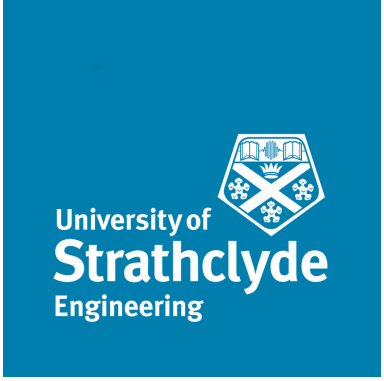


Department of Mechanical and Aerospace Engineering

University of Strathclyde, Glasgow

Toward automated design of Combined Cycle Propulsion.

Alessandro Mogavero

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

2016

This thesis is the result of the authors original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree. The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Abstract

One means to reduce both the cost and the risk associated with space missions is to employ a vehicle designed within the re-usable, airliner-like 'space plane' paradigm. Key to the practicality of such vehicles is the further development of Combined Cycle Propulsion technology. In this thesis, a new engineering tool called the HYbrid PRopulsion Optimizer (HyPro) is presented, with the aim of analysing the performance of diverse engine concepts. The tool is conceived to be modular and flexible, and makes use of parsimonious modelling, in order to describe the engine at an high level of abstraction and to be fast in execution.

A configurational optimizer has also been developed in order to automatically generate new design concepts, optimizing the engine cycle structure. It is conceived to be used at the beginning of development in order to perform an automatic and objective trade-off of possible propulsion solutions.

In this work the model has been implemented for Rocket-Based Combined Cycle, and it has been verified and validated against analytical models, computational fluid dyanamic analyses and experimental data. The design proposed by the optimizer in these conditions was coherent with manually designed Combined Cycle Propulsion engines, demonstrating the HyPro's capability to converge on good solutions.

The results, although preliminary, are very promising and represent a novelty in the field, since a configurational optimization, in the field of propulsion concepts, has never been attempted before. The results presented here demonstrate that the configurational optimization of engine design is viable.

The next steps to produce a practical optimizer, which delivers robust and innovative engine solutions, are the addition of modelling capabilities beyond the Rocket-Based Combined Cycle and analysis discipline beyond the pure performances.

Acknowledgements

After my first work experience in industry, this PhD has been a really pleasant coming back to the world of academia. First of all I wish to thank Prof. Richard Brown for encouraging me to take this step in the first place. His guide has been fundamental in yielding the results presented here and in building in me the right confidence. I have really appreciated his trust in me and my expertise, demonstrated by letting me supervise many of his students, thus allowing me to gain experience in teaching and supervising.

During these three years I enjoyed the work environment, mainly thanks to my colleagues at the Centre for Future Air-Space Transportation Technology. We have been more friends than colleagues, even if having different PhD projects we have always exchanged information and suggestions, so that part of this work is also their accomplishment. In particular I would like to thank Jimmy John Hoste for giving me the undeserved nick name of "Master" that makes me very proud, to the point of making me a little presumptuous. That nick name is meant to underline my tendency to help and giving suggestions (perhaps underlying how I do it?), it has however been for me a pleasure to help my colleagues and therefore I thank them all for letting me do it. I thank also Javier Herrera for every time he helped me with Linux and the computer in general, I guess he enjoys helping as much as me, but I want to thank him anyway for all the things he taught me (also when hacking my computer). Thank you also to Federico Toso and Viola Renato for making our office a bit more Italian (I was clearly not enough alone), moreover I will never end to thank them for their perfect hospitality in time of trouble. I would like to thank also Robert Garner especially

for his help in correcting this thesis. He has been fundamental to give decency to my poor English. Finally I thank Mark De Luca, one of the undergraduate students I supervised, for helping me in the validation of the model presented in this thesis. I would like also to send a special thank you to all my other colleagues not mentioned here for a matter of space.

The results achieved at work however would have never been possible without the support of my family. My parents have always supported me in my studies, providing all the resources when I was not autonomous and moral support after I found my first job. Moreover I had the privilege to grow up as the son of an engineer, experiencing the beauty of this profession since my childhood. I have to thank my father if I could disassemble my first engine at 14 or build my first model aircraft at 10 (citing only few of these experiences). While I have to thank my mother for letting me do all these 'dirt producing tasks' at home, even if she knew she was going to clean afterwards. Special mention here is due also to Fabio Mogavero and Sara Mogavero, respectively a cousin of mine and my sister, with which I made most of the aforementioned 'dirt producing tasks'. It was Fabio that first showed me the world of computer programming, letting me develop a skill vital for the modern world and this little PhD effort.

