Architectural Options and Optimization of Suborbital Space Tourism Vehicles

Markus Guerster
System Architecture Lab, MIT
77 Massachusetts Avenue.
Cambridge, MA 02139
857-999-6103
guerster@mit.edu

Edward F. Crawley
System Architecture Lab, MIT
77 Massachusetts Avenue.
Cambridge, MA 02139
617-230-6604
crawley@mit.edu

Abstract— **Since the creation of the Ansari X-Prize, a significant technical and commercial interest has developed in suborbital space tourism. An obvious question arises: what system architecture will provide the best combination of cost and safety for the performance defined by the prize? The objective of this paper is to address this question, by defining the design space and searching comprehensively through it with respect to launch mass (a proxy for cost) and safety. We have identified 33 architectures and visualized them in a single table. Of these, 26 have not earlier been proposed. A genetic algorithm optimized each of these 33 architectures for launch mass and safety. The launch mass was calculated by a design framework consisting of four modules: weight/size, propulsion, aerodynamics, and trajectory. To calculate the safety, a quantitative risk analysis is then developed. It is based on a hazard list with associated severities and likelihood factors. For each architectural feature, a mitigation factor is defined which quantifies the mitigation potential of an option during the design phase for a certain hazard. For a four-participant vehicle, six non-dominated architectures could be identified that merit a more refined design analysis.**

TABLE OF CONTENTS

1. INTRODUCTION

Generally, in the early stages of development, systems built for a specific function lie in a broad architectural space with numerous concepts being developed, built and tested. As the product matures, certain concepts become more dominant and the variety of concepts in use decreases [1]. Consider the wide range of "flying machines" in the decades before and after the Wrights. History teaches us that the original architectural decisions (e.g. biplane, pusher propeller, and canard) do not always survive as the dominant design [1]. This phenomenon of a wide variety of concepts can currently be observed in the suborbital tourism industry.

The birth of the suborbital space tourism dates to May 1996, where the Ansari XPrize was launched by the Ansari family. This competition challenged teams all around the world to build a reusable, private-funded and manned spaceship. The first team carrying three people to 100 km above the Earth's surface twice within two weeks received the $10 million price. 26 teams from 7 nations proposed their concept. Finally, the Mojave Aerospace Ventures team, which was led by Burt Rutan and his company Scaled Composites and financed from Paul Allen, won the competition on October 4, 2004. Richard Branson licensed Burt Rutan's technology and created Virgin Galactic. The competition launched a billion dollar market for suborbital space travel [2]. Besides Virgin Galactic, Blue Origin and XCOR have spent significant effort in developing suborbital vehicles. However, due to XCOR's adverse financial conditions, the company was forced to lay off most of its employees June 2017 [3, 4]. This delays the completion of their Mark 1 prototype for an indefinite period, leaving Blue Origin and Virgin Galactic as the major current commercial players. Their concepts are highly different, raising the obvious question:

What system architectures of suborbital space tourism vehicles will provide the best combination of performance, cost and safety?

System Architecture provides us with the methodology to approach this question. We will identify system architecture approaches and design frameworks that can be used to evaluate suborbital tourism architectures with respect to their benefit (performance and safety) and cost (with launch mass used as a proxy for cost). System Architecture emerged as a discipline in the late 1980s and has since been used by several industries. The methodology has proven its success in early phases of complex aerospace engineering system designs [5]. Like System Engineering, it is a system thinking approach that considers a system as *a set of interrelated entities which perform a function, whose functionality is greater than the sum of the parts* [6]. System Architecture is a process in the very early stages of system development, where the focus lies on carefully identifying and making the decisions that define the highest-level design. These architectural decisions determine much of the systems performance, cost, and safety. We use *architecture* as an abstract description of the entities of a system as well as the relationship between those entities. For man-made systems, decisions can represent architectures

[7, 8]. This idea of *architecture as decisions* is used later to describe suborbital tourism vehicles. [6]

To rank architectures and call one "better" than the other, the sources of associated benefit and cost must be determined. Form is generally associated with cost and the function delivers benefit to customers. A good architecture is then defined by providing the most benefit with the least cost, or equivalent performance with least cost [9]. Solution-specific metrics must be defined for each problem statement. [10, 11]

As each architecture can be represented by a set of decisions [12], we can encode these decisions into mathematical design variables. If the benefit and cost of the system become the objective functions, we can view the system architecting process as an optimization problem. Reduction in computational costs allows us to explore larger architectural design spaces in great detail, which makes it more likely to find a better architecture with respect to the defined metrics for benefit and cost [13].

A design framework generates output parameters for given input design variables. The kind of input variables is defined by the architecture. The output parameters are used to calculate the metrics for which the architecture is then optimized. Our literature search on existing design frameworks matches with the one executed by Frank [14-16] and Burgaud [17]. Including the one from Frank, we found twelve sizing and synthesis codes [14-16, 18-29]. They were compared in the categories of: computational speed, availability, ease of use, ability to explore the design space and their level of abstraction suitable for conceptual studies. For a detailed literature review refer to Guerster's Master's thesis [30] and Frank's Ph.D. thesis [14]. The comparison of the design framework evaluation codes shows that Frank's code has the best overall performance. Fortunately, Frank provided us his design framework. We deeply thank him for sharing his code with us and gratefully acknowledge his contribution.

We will search comprehensively through the architectural design space for suborbital space tourism vehicles and evaluate optimized architectures for safety and launch mass. In doing so we will identify the limits of the plausible design space and identify decisions that define the space. Then, we will build a parametric model for each of the viable options, and optimize that architecture. Finally, we will assess the un-dominated architectures, and identify the small handful of designs that merit more refined design analysis.

The specific objective of this paper is to identify a set of good suborbital tourism architectures that for an equivalent performance provide the most safety for the least cost (where launch mass is used as a proxy for cost). We use a system architecture approach and Frank's design framework. In section 2 we define the plausible design space. Then we build a parametric model for each viable architecture and optimize them in section 3. Finally, we visualize and make sense of the result to support the decision-making process in section 4.

2. DEFINING THE PLAUSIBLE DESIGN SPACE

Before we can optimize each architecture and evaluate the design space, we must define our design space. A common axiom is that if a larger design space can be explored, it is more likely to find better architectures and designs. Based on this, the objective of this section is to define the design space as broadly as is feasible. This is limited by computational resources as well as limitations of the design framework used to calculate the metrics. We identify four main steps to define the design space:

A. Building a database of existing concepts in the design space of interest

B. Defining a morphological decision matrix based on the database

C. Defining of logical, reasonable and model-limitation constraints

D. Enumeration of the feasible architectures which represent the plausible design space

These four steps are described for our problem-specific case of suborbital tourism vehicles in the following four subsections A - D. The definition of the morphological decision matrix, the reasoning about the constraints and the enumeration of the feasible architectures are coupled and consequently an iterative process.

A. Building a database of existing concepts

The first step in defining the design space is to look at existing concepts that have similar requirements. These can be concepts from competitions, current competitors who have already decided for an architecture or existing studies on the market. This sort of market and competitor analysis should anyway be done during the decision process of forming a Suborbital Space Tourism company or creating a project inside an existing company. The literature research on existing concepts revealed team summaries of almost all the companies competed in the Ansari X-Prize, previous studies of the market, books about space tourism as well as extracted data from companies' websites. The used sources are:

- **1998:** Report from the Associate Administrator for Commercial Space Transportation [31]
- **2002:** Report "Suborbital reusable launch vehicles and applicable markets" from Martin J.C. [32]
- **2003:** Ansari X-Prize Team Summaries [2]
- **2011:** The U.S. Commercial Suborbital Industry: A Space Renaissance in the Making from the Tauri Group [33]
- **2012:** A report from the Tauri Group: Suborbital Reusable Vehicles: A 10-Year Forecast for Market Demand [34]

- **2014:** The book "Suborbital industry at the edge of space" from Seedhouse Erik [35]

- **2016:** Current websites from ARCA [36], Blue Origin [37], Virgin Galactic [38], XCOR [39], Copenhagen Suborbital [40], Dassault-Aviation [41], EADS [42], Cosmocourse [43] and Scaled Composites [44]

We found 37 companies, extracted the main parameters and consolidated them into one table (see Guerster [30] for complete table). We limited the information for each company to top-level systems attributes and concept of operations differences. Unintentionally, by deciding for a set of attributes, we already have chosen a first draft set of decisions. For example, the type of engine (rocket or jet) or the propellant type are obviously important enough that the sources provide information about it. For all companies, we could find information about the top-level design of the vehicle, either in form of pictures and drawings or in form of text. The next section describes how we use this database to identify the main architectural decisions.

B. Defining the morphological decision matrix

The concept of using a morphological matrix to study the design space of different configurations dates back to Zwicky in 1969 [6, 45]. Since then, it has been used often as a decision support tool [46]. In the morphological matrix we use, rows represent architectural decisions and columns are the possible options (see Table 1). The options for a certain decision must be mutually exclusive. An architecture is then defined by choosing one option for each decision. They do not have to be from the same column. Our final decision matrix is shown in Table 1. Together with the constraints from section C, it will define our design space. The matrix consists of one vehicle level decision about the number of the module, followed by five and three decisions for the first and second module, respectively. We define a module as a system that separates during flight and lands independently. The module with the higher number has the higher energy state. The vehicle can either consist of one or two modules. If the number of modules equals 1, the first module is crewed and no second module exist (see Table 2, architectures #1 - #5 in Table 5). For a two-module vehicle, the passengers are part of the second module and the first module is uncrewed (architectures #6 - #33 in Table 5). In our notation, passengers are pilots plus participants. This means that we do not consider pilots on the uncrewed module. The module level decisions cover all three operational aspects: launching, flying and landing. The take-off mode (TOmode) can be horizontal or vertical. This decision does not exist for the second module because the second module has the same initial take-off mode as the first module and then separates during the ascent phase. The flying decision covers four issues: how the vehicle is configured, if it has wings or not, if it has jet engines or not, and if it has a rocket engine or not. By excluding the decision about the jet engine for the second module, we implicitly integrate the constraints that the final ascent must be made by a rocket engine, thus demonstrating that it is not reasonable to add an additional jet engine to the second module. If a jet engine is part of the architecture, then it must be on the first module. We do limit the maximum velocity of jet engines to Ma = 2 and do not consider air-breathing supersonic propulsion like ram- or scramjets. A typical jet engine's Ma = 2 at absolute ceiling altitude does not provide enough energy to reach 100 km by pulling up the vehicle to transfer the kinetic into potential energy. Finally, the last decision for each module is on the landing mode (LAmode). The options cover powered and unpowered landing methods. Unpowered methods are gliding and parachute, where the orientation of the first one is horizontal and the latter vertical. Powered methods are horizontal powered landing (HPowered) and vertical powered landing with a rocket engine (Rocket). We assume a horizontal powered landing with a rocket engine is not feasible due to challenges associated with its throttling over a greater range. As the second module cannot have a jet engine, it has no HPowered option. However, we include None as an option which must be chosen if the vehicle has only one module. A full combinatorial enumeration (product of the numbers in the last column) leads to 2,048 unconstrained architectures.

