

Pteromys: Interactive Design and Optimization of Free-formed Free-flight Model Airplanes

Nobuyuki Umetani^{1,2*}

Yuki Koyama¹

Ryan Schmidt²

Takeo Igarashi¹

¹The University of Tokyo / JST ERATO

²Autodesk Research

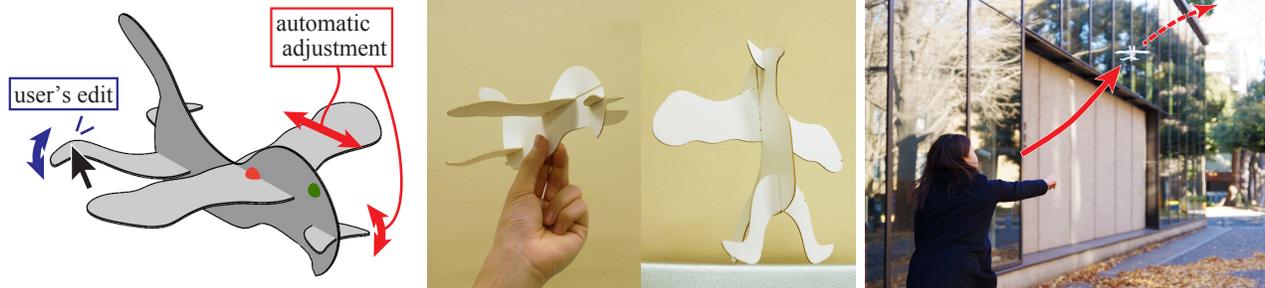


Figure 1: (Left) Our model airplane design tool analyzes the aerodynamic properties of a glider and optimizes while the user interactively designs the plane. (Center) The user fabricates the airplane. (Right) The airplane actually flies.

Abstract

This paper introduces novel interactive techniques for designing original hand-launched free-flight glider airplanes which can actually fly. The aerodynamic properties of a glider aircraft depend on their shape, imposing significant design constraints. We present a compact and efficient representation of glider aerodynamics that can be fit to real-world conditions using a data-driven method. To do so, we acquire a sample set of glider flight trajectories using a video camera and the system learns a nonlinear relationship between forces on the wing and wing shape. Our acquisition system is much simpler to construct than a wind tunnel, but using it we can efficiently discover a wing model for simple gliding aircraft. Our resulting model can handle general free-form wing shapes and yet agrees sufficiently well with the acquired airplane flight trajectories. Based on this compact aerodynamics model, we present a design tool in which the wing configuration created by a user is interactively optimized to maximize flight-ability. To demonstrate the effectiveness of our tool for glider design by novice users, we compare it with a traditional design workflow.

CR Categories: I.3.3 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling;

Keywords: fabrication, aerodynamics, optimization

Links: [DL](#) [PDF](#) [WEB](#) [VIDEO](#)

*e-mail:n.umetani@gmail.com

1 Introduction

Humanity has always been fascinated with flight, and most of us have experienced the joy of creating simple flying machines in the form of paper airplanes. One only has to take model aircraft slightly more seriously to discover the wide range of forms that simple hand-launched gliders can take on (Figure 2). The advent of personal fabrication devices, such as desktop 3D printers and CNC cutters, has greatly expanded the ability of individuals to construct precise glider aircrafts. However, it is surprisingly complicated to design a simple glider that will actually fly (let alone fly well). There are complex constraints between the shapes, positions, and mass of the various plane components, which must obviously be satisfied. Without domain knowledge of advanced aerodynamics, or extensive experimentation, novices (such as the authors) find designing even simple gliders to be quite frustrating. Designing more complicated free-formed gliders is prohibitively difficult for novices because the data and required knowledge for estimating their aerodynamics have not yet been well established.

We introduce a novel computation technique that can model the aerodynamics of free-flight gliders, and then use that model to automatically optimize a glider to maximize the flight-ability in order to aid in the free-form glider design process. This optimization is efficient enough to run in real time, providing guidance when using our interactive glider design tool. Figure 1 illustrates an example of optimization during interactive shape editing. After the user sketches the glider components (wings and fuselage), then our tool optimizes the design so that it will actually fly. This design and optimization cycle can be quickly iterated to reach a shape that is both aesthetically appealing and aerodynamically effective. Finally, the tool outputs a blueprint of the glider, which can be sent to a CNC cutter or 3D printer. The key technical challenge is efficiently estimating the aerodynamics forces, which depend on the shape, orientation,



Figure 2: Example of existing hand launched gliders.

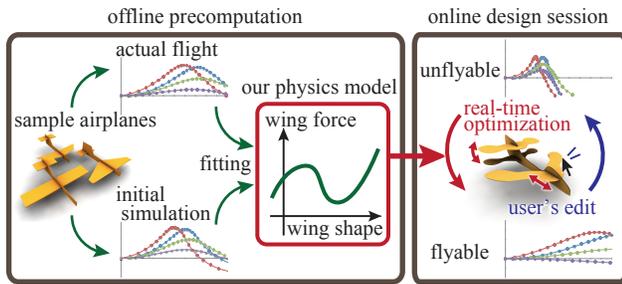


Figure 3: Our overall procedure consists of offline precomputation and online real-time design session. In the precomputation stage, a compact physics model that relates the wing’s shape to its aerodynamics force is represented using radial basis functions (RBFs), in which the parameters are obtained by fitting the trajectories from a physics simulation to the acquired actual flights. This compact physics representation is used in the interactive design session to optimize the design of the glider in real-time.

and velocity of the aircraft. Although existing 3D computational fluid dynamics simulations can provide this analysis, they are much too expensive to compute in real-time.

We present a novel data-driven compact aerodynamics representation that can be acquired very inexpensively through simple experimentation to create a real-time simulation and optimization. Figure 3 shows an overview of our approach. First, we construct a set of actual gliders and take several flight videos for each glider. Then, we optimize the parameters of our aerodynamics model to minimize any deviation between the simulated trajectories and the set of acquired trajectories. Encoding the relationship between the planform (i.e., the wing shape from a top view) and the aerodynamic forces is difficult because the dimensionality of the shape representation is inherently high. We slice the wing into *wing elements* that are parallel to the streamlines to simplify this problem, and define the total glider forces as an aggregate of per-element forces. Our wing elements have a low-dimensional parameterization, and as a result, a data-driven approach to the entire system becomes tractable.

Based on our aerodynamic force model, we can optimize the design of a glider to maximize flight characteristics such as long, stable free flight. We must ensure that the optimized design will fly based on as many initial conditions as possible, since a hand-launched glider could have a wide range of initial velocities and orientations. We propose a novel technique to evaluate the aerodynamic forces without conducting an actual trajectory simulation, which allows for the real-time optimization in our interactive design tool, in order to efficiently consider many launching parameters. We performed a user study to compare the design experience using our tool with that of a traditional design workflow, where the user iterates the design and evaluation several times. Our experiment shows that, using our tool, even novice users can design original functional free-form gliders that are difficult to design using a traditional workflow.

To summarize, our contributions include (1) a physics model that efficiently predicts the aerodynamics forces on arbitrary planforms, (2) a data-driven framework to estimate the parameters of our physics model from the acquired data, (3) an interactive optimization-guided glider design tool.

2 Related Work

Airplane design. The study of aerodynamics rapidly advanced after the Wright brothers’ first successful demonstration of heavier-

than-air flight in the beginning of the 20th century. Modern airplane design methodology is driven by *wing theory*, the study of wing performance under different shapes and conditions. Textbooks such as [Abbott 1959] and [Perkins and Hage 1949] provide good introductions to this topic, but focus on conventional planforms, such as the rectangular, ellipsoid, and swept-back wings seen on virtually all modern aircraft. These simple shapes are empirically known to perform well, and aerodynamic optimization based on computational fluid dynamics has enabled these well-understood wings to be fine-tuned to satisfy various performance goals [Sobieszczanski-Sobieski and Haftka 1997]. However, these elaborate, computationally intensive optimizations generally focus on small variations in the conventional shapes. Our goal is not to optimize the performance, but rather to enable for the creative exploration of airplane shapes while ensuring a basic flying capacity.