A special final thanks goes to the first component of my newest family, my wife Francesca Romana Auciello. She is by a matter of fact the main reason why I boldly choose this career path. Without her I probably would not have been so bold, I would do the same job in the same place and I would not have had the amazing experience I had. She has always pushed me to follow my passions, always telling me what I wanted to hear: that I am good. She is there when everything goes well, but she is there especially when things do not go smoothly as they should. If I arrived at this point of my life with this confidence and peace of mind is because I know that whatever happens she is with me. I cannot wait to go on the next chapter of my life with her!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The space vision

Space exploration is certainly fascinating to both experts and the general public. Probably many of the engineers and scientists that decide to dedicate their life to this job do that in order to satisfy their hunger for the exploration of the very last frontier. Although it is a very romantic and noble motivation, in applied research the paradigm of curiosity driven research cannot justify the amount of resources needed. New York Congressman Ed Koch once stated: "I just for the life of me can't see voting for monies to find out whether or not there is some microbe on Mars, when in fact I know there are rats in the Harlem apartments." [17]. The rhetoric of that politician well explains the thoughts of the general public about space activities, albeit scientists recognize unanimously the importance of studying the planetary evolution of other planets in order for instance to predict that of the Earth. Beside politicians, critics of the space exploration effort have come even from the most innocent sources. Sister Mary Jucunda, a nun who worked in Zambia among starving children, questioned the value of space exploration and asked Ernst Stuhlinger, Associate Director for Science at the Marshall Space Flight Center, why he wanted to spend billions for a voyage to Mars when children were starving on Earth. Stuhlinger answered the nun with a famous letter, in which he

explained many reasons why a voyage to Mars may help the Earth more than many humanitarian programs [18]. Recently I had the pleasure of explaining the advancements of European launcher programs to a class of primary school children in Paisley, near Glasgow. Even the children, among the various curiosities, concerned asked me: "How much does it cost?".

Engineers and scientists must therefore have a clear space vision of the future of space exploration and its impact on our lives. I believe that the arguments of technological dissemination, around which the letter of Stuhlinger was centred [18], are not enough to convince the general public, although I am convinced that alone would justify the investment. The scientist that is first in line in advocating a primary objective of space exploration different than technological dissemination is Stephen Hawking. He firmly believes humanity should colonize other worlds in order to avoid extinction. In many TV shows and interviews he said: "The human race shouldn't have all its eggs in one basket, or on one planet. Let's hope we can avoid dropping the basket until we have spread the load." [19]. Colonization might seem a very long term objective, but considering the pressure we are imposing on Earth today and the trend of population growth, it might be actually more immediate than we think.

A very comprehensive vision for the future of space exploration and exploitation is given in Fig. 1.1. A whole space infrastructure is foreseen, including a space station, fuel station spaceport network, orbital servicing vehicle, a lunar spaceport, even a waste storage, some manufacturing facilities and colonies. Such an infrastructure could trigger a whole new economy based in space, with potentially an unthinkable amount of resources and possibilities to be exploited. It is worth mentioning that Johann-Dietrich Woerner, new Director General of the European Space Agency (ESA), recently, proposed to build a village on the far side of the moon [20]. He proposed to use the moon as a stepping stone for a base on Mars, in order to test more easily the technologies that would be needed for a long presence on another planet. These and other signals, such as the commitment to asteroid exploitation of the Obama administration, might mean the time is ready for the aforementioned vision to be put into practice. Only time will tell.

