Parameterization and Geometric Optimization of Balloon Launched Sensorcraft for Atmospheric Research Missions

Christopher A. Paulson* and András Sóbester†

University of Southampton, Southampton, England SO17 1BJ, United Kingdom

We present a method for the payload centric automated design and manufacturing of balloon launched, high altitude gliders. The purpose of these gliders is to conduct directed measurements of atmospheric phenomena with a variety of payloads. A bespoke airframe design is generated that can protect the payload, ensure recoverability and extend sampling times. A manufacturing technique, that relies heavily on rapid prototyping, allows for rapid realization of the aircraft design. This allows atmospheric scientists and researchers unprecedented access to a broad range of altitudes.

Nomenclature

$ASTRA$	Atmospheric Science Through Robotic Aircraft
SLS	Selective Laser Sintering
SLA	Stereolithography
FDM	Fused Deposition Modeling
P	Power, Watts (W)
t	Time, seconds (s)
M	Mass, Kilograms (kg)
$COTS$	Commercial Off the Shelf
Subscript	
i	Variable number

I. Introduction

A. Atmospheric Science Through Robotic Aircraft (ASTRA)

The University of Southampton ASTRA project seeks to investigate new technologies for enabling low cost observations of atmospheric phenomena. This is done using a variety of development platforms, from radiosondes to robotic aircraft, operating up to $100,000$ ft.

The ASTRA program aims to deliver scientific payloads to extreme altitudes at very low cost.

*Postgraduate Research Student, Computational Engineering and Design Group, AIAA Student Member.

†Lecturer, Computational Engineering and Design Group, AIAA member.

Potential applications for this technology include measuring atmospheric pollutants, mapping particulate concentrations and exploring extreme weather phenomena.

To enable low cost missions, ASTRA uses consumer grade electronics, low cost manufacturing, rapid prototyping and rapid prototyped electronics. Additionally, automated design and automated flight planning algorithms are being developed to enhance the functionality of the radiosondes and aircraft developed for the ASTRA project.[4]

Balloon Trajectory Modeling

An important component of balloon launched systems is an accurate balloon trajectory prediction model. As part of the ASTRA program, a high accuracy Monte Carlo balloon flight simulation capability has been developed and tested[2].