We apply this matrix to Blue Origin's New Shepard to provide an example of how this matrix can be used to represent an architecture. By our definition, this vehicle is a two-module vehicle. The first module is a vertical launching rocket with a rocket engine. It has no wings and

Table 2: Matching of crewed and uncrewed module to Module 1 and 2

IF nModule == 1 (architectures #1 - #5)	**IF nModule == 2** (architectures #6 - #33)
 THEN Module 2 = N/A THEN Module 1 = crewed	 THEN Module 2 = crewed THEN Module 1 = uncrewed

Table 1. Morphological architecture decision matrix

		shortID	Option 1	Option 2	Option 3	Option 4	number of options
		nModules	1	2			2
Module 1	Launching	TOmode1	Horizontal	Vertical			2
	Flying	wing1	No	Yes			2
		JetEngine1	No	Yes			2
		RocketEngine1	No	Yes			2
	Landing	LAmode1	Gliding	HPowered	Parachute	Rocket	4
Module 2	Flying	wing2	No	Yes			2
		RocketEngine2	No	Yes			2
	Landing	LAmode2	Gliding	Parachute	Rocket	None	4

no jet engine. The landing mode is vertically powered. The capsule separates during flight, has no propulsion and wings, and lands with a parachute. This encodes results in the blue marked options in Table 1. Note, that our matrix does not include scaling attributes like the number of participants, passengers or crew. We treat the number of participants as a requirement variable and the number of pilots as a design variable, which is optimized.

Table 3. Part of the constraint matrix

			Module 1									
			nModules		TOmode		wing		JetEngine		RocketEngine	
			1	2	Hori-zontal	Vertical	No	Yes	No	Yes	No	Yes
Module 1	nModules	1	-	-	0	0	0	0	0	0	a	0
		2		-	0	0	0	0	0	0	0	0
	TOmode	Horizontal			-	-	h	0	0	0	0	0
		Vertical				-	0	0	0	i	j	0
	wing	No					-	-	0	0	0	0
		Yes						-	0	0	0	0
	JetEngine	No							-	-	o	0
		Yes								-	0	0
	RocketEngine	No									-	-
		Yes										-

C. Defining of logical, reasonable and model-limitation constraints

With the morphological matrix presented in Table 1, we can define almost all concepts from our database. The ones which cannot be characterized are these which are very likely to be dominated by others, including architectures that launch from underneath the water or architectures with more than two modules. The three constraints applied to narrow down the design space are:

1. The maximum number of modules is 2
36 out of 37 companies proposed a concept with one or two modules, and only one uses a three-module rocket. Most of the current orbital launch vehicles use two stage rockets. The energy needed to reach the Kármán line is well below the one for orbital trajectories. We reason that a suborbital tourism vehicle with three modules is not competitive due to the increasing complexity, development and maintenance cost accompanied with the third module.

2. No launching from water
Only one company proposes a launch from out of the water. One can grasp that this type of launch adds complexity, operational challenges, and cost without an obvious benefit.

3. No balloon as ascent method
We excluded the balloon option as ascent method as our design framework does not have the model to optimize and evaluate the metrics for a vehicle with this feature. Moreover, we can argue that it is hardly feasible to reach 100 km with a balloon. This would mean that the second module needs to have a rocket engine. This whole module must be carried by the balloon. Given the weight of the second module, it is questionable if this approach is feasible.

Our final morphological matrix from Table 1 defines the design space for all 2,048 unconstraint architectures. Some of them are logically not possible, like the combination of jet engine *yes*, rocket engine *yes* and the landing mode *none* for the second module. The option *none* as landing mode indicates that the vehicle has one module and therefore, by default, the decisions about the engines are *no*. This is a logical constraint or also called objective by the definition of Guest [47]. Additional constraints are more subjective or reasonable as we call them in this paper. They cover aspects where domain experts can reasonably argue that the implementation of this combination of option results in higher complexity and development challenges without providing benefit in the metrics of interest. The previously applied constraints about the number of modules as well as the launch from water are illustrations of this type. Another example is a horizontal takeoff without wings. One can think of systems where a horizontal takeoff without a lifting wing may be achievable (e.g. by providing lift with vertically orientated engines). However, we can argue with engineering judgment that this increases complexity without providing a benefit in the mass and risk dimension. The no-balloon limitation previously mentioned is of the model-limitation and of the reasonable kind. The model-limitation constraint is a constraint due to limitations of the model to evaluate the metrics for this architecture. This can be, for example, the lack of the design framework to evaluate a balloon ascent.

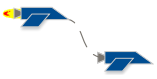
To systematically implement the constraints between the decisions and their options from the matrix in Table 1 we created a Design Structure Matrix (DSM) with the options as rows and columns. A part of this constraint matrix is shown in Table 3. Since the DSM is a symmetrical matrix, we only populated the top-right half. The number 0 with gray background stands for no interconnection and constraint between the options. If the background is red, there are constraints and the letter references to the justification in Guerster [30]. If there is a dash instead of a letter, this combination is per definition not allowed due to the mutually exclusive requirement of all options for one decision. We found in total 23 constraints of the three different types:

logical, reasonable and model limitation. For example, for the constraint identified with *a,* the table reads:

IF (nModules == 1)
THEN (RocketEngine1 != No)

Or in text form: if the option for the decision *nModules* is chosen to be *1*, then the *RocketEngine1* decision cannot be *No*. Since the RocketEngine1 decision only has two options, forbidding one option is equal to saying that it must be the other option. In this case, the previous statement is equal to the one saying: if the vehicle has one module, there must be a rocket engine on the first stage. The justification for this constraint is of the model-limitation kind (limitation of jet engines to Ma = 2 and therefore the need of at least one rocket engine). The syntax to read the other 22 constraints is equal to the example provided above.

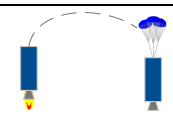
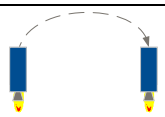
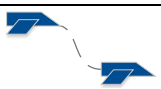
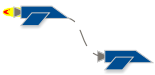
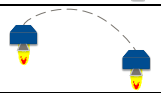
Table 4. Description of the 12 pictograms

1st module			2nd module		
	TOmode1: wing1: JetEngine1: RocketEngine1: LAmode1:	Vertical No No Yes Parachute	None	wing2: RocketEngine2: LAmode2:	No No None
	TOmode1: wing1: JetEngine1: RocketEngine1: LAmode1:	Vertical No No Yes Rocket		wing2: RocketEngine2: LAmode2:	Yes No Gliding
	TOmode1: wing1: JetEngine1: RocketEngine1: LAmode1:	Vertical Yes No Yes Gliding		wing2: RocketEngine2: LAmode2:	Yes Yes Gliding
	TOmode1: wing1: JetEngine1: RocketEngine1: LAmode1:	Horizontal Yes Yes No HPowered		wing2: RocketEngine2: LAmode2:	No No Parachute
	TOmode1: wing1: JetEngine1: RocketEngine1: LAmode1:	Horizontal Yes No Yes Gliding		wing2: RocketEngine2: LAmode2:	No Yes Parachute
	TOmode1: wing1: JetEngine1: RocketEngine1: LAmode1:	Horizontal Yes Yes Yes HPowered		wing2: RocketEngine2: LAmode2:	No Yes Rocket

D. Enumeration of the feasible architectures

We have created the morphological matrix and identified all constraints between the decisions' options. With this information, we can now define our design space by enumerating all unconstraint architectures and compare each of them within the constraints matrix. If the architecture contains a set of options which is restricted by one of the 23 constraints, it is no longer included in the list. The resulting list covers 33 distinguish feasible architecture out of the 2,048 unconstraint ones. This list of 33 architectures defines our design space. We have developed a matrix to visually represent these 33-feasible architectures. The matrix shown in Table 5 has the concept of the first module as columns and for the second module as rows. There are six possible concepts for the first and the same number for the second module (including the no second module choice). This yields to 36 architectures, but the constraint *a* discussed before excludes three of them. There must be a rocket engine either

Table 5: Overview of the 33 architectures defining the design space

1st module / 2nd module						
None	**1** e.g. Copenhagen Suborbital	**2**	**3**	x	**4** e.g. XCOR	**5** e.g. Rocketplane
	**6**	**7**	**8**	x	**9**	**10**
	**11**	**12**	**13**	**14** e.g. Virgin Galactic	**15**	**16**
	**17** e.g. ARCA	**18** e.g. Blue Origin	**19**	x	**20**	**21**
	**22** e.g. Canadian Arrow	**23**	**24**	**25**	**26**	**27**
	**28**	**29**	**30**	**31**	**32**	**33**

on the first or on the second module to reach 100 km. The pictograms visualize the concepts of the modules and their attributes. A description of the meaning of these pictograms is provided in Table 4. We use the decision ID and their options from the morphological matrix displayed in Table 1.

The 33 architectures are numbered starting from the top left to the bottom right. The three not feasible ones are marked by an *x*. If we screen our database of 37 proposed concepts and match them to one of the 33 architectures, we find that there are just 7 distinguished ones. For each of these, an example company is matched with respect to our decisions. This does not include design variables, like the number of pilots, participants or propellant type. For example, Virgin Galactic's WhiteKnightTwo and SpaceShipTwo are assigned to #14. Virgin Galactic uses this architecture with a specific set of design variables resulting in what we call in this paper a *design*. Or in other words, Virgin Galactic developed a design within the group of architecture #14[1]. There is a huge number of other designs with a different combination of design variables. This can be, for example, different rocket engine propellant types, number of participants, the wing's aspect ratio or turbine inlet temperature. For a full description of the 27 design variables with their limits refer to Guerster [30]. The optimization procedure described in section 3 varies these design variables to find a set of designs with the lowest mass and risk within each architecture.

For the remaining 26 architectures, no proposed design could be identified. This means that we could discover them as new architectures and describe their main attributes in one matrix together with the existing proposals. One representative of each architecture was chosen and included in the matrix. Christopher Frank investigated in his Ph.D. thesis four suborbital vehicle architectures [14-16]. We can match them into our overview matrix and see that he covers by our notation architecture #1, #4, #5 and #14. They are all part of the Ansari X-Prize design space. With the assumption that a "better" design can be found if the design space is larger, we have laid the foundation to discover new designs to make suborbital space tourism more affordable and safer.