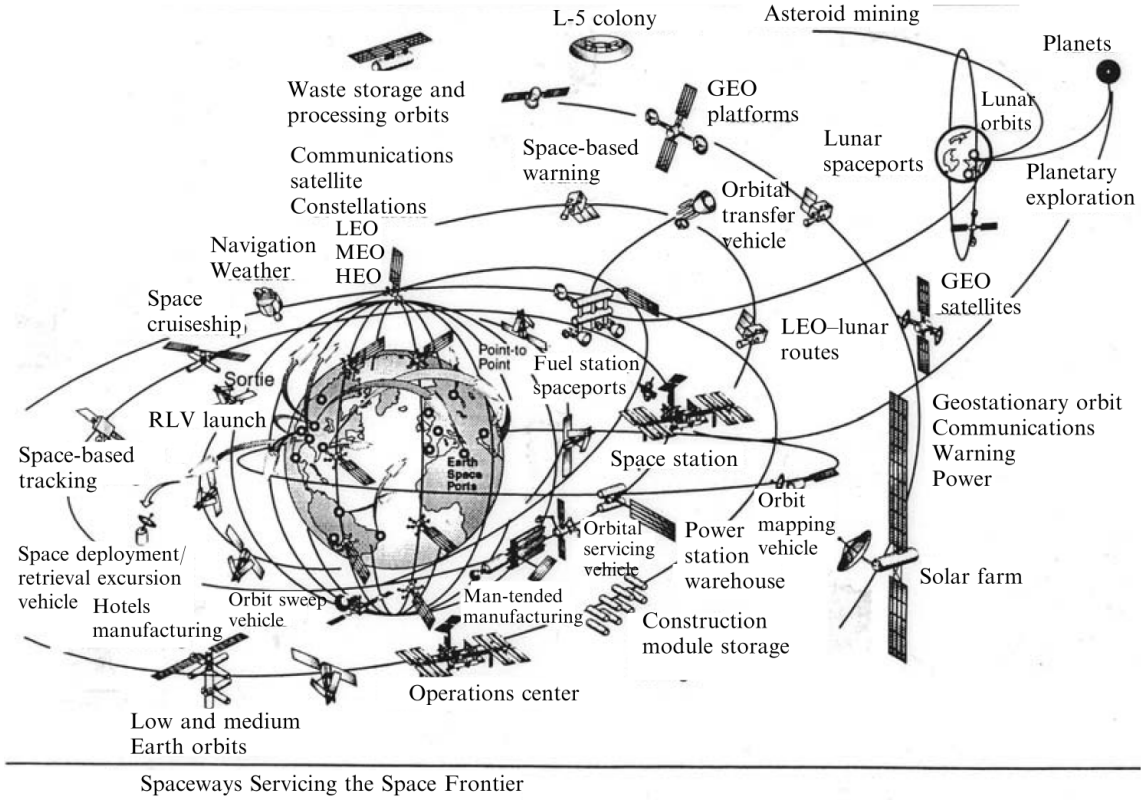


Figure 1.1: Space infrastructure envisioned by Dr William Gaubatz when director of the McDonnell Douglas Astronautics Delta Clipper Program. [1]

1.2 The first 400 kilometres

1.2.1 The problem

The space infrastructure depicted in Fig. 1.1 is almost impossible to build if the cost of access to space is not drastically reduced. Paradoxically the bottle neck of the space exploration remains the first 400 kilometres that separate the surface of the Earth from the Low Earth Orbit (LEO). For instance in the case of the Space Shuttle about 29,000$ was required per pound of payload delivered to LEO [1].

Many studies have however shown that, as often happens in industrial production, the costs decrease substantially if the launch rate increases. In Fig. 1.2 for instance the cost to launch a unit weight of payload to LEO is plotted for a Two Stage to Orbit (TSTO) vehicle [1, 2]. In Fig. 1.2 the total cost is also split into several categories: propellant, infrastructure, insurance, maintenance, production and Research, Development, Technology and Engineering (RDT&E). It is evident that the main recurrent cost is obviously the propellant. It however becomes predominant only at a launch rate much bigger than the practice today (i.e. less than 10 launches a year). At 10 launches a year the cost of this TSTO is still the same order of magnitude of that of the Space Shuttle, mainly due to the infrastructure cost, which will always be present no matter which launch technology is selected.

Due to the aforementioned reason many sceptics are convinced that the effort to reduce the space access cost is worthless, simply because the demand for such an amount of launches doesn't exist. Czysz and Bruno in [1] notice however, also that when the US Transcontinental Railroad was built there was no market for its usage. The commerce between the west and the east of the United States flourished only after the Railroad was completed. To be completely honest this comparison is not very strong, since in space there are not colonies yet, whereas in the West, at the time Railroad building effort started, there were plenty of colonies to trade with. It is however true that "all of projections of future space business for expendable or limited reuse launchers are as valid for future