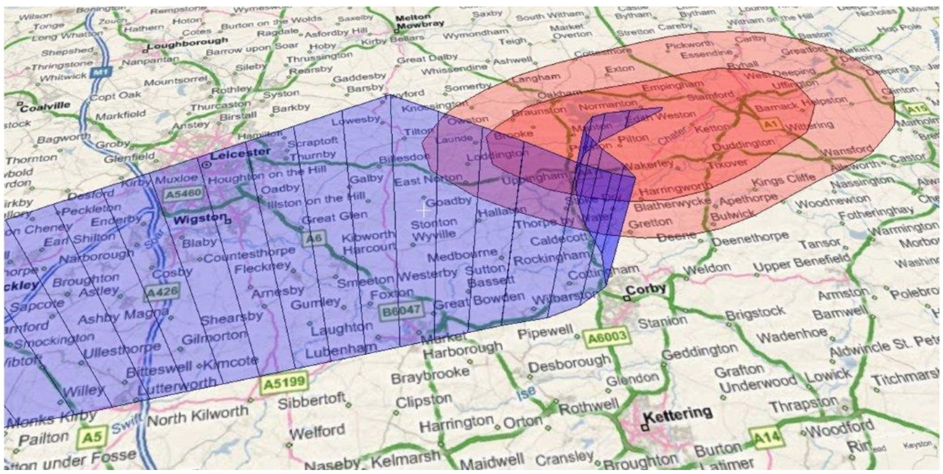


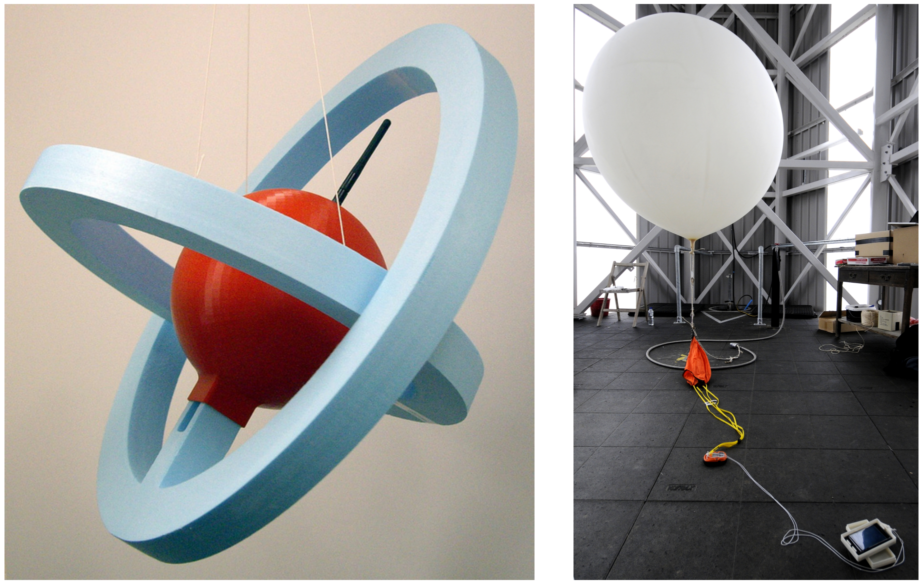
Figure 1. This figure contains the output of the Monte Carlo based balloon trajectory program for a balloon payload with a parachute descent. The red contours mapped on the ground are landing probability contours.

Radiosonde Development

Hundreds of radiosondes are launched daily. These sondes measure and collect data for meteorological models. The ASTRA program has investigated the deployment of consumer electronics based radiosondes to allow for measuring of extended atmospheric properties. Figure 2 shows two such balloon payloads. The radiosondes shown in both images carry Windows 7 Phones and the Microsoft .NET Gadgeteer rapid electronic prototyping environment. This combination allows an end user to rapidly deploy sensors, including a wide variety of existing COTS sensors, process data onboard and use a wide variety of data transmission methods to relay the collected data to a ground control station. In addition, these technologies improve the chances of recovery, as more position information can be transmitted near the ground over cellular communications.

High Altitude Glider Development

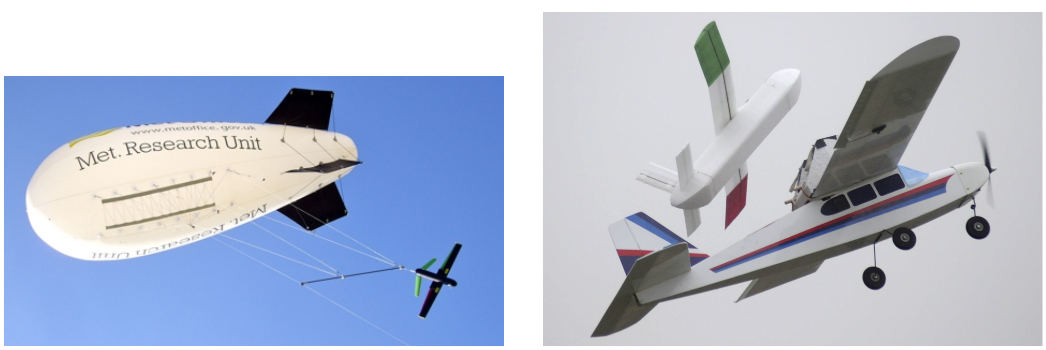
Gliders are now being developed as an extension of the radiosonde research. They offer several advantages when compared to traditional radiosondes. The most signifiant difference is the ability to actively manipulate trajectory. This allows for better sampling of airspace and improved recoverability. Ideally, this glider would be capable of a return to home flight mode, in which,



(a) (b)

Figure 2. Radiosonde related research: Both of these radiosondes use Microsoft Windows 7 smartphones for data acquisition, storage and communication. This is an easily expandable data acquisition system capable of loading data directly to the internet once in cellular datalink range.

the aircraft returns to the point of launch after its sampling mission is completed. If the return to home feature is not an option, alternate recovery sites are used. Figure 3 shows two such gliders developed previously as part of the ASTRA initiative.