3. Design Space Exploration

We have defined our design space in section 2 and describe in this section the methodology of how we are going to evaluate this design space (subsection A), the used design framework (subsection B) and the safety metric (subsection C).

A. Methodology

The methodology of the design space exploration is shown in Figure 1. An architecture definition class takes the in-decisions-encoded architectures as an input. As all architectures have a different list of design variables, the function extracts the active design variables with their lower and upper boundaries as well as their design variable type (continuous or discrete) from a generic list. For example, one design variable is the number of pilots which can be 0, 1, or 2. A second example is the type of rocket engine and its fuel combination. The rocket engine can be solid, liquid (LOX/LH2, or LOX/RP1), or hybrid (hypergolic,

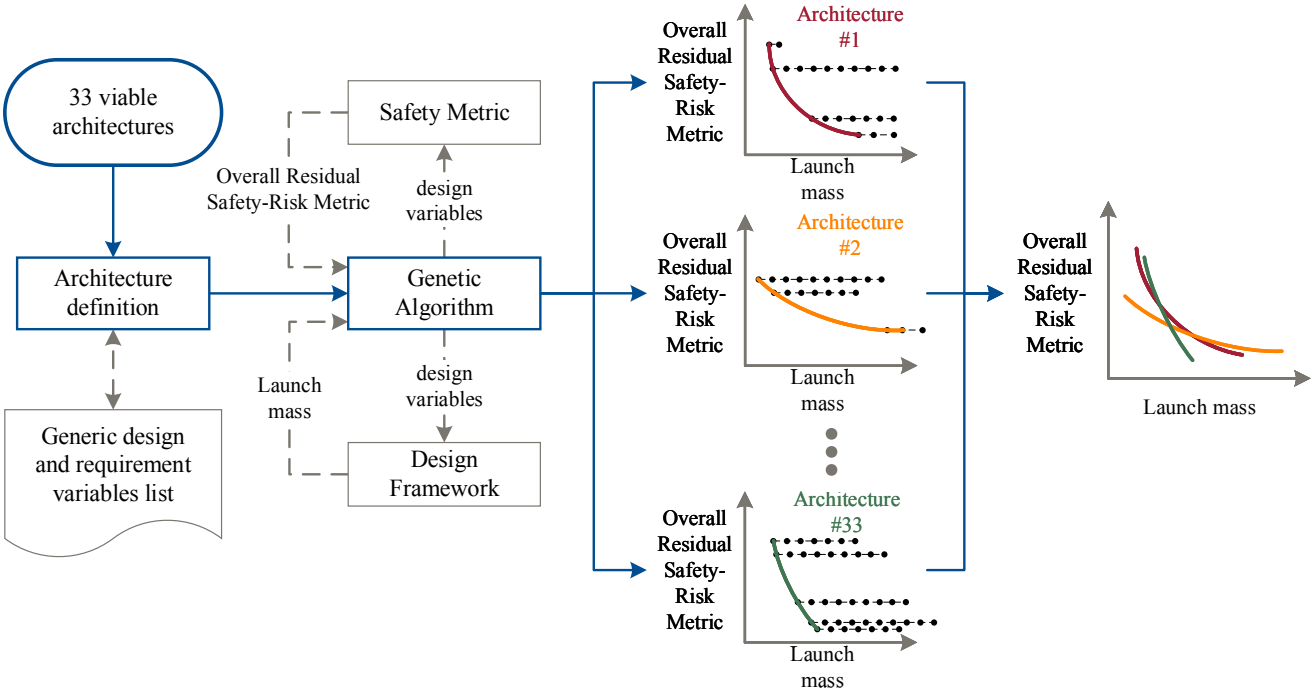


Figure 1: Schematic overview of the design space exploration methodology

[1] As defined in Table 2 we do not allow pilots on the uncrewed module 1. Hence, the WhiteKnightTwo carrier aircraft is slightly outside of our design space since it has 2 pilots. Nevertheless, it is an example design for architecture #14.

LOX/HTPB, LOX/Paraffin, N20/HTPB, or N20/Paraffin). In addition, there are requirement variables as the maximum altitude, acceptable load factor, seat-pitch as well as the number of participants. These active design variables are then populated by an NSGA II Genetic Algorithm (GA) [48]. The design framework takes the design variables from the GA as input and outputs the total launch mass. It sizes the module, calculates the propulsion, the aerodynamic properties, and the trajectory. This is an iterative optimization process as all four calculations depend on each other. For example, the trajectory depends on the weight of the module and the weight of module depends on the propellant needed for the trajectory. Depending on the vehicle's number of modules, the design framework must be executed once or twice. As a next step, the safety submodule calculates the Overall Residual Safety-Risk Metric (ORSRM) based on the design variables. This safety metric together with the total launch mass are the objective functions of the GA and the new generation is generated by crossover and mutation. The new population is then evaluated again by the design framework until the maximum number of generations is reached. This optimization process is done for all 33 architectures, resulting in a Pareto front with the most promising designs for each architecture. As shown in subsection C, the safety metric is discrete with up to 27 different values for each architecture. This discrete Pareto space is shown by the black dotted horizontal lines in Figure 1. Overlying the individual Pareto fronts results in the plot discussed in section 4 and displayed in Figure 4.

Frank's methodology does not consider the optimization of two module vehicles. Since our design space covers both one and two module vehicles, we need to extend his methodology. To evaluate a two-module vehicle, the design framework must be executed twice for each vehicle and the trajectory split up before the evaluation. The safety metric is calculated once on vehicle level. The separation point of the two modules is defined by the altitude and the velocity as can be seen in Figure 2. Both variables are active design variables for a two-module vehicle and optimized by the GA. If there is just a one module vehicle, then this altitude does not exist and is not a design variable during the optimization.

First, the second module and then the first module is evaluated with the design framework. The first module takes the mass and the aerodynamic parameter of the second module into account. If the second module has no propulsion, only the separation altitude is an actives design variable for the GA and the velocity is calculated by the design framework of the second module. In this case, the second module is on a ballistic trajectory, which is fully defined by the maximum and the separation altitude. The separation velocity is an output variable of this trajectory calculation and an input variable for the first module (together with the mass of the second module and its aerodynamic properties).

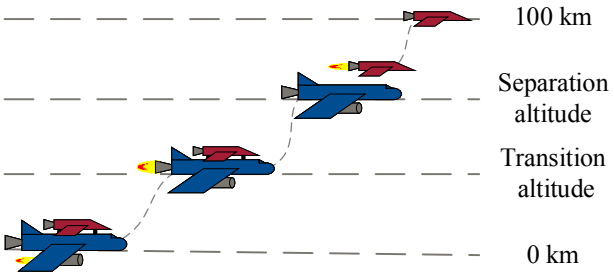


Figure 2: Separation and transition altitude for a two-module vehicle

Furthermore, if a module has both, a rocket and a jet engine, there is an additional transition velocity and altitude which is optimized. The definition of the corresponding variables is shown in Figure 2. It is assumed that if the module has a jet engine and a rocket engine, it launches with just the jet engines until the transition altitude, shuts down the jet engines and uses the rocket engine for the remaining ascent. We do not consider the simultaneous operation of both engines types. If the first module only has a jet engine or a rocket engine, there is no active design variable for the transition altitude and velocity. If the first module does only have jet engines, the upper boundary of the separation velocity is set to the maximum design velocity of jet engines which is Ma = 2 or around v = 600 m/s for a typical service ceiling altitude of 12,000m [49].

B. Design framework

The design framework evaluates the module according to the input design variables. It consists of four submodules, the weight/size, the propulsion, the aerodynamics and the trajectory submodule (marked blue in Figure 3). The propulsion, as well as the trajectory calculations, are further depicted in the jet and rocket engine part. Our design

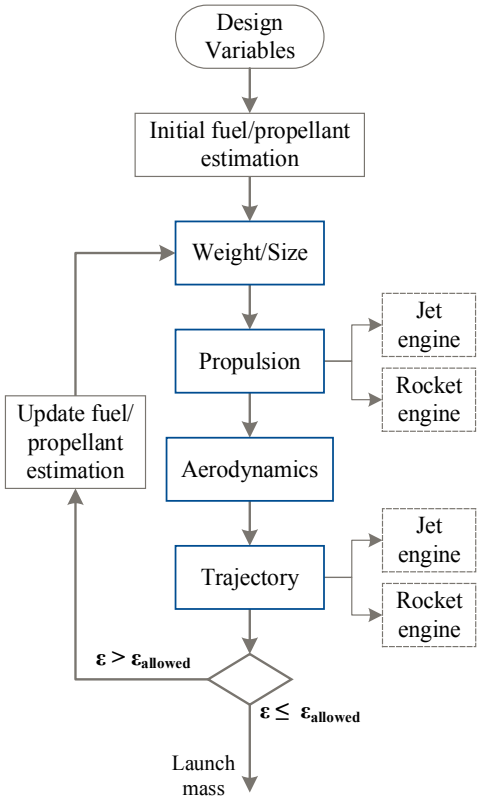


Figure 3: Overview design framework

framework is based on Frank's code. For details on the aspects that differ from Frank's design framework refer to Guerster [30]. The

Ph.D. Thesis of Frank [14] describes and validates each individual part. The modules can have wings or not which influence the gravity and drag losses. In the case of wings, the gravity losses are set to zero and the drag losses include the drag due to lift production of the wing. If the module has no wing, the gravity losses are considered and the drag loss is limited to the parasite drag. The optimizer for the design framework iterates the propellant/fuel masses and the burn time for the jet and rocket engines until the relative deviations with the previous guesses are below a defined limit. For each iteration, the weight and size are recalculated, the propulsion system is sized, the aerodynamic properties extracted and finally, the trajectory obtained. If the module has converged, the design framework outputs the launch mass including fuel and propellant.

A run time investigation of Frank's [14] initial code shows that around 95% of the design framework evaluation are caused by optimizing the trajectory. With this, it is impractical to explore a large architectural design space. We have developed a new approach to estimate the propellant mass with a fraction of the computational effort while retaining accuracy. The time for the overall design framework evaluation could be reduced by a factor of 95 [30]. This allows us to search through a much larger design space.

Validation of the Design Framework

The three projects of Virgin Galactic, Blue Origin, and Rocketplane XP are used to validate the design framework's weight, sizing, jet, and rocket trajectory optimization features. We use the take-off mass of these vehicles to compare them with the results from the design framework (see Table 7). Due to the lack of funding, no prototype of the Rocketplane XP exists and therefore no measured take-off mass is available. However, the company did a more detailed design study and data is available on their official website and other sources. Both, the New Shepard and the SpaceShipTwo are in the test phase but no take-off weights are publicly available. We have cited several sources for each project in the table, they vary from newspaper articles to online forum discussions. The table shows relative deviations between 1 and 7% depending on the vehicle. Given the range and uncertainty in the sources, this is a representative result, especially for conceptual design space exploration methods with this variety of features.