Functional fabrication. Since desktop fabrication devices such as 3D printers or CNC cutters have become more widely available, accessible design tools for functional, easily fabricable objects are becoming more important [Schmidt and Ratto 2013]. Advanced computer graphics have assisted recent works in designing more interesting functional objects such as mechanical contraptions [Zhu et al. 2012; Coros et al. 2013; Ceylan et al. 2013], beadwork [Igarashi et al. 2012], pop-up cards [Li et al. 2011]. We assemble our gliders out of the interlocking planar sections, a fabrication technique which has recently been studied in some detail [McCrae et al. 2011; Hildebrand et al. 2012; Schwartzburg and Pauly 2013]. Other recent works have taken into consideration the optimization in the context of a particular fabrication technique, to satisfy user-defined goals [Chen et al. 2013] or enhance the mechanical robustness of 3D-printed objects [Stava et al. 2012; Luo et al. 2012]. In the field of architecture, there are studies to design fabricable structure such as [Panozzo et al. 2013; Song et al. 2013].

Interactive design using physical simulation. Various prior works have incorporated real-time physical simulation into their interactive design tools. Umetani et al.’s system [2011] assists the user in designing physically viable objects using the real-time visualization of simulation results, while Umetani et al.’s system [2012] additionally suggests alternative physically-valid designs when the current design is invalid. In architectural design, Vouga et al. [2012] and Whiting et al. [2012] used design optimization inside the modeling environment to create feasible masonry structures. *Make it Stand* [Prévost et al. 2013] automatically optimizes the shape of a 3D-printed object so that it will be statically balanced when resting on flat ground. Our work applies a similar approach to a different aerodynamics problem domain and also presents a calibration method for acquiring the parameters necessary for simulation.

Measurement-based parameter fitting. Directly simulating complex physical phenomena such as the deformation of the microstructures of heterogeneous material is often prohibitively expensive for computer graphics applications. A standard way to handle this problem, pioneered in computer graphics by Pai et al. [2001], is to propose a simpler physical model and fit its parameters to the empirical data. The frameworks to capture complex physical phenomena using inexpensive experimental setups have been widely studied. Recent works have acquired images of deformed cloth and then used optimization techniques to fit the parameters of their simulation models to this acquired data to replicate the physically-realistic cloth behavior [Wang et al. 2011; Otaduy et al. 2012]. These sorts of approaches can be extended to quite complex situations, such as the history-dependent plastic deformation of clothing [Miguel et al. 2013]. Similarly, Bickel et al. [2009; 2010] have modeled the complex nonlinear deformation behavior

of isotropic, heterogeneous materials based on captured experimental data. In aerodynamics applications, controllable birds' flapping motions are synthesized by capturing their flights and modeling the aerodynamics of their flapping motion [Ju et al. 2013]. In this study, the true airflow around an airplane is too complex to simulate in real-time, we must use a simpler physical representation of the aerodynamic forces, and therefore, also use this approach of measurement-based parameter fitting.

3 User Interface

This section motivates the development of our simulation techniques described in the following sections with a brief description of our interactive glider design tool. Please see the video materials for an interactive demonstration. As seen in the screen capture shown in Figure 4, the glider consists of wings and a fuselage constructed from rigid planar sections of a predefined thickness and material. The user can edit the wings and fuselage by dragging the polyline boundaries using as-rigid-as-possible curve manipulation [Igarashi et al. 2005]. Several simulated flight trajectories (§ 4) are displayed below the shape view using a 2D height/distance plot to show how a glider should fly. We show the possible trajectories for six different angle/velocity launch conditions (velocity 7m/s, 9.0m/s and 11m/s horizontally and 10° vertically) to provide the user with a sense of the flight characteristics of the current design. As the wings and fuselage are manipulated, these trajectories are updated in real-time.

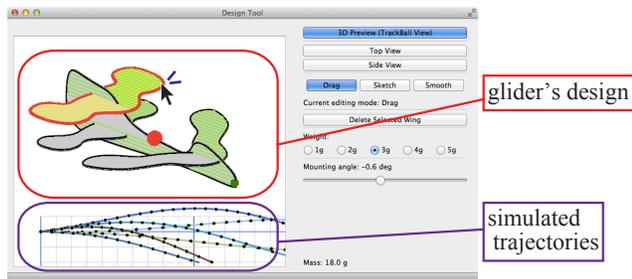


Figure 4: Screen shot of our design tool.

The most difficult task in the design process is adjusting the angle and position of the wings (§ 6), because the aerodynamic effect of changing each parameter can be unintuitive. So, if the current design would not fly, our tool automatically optimizes the mounting angles and positions of the wings without changing their shape. This optimization also happens at interactive rates, and therefore, the design space is effectively constrained to the functional, physically-viable regions. However, from the users' perspective, there is complete freedom to change the design and experiment with corrections that our tool suggests to ensure flyability.

4 Aerodynamic Model

In this section, we explain how free-flights of gliders are simulated given their design. During flight, the airflow interacts with the outer shape of the glider and generates a stress distribution on the glider surface. The aggregate of these stresses produces lift force, making flight possible. In addition to the lift forces, there is drag force that resists the airflow, and torque force, which changes the orientation of the glider. Stable flight is possible only when these aerodynamic forces are in balance. Therefore, glider design involves manipulating the shape to balance the relationship between these forces.

Obtaining aerodynamic forces through 3D fluid simulation is too

expensive for real-time use, so we leverage *wing theory*, which is an empirical method to estimate the aerodynamic forces on a wing. We briefly review wing theory in § 4.1. The original wing theory is limited to simple *planforms* (the wing configuration as viewed from above), and cannot be directly applied to free-form planforms. We extend wing theory to more complex shapes by introducing the *wing element* in § 4.2. In § 4.3, we explain how to simulate glider flight based on these wing forces.

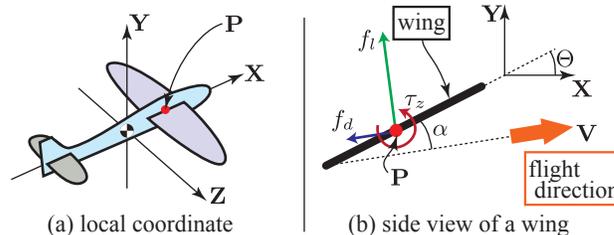


Figure 5: (a) Glider configuration. The center of gravity is placed at the origin, and the XY -plane is the plane of symmetry. (b) Aerodynamic forces on mounting position P : f_d , f_l and torque τ_z .

Glider local coordinates. Most gliders are mirror-symmetric around the fuselage, and we follow this convention. We orient the glider so that the XY -plane is the plane of symmetry, and the glider flies in the direction of the X axis, and Y is the vertical axis (Figure 5-(a)). The origin is placed at the center of gravity of the glider. The rotation of the glider about the Z -axis is called the *pitch*. We assume that the glider is a combination of rigid planar plates, each of which is either a wing or a fuselage. The fuselage is perpendicular to the Z -axis, while the wings rotate around the Z -axis, and can translate within the XY -plane. The angle between the wing and the X -axis is called the *mounting angle* Θ (see Figure 5-(b)).

4.1 Wing Theory

Our approach is based on wing theory, which estimates the aerodynamics forces on a wing using a scaling law and experimental data. This theory is easy to apply for many *airfoils* (the sectional shape of a wing) and planforms, so is commonly used in the initial design stage of flying objects. We can only summarize this theory, see [Abbott 1959] for complete details.