(a) (b)

Figure 3. ASTRA atmospheric research sensorcraft on test. Test launches from static, thethered balloons (left) or radio-controlled 'mother ships' (right) reduce development cost (compared to multiple free balloon launches).

B. Rapid Prototyping

A rapid prototyping technology is any technology that provides a high level abstraction to a sophisticated process. The authors feel that this type of technology enables radical new design and manufacturing approaches to solving difficult problems. Three categories of rapid prototyping are used in this research: object oriented programming, additive manufacturing and modular electronics.

Programming Languages

Object oriented languages allow for the quick prototyping of ideas and provide simplified access to a wide variety of existing tools. A majority of the design code described in Section II. is written in the Python language. This allows for easy access to existing geometry codes, system commands, modeling tools and high-performance computing tools. Additionally, testing new approaches takes hours rather than days. The modularity of object oriented languages allows for on-the-fly selection of packaging, geometry design and optimization codes. This greatly expands the system's ability to design for unique situations and to provide access to the most up-to-date algorithms available.

Object-oriented programming also allows for the rapid development of mission payload firmware. The .NET Gadgeteer platform uses C# to simplify development for high-performance code for ARM processors. This simplifies tasks like sensor reading, data processing, data storage and data transmission over wide variety of wireless interfaces. In addition, it simplifies the compiling and flashing process of the modules. Changes the to the mission sampling firmware can be made by scientists and engineers without requiring highly specialized knowledge.

Additive Manufacturing

Additive Manufacturing, or 3D printing, enables a paradigm shift in manufacturing. Design complexity comes at little or no cost. Low cost 3D printers will soon become common. One goal of the ASTRA project is to create content that can be printed on these printers, enabling scientific missions that are, at the moment, severely limited by cost constraints.

The authors' research group has previously demonstrated that entire aircraft can be 3D printed using an SLS process. The aircraft was entirely 3D printed, including all fasteners, control surfaces and hinges. The aircraft preformed well in flight testing, however, demonstrated that the material set current available in 3D printers is not yet adequate for high altitude gliders. The primary material deficiencies are cost and weight.

Electronics

The ASTRA project primarily uses the .NET Gadgeteer electronics development framework for data acquisition systems. The modular motherboards allow for quick development of entire systems, without the extra burden of PCB design, layout, manufacturing, assembly and programming. Many sensors have already been integrated with the platform, and most have been released under an Open Source Hardware License.

The system is comprised of a main board that holds the processor. This board typically has an ARM processor running at clock speeds between 72 MHz and 240 MHz. Several standardized connectors around the processor allow easy peripheral access.

Modules are connected to these peripheral access ports. The user then draws an equivalent digital connection in the .NET Gadgeteer integrated development environment. Once the sensor layout is complete, the .NET Gadgeteer IDE produces skeleton code. A user simply uses object oriented programming to define data flow, rate, storage and transmission. The program is flashed to the processor over a USB link.

II. Design Process

This section will provide an overview of the end to end process of creating a bespoke sensorcraft. The details for each of the process steps will be explained in greater depth below. The overall design process is shown in Figure 4.