Table 7: Overview design framework validation

	Reference [kg]	design frame-work [kg]	relative deviation [-]
Blue Origin's New Shepard	35,000 [50, 51]	32,710	6.5%
SpaceShip Two	9,740 - 13,608 [52, 53]	13,495	0.8% w.r.t. 13,608 kg
Rocketplane XP	8,840 - 9,072 [54-56]	9,641	6.3% w.r.t. 9.072 kg

C. Safety metric

The architectural decisions have a significant and lasting impact on how safe the system can be built. However, practical methods for quantitative analysis of the safety, like the top-down Fault Tree Analysis (FTA) or the bottom-up Failure Mode and Effect Analysis (FMEA) are hardly applicable in conceptual scenario trades [57-59]. The tragic example of the *Challenger* Space Shuttle's accident shows how the early architectural decision of not including a crew escape system later influenced the overall safety of the launch system. There was no cost-effective way to add a crew escape system after the orbiter was already developed [58, 60, 61].

Our approach is an adaption of Dulac's [59] and Leveson's [58] risk analysis of early system architecture trade studies. We define a top-level hazard list and assign to each hazard a **severity** and **likelihood** factor to obtain a risk value. Then, for each architectural decision and for two important design variables, a **mitigation** factor depending on the chosen option is used to calculate the Overall Residual Safety-Risk Metric (ORSRM). Subsection 1 describes the used scale for the severity and likelihood factor as well as their combined contributions to the risk factor. These values are then assigned to the hazard list in subsection 2. Subsequently, subsection 3 depicts the scale for the mitigation factor as well as their assignments. The calculation instruction with an example is shown in subsection 4.

1. Severity and likelihood factor scale

Common assessment methods treat risk as a combination of likelihood and severity, where the operand x somehow combines both values into a risk factor: $f_R = f_S \, x \, f_L$. We define severity as the worst possible consequence of a hazard and likelihood as the probability that this hazard occurs. Leveson [58] developed a custom severity scale for an early manned space exploration risk analysis. Table 6 shows this scale with the corresponding effect on the participants. The higher the factor, the more severe the hazard. The six likelihood categories are based on the standard risk matrix [58]. The scale together with a detailed definition is shown in Table 8. We anchored the definition of the likelihood to the occurrence of the hazard once over the system's lifespan.

Table 6: Severity scale

Severity factor f_S	Effect on participants
4	Loss of life
3	Severe injury
2	Minor injury
1	Less than minor injury

Often, a standard risk matrix is used to qualitatively combine both metrics to a risk value. In our assessment, we use the mathematical product and therefore obtain Table 9 with the

Table 8: Likelihood scale

Likelihood factor f_L	Definition	
5	Frequent	Guaranteed to occur multiple times over lifespan
4	Periodic	Guaranteed to occur at least once over lifespan
3	Occasional	Likely to occur once over lifespan
2	Remote	Possible, but unlikely to occur once over lifespan
1	Unlikely	Highly unlikely to occur once over lifespan
0	Impossible	Impossible to occur once over lifespan

severities as columns and likelihoods on the rows. The resulting risk value is colored with red indicating riskier combinations. The absolute value of our scale has no further meaning, it rather allows for a comparison of the hazards.

2. Hazard list with risk factors

The first step is to identify the hazards that might occur during system operation. For some specific systems, the government agencies have mandated hazards. For example, the U.S. Department of Defense prescribes a minimum baseline of hazards which must be considered when constructing nuclear weapon system [58]. However, in most cases, the hazards must be determined by the analyst. There are a few structured ways to enable individuals or a group of people to apply their knowledge (e.g. with *what-if* questions). For an extensive list of possible activities refer to *Safeware* [57]. Especially in the space sector, there are public available generic hazards lists, which can be used to identify high-level system hazards. Grounded in NASA's generic hazards lists for the Space Shuttle [62] and the Constellation Program [63], we derived a list of system-level hazards for each module. Table 10 shows them organized by mission phase, starting with the generally applicable hazards and followed by the ascent as well as the descent phase. The descent phase of the uncrewed module is not considered since there are no hazards associated with the threat to participants. For example, a fire or explosion of the uncrewed module during descent does not directly harm the participants. However, it is not desirable, as this hazard jeopardize the mission success and destroys the equipment. Compromises to these other categories are not taken into account in our safety-risk assessment.

Table 9: Risk matrix

Likelihood \ Severity	**1**	**2**	**3**	**4**
5	5	10	15	20
4	4	8	12	16
3	3	6	9	12
2	2	4	6	8
1	1	2	3	4
0	0	0	0	0

The loss of life support hazard C-G1 has the lowest value caused by a relatively moderate severity 3 combined with a low likelihood 2. Some of the hazards have equal risk values but different individual entries for the severity and likelihood. For example, the fire and explosion U-A1 and C-A1 compare with the incorrect propulsion/trajectory/control U-A2 and C-A2. If the module explodes, the participants will lose their lives. Compared to this, the severity of an incorrect propulsion/trajectory/control is lower as there are possibilities to reestablish a secure state. However, an engine

Table 10: Hazard list with associated severity factor f_S, likelihood factor f_L and resulting risk factor f_R

	Phase	ID	Hazard	f_S	f_L	f_R
Crewed module	General	**C-G1**	Loss of life support of crewed module (including power, temperature, oxygen, air pressure, CO2, food, water, etc.)	3	2	**6**
	Ascent	**C-A1**	Flammable substance in presence of ignition source during ascent of crewed module (Fire + Explosion)	4	3	**12**
	Ascent	**C-A2**	Incorrect propulsion/trajectory/control during ascend of crewed module	3	4	**12**
	Ascent	**C-A3**	Loss of structural integrity during ascent of crewed module (due to aerodynamic loads, vibrations, etc.)	4	2	**8**
	Descent	**C-D1**	Flammable substance in presence of ignition source during descent of crewed module (Fire + Explosion)	4	3	**12**
	Descent	**C-D2**	Incorrect propulsion/trajectory/control during descent of crewed module	4	3	**12**
	Descent	**C-D3**	Loss of structural integrity during descent of crewed module (due to aerodynamic loads, vibrations, etc.)	4	2	**8**
Uncrewed module	General	**U-G1**	Incorrect stage separation between uncrewed and crewed module	4	2	**8**
	Ascent	**U-A1**	Flammable substance in presence of ignition source during ascent of uncrewed module (Fire + Explosion)	4	3	**12**
	Ascent	**U-A2**	Incorrect propulsion/trajectory/control during ascend of uncrewed module	3	4	**12**
	Ascent	**U-A3**	Loss of structural integrity during ascent of uncrewed module (due to aerodynamic loads, vibrations, etc.)	4	2	**8**

loss or incorrect trajectory (U-A2, C-A2) is more likely to occur than an explosion. The higher likelihood combined with the lower severity lead to the same risk amplitude. Excluding the likelihood factor at this stage would lead to unreasonable higher values for the high severe but less likely hazards compared to the low severe but high likely ones. The risk factor varies from 6 to 12 which is in the orange region of the risk matrix from Table 9 (gray ellipse). As described in subsection 4 below, the risk factor weights the residual risk for each hazard before averaging leads to the ORSRM. Except of C-G1, the risk factors vary around ± 20% from 10. Due to the linear weighting, this also means that the ORSRM is effected by around 20%. As seen in the result section 4, the ORSRM varies approximately from 1 to 9. This means that the mitigation scale has significantly more influence on the safety metric than the severity and likelihood factors (which result in the risk factor). In standard space project, a mitigation strategy is developed during the design to reduce these risks (moving the gray ellipse from Table 9 towards the bottom left). We account for this effect by estimating this mitigation potential using the mitigation factor described in the following section.

3. Mitigation factor

After building the hazard list and calculating the risk factors, we determine the mitigation potential depending on the architectural decisions. Leveson developed the mitigation impact scale shown in Table 11. It ranges from 1 to 4 with higher numbers meaning more impact. The highest factor means that the design choice can completely mitigate the hazard, e.g. the choice of a single module vehicle mitigates the incorrect stage separation U-G1 (see Table 12). The dash means that the design choice has no mitigation potential. With this scale and the hazards, we build a matrix called hazard mitigation database. It is shown in Table 12. The columns show the hazards identified in Table 10 above. The rows are the architectural decision with their options, as well as two important design variables, which impact the safety of the vehicle. These are the number of pilots varying from 0 to 2 and the type of propellant, solid, liquid or hybrid. Including these design variables result in up to 27 different values within an architecture (3 options for the number pilots, 3 options for module one propellant type and 3 options for module two propellant type). However, not each combination leads to a design on the Pareto front. If we would exclude the design variables from the risk assessment, each architecture has a design independent ORSRM value.

Table 11: Mitigation impact scale[58]

Impact factor	Description
4	Complete elimination of the hazard from the design
3	Reduction of the likelihood that the hazard will occur
2	Reduction of the likelihood that the hazard results in an accident
1	Reduction of damage if an accident does occur
-	No mitigation potential

To populate the table, we first decide on whether the decision has a mitigation potential on the hazard or not. For example, the number of modules has no influence on the fire and explosion hazard of the crewed module C-G1. Every non-influenced combination was marked by a dash. For the other fields, we discussed the factors and compared the options inside a decision. The values shown in Table 12 are called mitigation factors (MF).

Since the hazards differ for the uncrewed and crewed module, the decision of module 1 and 2 have to be matched depending on the number of modules. If the architecture is a single module vehicle, the module number 1 is the crewed module and the uncrewed module does not exist (see discussion on Table 2). And therefore, hazards U-G1 until U-A3 are not applicable (red shaded box). For a two-module vehicle, the uncrewed module is module 1 and the module 2 matches to the crewed module. In this case, all hazards are applicable (blue shaded boxes). The vehicle decision is independent of the number of modules and the mitigation factors always apply to the hazards (gray box). However, if the vehicle is a single module architecture, the right four columns with the uncrewed module hazards are not considered in the assessment.

4. Calculation instruction with example

This section describes the six steps necessary to calculate the ORSRM from the risk and the mitigation factors introduced above. The instructions are based on Dulac [59]. We first introduce each step with its mathematical representation and illustrate its application. We use architecture #1 with a defined set of design variables as an example. Table 13 shows the chosen design variables together with the decisions which define architecture #1 (see Table 4 and Table 5 for reference, decisions for module 2 are not included).

