biggest limitation by far is that they require vast amounts of flight time to obtain a sufficient number of data points to characterize the aircraft’s performance.

Furthermore, the aircraft must be flown at precisely-controlled conditions. Traditional methods are not well-suited to estimating the performance of an aircraft that is not in steady flight – in the methods described here, only steady data (e.g., a stabilized glide or cruise) can be used. For example, data recorded during an aircraft’s climb or descent to flight test altitude is discarded, which is wasteful; ideally, every single second of data should contribute to refining our estimate of the aircraft’s performance. This requirement to maintain steady conditions can be frustrating and tedious for the pilot to maintain, 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 that leaves it difficult to quantify the uncertainty in the resulting performance estimates.

Nevertheless, the methodology described here provides a useful existence proof that airplane performance curves are fundamentally observable (and hence, able to be recovered) given airspeed, altitude, and power data of sufficiently varied flight. A primary goal of this work is to develop a methodology that can estimate these performance curves from a much smaller amount of flight data, which would decrease the cost and intentionality required to obtain these performance estimates.

## B. Inference-Based Flight Data Reconstruction Methods

We propose a method of estimating these performance characteristics that views the aircraft flight test as a data-generating process, from which we can infer performance relationships. Figure 1 broadly illustrates the toolchain for performance reconstruction that represents this work’s central contribution. This approach contrasts with traditional methods, which view flight testing as a process of pure measurement, based only on analyzing the data itself. This

---

\*e.g., rivets, sheet metal gaps, control surface gaps & hinges, engine and radiator thermal effects, inlet flow distortion, antennae, sensors, lights, guns, holes for ventilation and cooling, etc.

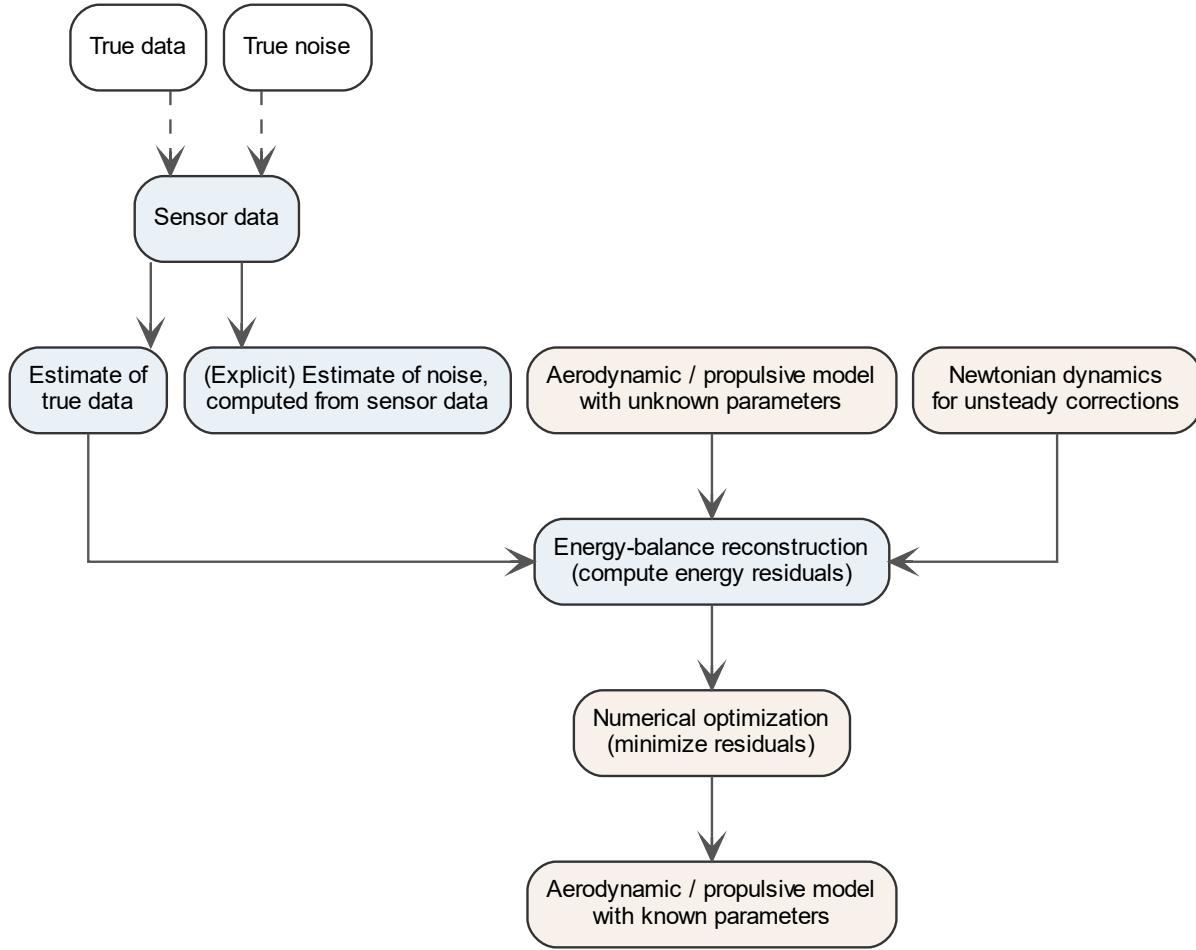
distinction manifests in a few ways, each of which recognizes that we have process- or domain-specific information that collapses the possible solution space of possible performance models:

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 IV.A. 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 aiding uncertainty quantification. In Figure 1, this is represented by the separation of sensor data into its signal and noise components.

We also know certain characteristics of the aircraft’s performance curve based on first-principles physics, which can further increase the effective information density and our resulting performance estimates. Another way to think about this is that the function space of performance curves is restricted to functions that generally follow first-principles physics, allowing us to reduce uncertainty on these resulting inferred relations. For example, we know that the true state of the aircraft must follow invariants from Newtonian physics (e.g., conservation of energy). Likewise, propulsive efficiency is a bounded function that must not exceed unity to be physically plausible. Embedding these constraints and invariants from physics reduces the number of unknowns, enabling more robust correction for unsteady effects. This is described in Section V.

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.

The example given in the subsequent sections is for a small electric-driven propeller aircraft, although the methodology is applicable to any aircraft type. Some considerations for applying this methodology to other aircraft types are discussed in Section VII.