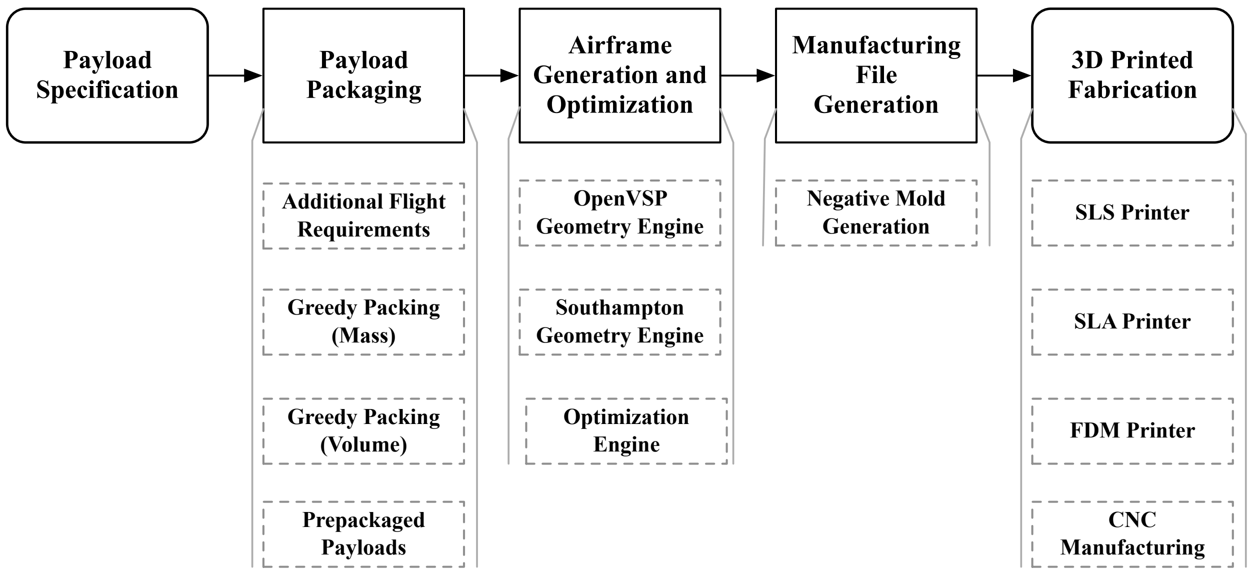


Figure 4. A graphical representation of the design process uses to create bespoke sensorcraft. The dashed boxes indicate different modules that can be used in the development process.

A. Payload Specification

The design process is initiated by uploading CAD models of the desired payload using a simple webpage form. Once the upload is complete, the bounding box of each object is extracted from the model data. A second, dynamically generated website is then shown to the user. On this page, additional information about each of the payload items is entered. A fully defined payload item must include the following information:

1. Valid CAD Model
2. Object Mass
3. Object Dimensions
4. Power Consumption
5. Operating Voltage
6. Object Quantity

Additionally, the second webpage allows for the description of the flight mission requirements. This includes flight time, release altitude, range and specific regions of interest.

B. Payload Packaging

After the user has input all required data for the payload, the design algorithm takes control. The first step is to add additional hardware that is required for control of the aircraft. These items added include the flight control system, actuators, GPS, inertial sensors, power conversion electronics,

Item	Mass (g)	Dimensions			Voltage (V)	P (W)	Dynamically Generated
		x (mm)	y (mm)	z (mm)			
Sensor 1	10	20	20	15	5	0.5	No
Sensor 2	10	20	20	15	5	0.5	No
Sensor 3	10	20	20	15	5	0.5	No
Autopilot	8	36	50	8	5	0.1	No
GPS (w/ antenna)	9.3	38	38	7.8	5	0.25	No
433 MHz Radio	6.5	32	12	5.5	5	0.112	No
Zigbee Radio	8	33	25	3	3.3	0.6	No
Gadgeteer	8	60	55	8	5	0.5	No
Servo 1	9	23	26	12	5	0.5	No
Servo 2	9	23	26	12	5	0.5	No
Battery							Yes

Table 1. An example glider payload. Only Sensor 1, Sensor 2 and Sensor 3 were added by the user. All other payload items were automatically populated by the design algorithm.

communication modems, batteries and possibly a propulsion system. Table 1 lists a full payload for a glider.

Certain items added to the payload, such as batteries, are specific to each payload and mission. These are called dynamically generated payload items.

Based on the payload power requirements and the estimated flight time, the total amount of energy needed on the aircraft is calculated. This is simply given by the summation of all payload energy requirements:

$$E_{total} = \sum_{i}^{payload} P_i t_i$$

Once the total energy has been calculated, an additional 15%-20% is added as a reserve supply. Custom battery packs are generated for each aircraft that can meet the minimum energy requirements.

Lithium Polymer batteries are used for the glider's power supply base on availability, high energy densities and reasonable low temperature performance. Commercial off-the-shelf prismatic Lithium Polymer batteries are rated at a nominal voltage of 3.7 volts per cell. The available voltages for the battery are quantized and is given by $V_{system} = (3.7\ V)n$ where n is the number of cells in series. For battery pack design, the number of cells, or n, is increased until V_{system} exceeds the maximum voltage requirement for the payload. For example, if the maximum voltage required by the payload is $5V$, two battery cells, or $n = 2$, in series are required. Once the series configuration of the battery is determined, the number of cells required in parallel is found such that the battery can supply the required amount of energy. This calculation is done for each battery in a database of COTS battery cells. For each battery in the database, the minimum number of cells is calculated, as is the total mass of the cells. The battery configuration with the lowest mass is added to the payload.