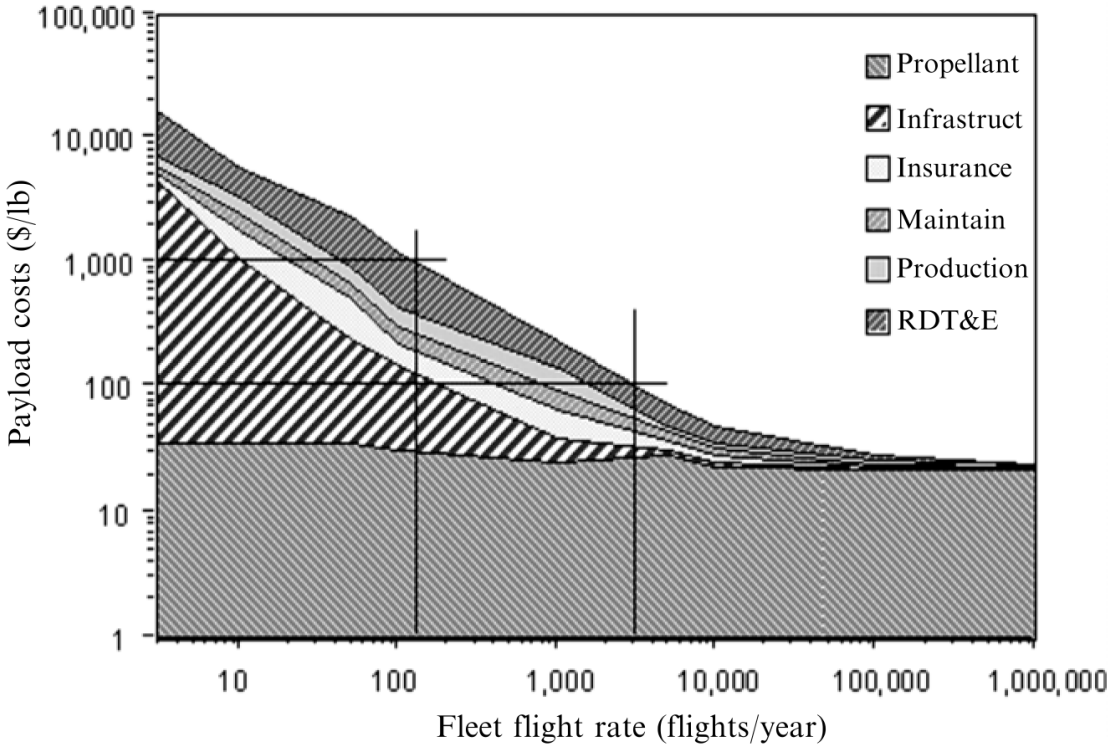


Figure 1.2: Cost estimate for a TSTO launcher by Jay Penn and Dr Charles Lindley [1, 2].

space business as the business projections for the future Railroad based on Conestoga wagons" [1].

Probably a better projection of the future market for space access is given by the Commercial Space Transportation Study (CSTS), conducted by National Aeronautics and Space Administration (NASA) and other aerospace companies, during 1993–1994 [3]. In the CSTS the demand increase with a possible reduction in the cost of access to space was predicted considering many different potential markets: communications, space manufacturing, remote sensing, government missions, new transportation service sectors (including space tourism and hazardous waste disposal in space), entertainment, new missions (including a space business park), space utilities (in particular, space solar power for use on Earth); commercial utilization of extraterrestrial resources and commercial advertising.

The results of the CSTS study are reported in Fig. 1.3. As can be observed the demand is predicted to increase drastically if the cost of access to space is reduced. A space access cost of 1,000$ per pound in the worst case scenario would stimulate a demand of more than 1 million pounds per year which, depending on the payload size, would require between 25 and 250 flights per year. This flight rate in turn can justify a space access cost ranging between 700$ and 4,000$ per pound, thus confirming the starting hypothesis (see Fig. 1.2).

This supply demand comparison is obviously far from conclusive, due to the high uncertainty of the data. In general it is very difficult to predict the costs of a complex system such as launch transportation and the market forecasts are even more difficult, due to the long time delay between the time the market study is performed (usually at the inception of the launch system development) and the time the launch system is operational. Compared to the Transcontinental Railroad, a space transportation infrastructure with high flight rate is a much more risky investment even if, as Czysz and Bruno observe in [1], it is much smaller in terms of total resources invested.

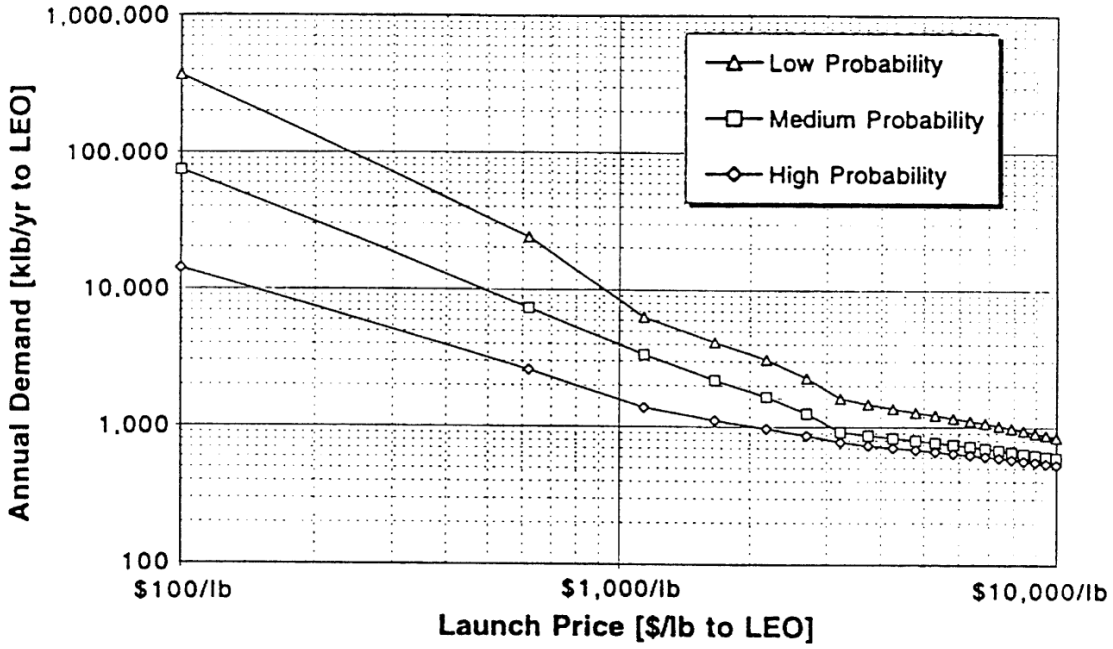


Figure 1.3: The elasticity of the Earth-to-Orbit launch market with price (CSTS Study) [3].