**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 VIII.A. 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.



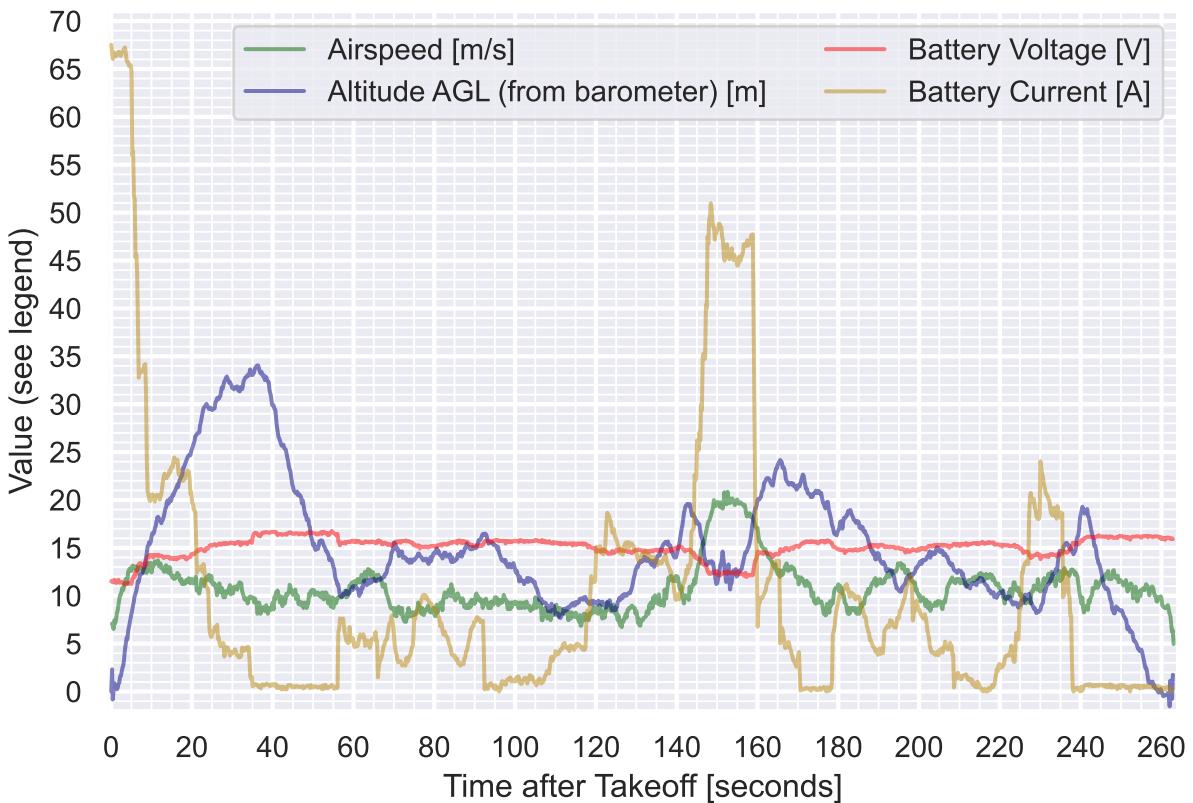
(a) *Solar Surfer* in flight



(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.



**Fig. 3** Raw sensor data from *Solar Surfer* aircraft test flight

Summary statistics of this data are given in Table 1.

**Table 1** Summary statistics of raw sensor data from *Solar Surfer* aircraft test flight. Based on 1,315 samples per measured quantity.

Description	Airspeed [m/s]	Altitude AGL [m]	Voltage [V]	Current [A]
Mean	10.8	16.4	15.05	10.31
Standard Dev.	2.52	6.81	1.08	13.73
Min	5.0	0	11.13	0.00
25%	9.1	12.3	14.69	0.94
50%	10.3	15.4	15.28	5.25
75%	11.7	19.3	15.67	13.28
Max	20.9	35.6	16.77	67.53

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 1,315 data points *in total* for each of the plotted sensor traces. 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 quite limited. While traditional methods might be able to extract general aircraft flight performance based on this quantity of data (i.e., average required power over the flight), a detailed airspeed-based power curve is not typically achievable with such limited data.

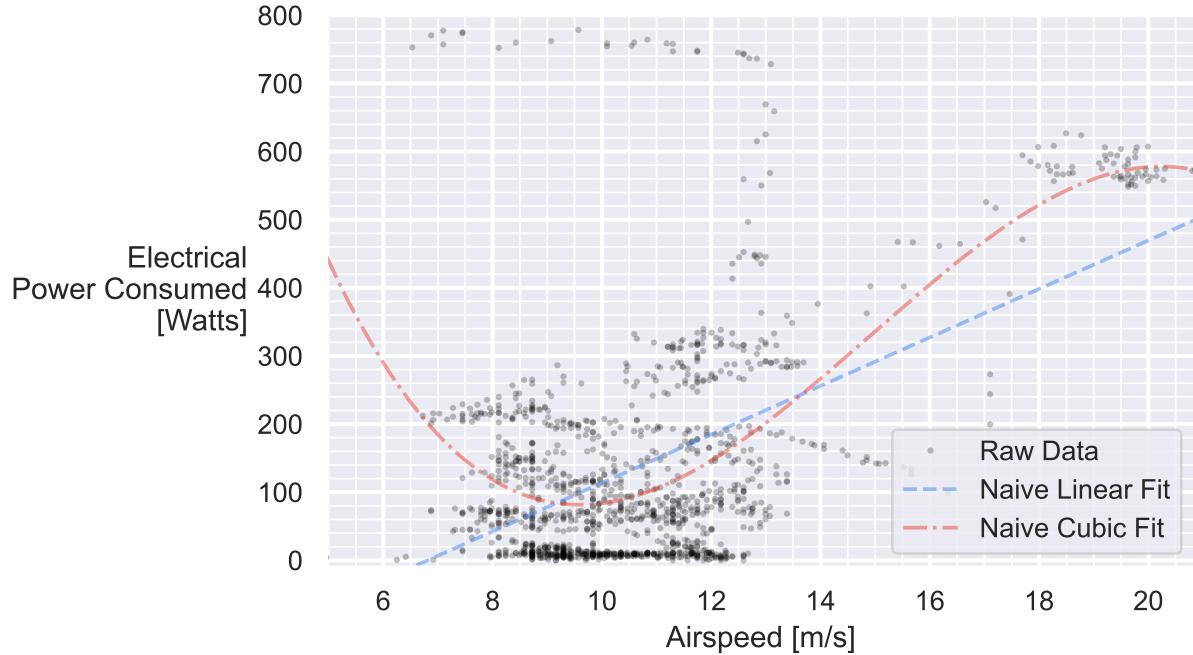
Consider that even the most sophisticated techniques for system identification via statistical inference require the system to be observable in some form - no amount of math can recover a signal that is simply not present in the data. This is difficult, because after noise removal, the data really no more than perhaps a half-dozen “system excitations” from which we can infer aircraft performance. 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.

## 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 to demonstrate the infeasibility of this approach. 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