Assumptions. Basic wing theory assumes that there is no interference between the wings and fuselage, thus each wing's force can be independently computed. Of course, the wings generate airflow and pressure distributions that have a global structure, but if the distance between the wings is sufficiently large, this interference is usually not significant.

Similarly, wing theory assumes that the forces on the glider are determined only by the current velocity and orientation (this is the *quasi-static assumption*). During stable flight, the velocity and orientation generally change little and can be ignored. We emphasize that our simulation framework is only suitable for predicting stable flight, where the temporal change in angle and velocity is small. During trajectory simulation, large changes in these values indicate unstable flight, and the simulation results are no longer valid.

Forces on a wing. Since we assume that the glider is perfectly rigid, flight is not affected by the shape of the wing/fuselage connection. Hence we assume that the wing and fuselage are mechanically coupled at a single *mounting point* P , and forces transfer at this point. The mounting points are purely virtual and can be placed

at arbitrary locations. We place \mathbf{P} inside both the plane of symmetry and the plane of the wing.

We assume the wing is moving in still air at a constant velocity \mathbf{V} that has a zero \mathbf{Z} -directional component. Combined with our symmetric plane configuration, the airflow around the wing and its stress distribution are also mirror-symmetric. At the mounting point, the representative force is the integral of the stress distribution on the wing surface. In the symmetric configuration the translational force stays inside the plane of symmetry. Torque at the mounting point is the integral of the cross product of the surface stress and displacement of the surface from the mounting point. The torque has only a \mathbf{Z} -directional component since the other components cancel out.

Therefore, the forces on the mounting point can be described using three components $\mathbf{f} = (f_d, f_l, \tau_z)$, as shown in Figure 5-(b). The first component f_d is the *drag force*, and is in the opposite direction of the velocity. The second component f_l is the *lift force*, which is the perpendicular component of the aerodynamic force. The last component τ_z is the torque around the \mathbf{Z} direction, which is called the *pitching moment*, measured at the mounting point.

These forces on a wing change due to many different parameters, such as the density of air ρ , the wing's speed $V = |\mathbf{V}|$, the wing's shape \mathbf{S} , and the *angle of attack* α , which is the relative pitch angle between the tangential direction of the wing and the wing's velocity \mathbf{V} . We can reduce the dimensionality of these parameters by introducing Reynolds's scaling law [Shevell 1988]. Let us assume that the two-dimensional planform \mathbf{S} is uniformly scaled with a scaling coefficient $L \in \mathbb{R}$ of $\mathbf{S}(L)$. This law states that the stress distribution on a wing's surface is proportional to $\rho V^2/2$ when the Reynolds's number $Re = \rho V L / \mu$ is constant, where the μ is the viscosity coefficient. Aerodynamic forces f_d, f_l are the stress integrated over the wing's surface and thus are proportional to L^2 , and thus to the wing's area A . Torque τ_z is the integral of the cross product between the surface stress and moment arm, thus it is proportional to L^3 , or $L \cdot A$. Therefore, the drag and lift forces can be compactly written as

$$(f_d, f_l, \tau_z)^T = \frac{1}{2} (C_d, C_l, C_m L)^T \rho V^2 A, \quad (1)$$

where C_d is the *drag coefficient*, C_l is the *lift coefficient*, and C_m is the *pitching moment coefficient*. These coefficients are dimensionless and are parameterized with the angle of attack α , Reynolds number Re , and normalized planform $\bar{\mathbf{S}}$ with an area of $A = 1$.

The field of aerodynamics engineering has extensively studied how these coefficients change under different conditions $(\alpha, Re, \bar{\mathbf{S}})$. However, the variation in wing shape $\bar{\mathbf{S}}$ in these analyses is limited to simple planforms such as rectangular, taper, back-sweep, and ellipsoid wings. Typical approaches for obtaining these coefficients for complex planforms are using wind-tunnel or large-scale fluid simulations. In § 5, we describe a simpler approach.

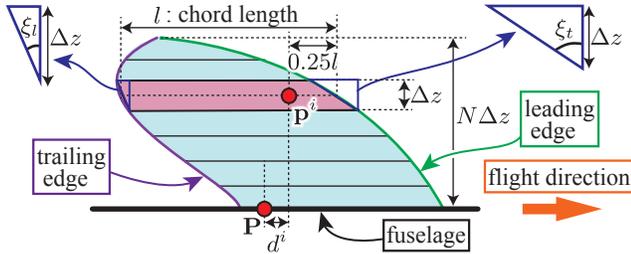


Figure 6: Our wing element discretization of wings.

4.2 Wing Element Discretization

We introduce a new discretization of a wing that is based on the *wing element* (see Figure 6) to efficiently handle a wide range of planforms. In aerodynamics terminology, the upwind edge of a wing is called the *leading edge*, the downwind edge is called the *trailing edge* and the length in between is called the *chord length*. A wing is discretized into wing elements by slicing it with a plane perpendicular to the \mathbf{Z} -axis at a constant interval Δz . The wing element then takes on a trapezoidal shape to approximate the slice. The average horizontal length l of the trapezoid corresponds to the chord length of the wing. The angles at the leading and trailing edges of the trapezoid are denoted as ξ_l and ξ_t .

Our assumption behind the wing element discretization is that the air velocity relative to wings is much faster in the (\mathbf{X}, \mathbf{Y}) direction, compared to that of \mathbf{Z} . Therefore, we can use a simplified model for the shape difference in the \mathbf{Z} -direction. If the shape is sufficiently smooth, the planform of the wing element can be captured using these two angle parameters, ξ_l and ξ_t .

Let wing \mathbf{S} have wing elements \mathbf{s}^i ($i = 1, \dots, N$). For each \mathbf{s}^i , we compute the amount of torque at chosen point \mathbf{p}^i be $0.25l$ downwind from the leading edge. For a rectangular wing, this point is called the *aerodynamic center* and it is empirically known that the pitching moment here is insensitive to the angle of attack [Abbott 1959]. Each wing element contributes drag, lift, and pitching moment to mounting point \mathbf{P} as

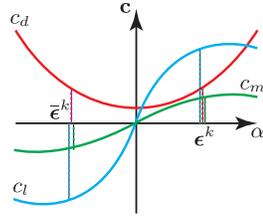
$$\begin{pmatrix} f_d^i \\ f_l^i \\ \tau_z^i \end{pmatrix} = \frac{1}{2} \begin{pmatrix} c_d \\ c_l \\ c_m l^i + c_d d^i \sin \alpha + c_l d^i \cos \alpha \end{pmatrix} \rho V^2 a^i, \quad (2)$$

where a^i is the area of the wing element and d^i is how far \mathbf{p}^i is from the mounting point in the upwind direction. These drag, lift, and pitching moment coefficients $\mathbf{c} = (c_d, c_l, c_m) \in \mathbb{R}^3$ for a wing element are parameterized with a four-dimensional parameter vector $\epsilon = (\alpha, Re, \xi_l, \xi_t) \in \mathbb{R}^4$, where $Re = \rho V l / \mu$ is the Reynolds' number for the element. We assume the use of planar wings, and therefore, the sectional shape is rectangular and thus symmetric, so c_d becomes an even function and c_l, c_m become odd functions, with respect to the angle of attack. These forces are aggregated to compute the total forces at the mounting point.

Our discretization is motivated by the *blade element* [Perkins and Hage 1949] used in propeller design. Propeller blades are decomposed into blade elements, and per-element forces integrated to estimate forces on the entire blade. This technique is common for propeller blades because the sectional shape and angle of attack significantly vary along the blades' span, whereas wings usually have a constant sectional shape and angle of attack. Our wing also has constant sectional shapes and angle of attack, but the element shape along the wing may significantly change. The main difference between the wing and blade elements is that we include parameters ξ_l, ξ_t to capture these planform differences along the span.