Once the battery selection has been made, the payload is tightly packed into a small volume. For this task, a custom packing algorithm has been developed. Two embodiments of the algorithm exist, both based on a Greedy Placement algorithm. The first algorithm positions and orientates payload items by mass, while the second algorithm aims to minimize the volume.

One challenge in packaging the payload is the unique cross section of the payload bay. Traditional bin packaging algorithms work with great successes for packaging objects into regular shapes[3]. However, the cross section of the payload bay is an irregular airfoil. The custom packaging algorithm uses an elliptical tube as a container for packaging. The ellipse was chosen for its well defined mathematical properties and because it can approximate an airfoil cross-section. The objects in the payload list are manipulated and oriented in a configuration that minimize the volume of the the elliptical container. Center of mass is a secondary optimization, used primarily to form an initial guess for positioning.

Each item must fit inside the elliptical cross section of the tube. However, if it exceeds that cross section, then the cross sectional area of the tube is increased. The length of the tube is also varied based on placement. The optimal placement for an object is the one that minimize the volume added to the tube. It is also possible for an object to not increase the volume of the tube. In this circumstance, the resulting orientation and position is considered to be optimal and the placement algorithm moves to the next item in the payload list. The output of this algorithm is a list of bounding boxes and the center position of the bounding box. Additionally, the ellipse parameters and tube length parameters provide an accurate initial guess for the sizing of the payload bay section of the aircraft.

C. Airframe Design

Airframe design consists of two components: a geometry engine and an optimizer. Currently there are two options for geometry engines in the design algorithm. The first is a custom blended wing body geometry description and STL file generation engine. The second geometry engine option is OpenVSP[1]. Both geometry engines are similar for describing a blended wing body aircraft, however there are two key differences between these geometry engines. The geometry engine developed by the authors blended wing body geometry engine is written in Python and can be directly called. OpenVSP must be called using batch mode processing from a command line interface. The second difference is how both geometry engines can expand into broader airframe topologies. OpenVSP has a vast feature set that would enable sophisticated geometry creation. An example airframe designed using these methods is shown in Figure 5. Figure 6 shows two geometries that were created using both the authors' geometry optimization method and OpenVSP.

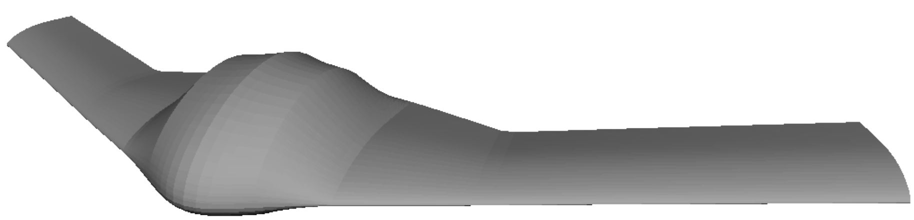


Figure 5. An example airframe. This glider is known in the authors' research group as the MkII. It was created as a platform to develop the airframe manufacturing process. It is also undergoing flight testing.