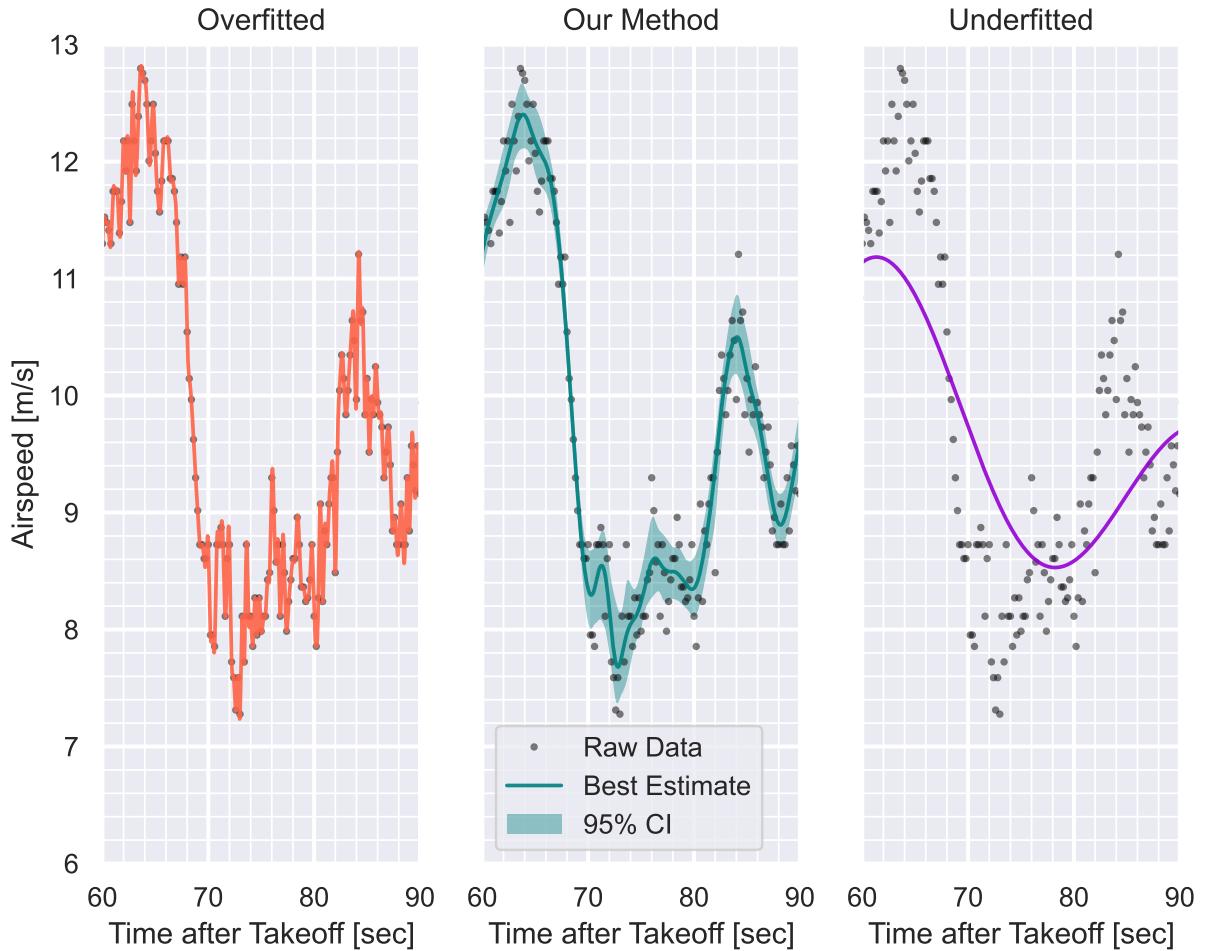
The first contribution here is to develop a means of pre-processing the sensor data to isolate noise; readers who are more interested in physics-informed regression than uncertainty quantification may skip forward to Section V. Traditionally, noise is removed by simple averaging; after this process is 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 cases where data is limited provide a strongly incentive 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 separate 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 is 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”, although it’s not immediately clear why.

## 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 aim to recover (i.e., the system is “slow” relative to the sample rate), with this assumption quantified later in the present work.

Under these assumptions above, the probabilistic properties of the noise reduce to a single parameter: the variance of the noise  $\sigma_n^2$ . 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

The variance of the noise is a useful quantity to estimate for several reasons. First, a probabilistic model of the sensor noise allows us to make optimal choices about the bias-variance tradeoff, armed 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 [2]. 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 difficult.

### 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, while the noise remains white. 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, consider a scenario where the measured data  $x(t)$  consists of a true signal  $s(t)$  and noise  $n(t)$ :

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

Assume the noise of the sample  $n(t)$  is drawn from  $\mathcal{N}(0, \sigma_n^2)$ , with initially-unknown  $\sigma_n$ . Now, consider two discrete measurements taken at adjacent points in time  $t_1$  and  $t_2$ :

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

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

If  $t_1 \approx t_2$  (as would be the case for subsequent samples), then  $s(t_1) \approx s(t_2)$ , contingent on the sample rate greatly exceeding the effective fastest frequency of the true signal. Hence:

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

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  $\sigma_n^2$  is the variance of the sensor noise. So, the following is then approximately true:

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

Therefore, taking the mean difference across all adjacent pairs of measured data  $x(t)$  provides an unbiased estimator of  $\sigma_n^2$ , the variance of the noise:

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

where  $N$  is the number of samples in the dataset. We call this a *first-order* noise estimator as it takes a first-order numerical derivative of the measured data.

### 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  $f_{\text{signal}}$  with added independent, normally-distributed noise. We then sample this sinusoid at some frequency  $f_{\text{sample}}$  (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 2. The higher-order estimators are described in the following sections.

**Table 2 Numerical demonstration of noise variance estimator performance**

$f_{\text{signal}}$	$f_{\text{sample}}$	True $\sigma_n$	Estimated $\sigma_n$		
			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

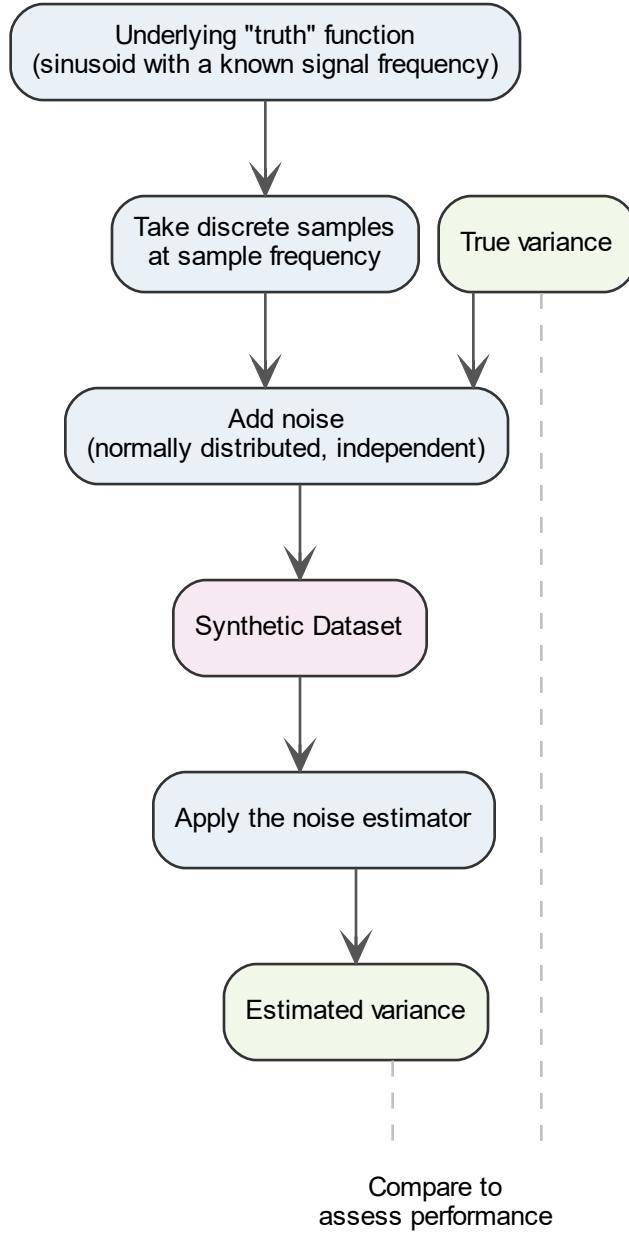
As shown in Table 2, the first-order noise estimator yields a  $\sigma_n$  value that is very close to the true value. We will also show later that this estimator is convergent to the true value as the number of samples increases.

However, this estimator is only first-order convergent with respect to the ratio between the sampling frequency  $f_{\text{sample}}$  and the underlying dynamics of the system  $f_{\text{signal}}$ . 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 2, where the estimator produces an incorrect result by a factor of 3.

### 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:



**Fig. 6 Procedure for numerical demonstration of estimator performance**

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

where  $N$  is the number of samples in the dataset. As shown in Table 2, 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  $f_{\text{signal}}$ ), above which the power spectral density gradually goes to zero.

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 - a key insight into this method.

### 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:

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

An example stable numerical implementation of this estimator is given in Appendix VIII.B.

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.

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 one can use extremely-high-order estimators to estimate the noise variance with very high accuracy.