Coefficient representation by scattered data interpolation. We take a scattered data interpolation approach obtain coefficient \mathbf{c} for our new wing element model. Let the relationship between input parameter ϵ and output coefficient \mathbf{c} be represented as function $\mathbf{c}(\epsilon) : \mathbb{R}^4 \rightarrow \mathbb{R}^3$. We discretely represent this function by smoothly interpolating the output coefficient value $\mathbf{c}(\epsilon^k) = (c_d^k, c_l^k, c_m^k)$ at several sampling points $\epsilon^k = (\alpha^k, Re^k, \xi_l^k, \xi_t^k)$, where $k \in K$ and K is a set of sampling points. We explain how to construct these discrete sampling sets in § 5.

Following [Bickel et al. 2009], we use Radial Basis Functions (RBFs) for the interpolation between sampling points (i.e., RBF centers). We carefully construct the RBF interpolant so that it automatically enforces that c_d is an even function and c_l, c_m are odd functions, with respect to the angle of attack α .



We enforce this parity by using a ghost sampling point $\bar{\epsilon}^k = (-\alpha^k, \text{Re}^k, \xi_l^k, \xi_t^k)$ for each sampling point ϵ^k that is located symmetric to the hyper plane $\alpha = 0$ in the input parameter space. The output coefficient value of the ghost sampling point is set at $c(\bar{\epsilon}^k) = (c_d^k, -c_l^k, -c_m^k)$. The interpolation becomes

$$\mathbf{c}(\epsilon) = \sum_{k \in K} \mathbf{w}^k \cdot \psi(\|\epsilon - \epsilon^k\|) + \bar{\mathbf{w}}^k \cdot \psi(\|\epsilon - \bar{\epsilon}^k\|), \quad (3)$$

where $\psi : \mathbb{R} \rightarrow \mathbb{R}$ is a scalar kernel function and $\mathbf{w}^k = (w_d^k, w_l^k, w_m^k) \in \mathbb{R}^3$ is the weight for the k -th sampling point. We solve the weights for the drag, lift, and pitching moment coefficients, respectively, using standard RBF techniques. The weights for the ghost sampling points become $\bar{w}_d^k = w_d^k$ for the drag coefficient and $\bar{w}_l^k = -w_l^k, \bar{w}_m^k = -w_m^k$ for the lift and moment coefficients, enforcing the parity of these coefficients.

We use the Gaussian kernel $\psi(r) = \exp(-r^2)$, where the distance r is the square norm of the difference of parameter ϵ . Smoother harmonic kernels $\psi(r) = r$ and $\psi(r) = r^3$ resulted in unstable flight simulations because they can output unrealistically large coefficients when the input parameter is far from the sampling points.

4.3 Free-flight Simulation of Glider

We have thus far illustrated the forces on an individual wing given the discrete sampling set of the relationship between a wing element and the coefficients of aerodynamic force. In this section, we simulate the flights of gliders using these parameterized wing forces. This simulation is used for the trajectory estimation during interactive shape design, as well as for real-time design optimization (§ 6), and our parameter acquisition procedure (§ 5.1).

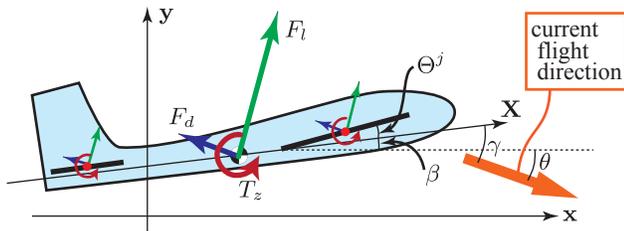


Figure 7: Configuration of forces on glider during flight.

World coordinates. As a glider flies, its position and orientation continuously change. We use 3D world coordinates (x, y, z) with the y axis being the upright, vertical axis to describe this flight trajectory. The glider is launched at time $t = t_0$, and we assume a perfectly symmetric launching condition so that the gliders fly inside the xy -plane. The transformation from the glider local coordinates to the world coordinates can be written as $(x(t), y(t))^T = \mathbf{R}(\beta(t))(X, Y)^T + (p_x(t), p_y(t))$ and $z(t) = Z$, where (X, Y, Z) and (x, y, z) are an identical point represented in local and world coordinates, respectively. \mathbf{R} is a two-dimensional rotation matrix, $\beta \in \mathbb{R}$ is the glider's pitching angle in world coordinates, and

(p_x, p_y) is the location of the center of gravity. The plane flies at the speed $V = \|\dot{\mathbf{p}}(t)\|$ in the direction $\theta(t) = -\tan^{-1}\{\dot{p}_y(t)/\dot{p}_x(t)\}$, where the sign of θ takes on positive values when the glider is descending. Since the glider is flying against still air at a relative angle $\gamma = \theta + \beta$, this is the angle of attack. Note that we set the signs of angles such as θ, β and Θ such that larger angles increase the angle of attack by following tradition of aerodynamics (Figure 7).

Equation of motion. Figure 7 shows the aerodynamic forces acting on a glider. Let \mathbf{S}^j ($j = 1, \dots, \#\text{wing}$) be a wing in the glider. The angle of attack for each wing is $\alpha^j = \gamma + \Theta^j$, where Θ^j is the mounting angle for wing j . We compute the (f_d^j, f_l^j, τ_z^j) forces for wing j using the method described in § 4.1. Let $F_d = \sum_{j=1}^{\#\text{wing}} f_d^j$ be the total drag force generated in the flying direction and $F_l = \sum_{j=1}^{\#\text{wing}} f_l^j$ be the total upward force generated perpendicular to the flying direction. We measure the pitching moment at the glider's center of gravity (i.e., the origin). Since the mounting point and the center of gravity are different, we take the linear forces into account for the total pitching moment $T_z = \sum_{j=1}^{\#\text{wing}} \tau_z^j + \mathbf{P}_x^j f_l^j + \mathbf{P}_y^j f_d^j$, where $(\mathbf{P}_x, \mathbf{P}_y)^T = \mathbf{R}(\gamma)(\mathbf{P}_X, \mathbf{P}_Y)^T$. Finally, the equation of motion can be written as

$$M\ddot{p}_x = F_x, \quad M\ddot{p}_y = F_y - Mg, \quad I\ddot{\beta} = T_z, \quad (4)$$

where $(F_x, F_y)^T = \mathbf{R}(\theta)(F_d, F_l)^T$, M is the mass of the glider, and I is the rotational inertia of the pitching rotation measured at the center of gravity.

The trajectory of a glider is obtained by numerically integrating (4). We use Verlet integration for this time integration. Weißmann et al. [2012] described a more accurate time integration scheme based on the Kirchhoff tensor, but we found a simple Verlet to be sufficient for our application.



Figure 8: Twenty gliders used to train our physics model.

5 Fitting Parameters

In the previous section, we explained how the trajectory of a glider is simulated given the glider's geometry and aerodynamic coefficients $\mathbf{c}(\epsilon)$ of the wing elements. This section describes how to obtain these coefficients by using a data-driven method. These coefficients are computed by constructing a set of training gliders, and then fitting simulated trajectories to those we acquired.