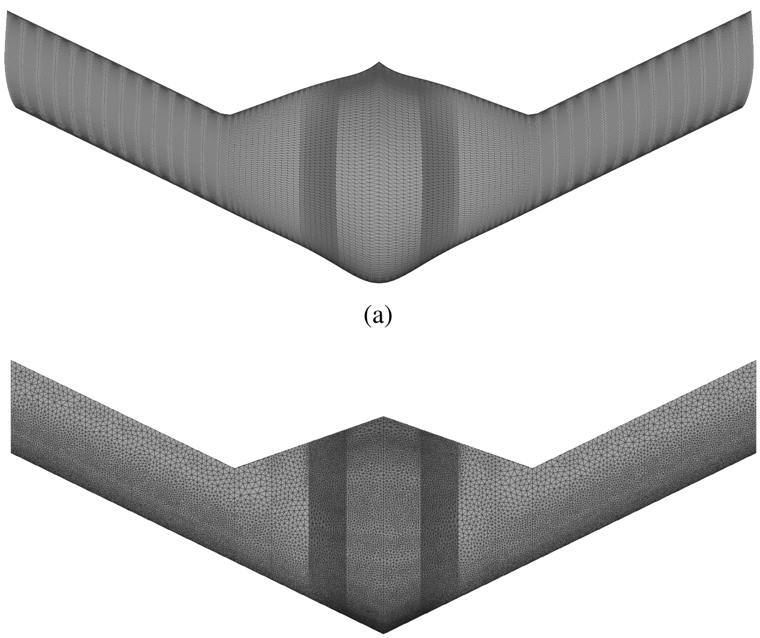
The authors' geometry engine is based on an XML description of the airframe. An airframe is defined by a series of nodes. Each node is defined by a vector $\vec{v}$, a rotation matrix, an airfoil profile and a scale factor. The vector indicates the position of the leading-edge with respect to the origin. The rotation matrix allows for washout to be added to the wing.

To form the geometry, a spline is fit through the nodes that define the leading edge. The spline curve is then used to create an array of subsampled points. At each point, an airfoil and scale factor are interpolated from the nearest two nodes. This approach allows for the mixing of airfoils in the aircraft which becomes important when determining the shape of the payload bay. This much larger array of points and airfoil cross sections is directly converted into an STL mesh, shown in Figure 6(a).

The optimization engine is used to maximize the aerodynamic performance and stability of the aircraft. Optimization is achieved through modification and evaluation using an iterative evolutionary algorithm. If the design performs well, traits of that design are passed on to future generations of designs. If the design underperforms, that it is suppressed from future generations.

Modeling performance consists of several factors that range from elementary sizing calculations to CFD modeling. The results from performance modeling are combined into a single score for the airframe that indicates its performance relative to other designs tested. This is done using a summation of normalized quadratic loss functions.

Modifications are made to the designs by making changes to a node. The modifications include modifying the offset vector, the rotation matrix, the scale or the airfoil profile.



(a)

(b)

Figure 6. Two airframe meshes produced using the authors' geometry engine (a) and OpenVSP (b). Both approaches have the ability to generate high quality meshes for simulation and manufacturing.

D. Additive Manufacturing Files

Once the aerodynamic surface has been optimized around the payload, manufacturing files are generated. These files are in STL format and ready to be printed in a wide variety of printers.

The mold surface is generated by extruding each face outward by a fixed distance. This produces a thick shell around the aerodynamic surface. In order to form the mold, this thick shelled structure is split along the leading and trailing edge using a NURBS surface. It is important that this surface does not intersect the aerodynamic surface except at the the leading edge and the trailing edge. To accomplish this, the NURBS surface is formed such that it follows the mean camber line of any given airfoil cross-section. Figure 7 shows a CAD rendering of the final mold. Also visible is the payload tray and payload bay, which will be molded into the aircraft. The cubes protruding from the outer surface of the mold are clamping features. These short stubs are held together with 3D printed clamps during the molding process.

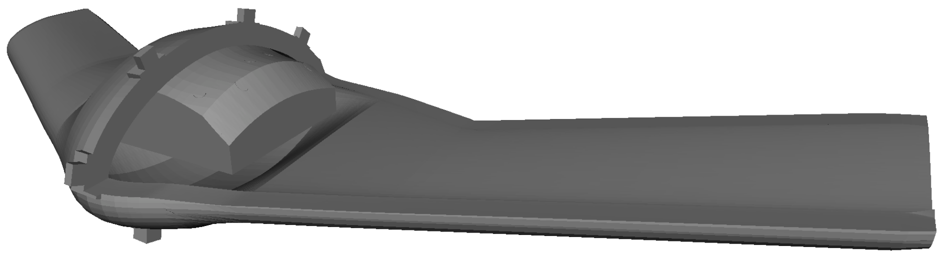


Figure 7. An example mold generated by thickening and splitting the airframe surface. The placement of the payload tray and payload tray is visible inside the mold. The mold shown in the rendering is used to create the aircraft shown in Figure 5

Additionally, the payload tray to hold the payload items is designed. Currently, this is done in a non-automated process. An automated approach will be developed in the near future. Payload items are positioned as described in the payload packaging section. Supports are added to hold payload items and it is integrated with a lid that forms part of the aerodynamic surface. Figure 8 shows a designed payload tray.