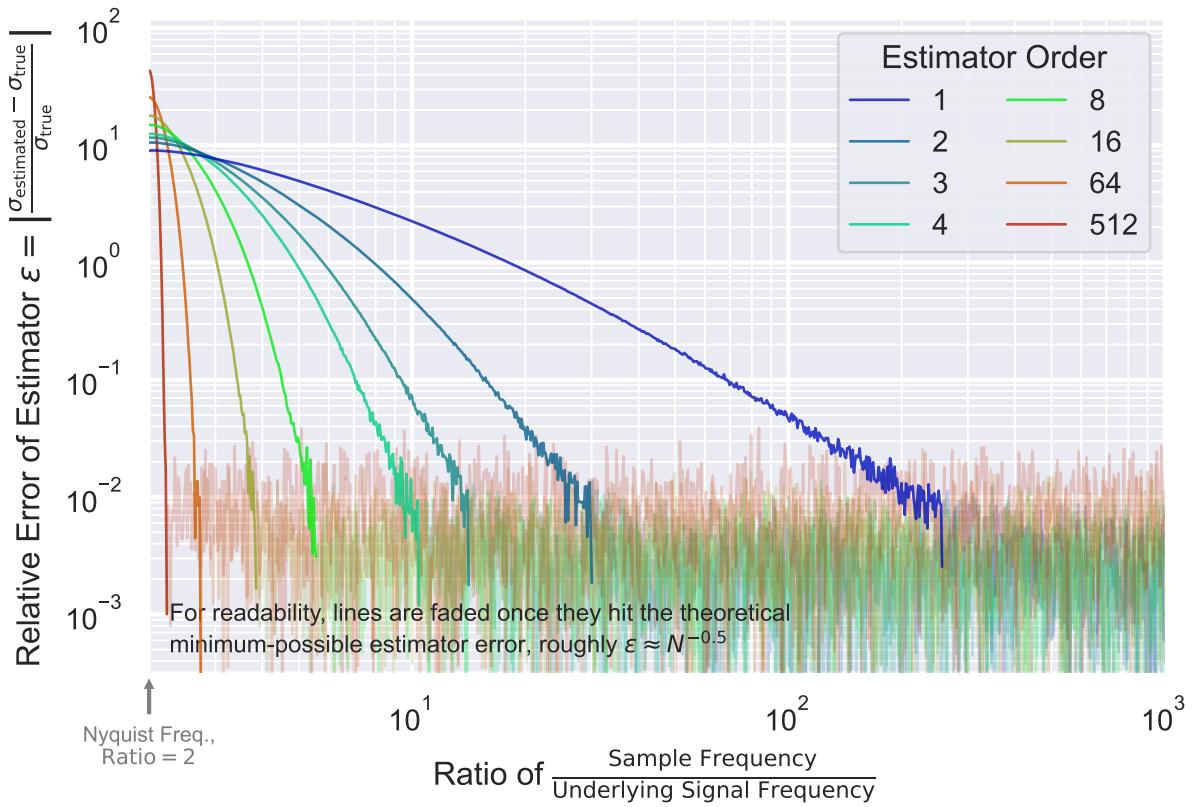
## B. Uncertainty Quantification Using Noise Variance Estimators

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 [3, 4]) with a noise-variance-aware spline interpolator (described in [2, 3]). 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.

Figure 8 certainly 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 our propensity to make statistical conclusions that go beyond what the data can truly support.

Furthermore, this uncertainty quantification allows us to determine where in the flight envelope more data should be collected (i.e., serve as an acquisition function). For example, to refine the power curve of Figure 8, we might aim to fly future test flights at speeds between 13 and 17 m/s.

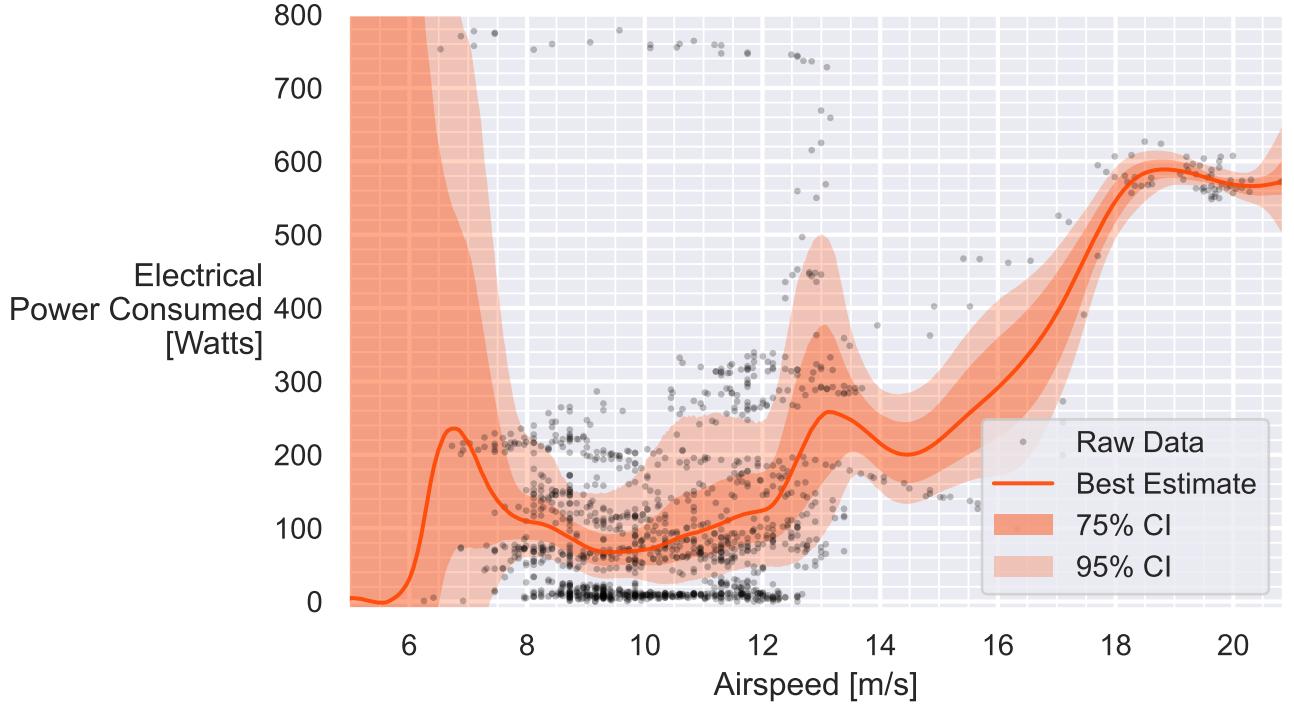
With this said, two major problems remain with the approach of Figure 8. First, the resulting model uncertainty is



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

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.

Second, the model still allows physically-imausible relationships, such as the unexplained power curve increase near 13 m/s and the flattening power curve near 20 m/s. Clearly, the uncertainty bounds here are wider than what they would be if our physical insights were incorporated into the model.



**Fig. 8** Raw uncorrected data for the power curve, with a noise-variance-aware spline model. A resampling bootstrap allows uncertainty quantification. This improves over the naive approach of Figure 4, although, the lack of physics-based model structure still leads to the recovery of a physically-implausible result.

## V. Incorporating Physics-Based Data Corrections and Domain-Specific Knowledge

When inferring the performance characteristics of a physical system, the laws governing that system can be used to significantly narrow the solution space and reduce the unexplainable variance in the data.

### A. The Aircraft as an Energy System

In the example of the power curve regression, one way to do this is to consider the aircraft as an energy system and exploit the fact that energy should be conserved. Consider the following basic accounting equation for the total energy of an aircraft at some time  $t$ :

$$E_{\text{total}}(t) = \frac{1}{2} m U(t)^2 + m g h(t) + \int_0^t [P_{\text{loss}}(\tau) - P_{\text{thrust}}(\tau)] d\tau = \text{constant} \quad (5)$$

where:  $E_{\text{total}}$  = Total energy of the aircraft system. System boundary includes the aircraft but not the onboard stored energy (e.g., battery) or power-to-air (e.g., drag); hence, the representation of these as flow terms. Sign convention is that energy added to the airplane is positive.