1.2.2 The solution

As explained in section 1.2.1, the main road to low space access cost is increasing the flight rate. The debate about the launch system that is best suited to achieve this goal is, however, very much open. In 1985 NASA and the Department of Defence (DOD) studied a common plan, called Space Transportation Architecture Study (STAS), for the development of a second-generation of space transportation systems to follow the Space Shuttle [3]. Among the various findings, the STAS concluded that Expendable Launch Vehicles (ELVs) are not competitive for low cost launch and that air-breathing propulsion technology deserves the effort paid for its maturation. However, NASA and the DOD, probably influenced by tragedy of the Challenger, went back toward the ELV concept, with two new programs with the same aim of STAS, dubbed Advanced Launch System (ALS) and National Launch System (NLS) [3]. This time they concluded that low cost could be achieved using ELVs, if the system procurement rates were increased by means of common technologies and by

systems commonality between vehicles of various sizes. A similar strategy has been more recently employed for the new launchers foreseen by ESA [21]. Ariane 6 will employ a modular architecture using either two (Ariane 62) or four (Ariane 64) P120 solid-propellant boosters. While VEGA-C, an evolution of VEGA, will use the same P120 as first stage. Time will tell if this will be sufficient to increase the launch frequency and consequently lower launch costs.

This abrupt change in strategy adopted by NASA and also followed by other agencies such as ESA surely demonstrates that the debate is open, since the prevalence of ELV architecture seems only a reaction to the problems of the Space Shuttle and a strategy to minimize risks. The Space Shuttle has indeed been the only example of a partial re-usability in space transportation, that many consider a proof of the non-viability of the Re-usable Launch Vehicle (RLV) concept. The Shuttle program was however undertaken during an historical period in which "the mood of the country demanded major cuts in the space program" [17]. As a consequence NASA had to compromise on the Shuttle configuration in order to meet very strict requirements in terms of development costs and to satisfy some specifications dictated by the DOD, whose backing was vital to receive political commitment. The result was a Shuttle not able to meet the launch frequency required to have really cheap access to space and it was actually even more expensive and less reliable than ELV. It is of course impossible to be sure that a different Shuttle, a fully re-usable TSTO as was originally conceived by NASA, would have been more successful or even possible in the first place. The bad final results of the Shuttle experience (also this severe judgement is debatable) were however not only caused by technical problems, but in large part by a lack of political commitment and also by the NASA failure to engage with a changed society [17].

Despite this general disbelief toward re-usability followed from the problems of the Space Shuttle, many still believe that the future is in RLV. In 1995 for instance NASA conducted a study called Highly Reusable Space Transportation (HRST), with the clear aim to find new concepts based on the RLV, that would allow low cost space access [3]. Several different strategic approaches were considered and analysed separately, a solution using ELV was also analysed in order to take into

consideration the possibility that the idea at the base of the project was wrong. The results of HRST priced the solutions using advanced Combined Cycle Propulsion (CCP) discarding Combination Propulsion System (CPS) (see section 1.3), all rocket systems and also ELV.

CCP is advocated by many as the only way to really reduce the launch cost and have reusable systems. But the debate is also in this case open and the convenience of such a technology depends strongly on the details of its design. Recently at Japan Aerospace Exploration Agency (JAXA) research and development activities are being performed on a Rocket-Based Combined Cycle (RBCC) (see section 1.3) for a TSTO vehicle [22]. RBCC is also considered as the main propulsion system of the first stage of a TSTO vehicle at the Northwestern Polytechnical University in China [23] and as propulsion of a Single-Stage to Orbit (SSTO) by Olds et al. in [24]. Others instead prefer to consider all rocket RLV as a more feasible intermediate solution. At the German Aerospace Centre (DLR) for instance an all rocket fully reusable TSTO concept is under development [25]. While NASA in 1994 pointed at an all rocket SSTO with the Venturestar vehicle and its small scale demonstrator X-33 [26]. However, the program was closed after the occurrence of major problems with the cryogenic tank development [27].