5.1 Data Acquisition

We manually designed a set of 20 gliders, shown in Figure 8, to fit with the physically-valid parameters for our wing element model. The gliders have varying planforms (rectangular, tapered, swept, etc.), which we hoped would result in good coverage of the parameter space ξ_l, ξ_t of the wing element. Flight data cannot be acquired

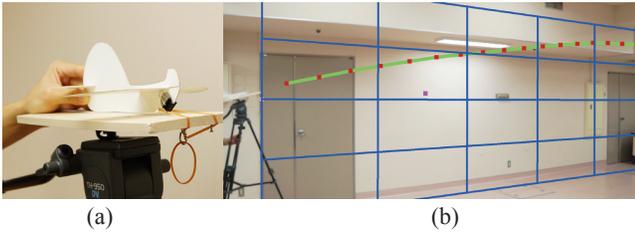


Figure 9: (a) Our launching device and (b) Our flight trajectory tracking interface.

for invalid gliders, so none were included. Note that eliminating invalid gliders does not bias the sampling because we do not selectively exclude specific wing element shapes from sampling.

Glider construction. We constructed our gliders using 1mm thick foam core board, which is a polystyrene sheet clad with paper. This material is light and easily cut using a laser cutter. We added slits to the wings and fuselage so that the planar plates can interlock with each other, and glued small L-shaped plastic braces at the intersections to reinforce the joint. We attached a paperclip of a known weight at the head of the glider to add mass. The density of the board is measured beforehand to compute the moment of inertia from the glider’s geometry and the paperclip’s position.

Launching device. We constructed the simple launcher shown in Figure 9-(a) to produce consistent glider launches. The launch force is provided by a rubber band. During a launch, the glider slides on a rail that is placed on a flat plate attached to a tripod, allowing the launch pitch angle to be adjusted. Note that our optimization takes care of determining the exact launch velocity \mathbf{V}_0 , pitch angle β_0 , and time t_0 , so we do not need to precisely measure these values.

Capturing. We used video captures from the side and back of the launching device to capture the sample glider trajectories. We used off-the-shelf SONY $\alpha 77$ DSLR cameras to capture 1920×1180 video at 30 frames per second. As the gliders should fly within the plane of symmetry, we only needed the side camera to capture the trajectories; the rear camera feed was used to verify this assumption. Several of our gliders did initially veer off-course, but manual symmetrization of the gliders corrected these artifacts.

Trajectory tracking. The camera parameters (angle of view, relative position to launcher, etc.) were determined based on the reference points of an already known configuration. We developed a simple tool in which the user manually annotates the location of the center of gravity in each frame of the video to recover the flight trajectories, as shown in Figure 9-(b). We estimated the acquisition precision to be $\pm 5\text{cm}$. See the supplementary material for the actual captured video and acquired trajectories.

5.2 Parameter Estimation Algorithm

We fit our simulated trajectories to the acquired trajectory to obtain the aerodynamic coefficients of the wing elements. As explained in § 4.2, we model the aerodynamic coefficients of a wing element as an interpolated function $\mathbf{c}(\epsilon)$ that is sampled at discrete parameters ϵ^k . Therefore, we can obtain the function by optimizing the values of the coefficients at the sampling points $\mathbf{c}^k = \mathbf{c}(\epsilon^k)$.

In the previous section, the trajectories of actual gliders were tracked for each launch $l \in L$ as discrete trajectory points $\mathbf{q}_m^l \in \mathbb{R}^2$ $\{m = 1, \dots, N_l\}$ with the camera’s frame rate $\Delta t = 1/30$,

where L is the set of launches across all the training gliders. The number of tracked point N_l for each launch generally ranged from 15 to 20. However, the exact launch conditions were difficult to measure using our simple launcher. Therefore, we also obtained (i) the launch conditions (velocity, orientation, and timing) from the optimization together with (ii) the aerodynamic coefficients of the wing elements. These two optimizations are coupled; if we change the aerodynamic coefficients of the wing elements the optimized launch condition estimates will change, and vice versa. So, we interleave these two optimizations to simultaneously optimize both sets of parameters.

For each launch $l \in L$, the launch parameters $\sigma_l = (V_0, \theta_0, \beta_0, t_0)$ are estimated by fitting the initial trajectory points \mathbf{q}_i :

$$\arg \min_{\sigma_l} \sum_{m=1}^M \|\mathbf{p}^l(t_0 + m\Delta t) - \mathbf{q}_m^l\|^2, \quad (5)$$

where $\mathbf{p}^l(t_0 + m\Delta t)$ denotes the simulated glider position after time $t = t_0 + m\Delta t$ and we chose $M = 10$. The optimization of (5) starts from a guess where V_0 and β_0 are computed from two points right after the launch, t_0 is zero, and θ_0 is equal to β_0 .

Given the initial launch parameters, the coefficients at the sampling points \mathbf{c}^k are optimized to fit the trajectory for all the acquired launches, across all the training glider shapes.

$$\arg \min_{\mathbf{c}^k, k \in K} E, \quad E = \sum_{l \in L} \sum_{m=1}^{N_l} \|\mathbf{p}^l(t_0 + m\Delta t) - \mathbf{q}_m^l\|^2. \quad (6)$$

The two optimization problems in (5) and (6) are solved using the gradient descent method. The gradients were computed using the forward-differencing, where E is computed at two different parameters with an empirically chosen difference. Even with this simple approach, we observed a rapid convergence to the optimal parameters for both the launch conditions and wing element parameters.

One remaining question is how to obtain the sample points ϵ^k . We used a greedy method created by [Bickel et al. 2009]. We start with a single point ϵ^0 that has the averaged parameters of the wing elements obtained for all the training gliders. We find the launch conditions and wing element coefficients for this single sampling point using the above optimization. After convergence, we add the next sample point ϵ^k at the location where the largest difference between the simulated and acquired trajectories occurs. We thus reduce the approximation error until it converges by iteratively adding the sampling points in this fashion.

6 Design Optimization

In the previous sections, we explained how to model the physics of a glider to simulate its flying trajectories. We now illustrate our design optimization based on the physics model. The optimization changes the mounting angle Θ^j and the mounting positions \mathbf{P}_X^j and \mathbf{P}_Y^j for all the wings. Therefore, the optimization searches the design in a $3 \times (\#wing)$ -dimensional design space. The launching conditions such as the glider’s velocity and pitch angle for a hand launch are different for each launch. So, we optimize the glider’s performance for a certain range of velocities and orientations, instead of optimizing performance for a specific launching condition.

Characteristic function of glider. In § 4.3, the glider’s dynamics were simulated from the forces of the wing. We denoted function $\mathcal{F}(V, \gamma) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, which maps the glider’s speed V and angle of attack γ to (F_d, F_l, T_z) . With this function, from (4), a

glider's flight path can be computed given the launching configuration. Therefore, we assume that this function characterizes the aerodynamics property of the entire glider. Since we use hand launching to fly the glider, the initial velocity and initial angle varies a lot. So, we take into account the region of interest in the input parameters $\Omega = [V_{\min}, V_{\max}] \times [\gamma_{\min}, \gamma_{\max}]$, which is the rectangular region of the range of speed and angle of attack, and then try to optimize the overall glider's performance in this region instead of taking into consideration the specific launching configurations. We chose $V_{\min} = 6\text{m/s}$, $V_{\max} = 9\text{m/s}$, $\gamma_{\min} = -10^\circ$, and $\gamma_{\max} = 10^\circ$, which are typical in hand launching.

What property in this characteristic function \mathcal{F} in region Ω makes a better glider? The most important observation is that when a glider flies stably the pitching moment approaches zero: $T_z = 0$. This is because the aerodynamic forces F_d, F_l are very sensitive to the glider's angle of attack, and if the pitching moment constantly has a large value, it quickly changes the angle of attack and then the forces are difficult to balance. We constructed a cost function to evaluate a glider's performance based on this observation. The cost function consists of three different performance evaluations: (i) pitch stability, (ii) vertical force balance, and (iii) less drag. Each of these three properties are represented as a cost function and the overall cost function is the sum of them. We respectively explain these properties in the following paragraphs.