$m$  = Mass of the aircraft

$U(t)$  = Airspeed of the aircraft at time  $t$

$g$  = Local gravitational acceleration

$h(t)$  = Altitude of the aircraft at time  $t$

$P_{\text{thrust}}(t)$  = Thrust power ( $T \cdot U$ ) to the aircraft at time  $t$

$P_{\text{loss}}(t)$  = Power lost from the aircraft at time  $t$ ; primarily drag, but also other system losses

Note that the above equation is “fully-accounted” in that it includes all energy flows into and out of the airplane, including from onboard sources. Hence,  $E_{\text{total}}$  should remain constant throughout the flight.

To exploit this knowledge that  $E_{\text{total}}$  is constant, we differentiate Equation 5 with respect to time to obtain an expression for the total power. The total power must equal zero for energy conservation to hold. In writing the following expression, we can also expand the power gains and losses into constituent terms as appropriate for an electric aircraft:

$$P_{\text{total}}(t) = m U(t) \dot{U}(t) + m g \dot{h}(t) + [U(t) D(\theta) + P_{\text{avionics}} - V(t) i(t) \eta_{\text{propulsive}}(\theta)] = 0 \quad (6)$$

where:  $(\cdot)$  = Time derivative

$P_{\text{total}}(t)$  = Total power at time  $t$ . Just as with  $E_{\text{total}}$ , this is a “fully-accounted” quantity, and hence  $P_{\text{total}}$  should equal zero.

$V(t)$  = Battery voltage at time  $t$

$i(t)$  = Battery current at time  $t$

$\eta_{\text{propulsive}}(\theta)$  = Propulsive efficiency ( $P_{\text{thrust}}/P_{\text{in}}$ ) as a function of both flight data and unknown parameters  $\theta$

$P_{\text{avionics}}$  = Power consumed by avionics. In the case of the example dataset, this is assumed to be constant and equal to the average measured power consumption during motor-off periods.

$D(\theta)$  = Drag as a function of both flight data and unknown parameters  $\theta$

Ideally, this equation would be used to simply correct the data for the unsteady terms (i.e., acceleration and climb terms) in Equation 6, at which point we would directly use that data and apply traditional curve-fitting techniques. However, the observant reader will notice that Equation 6 cannot, on its own, be used to correct the data. Both the propulsive efficiency and drag terms are unknown functions of the flight data and unknown parameters  $\theta$ , and hence these need to be solved for.

These parameters can be defined as the solution of a residual minimization problem (i.e., optimization) between our data and model—effectively, system identification. Here, the residual in question is the instantaneous  $P_{\text{total}}$  given in Equation 6, and we aim to minimize some norm<sup>†</sup> of the residual vector, as resampled from the data reconstruction shown in Section IV.A. However, the interesting insight here is that changes to our model parameters affect *both* the model and the corrected data, and hence the model and data must be optimized simultaneously. Stated equivalently: we are not only correcting the data using our models; we are also simultaneously fitting the models to the data.

One other consideration to note with this energy-corrected formulation is that it relies on the time derivative of sensor data. Because our data reconstructors from Section IV.A are based on splines, these can be conveniently differentiated by conducting simple polynomial differentiation piecewise between knot points. However, the derivative of the best estimator is not necessarily the bias-variance-optimal estimator of the derivative, due to the amplification of high-frequency variation during the differentiation process. To combat this, we add a Gaussian low-pass-filter with time constant  $\sigma = 4$  sec to the derivative estimator. Developing a rigorous method for determining the optimal filter time constant is left as future work, but is theoretically possible using a cross-validation approach.

## B. Modeling the Unknown Functions

There are two unknown models in Equation 6: the propulsive efficiency and the drag. We replace these with parametric models with forms taken from domain-specific knowledge. In this example, we use simple mathematical relations to demonstrate that even basic restrictions on model form can significantly improve the quality of the resulting model. In practice, more sophisticated models based on parameterized low-fidelity analyses could be used, although this may come at the expense of reduced interpretability.

### 1. Drag Model

The aerodynamic drag can be modeled as a function of airspeed, as required to close Equation 6. To do this, we use the following procedure:

- 1) Compute the instantaneous lift coefficient  $C_L(t)$  from the airspeed  $U(t)$ , assuming a load factor of 1 and known aircraft properties (wing area, mass, etc.).
- 2) Map the  $C_L$  to a drag coefficient  $C_D$  using a parametric model with unknown parameters  $\theta_1$  to  $\theta_6$ . The form of this model is a piecewise-quadratic  $C_D(C_L)$  function, as shown in Figure 9. This model form is identical to the assumed form of the profile drag function in the AVL [5] aerodynamics code. In this model form:

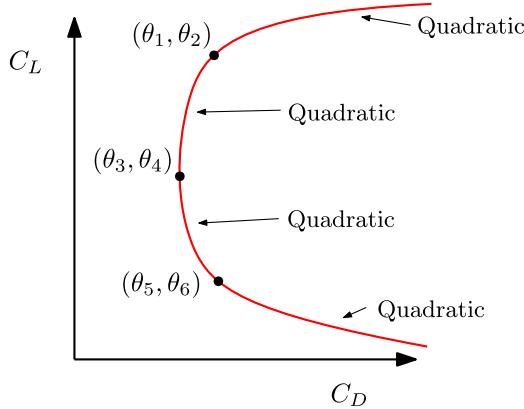
---

<sup>†</sup>Typically the  $L_1$  or  $L_2$  norm

- $(\theta_1, \theta_2)$  represents the point of positive stall onset
- $(\theta_3, \theta_4)$  represents the minimum drag point
- $(\theta_5, \theta_6)$  represents the point of negative stall onset

The piecewise-quadratic function is constructed to be  $C^1$ -continuous between the four regions delineated by these points. This requirement fully-constrains the non-stalled regions of the curve. The regions above positive stall and below negative stall use an assumed curvature (identical to the values used in AVL [5]) that generally models post-stall drag rise.

3) Dimensionalize the computed  $C_D$  to obtain the drag.



**Fig. 9 Assumed parameterized form of the aerodynamic polar, where drag coefficient is a piecewise-quadratic function of the lift coefficient. Same mathematical form as the profile drag function used by AVL [5].**

## 2. Propulsive Efficiency Model

Similarly, a characteristic model for the propulsive efficiency is defined based on domain-specific knowledge. In an ideal case, this model would take the propeller advance ratio  $J$  as an input and yield the propulsive efficiency  $\eta_{\text{propulsive}}$  as an output. However, the advance ratio is not directly measurable in this *Solar Surfer* case study, as this requires propeller RPM data which is not available. However, we do have sufficient data to infer a quantity known to differ from the advance ratio  $J$  by a constant factor; we denote this proportional quantity  $c_J$ .

Therefore, the propulsive efficiency model is a mapping from  $c_J$  to  $\eta_{\text{propulsive}}$ , rather than from  $J$  to  $\eta_{\text{propulsive}}$ . Although this means that the maximum-efficiency advance ratio  $J$  cannot be estimated, the peak propulsive efficiency itself can be estimated; ultimately, it is the peak efficiency that is the primary performance metric of interest.