We see from the decisions that architecture #1 is a single module vehicle. This means that looking at the hazard mitigation database in Table 12, only the hazard C-G1 to C-D3 are applicable and we can neglect the right four columns. This also means that the crewed module hazards are matched to the module 1 decisions and design variables. This reduced the matrix to the gray and red shaded box on the top-left and middle-left. The following steps described an exemplary calculation of the safety metric for the first hazard C-G1.

1. We begin with step 1 based on Dulac [59]. It calculates the maximum possible mitigation potential for each hazard *h*. The so-called Total Maximum Mitigation Factor (TMMF) is computed by summing the Maximum Mitigation Factors (MMF) for each decision and design variable *d* (see Eq. (1)). The MMF is obtained by taking the highest mitigation factor for a certain hazard and decision. It is a measurement of the maximum mitigation potential of a decision towards a specific hazard. Starting with the first applicable hazard C-G1 in the first column of Table 12, we go down to the

Table 12: Hazard mitigation database

		Always applicable							Applicable IF nModule == 2			
		Crewed module							Uncrewed module			
		C-G1 Loss of life support	C-A1 Fire + Explosion during ascent	C-A2 Incorrect prop/traj during ascent	C-A3 Loss of structural integrity ascent	C-D1 Fire + Explosion during descent	C-D2 Incorrect prop/traj during descent	C-D3 Loss of structural integrity descent	U-G1 Incorrect stage sepa-ration	U-A1 Fire + Explosion during ascent	U-G1 Incorrect stage sepa-ration	U-A1 Fire + Explosion during ascent
Risk factor f_R:		**6**	**12**	**12**	**8**	**12**	**12**	**8**	**8**	**12**	**8**	**12**
Vehicle		**Always applicable**							**Applicable IF nModule == 2**			
Decisions												
nModule	1	-	-	-	-	-	-	-	4	-	-	-
	2	-	-	-	-	-	-	-	-	2	2	2
TOmode	Horizontal	1	-	1	1	-	-	-	-	-	1	1
	Vertical	-	-	-	-	-	-	-	-	-	-	-
DesignVariables												
nPilots	0	-	-	-	-	-	-	-	-	-	-	-
	1	1	-	1	-	-	1	-	-	-	1	-
	2	2	-	2	-	-	2	-	-	-	2	-
Module 1		**Applicable IF nModule == 1 THEN Module 1 = crewed**							**Applicable IF nModule == 2 THEN Module 1 = uncrewed**			
Decisions												
wing1	No	-	-	-	-	-	-	-	-	-	-	-
	Yes	-	-	2	-	-	2	-	-	-	2	-
JetEngine1	No	-	2	-	-	-	-	-	-	2	-	-
	Yes	-	-	1	-	-	2	-	-	-	1	-
RocketEngine1	No	-	3	1	-	-	-	-	-	3	1	-
	Yes	-	-	-	-	-	-	-	-	-	-	-
LAmode1	Gliding	-	-	-	-	3	2	2	-	-	-	-
	HPowered	-	-	-	-	2	3	3	-	-	-	-
	Parachute	-	-	-	-	3	-	1	-	-	-	-
	Rocket	-	-	-	-	-	2	-	-	-	-	-
DesignVariables												
Propellant1	Solid	-	2	-	-	-	-	-	-	2	-	-
	Liquid	-	-	2	-	-	-	-	-	-	2	-
	Hybrid	-	1	1	-	-	-	-	-	1	1	-
Module 2		**Applicable IF nModule == 2 THEN Module 2 = crewed**							**N/A**			
Decisions												
wing2	No	-	-	-	-	-	-	-				
	Yes	-	-	2	-	-	2	-				
RocketEngine2	No	-	3	1	-	-	-	-				
	Yes	-	-	-	-	-	-	-				
LAmode2	Gliding	-	-	-	-	3	2	2				
	Parachute	-	-	-	-	3	-	1				
	Rocket	-	-	-	-	-	2	-				
	None	-	-	-	-	-	-	-				
DesignVariables												
Propellant2	Solid	-	2	-	-	-	-	-				
	Liquid	-	-	2	-	-	-	-				
	Hybrid	-	1	1	-	-	-	-				

nModule decision and see that there are only dashes and no values. This means that the MMF for this combination of hazard and decision $x_{MMF_{C-G1,nModule}}$ equals 0 (no mitigation potential). The next decision for the same hazard is the TOmode. We found that there is one single value (horizontal), resulting in $x_{MMF_{C-G1,TOmode}} = 1$. Then, the design variable about the number of pilots has two mitigation factors for the hazard C-G1, one pilot result in a MF of 1 and two pilots in a MF of 2. As we want to know the maximum mitigation potential of a certain decision for a certain hazard, we obtain $x_{MMF_{C-G1,nPilots}} = \max(1,2) = 2$. All applicable MFs in the red box are dashes and therefore the MMFs are all 0. Hence, using Eq. (1), the TMMF for hazard C-G1 is calculated to 1 + 2 = 3. This means that the maximum potential mitigation for the C-G1 hazard from Table 12 is 3. If a design reaches this value, the hazard is mitigated to the greatest possible extend. Using the same procedure, the other

Table 13: Decisions and design variables for the example architecture #1

Architecture #1	nModule	TOmode	wing1	Jet-Engine1	Rocket-Engine1	LAmode1	nPilots	Pro-pellant1
	1	Vertical	No	No	Yes	Parachute	1	Liquid

Table 14: Risk assessment of example architecture #1

	C-G1	C-A1	C-A2	C-A3	C-D1	C-D2	C-D3
	Loss of life support	Fire + Explosion during ascent	Incorrect prop/traj during ascent	Loss of structural integrity ascent	Fire + Explosion during descent	Incorrect prop/traj during descent	Loss of structural integrity descent
TMMF	3	7	9	1	3	9	3
HMI	1	2	3	0	3	1	1
RRR	0.67	0.71	0.67	1	0	0.89	0.67
RSI	4	8.57	8	8	0	10.7	5.33
ORSRM				**6.37**			

six TMMF can be obtained. They are shown in Table 14 in the second row. These values are independent from the chose decision and equal for all single module vehicles.

$$x_{TMMF_h} = \sum_d x_{MMF_{h,d}} \tag{1}$$

2. In the second step of Dulac's [59] procedure, the mitigation potential of the actual design choice is calculated. He calls this the Hazard Mitigation Indices (HMIs). It is calculated for each hazard *h* by summing the MFs which correspond to the chosen design (see Eq. (2)). Examining the first column from Table 12 with our example from Table 13, only for the decision about the number of pilots (=1), a chosen option leads to some mitigation of the hazard C-G1 (the vertical TOmode decision leads to no mitigation of C-G1, indicated by the dash). This leads to an HMI of 1 for the C-G1 hazard, meaning that the example design has a mitigation potential of 1 compared to the maximum reachable mitigation of 3 (TMMF).

$$x_{HMI_h} = \sum_a x_{MF_{h,a}} \tag{2}$$

3. In the third step, the residual risk for each hazard *h* is calculated using the HMI and TMMF from the previous two steps. Dulac [59] names this the Relative Residual Risk (RRR). It is calculated by subtracting from 1 the fraction of the HMI over the TMMF (see Eq. (3)). This reads for the C-G1 hazard $x_{RRR_{C-G1}} = 1 - 1/3 = 2/3$, which means that the example design shown in Table 13 can mitigate 1/3 of the risk of hazard C-G1 leaving an unmitigated residual risk of 2/3. Or in other words, the risk factor $f_R = 6$ (see Table 10) can be reduced by 2/3 as shown in Table 14.

$$x_{RRR_h} = 1 - \frac{x_{HMI_h}}{x_{TMMF_h}} \tag{3}$$

4. In the fourth step of Dulac's [59] approach, we check if any MF equals 4 (see Eq. (4)). According to Table 11, a MF of 4 means that the hazard can be eliminated, leaving no residual risk and therefore the RRR is set to zero. This is not the case for our example design from Table 13.

$$IF\ (x_{MF_h} == 4)\ THEN\ (x_{RRR_h} = 0) \tag{4}$$

5. The fifth step is weighting the residual risk RRR with the risk factor f_R from Table 10 to the power of *k*. This results in the Relative Severity Index (RSI). Dulac [59] proposed a quadratic dependency (k = 2) to increase the influence of higher severity levels (see Eq. (5)). However, we have chosen to include the likelihood into the risk factor. Since the values of this scale are close to the severity ones, we have already approximately squared the values. This is why we use a linear dependency, k = 1. For our example design from Table 13 and C-G1, the RRR is 2/3 and the risk factor $f_R = 6$. Multiplication of these two values leads to the weighted RSI $x_{RSI,C-G1} = 4$ as shown in Table 14.

$$x_{RSI_h} = x_{RRR_h} \cdot f_R^k = x_{RRR_h} \cdot f_R \tag{5}$$

6. After repeating steps 1-5 for the remaining six hazards, the ORSRM can be calculated by averaging over all RSIs (see Eq. (6) with N being the number of hazards). The values for the other hazards are shown in Table 14. The ORSRM for architecture #1 with this specific set of design variables is calculated to be 6.37.

$$ORSRM = \frac{\sum_h^N x_{RSI_h}}{N} \tag{6}$$

The RRR is a measurement of how well the design can mitigate the hazards. It is the remaining risk and it has to be between 0 and 1 with higher values meaning riskier. The values for C-G1, C-A1, C-A2, C-D2, and C-D3 are in the upper third of the possible mitigation potential from Table 14. Leaving a fair risk potential for these hazards. The loss of structural integrity during ascent hazard C-A3 remains unmitigated since only a horizontal TOmode can mitigate it. In contrast, the Fire + Explosion during descent C-D1 is mitigated maximally. The LAmode is parachute which has the highest MF for this decision. Since this is the only influence on the C-D1 hazard, the RRR equals 0.

Following this procedure, the risk assessment for all 33 architectures can be calculated. Besides the total launch mass, it is the second objective function during the GA optimization.

4. Results and Discussion

We have defined the design space in section 2 and described the exploration methodology including the safety metric in section 3. This section 4 presents and discusses the optimized results. The design optimization uses the number of participants as a prescribed requirement variable. The number of pilots is chosen by the algorithm. We ran the optimization three times with a fixed number of participants *1, 4* and *8*. To keep this section concise, only the results for *4* participants are shown and described.

Each architecture was optimized by the GA. To explore a broad design space, the population was set to 2,000. This increases the likelihood that the converged designs are global optimums. For each generation, the script calculated the

relative deviation of the best individual with respect to the latest generation. This assumes that the latest generation is the one closest to the "real" solution. The relative deviation is calculated for both metrics, ORSRM and total launch mass. It is tracked over the whole optimization and is used as a measurement for the level of convergence. A study on this has shown that 50 generations is an appropriate setting. Since the optimization of the various architectures is not coupled, the problem could be parallelized at the very beginning to reduce the computational time. We have broken down the problem into 33 sub-problems, with each architecture optimization being one problem.