Pitch stability. Pitch stability means the pitching angle is stable under a small perturbation. This is the most important property for a long and stable flight. Without pitch stability, the glider rapidly changes its pitching angle and angle of attack, resulting in a sharp plummet or a quick flip after launch. The pitch stability is achieved by producing restoring torque to make the deviation small when the angle of attack deviates from the angle that makes $T_z = 0$. That is, a small angle of attack should produce a positive pitching moment and large angle of attack should produce a negative pitching moment. In the Ω region, this property is enforced if the angle of attack is too small $\gamma = \gamma_{\min}$, then there is a positive pitching moment $T_z(V, \gamma_{\min}) > 0$, or if the angle of attack is too large $\gamma = \gamma_{\max}$, then there is a negative pitching moment (see Figure 10). This can be represented with the cost function:

$$C_1 = \int_{V_{\min}}^{V_{\max}} T_z^-(V, \gamma_{\min}) + T_z^+(V, \gamma_{\max}) dV, \quad (7)$$

where $T_z^-(V, \gamma)$ takes $-T_z(V, \gamma)$ if the value T_z is negative and takes on a zero if the value is positive. In the same manner, $T_z^+(V, \gamma)$ takes on $T_z(V, \gamma)$ if the value T_z is negative and zero if positive.

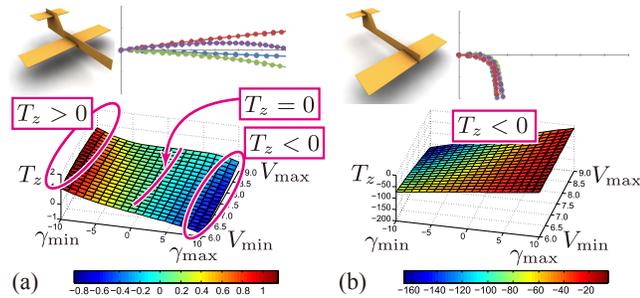


Figure 10: (a) Stable and (b) unstable glider design.

Vertical force balance. Pitch stability ensures that T_z is near-zero and the glider flies stably. However, with insufficient lift forces the glider quickly descends, but with too much lift force the glider

ascends and the possibility of a stall increases. Therefore, the lift force should be equal to the gravitational force in the given range. The cost function is described as

$$C_2 = \int_{V_{\min}}^{V_{\max}} \|F_l(V, \gamma) - Mg\|^2 dV, \quad \text{where } T_z(V, \gamma) = 0. \quad (8)$$

Note that the actual vertical aerodynamic force is not F_l but $F_y = F_l \cos \theta + F_d \sin \theta$, where θ is the defensing angle (§ 4.3). However, if the glider flies horizontally the defensing angle becomes small and the difference between the lift and vertical forces also becomes small.

Less drag. As the glider flies, the drag force pulls the glider backward and the speed attenuates. If the drag force is small the glider flies longer. Therefore, we designed the cost function as

$$C_3 = \int_{V_{\min}}^{V_{\max}} \|F_d(V, \gamma)\|^2 dV, \quad \text{where } T_z(V, \gamma) = 0 \quad (9)$$

The *lift-drag ratio* and is frequently used for evaluating an airplane's quasi-static performance, where all the forces are in balance and the lift and drag forces do not change. On the other hand, during actual glider flight the lift and drag forces constantly change, and it is difficult to directly apply this lift-drag-ratio. However, with a small C_2 , the lift and gravity forces are similar. So, minimizing C_3 and C_2 together means the overall lift-drag ratio is improved.

Minimizing cost function. The second and third cost functions are difficult to compute because they are the integrations of non-linear function \mathcal{F} over the line, where the pair of V and γ make $T_z = 0$. Therefore, we discretely sample the values of the forces inside Ω and interpolate them to make it easier to handle. We use one-dimensional interpolation; we sample several fixed velocities, and then with that given velocity, we linearly interpolate the value of the forces that are sampled with different γ .

We set up a total cost function as $C(\mathcal{F}) = w_1 C_1 + w_2 C_2 + C_3$, where we choose $w_1 = 100$ and $w_2 = 1$. Note that the first cost function is the most important one for the plane's flight, so we choose a large number. We minimize this cost function online during editing using the steepest descent method. We use the numerical differentiation to compute gradients of the total cost function with respect to design parameters: the wing positions, mounting angles.

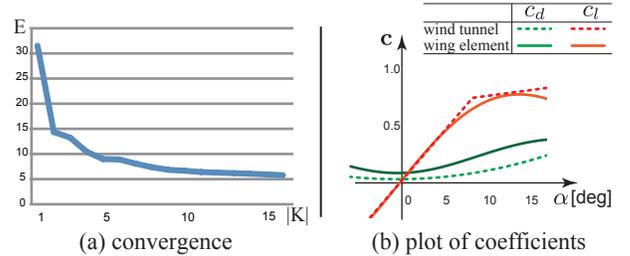


Figure 11: (a) Convergence of fitting error E with increasing number of sampling points and (b) change of drag and lift coefficients with respect to angle of attack α for wind tunnel data and the wing element model.

7 Results

7.1 Validation of Aerodynamics Model

We introduced *wing elements* to represent the aerodynamic properties of free-form wings. The parameters of our wing element

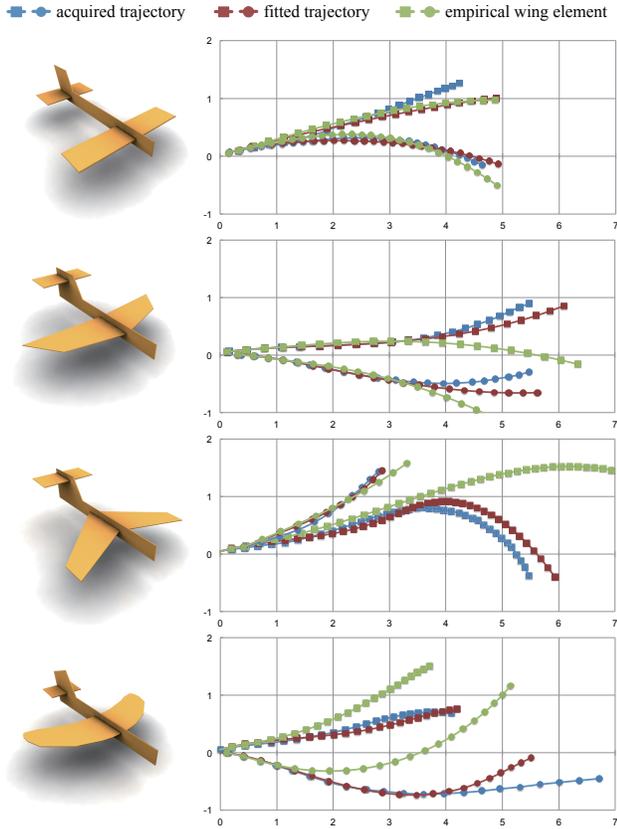


Figure 12: Simulated trajectories (red) produced after fitting our wing element model to the acquired trajectories (blue). Green trajectories shows simulation using empirical wing elements.

model are learned from empirical trajectory data, which consists of 113 flights of the 20 gliders. Figure 11-(a) plots the differences between the measured trajectories and the simulated trajectories resulting from the parameter-fitting procedure illustrated in § 5.2. We observe that as the number of sampling points increases, the error steadily decreases, until approximate convergence. In all further experiments we use 10 RBF centers.