To construct this model, we use the following procedure:

- 1) Use the measured current  $i(t)$  to estimate  $c_n$ , a quantity proportional to the propeller rotational rate  $n$ , using the  $P_{\text{in}} \propto n^3$  relationship found using the Buckingham  $\pi$  theorem [6]:

$$c_n = i(t)^{1/3} \propto n$$

- 2) Propagate this to estimate  $c_J$ , a quantity proportional to the propeller advance ratio  $J$ :

$$c_J = \frac{U(t)}{c_n} \propto J$$

- 3) Map  $c_J$  to a propulsive efficiency  $\eta_{\text{propulsive}}$  using the following model form and unknown parameters  $\theta$ :

$$\eta_{\text{propulsive}} = \theta_7 \cdot \text{softmin} \left( c_J / \theta_8, \frac{c_J - \theta_9}{\theta_8 - \theta_9} \right) \quad (7)$$

where softmin is a smooth approximation of the minimum function, analogous to the logsumexp definition of softmax by Cook and others [7]:

$$\text{softmin}(x, y) = -\theta_{10} \cdot \ln \left( e^{-x/\theta_{10}} + e^{-y/\theta_{10}} \right)$$

Note the physical interpretation of several parameters here:

- $\theta_7$  provides an upper bound on the peak propulsive efficiency
- $\theta_8$  represents the  $c_J$  value corresponding to peak efficiency
- $\theta_9$  represents the  $c_J$  value corresponding to the “pitch speed” of the propeller, where the thrust (and hence efficiency) goes to zero.
- $\theta_{10}$  is a smoothing parameter that controls the sharpness of the transition between the two regions.

This model form captures expected behavior of the propulsive efficiency curve, including the fact that the propulsive efficiency is zero at zero airspeed and zero at the pitch speed of the propeller. The  $\theta_7$  parameter is constrained to be less than one to bound efficiency to physically-plausible values. The resulting model form matches well with expected propeller efficiency trends, such as the curves seen in [6].

### C. Solving the Optimization Problem

With these models in place, a residual minimization problem of the following form can be solved to yield optimal  $\theta$  values, where  $P_{\text{total}}$  (the net power into the system, after all inflows and outflows should be accounted for) from Equation 6 is used as the residual:

$$\begin{aligned} & \underset{\theta}{\text{minimize}} \quad \sum (P_{\text{total}})^2 \\ & \text{subject to} \quad \theta \text{ constraints described in Section V.B} \end{aligned} \tag{8}$$

Optionally, other norms of the  $P_{\text{total}}$  residual vector can be minimized instead. The  $L_1$  norm makes a particularly compelling choice here due to its robustness to outliers [8]. In the example problem here, we solve this optimization problem numerically using IPOPT [9] via the AeroSandbox [10] framework for engineering design optimization.

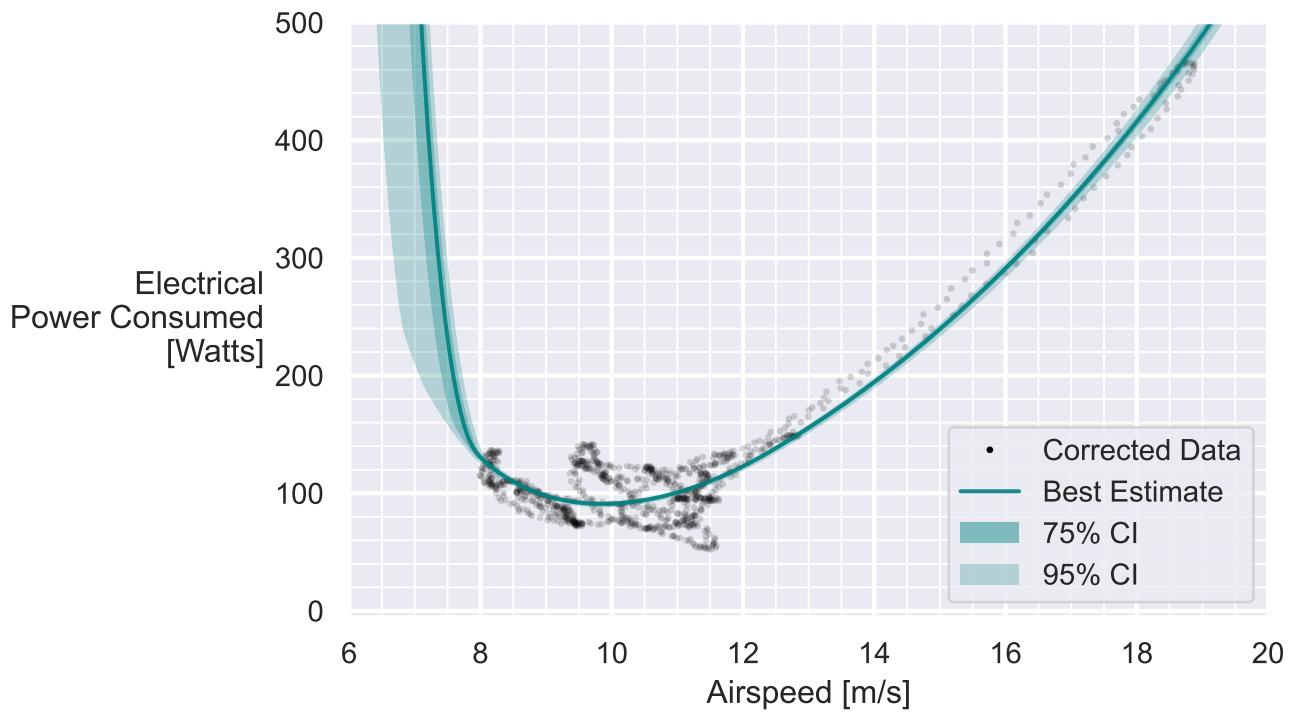
### D. Results

With the optimization problem solved, we can now use the resulting  $\theta$  values to compute the corrected data and model. This is shown in Figure 10. The corrected data and model are now in excellent agreement, and the corrected data has significantly less unexplained variance than the raw data. The remaining unexplained variance is attributable to a variety of factors, such as wind updrafts and downdrafts and non-unity load factors during banked flight. In theory, both these effects could be corrected out using a similar inference procedure that is shown in this work (which could be augmented by, for example, IMU data). This is left as future work, which, if successful, would reduce the data spread in Figure 10 further.