For each architecture, we can identify the individual designs on the Pareto front (colored lines Figure 1). If we combine these lines of the 33 architectures into one chart, we obtain Figure 4 with the ORSRM on the ordinate and the total launch mass on the abscissa. Each Pareto front is labeled with the architecture number defined in Table 5. The number is plotted to the left of the optimized design of the architecture with the highest ORSRM, i.e. on the top-left of the line. The dashed black line combines the non-dominated solution of all architectures and displays the overall Pareto front. Furthermore, the architectures are colored depending on their wing attribute. If both modules have a wing, they are colored blue; if only the first module has wings but not the second, they are colored red; if only the second module has wings but not the first, they are colored orange and if neither of the modules has wings, they are colored gray. In addition, the four architectures of Blue Origin, Virgin Galactic, XCOR and Rocketplane XP are encoded with markers. The Blue Origin architecture #18 has larger dots (•), the Virgin Galactic architecture #14 has a downward-pointing triangle (▼), the XCOR architecture #4 has crosses (x) and the Rocketplane XP architecture #5 has filled diamonds as markers (♦).

The XCOR architecture is on the Pareto front for medium risk and masses. The Rocketplane XP architecture is around 1 t heavier but slightly safer. The Blue Origin #18 has an additional 1 t launch mass compared to the Rocketplane XP and is riskier. Virgin Galactic #14 has almost twice the same mass as the Rocketplane XP and is slightly safer. Architecture #17 and #18 differ in the landing mode of the first module, parachute and rocket powered, respectively. As can be seen from the figure, the rocket powered landing increases the mass and slightly increases the risk for the vehicle. This is due to the additional systems and propellant needed. The same reasoning explains the difference between #1 and #2.

The minimum total launch mass for a vehicle with four participants is 4.7 metric tons. With decreasing mass, the risk increases and vise-versa. This is an expected behavior and true for most technical systems [6]. There is a trade-off between risk and mass (which is often a proxy for cost). The colors show a clear trend from the heavier and safer options on the bottom-right to the lighter and riskier architectures on the top-left. The centroids indicate the mean value of the architecture within one color (and therefore the wing attribute) for both dimensions. Wings add weight but also reduce risk. The architectures with two wings are the heaviest but the safest. If the vehicles have no wings (gray lines) they are riskier but lighter. The architectures with either a wing on the first or second modules are in between. According to our

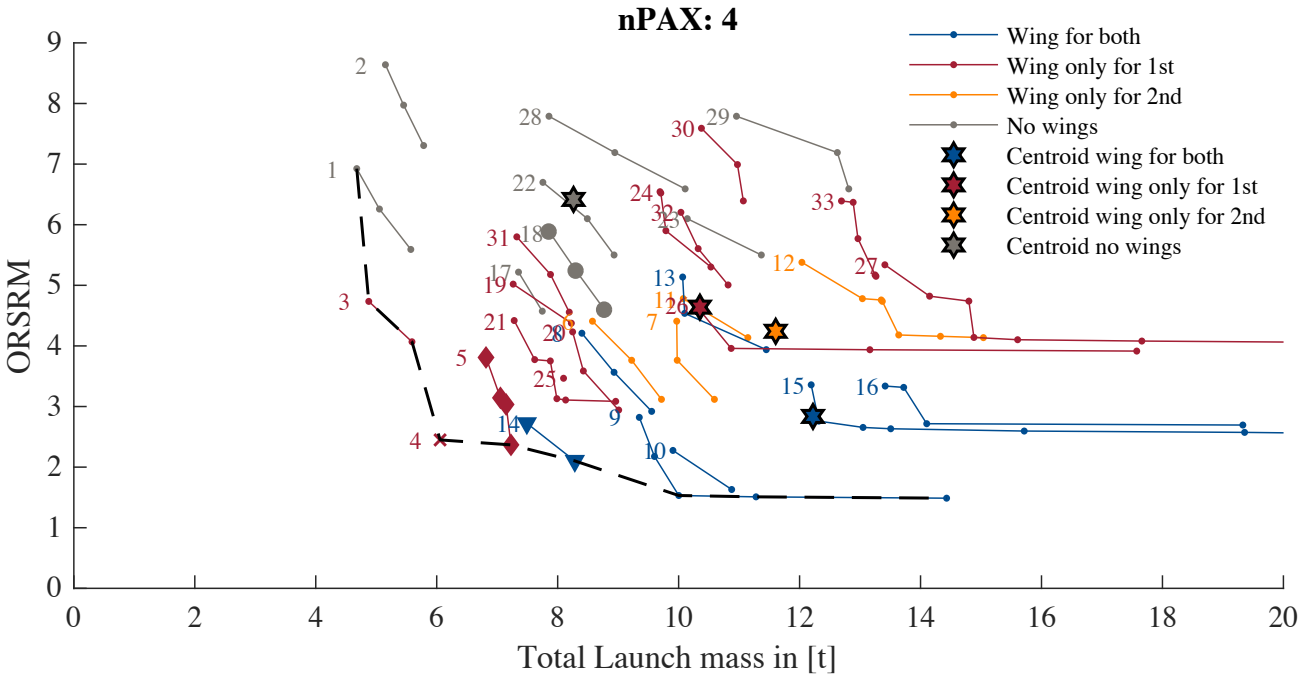


Figure 4: Pareto front of the 33 architectures with respect to the ORSRM and total launch mass; number of participants is set to 4; the Blue Origin architecture #18 has larger dots (•), the Virgin Galactic architecture #14 has a downward-pointing triangle (▼), the XCOR architecture #4 has crosses (x) and the Rocketplane XP architecture #5 has filled diamonds as markers (♦)

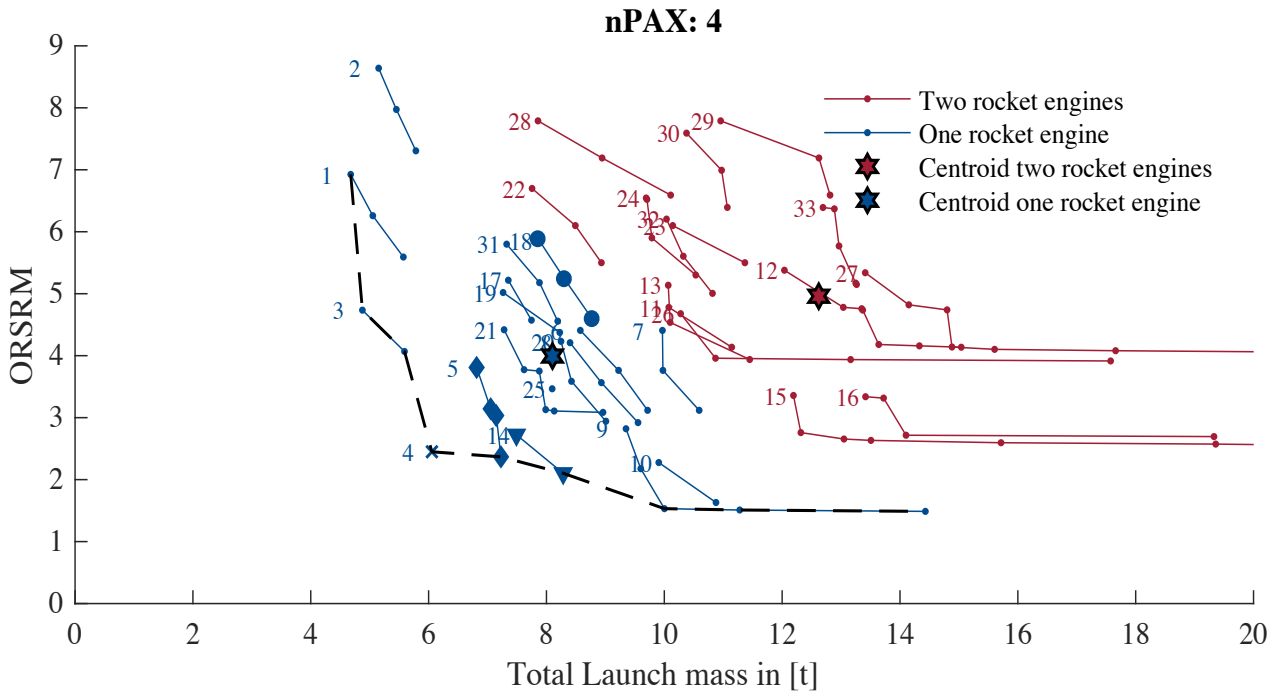


Figure 5: Pareto front of the 33 architectures colored by their number of rocket engines. The centroids indicate the mean value of one colored group

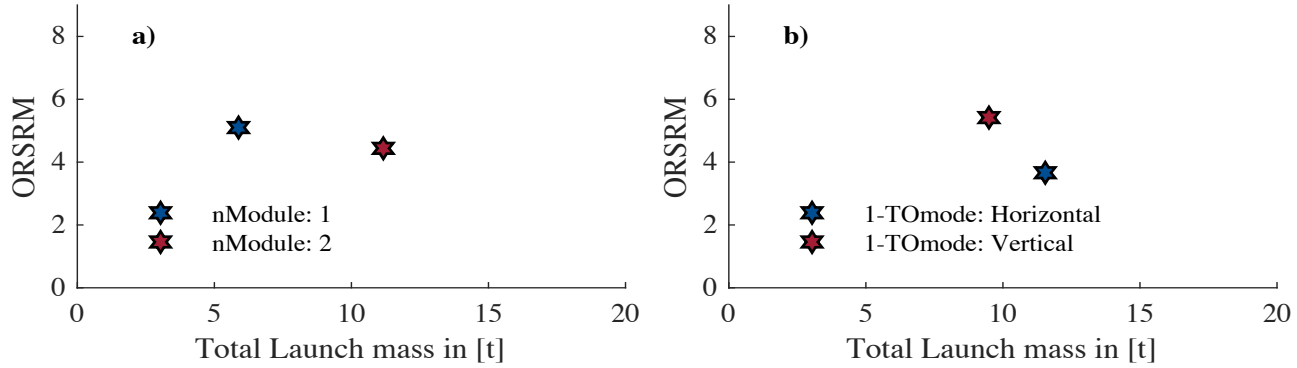


Figure 6: Centroids for the different number of modules on the left and the take-off modes on the right

reasoning of the risk mitigation factors in Table 12, wings have the potential to mitigate trajectory and steering hazards. In addition, they open up the possibility to use jet engines instead of rocket engines, which are considered safer. As the red and orange centroids show, having a wing on the second module is safer but heavier. This makes sense, since the weight of the wings on the second module has to be carried by the first module, increasing its size and hence increasing the overall mass of the vehicle.