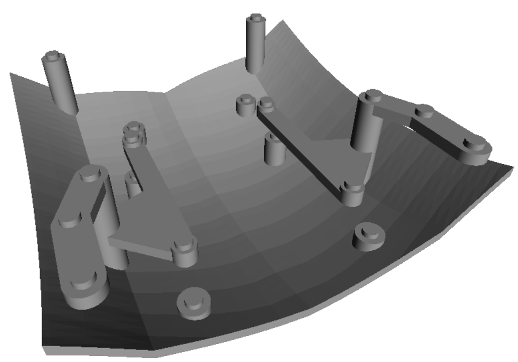


Figure 8. An example payload tray generated from payload packaging information.

III. Manufacturing

The airframe manufacturing process was derived from existing composite material techniques. A negative mold is 3D printed from the manufacturing files produced by the design algorithm. If necessary, the mold surfaces must be sanded until smooth. Once the mold is ready, layers of mold release wax are added and buffed to a mirror finish.

The first step in manufacturing is to position the 3-D printed in the tray into the mold. Clay is used as a gasket material to prevent the gelcoat and foam from filling the payload bay. Once the sealing of the payload bays complete, a polyester gelcoat is mixed and painted into the mold. The gelcoat is left until it hardens.

After the gelcoat preparation is finished, items to be embedded in the foam are added to the mold. These items can include support structures, sensors, or any other object that needs to be embedded in foam.

At this point, an A-B polyurethane expanding foam is mixed and poured into the mold. The mold is clamped shut using 3D printed braces and locks. After the foam has cured, the aircraft is carefully demolded.

This process takes approximately two hours of labor and two hours of curing time. The materials cost breakdown for an individual aircraft is shown in Table 2.

Item	Material Cost	Required (g)	Total Cost	Cost per Airframe
3D Printed ABS Mold	0.236 $^{\$}/_{g}$ *	450 g	\$108	\$4.32†
Polystyrene Gelcoat	0.016 $^{\$}/_{g}$	31 g	—	\$0.50
Polyurathane Expanding Foam	0.011 $^{\$}/_{g}$	33 g	—	\$0.36
Payload Tray	0.236 $^{\$}/_{g}$ *	3 g	—	\$0.71
Mold Release	—	—	—	\$0.03
Misc	—	—	—	\$0.10
Total Airframe Cost				\$6.02

Table 2. Material Cost Breakdown per Airframe

IV. Planned Future Work

The development of the semi-automated design and manufacture process described here is the first phase in the construction of a comprehensive 'UAV generator'. The ultimate aim is to enable the science user of the system to specify the details of the mission through a simple web interface. The output is an atmospheric sampling system, which is fully customized to meet the requirements of the mission at hand, thus minimizing costs and maximizing sampling performance. The first generation of aircraft produced by the prototype system described here is currently undergoing flight testing.

Acknowledgments

The authors are grateful to the NEMODE network and the Royal Academy of Engineering for the financial support of this work. We would also like to thank Microsoft Research and Dr. Steven Johnston for their assistance with the development of the .NET Gadgeteer-based on-board systems of the aircraft described here.

*This material cost comes from the price of 3D printable ABS plastics.

†This assumes the mold will produce 25 aircraft in its lifetime.

References

[1]Openvsp. http://www.openvsp.org.

[2]Sobester A., Castro I., and et al. Notes on meteorological balloon mission planning. AIAA, 2013. Work to be presented March 2013.

[3]S. Martello, D. Pisinger, and D. Vigo. The three-dimensional bin packing problem. *Operations Research*, 48(2):256–267, 2000.

[4]A. Sobester, S. J. Johnston, J. Scanlan, N. S. O'Brien, E. E. Hart, C. I. Crispin, and S. J. Cox. High altitude unmanned air system for atmospheric science missions. 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference. AIAA, September 2011.