A much simpler solution to obtain a partially RLV is today pursued by private companies, such as Space-X. The first stage of a conventional ELV is designed to be recovered after launch and then reused. Recently the first stage of the Falcon-9 launcher, developed by Space-X, has been recovered successfully. The actual viability of this strategy is, however, still debatable and it will be judged over time.

1.2.3 Why we are still waiting?

As mentioned in section 1.2.2, the only partially reusable space transportation vehicle ever flown is the Space Shuttle. Despite all the effort, today the only means of access to space is through old fashioned ELV. So the question is why, after decades of efforts and billions of dollars spent, are we still using the same technology as the space pioneers?

From my point of view the answer to this question lays in the fact that moving from ELV to SSTO or fully RLV is a major technological leap. The resources and the time required to develop these technologies are enormous and the risk linked with such an endeavour is very high. Moreover the world today is very different to that of the cold war, and the political commitment to space exploration is not what it was half a century ago.

The challenge is thus to find a strategy capable of gathering the required political backing and at the same time produce advances in the technology. In the USA for instance, the National Aero-Space Plane (NASP) program was intended to deliver a SSTO by the 1990 [28]. It was designed to use CCP in order to be very efficient and reusable. The program however was cancelled in 1994 after more than 2 billions dollars were spent. Major technological breakthroughs were required in almost every field of its development, thus the costs skyrocketed with few practical results. The consequence of this failure was only an increased scepticism toward RLVs affecting the funding of future programs.

In the same period the director of NASA, Daniel Goldin, came up with a set of rules aimed to avoid similar problems in the future. Namely, the Goldin's rules defined the "faster, better, cheaper" approach opposed to the NASP or Space shuttle approach of huge, monolithic programs. Goldin's approach was initially designed for space probe missions, especially after the Cassini mission that ended up costing more than the Space Shuttle. "Faster, better, cheaper" objectives were implemented, imposing tight limits on the cost and schedule and accepting a greater risk. This led to the launch of many more scientific probes but, as a result of the higher risk, the failure rate increased [28].

Goldin then proposed employing his approach also for space transportation development, simplifying the requirements of the programs. Simple flight test demonstrators took the place of full scale vehicles such as those planned for NASP. The purpose of these programs was to gradually build the knowledge of the technologies required for hypersonic flight. The first program following this approach was the X-43A Hyper-X [28, 29]. The X-43A was a scramjet powered hypersonic vehicle

launched from a B52 and accelerated to hypersonic speeds by means of a solid rocket booster. The requirement was simply to perform a very short (matter of seconds) sustained hypersonic flight, with secondary objectives only added once the primary was successfully accomplished. In this way the costs were kept acceptably low and the program was considered a success. After X-43A a new similar program, called X-51A, with improved requirements was designed, and it is now still on going [30].

The aforementioned strategy is being pursued also by other nations. A joint effort of the U.S. Air Force Research Laboratory and the Defence Science Technology Organisation in Australia aims to investigate hypersonic flight with the program Hypersonic International Flight Research Experimentation Program (HIFiRE) [31]. In HIFiRE a scramjet powered hypersonic test vehicle is launched by means of a sounding rocket, thus lowering the costs even more than the X-43A approach. Similarly in Europe the Intermediate eXperimental Vehicle (IXV) program, meant to enhance Europe's knowledge of re-entry technologies, was recently concluded successfully [32]. The UK is investing in the development of new propulsion technologies, focusing for now only on the technological demonstration of some key components (i.e. the heat exchanger, see also 1.3.3), with the final long term goal of developing a SSTO.

In summary, the technology breakthroughs required for the development of RLVs, need large investments of time and resources characterised by high risk. Since the socio-political environment today makes it difficult to secure funding for such an endeavour, the common approach is to break the development of RLV into many technological milestones. Each milestone by itself does not deliver a technology ready to be employed in the actual space industry, but it does give a sensible outcome that demonstrates an advancement to both the experts and the general public. This strategy is probably more efficient in securing funding than that of the NASP or of the Space Shuttle, but it inevitably delays the delivery of the final product. Prediction of when we will actually have a RLV is almost impossible, since it depends on the amount of political commitment to the cause. So the answer to the question asked at the beginning of this section might just be

that we need to be patient.

1.3 Combined Cycle Propulsion

For hypersonic vehicles, with the exception of all rocket solutions, many different propulsion technologies are required to operate on the same vehicle, since the flight envelope spans multiple regimes (i.e. subsonic, supersonic, hypersonic or empty space). In the case where different engines employing different technologies are separately installed on the vehicle, the propulsion system is dubbed Combination Propulsion System (CPS). Where many propulsion technologies are integrated into a single engine that is able to operate in different propulsion modes, the propulsion system is called Combined Cycle Propulsion (CCP) [33, 4].