As can be seen from Table 5, the architectures #16, #27, and #33 have three engines, a rocket and jet engine on the first module and an additional rocket engine on the second module. Our engineering intuition is that these architectures are heavier and will be dominated by others. And indeed, the model confirms this intuition. They are the three heaviest architectures in Figure 4 with total launch masses around 13 to 14 t. In addition, due to the two rocket engines, which do not mitigate fire and explosion hazards, the ORSRM is in the medium range. To further investigate the influence of the number of engines on the mass and risk, we recolor the pareto fronts according to their number of rocket engines on the vehicle (see Figure 5**)**. Red indicates vehicles with two rocket engines, meaning that each module has one engine. The blue architectures have one rocket engine (either module 1 or module 2). None of the architectures with two rocket engines are on the pareto front; they are completely dominated by those with only one rocket engine. This leads us to the conclusion that not only three engines are unnecessary but also that vehicles with a single rocket engine provides the least mass with the least risk. For orbital ascents, splitting the required energy between stages (=staging) results in a performance benefit [64]. Due to the lower required Δv, this is different for suborbital vehicles as the results show.

Furthermore, we investigate the influence of the two architectural decisions about the number of modules as well as the take-off mode. The centroids are calculated with the same method as the ones for the wing and the number of rocket engines. The left plot a) of Figure 6 shows that the average one module vehicles are lighter but slightly riskier than the two-module architectures. The separation capability

of a two-module vehicle makes them safer. The mass difference is significant, the average two-module vehicle is around twice as heavy due to the additional dry mass. On the right plot b) of Figure 6 the influence of the take-off mode on the two objective functions can be observed. A vertically take-off results in a slightly lighter vehicle but increases the risk by almost 50%. Since the horizontal take-off requires wings, there is a dependency between the decision about the take-off mode and the wing. Some of the additional safety for the horizontal launch comes from the wings as discussed above.

For supplementary discussion, we look at the architectures on the Pareto front with the numbers #1, #3, #4, #5, #14, and #9, starting from lighter and riskier to heavier and safer. They are shown in Table 16 using the pictograms defined in Table 4. 5 out of 6 architectures on the Pareto front have wings. Since the architectures are ordered from the lighter and riskier to the heavier and safer, the previously discussed influence of the wing on the metrics can be seen graphically. The safer vehicles have wings on their modules. The two safest have an additional second module with wings. The second module adds safety due to separation capabilities. This must be traded-off against the higher mass. This graphical observation confirms the discussion on the left plot a) of Figure 6 above. The vertical take-off mode for a winged module leads to a lighter vehicle since the wings just must be sized for the lighter landing mass (#3, #4). However, the vertical launch method is riskier (confirms discussion on Figure 6 b)). If the vehicle consists of two-modules, having a rocket engine on the first uncrewed module is safer (#14 vs. #9). Comparing #9 and #14, the staging of the two modules in architecture #14 helps to reduce the weight. The rocket engine on the second module increases the risk of having a fire or explosion. Conversely, this hazard can be fairly mitigated in the second module of #9 since there is no engine.

Table 16: The six architectures on the Pareto front

	Module 1	Module 2	
#1 e.g. Copenhagen Suborbital		None	**Lighter & Riskier**
#3		None	
#4 e.g. XCOR		None	
#5 e.g. Rocketplane XP		None	
#14 e.g. Virgin Galactic			**Heavier & Safer**
#9			

Table 15: Overview of the most important design variables

		ORSRM [-]	Launch mass [kg]	nPilots [-]	Rocket propellant module 1	Rocket propellant module 2
#1	Design 1	6.9	4,678	0	Solid	-
	Design 2	6.2	5,053	1	Solid	-
	Design 3	5.6	5,572	2	Solid	-
#3	Design 1	4.7	4,876	1	Solid	-
	Design 2	4.1	5,591	2	Solid	-
#4	Design 1	2.4	6,056	2	Solid	-
#5	Design 1	3.8	6,815	0	LOX/LH2	-
	Design 2	3.1	7,054	1	LOX/LH2	-
	Design 3	3.0	7,147	1	Solid	-
	Design 4	2.4	7,228	2	Solid	-
#14	Design 1	2.7	7,490	1	-	Solid
	Design 2	2.1	8,284	2	-	Solid
#9	Design 1	2.8	9,350	0	LOX/LH2	-
	Design 2	2.2	9,600	1	LOX/LH2	-
	Design 3	1.53	10,002	2	LOX/LH2	-
	Design 4	1.51	11,280	2	Hypergolic	-
	Design 5	1.49	14,431	2	Solid	-

Table 17: Further design variables of the two designs within architecture #5

	#5 – Design 2 LOX/LH2	#5 – Design 3 Solid
Transition velocity	176 m/s	176 m/s
Transition altitude	5.96 km	7.88 km
Isp	450s	296s
Thrust	61 kN	137 kN
Burn time	150s	58s
Chamber pressure	5.6 MPa	4.1 MPa
Nozzle expansion ratio	63.6	61.6
Propellant mass	2,350 kg	3,060 kg
Mass rocket engine	832 kg	339 kg

Since the ORSRM depends not only on the architectural decisions but also on two design variables, there can be more than one design per architecture on the Pareto front (see black dotted horizontal lines in Figure 1 and Figure 4). All un-dominated designs for the identified six architectures on the Pareto front are listed in Table 15 with their ORSRM value, launch mass, number of pilots, as well as their rocket propellant type for module 1 and 2. The designs are numbered, with 1 being the riskiest within an architecture (design with the architectural number next to it in Figure 4).

Since we investigate the trade-off between risk and mass, this riskiest design is also the lightest. The three columns on the right show the design variables which influence the ORSRM.

With an increasing number of pilots, the designs become safer. This makes sense since we have defined in Table 12 that pilots have mitigation potential for some hazards. The propellant for all vertical launches with a one-module vehicle is solid (#1, #3), whereas, for the horizontal take-off #5, the lighter two designs have a liquid LOX/LH2 rocket engine. The safer but heavier two designs are powered by a solid rocket engine. In the following, we compare design 2 and 3 of architecture #5 in more detail (see Table 17 for a summary of further design variables). The system optimization of these two designs shows that the transition velocity is equal but the transition altitude is different (see Figure 2 for definition of transition velocity and altitude). The liquid rocket engine ignites at around 6 km compared to 8 km for the solid engine case. The higher Isp of LOX/LH2 (450s compared to 296s) makes the use of the rocket engine more preferable for a greater energy difference. Despite the lower thrust of the LOX/LH2 engine (61 kN compared to 137 kN), the longer burn time of 150s (compared to 58s) leads to a greater energy increase. In addition, due to the higher Isp of LOX/LH2, less propellant is burned. Even though the liquid engine has a lower thrust with similar chamber pressure and nozzle expansion ratio, the overall complexity is higher than the solid rocket engine and therefore its dry mass is more than twice as much. The sum of the propellant mass and rocket engine weight is 3,182 kg for the liquid case and 3,399 kg for the solid engine. This is the major contribution to the different launch masses in Table 15. Other minor effects are different jet engine sizing, jet fuel consumptions as well as general structural differences. This example shows that due to the lower required Δv for suborbital ballistic trajectories, the higher Isp of LOX/LH2 engines must be traded of precisely against the higher initial mass. This example illustrated how the model and the obtained data can be used to not only understand architectural decisions, but also the technical trade-offs on system level.

5. CONCLUSIONS

This paper aimed to address the question about which system architecture of suborbital space tourism vehicle will provide the best combination of launch mass and safety. We have approached this problem by defining the plausible design space. With our approach, we could find a set of 33 architectures and describe them in a single matrix. Of these, 26 have not earlier been proposed. Based on this description we could optimize each of the 33 architectures with respect to two objective functions, launch mass and safety. To quantitatively evaluate the safety, an approach based on a hazard list was used. Severity and likelihood factors defined the risk of each hazard and a mitigation factor quantified the potential of the individual architectural features to mitigate the hazard. We found that the major contribution to the Overall Residual Safety-Risk Metric is due to the different mitigation potentials of the designs. The depth in modelling allowed us to perform informed trade-offs on system level, like the type of rocket engine. Finally, on architectural level, we could compare the Pareto fronts of the architectures. This leads us to the following conclusions for four-participant suborbital space tourism vehicles:

- Wings are safer but heavier, wings on second module are safer but heavier compared to wings on first module
- Architectures with one rocket engine completely dominate architectures with two rocket engines and they dominated those with two rocket engines and an additional jet engine
- Single-module architectures weight on average around half of two-module architecture but are slightly riskier
- A vertically take-off results in a slightly lighter vehicle, but increases risk by 50% on average
- The six non-dominated architectures #1, #3, #4, #5, #14, and #9 merit a more refined design analysis

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Christopher P. Frank and Prof. Dimitri Mavris for providing the code for the design framework. This work was partially funded by the Heinrich und Lotte Mühlfenzl Stiftung.

REFERENCES

[1] W. J. Abernathy and J. M. Utterback, "Patterns of industrial innovation," *Technology review,* vol. 80, no. 7, pp. 40-47, 1978.

[2] (Oct 30 2016). *Ansari Xprize*. Available: http://ansari.xprize.org/teams

[3] J. Foust, "XCOR lays off employees to focus on engine development," in *Spacenews*, ed, 2016.

[4] J. Foust, "XCOR Aerospace lays off remaining employees," in *Spacenews*, ed, 2017.

[5] M. W. Maier, *The art of systems architecting*. CRC press, 2009.

[6] E. Crawley, B. Cameron, and D. Selva, *System architecture : strategy and product development for complex systems*. Hoboken, NJ: Pearson Higher Education, Inc., 2016, pp. xiii, 465 pages.

[7] G. A. Hazelrigg, "A framework for decision-based engineering design," (in English), *Journal of Mechanical Design,* vol. 120, no. 4, pp. 653-658, Dec 1998.

[8] F. Mistree, W. Smith, and B. Bras, "A decision-based approach to concurrent design," *Concurrent Engineering: Contemporary Issues and Modern Design Tools. HR Parsaei & WG Sullivan (eds.). Chapman & Hall, London,* pp. 127-158, 1993.

[9] O. L. de Weck and M. B. Jones, "Isoperformance: Analysis and design of complex systems with desired outcomes," *Systems Engineering,* vol. 9, no. 1, pp. 45-61, 2006.

[10] J. A. Battat, B. Cameron, A. Rudat, and E. F. Crawley, "Technology Decisions Under Architectural Uncertainty: Informing Investment Decisions Through Tradespace Exploration," *Journal of Spacecraft and Rockets,* vol. 51, no. 2, pp. 523-532, 2014.