Figure 11-(b) compares the aerodynamic lift and drag coefficients of a rectangular wing (modeled by our wing element with $\xi_l = 0$, $\xi_t = 0$) and experimental wind-tunnel data for a planar wing with 30mm chord length and 0.3mm thickness, taken from [Abbott 1959]. We choose this experimental data because the Reynolds’ number is similar to our problem ($Re \simeq 1.0 \times 10^4$). The coefficient data obtained from wind-tunnel tests with this Reynolds’ number is frequently used for construction of gliders with typical wing shapes. Since the airfoils are different, these plots should not be identical, but the similarities show that our obtained model captures basic traits of the aerodynamics coefficients, such as that the lift coefficient stops increasing when the angle of attack is around 10° (This phenomena is called *stall*). The drag coefficient is larger than in the wind tunnel data because our wing is thicker.

Figure 12 shows a comparison of the recorded flight and simulation results using the parameters obtained by using (5) and (6). In the figure, we also show a trajectory simulated with a method similar to the blade element, discretizing it with rectangular wing elements, and using the empirical relationship between the aerodynamic coefficients and attacking angle from Figure 11-(b). We refer to this

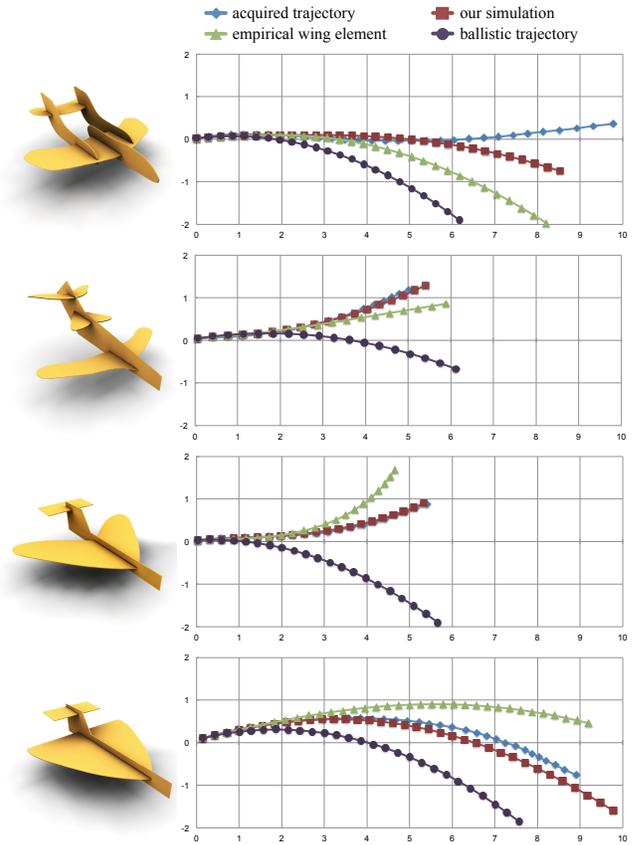


Figure 13: Trajectory plots produced from our acquired data, our simulation, simulation with the empirical wing element, and ballistic motion, for complex glider designs that are not included in the training data.

naïve approach as the *empirical wing element*. When the wing shape is close to rectangular (e.g., Fig. 12-first row), the trajectories simulated with our wing element and the empirical wing element produce similar trajectories, however if the wing element angles are large (e.g., the forward-swept wing glider in Fig. 12-third row), our approach produces a trajectory that better fits the acquired trajectory. These results, and those for our set of training gliders found in the supplemental materials, show that our simple wing element model is capable of reproducing many real glider flights.

We validated our wing element by simulating the trajectories of four complex gliders that were not included in the training data. Fig. 13 compares the acquired trajectories with those generated by conducting a simulation using our wing element and the empirical wing element. We also show the ballistic trajectories that were computed when there were no aerodynamics forces. The trajectories from our more advanced wing element show a closer approximation to the experimental data than those based on the empirical wing element. Fig. 14 shows more complex gliders that are created with our system. They each fly more than 10m with casual hand launches. Please see the accompanying video for their flight examples.

7.2 User Study

We performed an informal user study to demonstrate the effectiveness of our design tool. Our goal was to examine how significant an impact our automatic flyability correction had on the glider design process compared to traditional design/test iterations. Four com-



Figure 14: Airplanes designed with our system.

puter science students (A,B,C,D) participated in the study. They were familiar with 3D modeling but had no prior experience or knowledge with airplane design. We tested two different glider design environments. In the first environment, the participants designed gliders in a conventional way, without continuous feedback from the simulation and optimization. Since actually making the glider and observing its flight pattern would be very time consuming, instead the participants could preview the simulated flight results, but only three times per task, mimicking design/test iterations. In the second environment, the user designed gliders with our real-time optimization enabled, which automatically corrected their designs to ensure flyability. Each participant designed a different original glider in each design environment. The participants were divided into two groups; participants A and B tried the conventional design environment first, and then designed with our flyability correction enabled. Participants C and D completed the same tasks in the reverse order.

Each subject was given 15 minutes to design an original glider that could fly more than 5m horizontally when launched from a 1.5m height. The eight resulting gliders were constructed and tested using the launching device described in § 5.1, and we used a radar gun (D&M Pocket Radar™) to verify the actual launching velocity. We collected 10 flight distances for each plane, with the launching speeds ranging from 7 to 9m/s.

Figure 15 shows the designed gliders and their flight distances. All the participants used our optimization functionality frequently, to try to improve flight distances both during exploration and also while finalizing the design. With the traditional design/test iterative process, the plane designs not only stayed closer to the conventional glider shapes, but many of the planes also failed to fly long distances. With our optimization-guidance enabled, the subjects explored more design variations, and also achieved longer flights.

	A	B	C	D
Trad.				
max [m]	3.3	3.3	4.5	3.2
med [m]	2.2	2.0	2.6	2.4
Sim. + Opt.				
max [m]	8.5	(4.2)	9.8	11.0
med [m]	6.3	(3.3)	8.6	8.5

Figure 15: Result of the user study. The top row (Trad.) shows the user’s glider designs with traditional design/test iteration. The bottom row (Sim.+Opt.) shows designs with our tool. med and max stands for the maximum and median flight distances of 10 flights. Participant B’s design was unstable due to Y-axis rotation (discussed in the next section).

8 Discussion

We presented an interactive glider design tool that uses a real-time aerodynamics-driven optimization to constrain the design space to shape configurations that are likely to actually fly. To do so, we introduced a compact wing element model that allowed us to simulate the aerodynamics of free-form wings, and recovered physically-valid simulation parameters from the captured trajectories of a set of sample gliders. Our analysis results show that this simplified model is effective, and our user study shows that novice users can easily design novel flyable gliders.

We obtained a reasonable physics model by conducting inexpensive experiments, and this required only a video camera and many different sample gliders. This approach was much simpler and more accessible than wind tunnel experiments or expensive CFD simulations, and could be widely deployed in various design tasks. Our methodology is applicable under more complex conditions such as variable airfoil shapes and different flying conditions, and may even be useful for other functional design tasks such as for kites, windmills, submarines, or sailing boats.

Our techniques are suitable for designing hobby-grade hand-launched gliders with creative shapes which can be constructed easily and inexpensively. However, our techniques would need extensive validation to be applied to the design of human-scale manned gliders, where aerodynamic performance is more critical. We believe our prototype system could still be useful in those contexts as a tool for brainstorming or education.

Construction using other material.

Aerodynamic forces depend on the outer shape of the glider, and therefore, we can use the same optimized simulation parameters for different materials if the thickness remains consistent and the density is known. The right figure shows a glider constructed from laser-cut acrylic resin board. The acrylic resin plane flies roughly 9.5m at a horizontal launch speed of 8m/s. See the accompanying video for more detail.