CCP are usually preferred over CPS because they combine different propulsion technologies with less of an increase in engine dry mass. On the other hand the CCP solution is more complex and therefore requires more effort in development than a CPS one. CCP are indeed equipped with many mechanisms required to adapt the engine at a wide range of operative conditions and used during the engine mode switches. Beside increasing the overall development cost this complexity can produce reliability issues. The CPS solution however is usually not more reliable than CCP, since it requires the installation of several different simpler engines instead of a single complex one. Therefore, also from this point of view, CCP is usually preferred (see for instance the study in [3]).

Until today the only example of CCP ever flown is the J-58 engine mounted on the Blackbird [33, 6], however CPS solutions are even more rare (ignoring scramjet/ramjet test vehicles).

As explained in section 1.2.2, CCP is very well suited for space transportation applications, it is also very much preferred for future applications in hypersonic airliner concepts [34, 35].

1.3.1 Rocket-Based Combined Cycle

RBCC engines are conceived as an improvement of a rocket engine, which is the core of the cycle. In the most simple RBCC engine the rocket is framed into a duct, so that the mass flow moved by

the rocket is augmented with a secondary flow coming from the atmosphere. As in any propulsive system, the augmented mass flow increases the propulsive efficiency of the engine, using also the atmosphere as propellant and not only the propellant stored aboard. The process is equivalent to the concept of a bypass turbojet, but in this case momentum is transferred to the secondary flow simply by means of friction between the rocket plume and the air (ejector effect) rather than using a fan. RBCC can however be more complex than that adding a post combustor and thus the possibility to operate also in pure ramjet mode.

An example of RBCC is given by the early study of Marquardt about possible CCP configurations [8]. The engine, called Supercharged Ejector RamJet (SERJ), was envisioned to form the propulsion system of the first stage of a TSTO vehicle (see Fig. 1.4). An overall drawing of the engine is presented in Fig. 1.5. In Fig. 1.5 the inlet is omitted for clarity, it was conceived as a highly variable geometry inlet in order to adapt all over the wide operational range of the engine (see Fig. 2.5 for details). The engine uses liquid hydrogen and liquid oxygen as propellants.

The SERJ was developed specifically to work efficiently over a wide range of flight Mach numbers by changing its configuration to allow operation in one of several different modes: supercharged ejector mode, fan ramjet mode, pure ramjet mode, pure rocket mode and fan operation mode. At fixed point and low speed the SERJ operates in ejector mode. In ejector mode the main source of thrust is the rocket primary. Air from the inlet and fuel from the secondary fuel injectors is also burned in the post-combustor, the tip-driven fan located past the inlet is also active in order to increase the engine internal pressure. Once significant internal pressure is achieved, in supersonic regime, the rocket primary is shut while the fan remains operative, constituting the fan-ramjet mode. At higher speed, the fan is removed from the flow path, thus the engine operates at pure ramjet mode. At speed higher than the maximum operative range of the ramjet and in rarefied atmosphere or open space the post combustor is shut, the inlet is closed and the primary is activated again, thus the engine operates in pure rocket mode. In the fan operation mode only the fan is active providing thus a very low thrust, this operation mode is foreseen mainly for loiter phases.