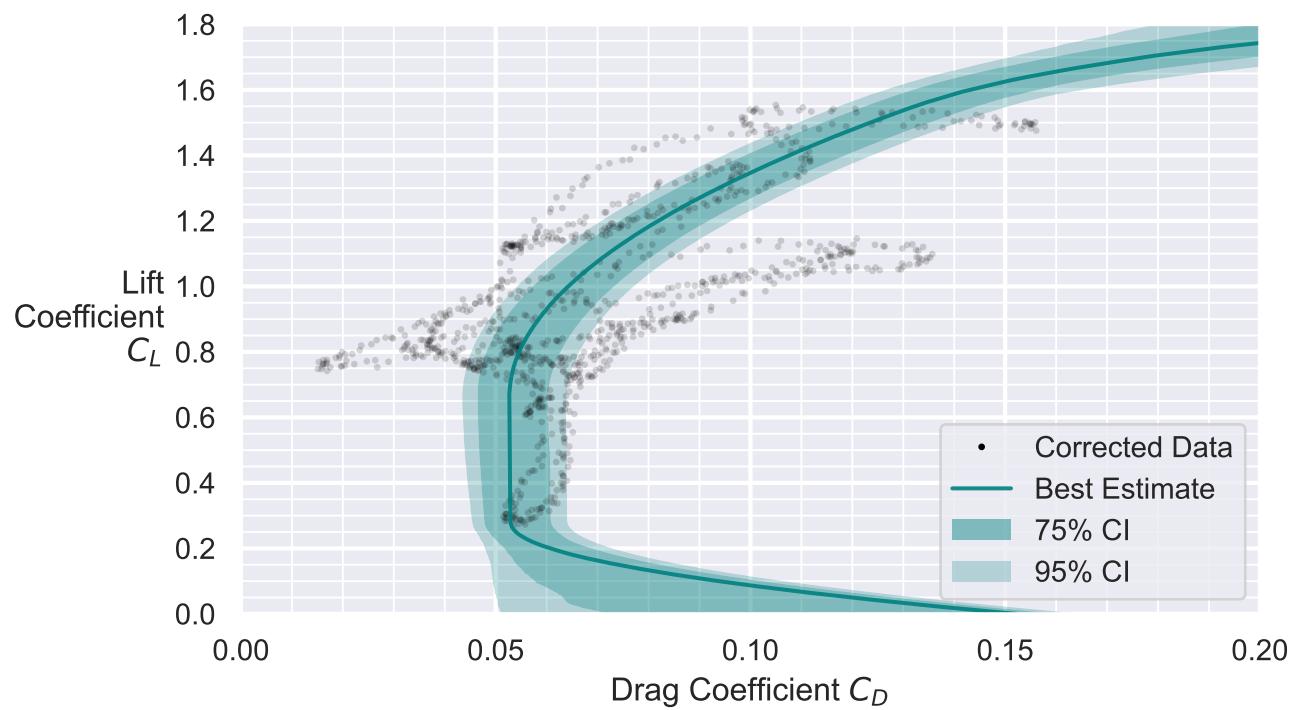
Conveniently, this process yields other critical performance information, like the aerodynamic polar and propulsive efficiency curve, “for free” as part of the fitting process. These are shown in Figures 11 and 12. Furthermore, we also obtain uncertainty estimates for all these quantities, which are critical for understanding the confidence we can have in the results.

A notable observation is that the inferred aerodynamic polar of Figure 11 and the propulsive efficiency curve of Figure 12 show relatively wide uncertainty bands, while the power curve of Figure 10 shows a relatively narrow uncertainty band. This is interesting, since in typical analysis the power curve is computed as a function of these two performance relations; hence, one would intuitively expect that the power curve would inherit the uncertainty of both relations. The reason this does not occur here is that the uncertainties in the aerodynamic polar and the propulsive efficiency curves are not independent. In other words, the power required can be inferred relatively precisely, but there is not quite enough data to rigorously determine whether power losses are more due to aerodynamic drag or propulsive inefficiency.

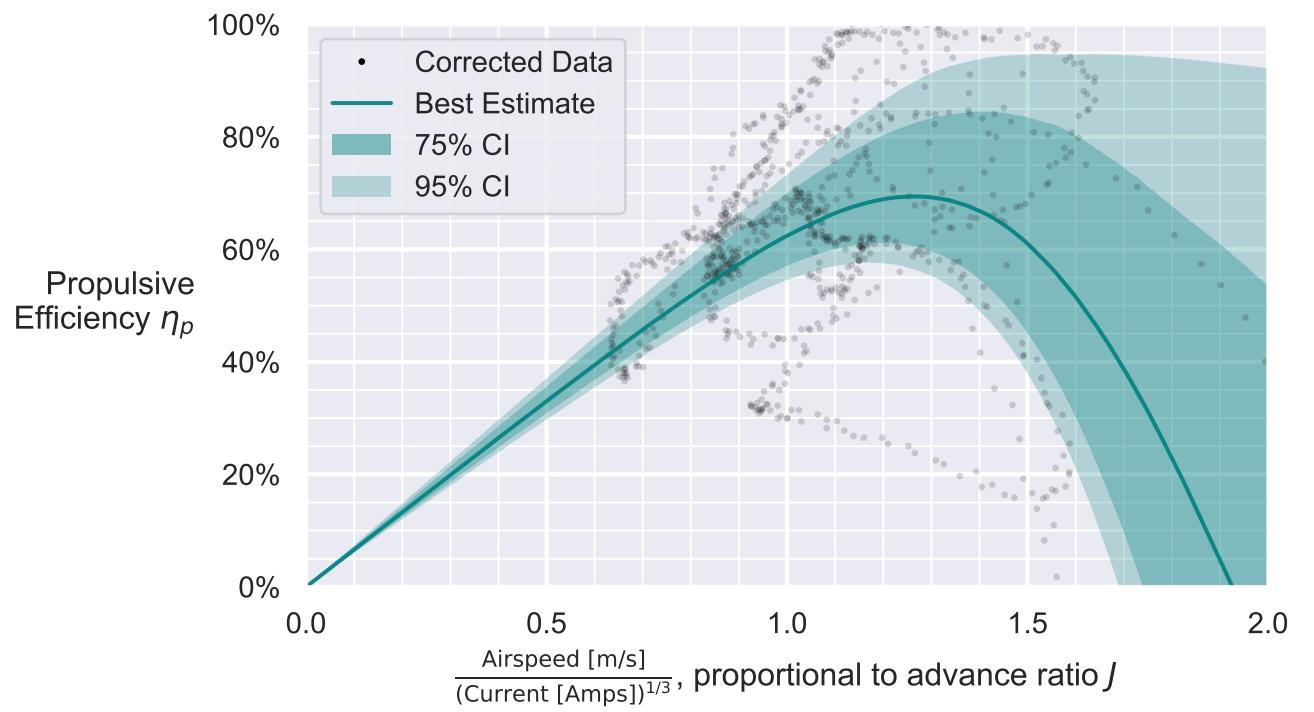
Compared to the naive approach in Figure 4, Figure 10 enables us to make very accurate, physically-grounded, and uncertainty-quantified assertions about the performance of the airplane, based on minimal data (here, only four minutes of flight time).



**Fig. 10** Inferred power curve with physics-informed corrections on data and model, and using noise estimator with a resampling bootstrap for uncertainty quantification. Power curve agrees closely with the data and shows physically-plausible behavior. Uncertainty quantification is significantly tighter than in Figure 8, as the function space has been restricted using embedded physics knowledge.



**Fig. 11** Inferred aerodynamic polar with physics-informed corrections on data and model, and using noise estimator with a resampling bootstrap for uncertainty quantification.



**Fig. 12** Inferred propulsive efficiency curve with physics-informed corrections on data and model, and using noise estimator with a resampling bootstrap for uncertainty quantification.

## VI. Reproducibility Statement

All code and data used in this paper is publicly available at <https://github.com/peterdsharpe/aircraft-polar-reconstruction-from-flight-test>.

## VII. Conclusion

In this paper, we have demonstrated a rigorous method for reconstructing the performance characteristics of an aircraft from flight test data. This method is based on a physics-informed correction of the data and model, and is shown to be effective at recovering the true power curve and aerodynamic polar of an example aircraft from noisy, limited, unsteady data. This method is also shown to be robust to the presence of outliers in the data, and to be able to provide uncertainty estimates for the resulting aerodynamic polar. This method is also shown to be able to recover the propulsive efficiency curve of the aircraft, which is a critical quantity for understanding the performance of the aircraft. Finally, this method is shown to be able to recover the aerodynamic polar of the aircraft from only four minutes of flight test data, which is a significant reduction in the amount of data required compared to the state of the art.

This work has focused on the ability of these inference-based reconstruction procedures to significantly reduce the required flight hours to characterize an airplane’s performance. However, an alternative way to “spend” this increased ability to extract information is on massively reducing instrumentation costs. For example, this approach makes it conceivable to “estimate out” the ambient wind field<sup>‡</sup> from groundspeed data and power measurement alone. If this is possible, it would enable flight test engineers to perform sufficient early-stage characterization of a new crewed airplane using nothing other than the sensors available on a modern smartphone. Here, GPS, IMU, and barometer sensors provide position and groundspeed information, and the phone’s microphone can be used to infer engine RPM and hence power and advance ratio from an audio spectrum. If successful, this would be an enormous reduction in instrumentation cost and time compared to the state of the art.