Limitations and future work. In the glider design tool, we only compute aerodynamic forces on a wing, ignoring contributions from the fuselage. This is because if a glider flies in the plane of symmetry, the airflow around the thin fuselage results in very small aerodynamic forces. However, fuselage shape does affect the rotational stability about the Y-axis (i.e., yaw stability). If the front part of the fuselage is significantly larger than the rear portion, the glider flips about the Y-axis and falls down. For example, in the user study, the glider created by user B in Figure 15 has a large head compared to its tail, thus it did not fly stably. Although it is not observed in our study, the same rotation instability can happen for the X-axis (i.e., roll instability) when the upper part of a glider is significantly larger than the lower part. Furthermore, guaranteeing stability in the presence of imperfect symmetry in construction and launching condition is an important future work.

Our aerodynamic model cannot handle the interference from multiple wing elements in a streamline (see the right



figure), in these cases the system gives unreliable results. Our tool is limited to symmetric glider designs assembled from thin rigid plates. Symmetric wings simplify the problem because there are only three independent component forces acting on the wing, and

the planar components simplify both the fabrication and aerodynamic analysis. Using a thin plate for the fuselage results in lower drag forces, and at the scale and velocity of hand-launched gliders, the performance of the planar wings is comparable to more complex airfoils. However, we would like to use sculpted 3D shapes to increase the users' design freedom. This will involve modeling and learning the aerodynamic coefficients of a 3D asymmetric object, which is a challenging problem to address in our future work.

Another potential direction is increasing the controllability. We currently optimize for gliders that fly straight and flat within the plane of symmetry. A more complex goal behavior, such as for a glider that flies along a designed trajectory or stays airborne for a maximum amount of time, is an interesting possibility. Moreover, thanks to the recent advances in battery technology, micro-motors, and wireless radios, it is possible to provide thrust and control even for small, lightweight gliders. These are interesting venues to explore in the future.

Acknowledgements. We thank the anonymous reviewers and Kenichi Rinoie for their comments. We appreciate the assistance of Daisuke Sakamoto, Yutaro Hiraoka and Junghyun Kim in our experiments. Yuki Koyama is funded by JSPS research fellowship.

References

- ABBOTT, I. H. 1959. *Theory of Wing Sections: Including a Summary of Airfoil Data*. Dover Publications.
- BICKEL, B., BÄCHER, M., OTADUY, M. A., MATUSIK, W., PFISTER, H., AND GROSS, M. 2009. Capture and modeling of non-linear heterogeneous soft tissue. *ACM TOG* 28, 3.
- BICKEL, B., BÄCHER, M., OTADUY, M. A., LEE, H. R., PFISTER, H., GROSS, M., AND MATUSIK, W. 2010. Design and fabrication of materials with desired deformation behavior. *ACM TOG* 29, 4.
- CEYLAN, D., LI, W., MITRA, N. J., AGRAWALA, M., AND PAULY, M. 2013. Designing and fabricating mechanical automata from mocap sequences. *ACM TOG* 32, 6.
- CHEN, D., LEVIN, D. I. W., DIDYK, P., SITHI-AMORN, P., AND MATUSIK, W. 2013. Spec2Fab: A reducer-tuner model for translating specifications to 3D prints. *ACM TOG* 32, 4.
- COROS, S., THOMASZEWSKI, B., NORIS, G., SUEDA, S., FORBERG, M., SUMNER, R. W., MATUSIK, W., AND BICKEL, B. 2013. Computational design of mechanical characters. *ACM TOG* 32, 4.
- HILDEBRAND, K., BICKEL, B., AND ALEXA, M. 2012. Crdbrd: Shape fabrication by sliding planar slices. *CGF* 31.
- IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM TOG* 24, 3.
- IGARASHI, Y., IGARASHI, T., AND MITANI, J. 2012. Beady: Interactive beadwork design and construction. *ACM TOG* 31, 4.
- JU, E., WON, J., LEE, J., CHOI, B., NOH, J., AND CHOI, M. G. 2013. Data-driven control of flapping flight. *ACM TOG* 32, 5.
- LI, X.-Y., JU, T., GU, Y., AND HU, S.-M. 2011. A geometric study of v-style pop-ups: Theories and algorithms. *ACM TOG* 30, 4.
- LUO, L., BARAN, I., RUSINKIEWICZ, S., AND MATUSIK, W. 2012. Chopper: Partitioning models into 3D-printable parts. *ACM TOG* 31, 6.
- MCCRAE, J., SINGH, K., AND MITRA, N. J. 2011. Slices: A shape-proxy based on planar sections. *ACM TOG* 30, 6.
- MIGUEL, E., TAMSTORF, R., BRADLEY, D., SCHVARTZMAN, S. C., THOMASZEWSKI, B., BICKEL, B., MATUSIK, W., MARSCHNER, S., AND OTADUY, M. A. 2013. Modeling and estimation of internal friction in cloth. *ACM TOG* 32, 6.
- OTADUY, M. A., BICKEL, B., BRADLEY, D., AND WANG, H. 2012. Data-driven simulation methods in computer graphics: Cloth, tissue and faces. In *SIGGRAPH Courses*.
- PAI, D. K., DOEL, K. V. D., JAMES, D. L., LANG, J., LLOYD, J. E., RICHMOND, J. L., AND YAU, S. H. 2001. Scanning physical interaction behavior of 3D objects. In *Proc. SIGGRAPH*.
- PANOZZO, D., BLOCK, P., AND SORKINE-HORNUNG, O. 2013. Designing unreinforced masonry models. *ACM TOG* 32, 4.
- PERKINS, C. D., AND HAGE, R. E. 1949. *Airplane Performance, Stability and Control*, 1 ed. Wiley, 1.
- PRÉVOST, R., WHITING, E., LEFEBVRE, S., AND SORKINE-HORNUNG, O. 2013. Make it stand: Balancing shapes for 3D fabrication. *ACM TOG* 32, 4.
- SCHMIDT, R., AND RATTO, M. 2013. Design-to-fabricate: Maker hardware requires maker software. *IEEE CG&A* 33, 6.
- SCHWARTZBURG, Y., AND PAULY, M. 2013. Fabrication-aware design with intersecting planar pieces. *CGF* 32, 2.
- SHEVELL, R. S. 1988. *Fundamentals of Flight (2nd Edition)*, 2 ed. Prentice Hall, 8.
- SOBIESZCZANSKI-SOBIESKI, J., AND HAFTKA, R. T. 1997. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural optimization* 14, 1.
- SONG, P., FU, C.-W., GOSWAMI, P., ZHENG, J., MITRA, N. J., AND COHEN-OR, D. 2013. Reciprocal frame structures made easy. *ACM TOG* 32, 4.
- STAVA, O., VANEK, J., BENES, B., CARR, N., AND MĚCH, R. 2012. Stress relief: Improving structural strength of 3D printable objects. *ACM TOG* 31, 4.
- UMETANI, N., KAUFMAN, D. M., IGARASHI, T., AND GRINSPUN, E. 2011. Sensitive couture for interactive garment modeling and editing. *ACM TOG* 30, 4.
- UMETANI, N., IGARASHI, T., AND MITRA, N. J. 2012. Guided exploration of physically valid shapes for furniture design. *ACM TOG* 31, 4.
- VOUGA, E., HÖBINGER, M., WALLNER, J., AND POTTMANN, H. 2012. Design of self-supporting surfaces. *ACM TOG* 31, 4.
- WANG, H., O'BRIEN, J. F., AND RAMAMOORTHY, R. 2011. Data-driven elastic models for cloth: Modeling and measurement. *ACM TOG* 30, 4.
- WEISSMANN, S., AND PINKALL, U. 2012. Underwater rigid body dynamics. *ACM TOG* 31, 4.
- WHITING, E., SHIN, H., WANG, R., OCHSENDORF, J., AND DURAND, F. 2012. Structural optimization of 3D masonry buildings. *ACM TOG* 31, 6.
- ZHU, L., XU, W., SNYDER, J., LIU, Y., WANG, G., AND GUO, B. 2012. Motion-guided mechanical toy modeling. *ACM TOG* 31, 6.