[11] M. S. Net, I. del Portillo, B. Cameron, E. F. Crawley, and D. Selva, "Integrated tradespace analysis of space network architectures," *Journal of Aerospace Information Systems,* 2015.
[12] W. Simmons, B. Koo, and E. Crawley, "Architecture Generation for Moon-Mars Exploration Using an Executable Meta-Language," 2005.
[13] R. R. Schaller, "Moore's law: past, present and future," *IEEE spectrum,* vol. 34, no. 6, pp. 52-59, 1997.
[14] C. P. Frank, "Design Space Exploration Methodology to Support Decisions Under Evolving Uncertainty in Requirements and Its Application to Advanced Vehicles," PhD, School of Aerospace Engineering, Georgia Institute of Technology, 2016.
[15] C. P. Frank, R. A. Marlier, O. J. Pinon-Fischer, and D. N. Mavris, "An Evolutionary Multi-Architecture Multi-Objective Optimization Algorithm for Design Space Exploration," presented at the San Diego, California, USA, 57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2016.
[16] C. P. Frank, M. F. Atanian, O. J. Pinon-Fischer, and D. N. Mavris, "A Conceptual Design Framework for Performance, Life-Cycle Cost, and Safety Evaluation of Suborbital Vehicles," presented at the 54th AIAA Aerospace Sciences Meeting, San Diego, California, USA, 2016.
[17] F. Burgaud, C. P. Frank, and D. N. Mavris, "A Business-Driven Optimization Methodology Applied to Suborbital Vehicle Programs," presented at the AIAA SPACE 2016, Long Beach, California, 2016.
[18] N. S.-K. Marti Sarigul-Klijn, "Flight Mechanics of Manned Sub-Orbital Reusable Launch Vehicles with Recommendations for Launch and Recovery," presented at the 41st Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2003.
[19] M. Buckley, K. Fertig, and D. Smith, "Design sheet-An environment for facilitating flexible trade studies during conceptual design," in *Aerospace design conference*, 1992, p. 1191.
[20] X. Huang and B. Chudoba, "A Trajectory Synthesis Simulation Program for the Conceptual Design of a Suborbital Tourism Vehicle," 2005.
[21] J. D. Mattingly, *Aircraft engine design*. Aiaa, 2002.
[22] T. Nam, "Introduction to FLOPS Modeling," U. Lecture, Ed., ed, 2012.
[23] A. S. T. rep., "ASTOS, Aerospace Trajectory Analysis Tool for Launch, Reentry and Orbit Vehicles," G. Tech, Ed., ed, 2008.
[24] F. Villeneuve and D. Mavris, "A new method of architecture selection for launch vehicles," in *AIAA/CIRA 13th International Space Planes and Hypersonics Systems and Technologies Conference*, 2005, p. 3361.
[25] D. O. Stanley, T. A. Talay, R. A. Lepsch, W. Morris, and K. E. Wurster, "Conceptual design of a fully reusable manned launch system," *Journal of Spacecraft and Rockets,* vol. 29, no. 4, pp. 529-537, 1992.
[26] J. R. Olds, "Multidisciplinary Design Techniques Applied to Conceptual Aerospace Vehicle Design," PhD, School of Aerospace Engineering, North Carolina State University, 1993.
[27] D. Kinney, J. Bowles, L. Yang, and C. Roberts, "Conceptual Design of a'SHARP'-CTV," in *35th AIAA thermophysics conference*, 2001, p. 2887.
[28] R. D. Braun, A. A. Moore, and I. M. Kroo, "Collaborative approach to launch vehicle design," *Journal of spacecraft and rockets,* vol. 34, no. 4, pp. 478-486, 1997.
[29] C. Frank *et al.*, "Preliminary Design of a New Hybrid and Technology Innovative Suborbital Vehicle for Space Tourism," 2014.
[30] M. Guerster, "Architectural Options and Optimization of Suborbital Space Tourism Vehicles," M.Sc., Department of Aeronautics and Astronautics, Technical University of Munich, Massachusetts Institute of Technology, 2017.
[31] A. A. f. C. S. T. AST, "Reusable Launch Vehicle Programs and Concepts," 1998.
[32] J. C. Martin, G. W. Law, and United States. Office of Space Commercialization., Suborbital reusable launch vehicles and applicable markets, Washington, D.C.: U.S. Dept. of Commerce, Office of Space Commercialization,, 2002. [Online]. Available: http://purl.access.gpo.gov/GPO/LPS34272.
[33] T. T. Group, "The U.S. Commercial Suborbital Industry: A Space Renaissance in the Making," Federal Aviation Administration2011.
[34] T. T. Group, "Suborbital Reusable Vehicles: A 10-Year Forecast for Market Demand," Federal Aviation Administration2012.
[35] E. Seedhouse, Suborbital industry at the edge of space, Cham: Springer,, 2014, p. 1 online resource. [Online]. Available: SpringerLink http://dx.doi.org/10.1007/978-3-319-03485-0 MIT Access Only.
[36] (Oct 20 2016). *ARCA*. Available: http://www.arcaspace.com/en/haas2b.htm
[37] (Oct 20 2016). *Blue Origin*. Available: https://www.blueorigin.com/technology
[38] (Oct 20 2016). *Virgin Galactic*. Available: https://en.wikipedia.org/wiki/SpaceShipTwo
[39] (Oct 20 2016). *XCOR*. Available: http://spaceexpeditions.xcor.com/spacecraft/
[40] (Oct 20 2016). *Copenhagen Suborbital*. Available: https://copenhagensuborbitals.com/roadmap/spica/
[41] (Oct 20 2016). *Dassault-Aviation*. Available: http://www.dassault-aviation.com/en/space/our-space-activities/aerospace-vehicles/suborbital-vehicles-vehra-vehicule-hypersonique-reutilisable-aeroporte-family/
[42] (Oct 20 2016). *EADS*. Available: http://www.space-airbusds.com/en/dossiers-ea0/the-spaceplane-rocketing-into-the-future.html
[43] (Oct 20 2016). *Cosmocourse*. Available: http://www.cosmocourse.com/sxema-polyota/
[44] (Oct 20 2016). *Scaled Composites*. Available: https://en.wikipedia.org/wiki/SpaceShipOne

[45] F. Zwicky and F. Zwicky, "Discovery, invention, research through the morphological approach," 1969.
[46] G. Pahl and W. Beitz, *Engineering design: a systematic approach*. Springer Science & Business Media, 2013.
[47] A. N. Guest, "Multi-level Decision-based System Architecting," Master, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2011.
[48] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation,* vol. 6, no. 2, pp. 182-197, 2002.
[49] J. D. Anderson, *Aircraft performance and design*. Boston: McGraw-Hill, 1999, pp. XVI, 580 S. graph. Darst.
[50] saabstory88. (2015, March). *Estimated masses and dimensions of the New Shepard PM. Estimates based on Thrust and Volume*. Available: https://redd.it/3u8m2h
[51] mlindner. (2015, March). *Blue Origin - New Shepard second developmental test flight and landing*. Available: https://forum.nasaspaceflight.com/index.php?topic=38873.280
[52] G. Norris. (2009, September 7) Galactic. *Aviation Week & Space Technology*.
[53] P. Vis. (Feb 20). *SpaceShipTwo Specifications*. Available: http://www.petervis.com/interests/published/Spaceshiptwo/Spaceshiptwo_Specifications.html
[54] I. Adventures. (March). *Suborbital Space Flights: The Space Plane*. Available: http://www.incredible-adventures.com/rocketplane-xp.html
[55] M. Wade. *Rocketplane XP*. Available: http://www.astronautix.com/r/rocketplanexp.html
[56] R. G. Inc. (20 Feb). *Rocketplane XP*. Available: http://rocketplane.ca/spec_and_description.html
[57] N. Leveson, *SafeWare : system safety and computers*. Reading, Mass.: Addison-Wesley, 1995, pp. xvii, 680 p.
[58] N. Leveson, *Engineering a safer world : systems thinking applied to safety* (Engineering systems). Cambridge, Mass.: MIT Press, 2011, pp. xx, 534 p.
[59] N. Dulac and N. Leveson, "Incorporating Safety Risk in Early System Architecture Trade Studies," (in English), *Journal of Spacecraft and Rockets,* vol. 46, no. 2, pp. 430-437, Mar-Apr 2009.
[60] H. E. McCurdy, "Inside NASA: High technology and organizational change in the US space program," *New Series in NASA History, Baltimore: Johns Hopkins University Press,| c1993,* 1993.
[61] R. Godwin, *Columbia accident investigation report*. Burlington, Ont.: Apogee Books, 2003.
[62] NASA, "Methodology for Conduct of Space Shuttle Program Hazard Analyses," 1993.
[63] NASA, "Constellation Program Hazard Analyses Methodology," 2009.
[64] U. Walter, *Astronautics the physics of space flight*, 2., enl. and improved ed. Weinheim: Wiley-VCH, 2012, pp. XXVIII, 568 S.

BIOGRAPHY

Markus Guerster *is a graduate student pursuing his PhD degree in the System Architecture Lab of the Department of Aeronautics and Astronautics at MIT. His interests include the development of tools to support the decision-making process for complex engineering systems under uncertainty. He received a M.Sc. degree (2017) with high distinction in Aerospace Engineering as well as a B.Sc. degree (2015) in Mechanical Engineering from the Technical University of Munich. He worked as a System Engineer at OHB System AG for 2 years besides his studies and recently conducted a scientific study with in-tech GmbH to transfer aerospace methodologies to the railway industry.*

Edward F. Crawley *is the Ford Professor of Engineering, and a Professor of Aeronautics and Astronautics at MIT. He has served as the founding President of the Skolkovo Institute of Science and Technology (Skoltech) in Moscow, the founding Director of the MIT Gordon Engineering Leadership Program, the Director of the Cambridge (UK) MIT Institute and the Head of the Department of Aeronautics and Astronautics at MIT. Dr. Crawley is a Fellow of the AIAA, the Royal Aeronautical Society (UK) and a member of the International Academy of Astronautics. He is a member of five national academies: in Sweden, the UK, China, Russia and the US. He received an S.B. (1976) and an S.M. (1978) in aeronautics and astronautics and a Sc.D. (1981) in aerospace structures, all from MIT, and has been awarded two degrees of Doctor Honoris Causa. Crawley's research has focused on the architecture, design, and decision support and optimization in complex technical systems subject to economic and stakeholder constraints. His work ranges from the development of underlying theory to the development of models for real systems. His recent book – System Architecture: Strategy and Product Development for Complex Systems – was published by Pearson (2016).*