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 somewhat less simple.

---

<sup>‡</sup>An intuitive explanation for this is that if an airplane loiters in a circle at constant power setting, the upwind and downwind groundspeeds yield information about the wind field. In practice, some additional assumptions about wind steadiness and uniformity would likely be required to collapse the  $(x, y, z, t)$  space; otherwise, it is likely too large to realistically span.

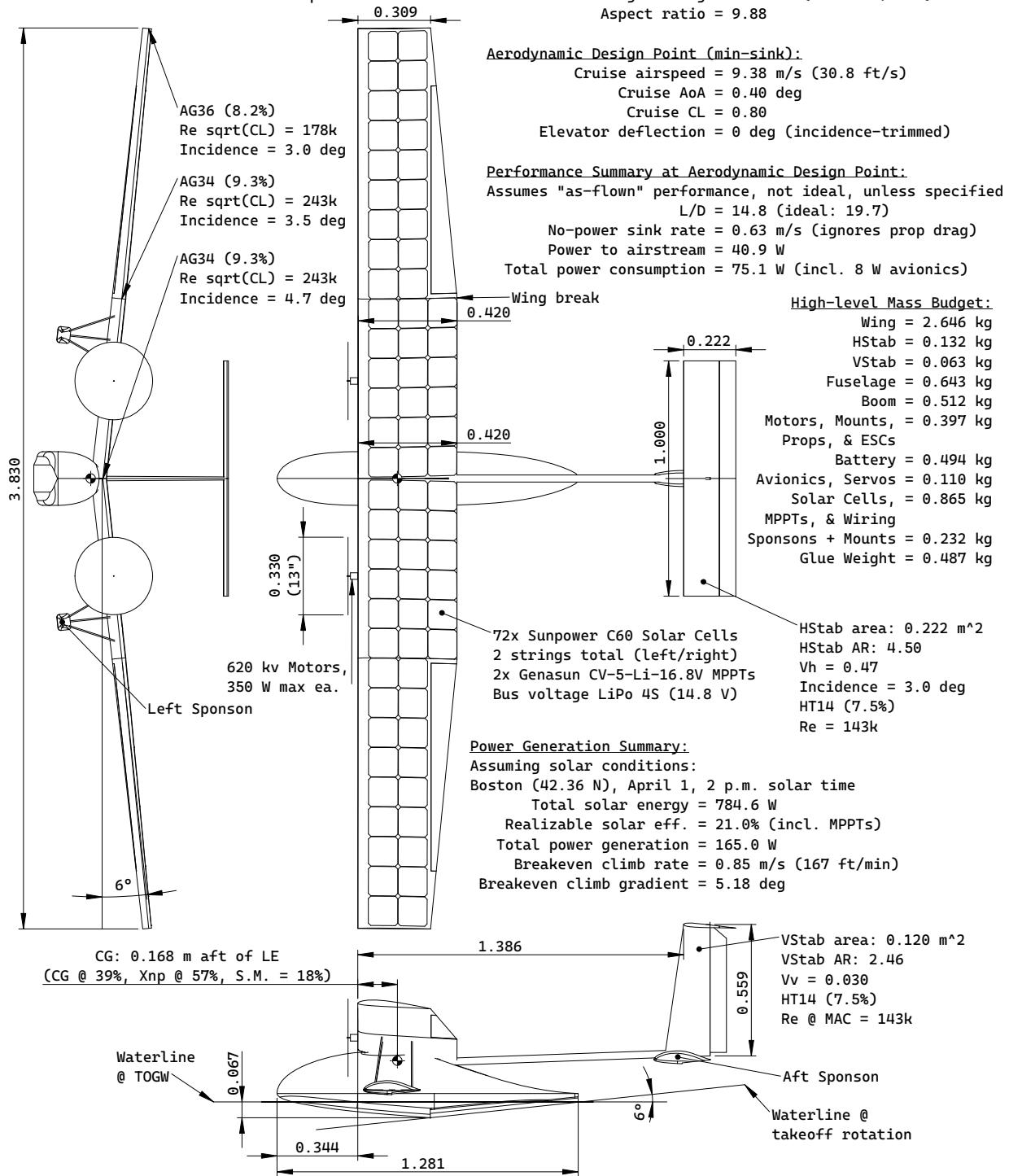


## VIII. Appendix

## 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 4) 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 is based on syntax for NumPy 1.24.3 and SciPy 1.11.1 within Python 3. This implementation uses the log-gamma function, which is much faster for high-order estimators and delays numerical overflow until much higher orders. The implementation also uses a convolutional kernel to vectorize the summation, enabling further speedups.

---

```
import numpy as np
from scipy.special import gammaln

ln_factorial = lambda x: gammaln(x + 1) # Returns ln(x!)

def estimate_noise_standard_deviation(data: np.ndarray, estimator_order: int = 10) -> float:
    """
    Estimates the standard deviation of noise in a time-series data set.

    Args:
        data: A 1D NumPy array of time-series data, assumed to consist of signal + noise.
        estimator_order: The order of the estimator to use. A positive integer.

    Returns: The estimated standard deviation of the noise in the data.
    """
    ln_f = ln_factorial(np.arange(estimator_order + 1)) # For speed, cache ln(x!) for x ∈ [0, estimator_order]
    coefficients = np.exp( # Create a convolutional kernel to vectorize the summation
        2 * ln_f[estimator_order] - ln_f - ln_f[::-1] - 0.5 * ln_factorial(2 * estimator_order)
    ) * (-1) ** np.arange(estimator_order + 1)
    coefficients -= np.mean(coefficients) # Remove any bias introduced by floating-point error
    sample_stdev = np.convolve(data, coefficients[::-1], 'valid') # Convolve the data with the kernel
    return np.mean(sample_stdev ** 2) ** 0.5 # Return the standard deviation of the result
```

---

**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] Hoerner, S. F., *Fluid-Dynamic Drag*, 1965.
- [2] Wahba, G., *Spline Models for Observational Data*, 1990.
- [3] Gramacy, R. B., *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*, Chapman Hall/CRC, Boca Raton, Florida, 2020. <http://bobby.gramacy.com/surrogates/>.

- [4] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2<sup>nd</sup> ed., Springer Series in Statistics, 2017. URL <https://hastie.su.domains/ElemStatLearn/>.
- [5] Drela, M., “Athena Vortex Lattice (AVL),” Massachusetts Institute of Technology, Sep. 2004. <https://web.mit.edu/drela/Public/web/avl/>.
- [6] Spakovszky, Z., Greitzer, E. M., and Waitz, I. A., “MIT Unified Engineering Course Notes, Thermodynamics and Propulsion: Performance of Propellers,” 2007. URL <https://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node86.html>.
- [7] Cook, J., “Basic Properties of the Soft Maximum,” [https://www.johndcook.com/soft\\_maximum.pdf](https://www.johndcook.com/soft_maximum.pdf), Sep. 2011.
- [8] Brunton, S. L., and Kutz, J. N., *Data Driven Science & Engineering*, 2017.
- [9] Wächter, A., and Biegler, L. T., “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57. <https://doi.org/10.1007/s10107-004-0559-y>.
- [10] Sharpe, P. D., “AeroSandbox: A Differentiable Framework for Aircraft Design Optimization,” Master’s thesis, Massachusetts Institute of Technology, 2021.