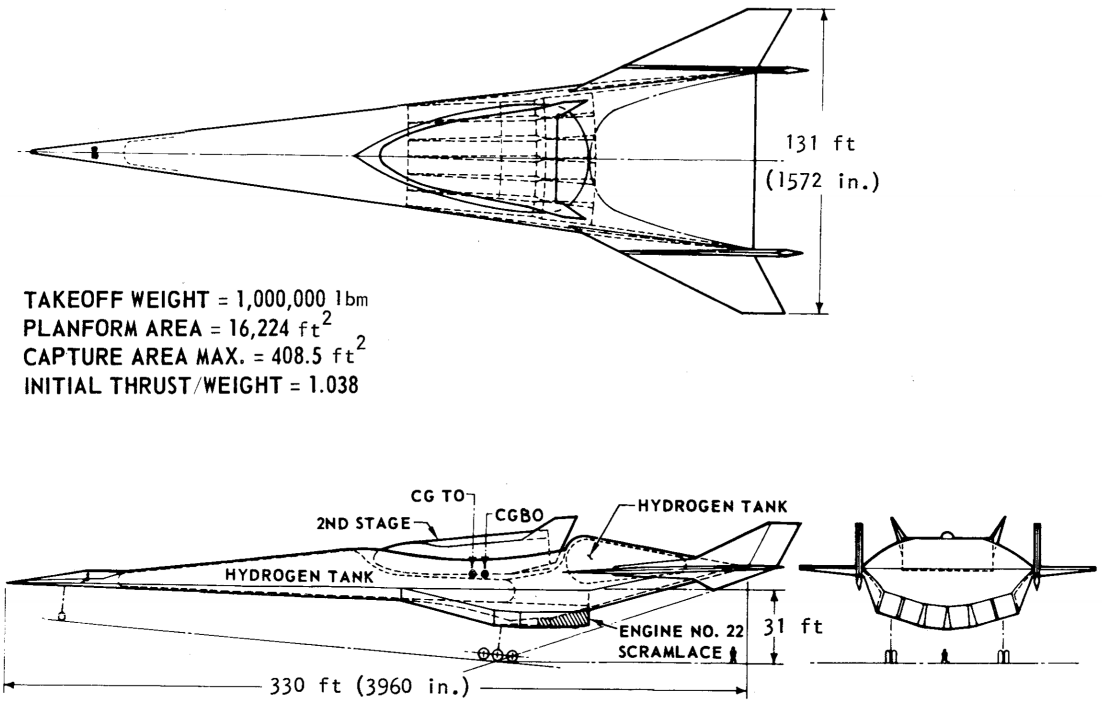


Figure 1.4: TSTO vehicle proposed by the Marquardt Corporation [4].

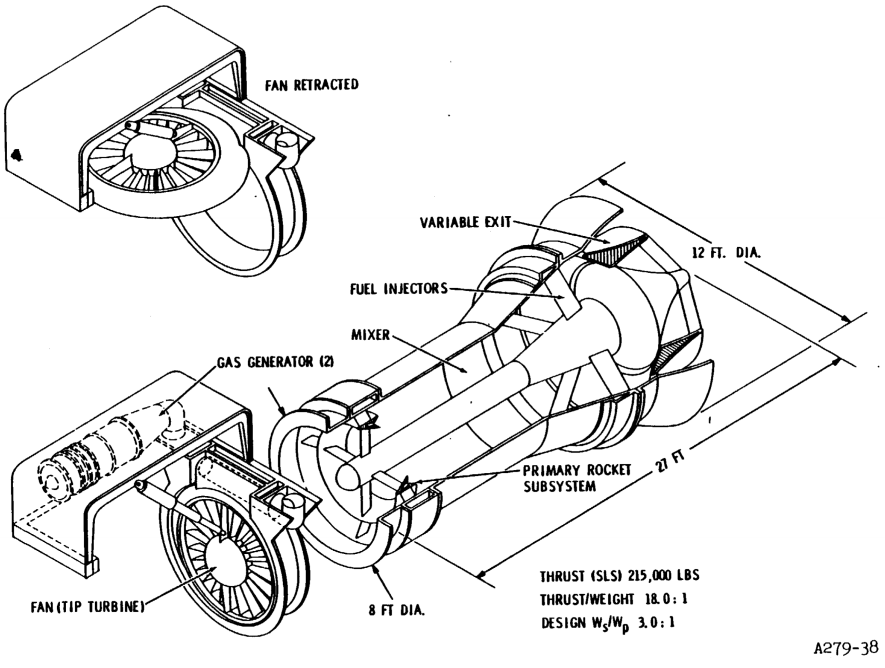


Figure 1.5: Overall view of the Marquardt SERJ engine[5].

Here, it is worth underlining the tip driven technology foreseen for the fan/gas-generator assembly. Two relatively small gas generators are installed just upstream the fan, the hot gas generated is then expanded at the tip of the fan that thus function as turbine. With this architecture a shaft between the fan and its driver is not needed and therefore the fan can be easily hinged to be removed when required. This solution is very attractive because the fan can be completely removed from the flow path when not needed, thus avoiding any pressure loss due to a wind milling architecture. The down side of this approach is however that only small power fans can be installed with this solution.

1.3.2 Turbine-Based Combined Cycle

In Turbine-Based Combined Cycle (TBCC) engines a conventional turbojet engine is coupled with other propulsive technologies such as the ramjet or the scramjet or both. Usually the turbojet core is used at low speed where the ram compression is not enough to guarantee propulsion. At higher speed the turbojet is then shut down, wind milled or simply throttle down and the main thrust is provided by the ramjet.

The classical example of TBCC engine is the J-58, which was installed on the Blackbird, a vehicle designed for military reconnaissance [6]. The J-58, thanks to its innovative design, was able to provide the required high endurance at high speed (Mach 3) and altitude, with a performance that is still unbeaten today. The J-58 configuration was defined a turbo-ramjet, since it coupled a turbojet core engine with a post combustor. The turbojet engine core could be bypassed in favour of the post combustor, thus letting the engine function as a ramjet.

In Fig. 1.6 the engine operations from fixed point to cruise speed are showed. The engine was equipped with a movable inlet spike followed by a set of bypass doors. The spike was used to control the position of the oblique shock, while the forward bypass doors, placed downstream the inlet, were used to control the position of the normal shock in the inlet divergent. A set of after bypass doors where then used to control the air flow passing through the core turbojet engine and

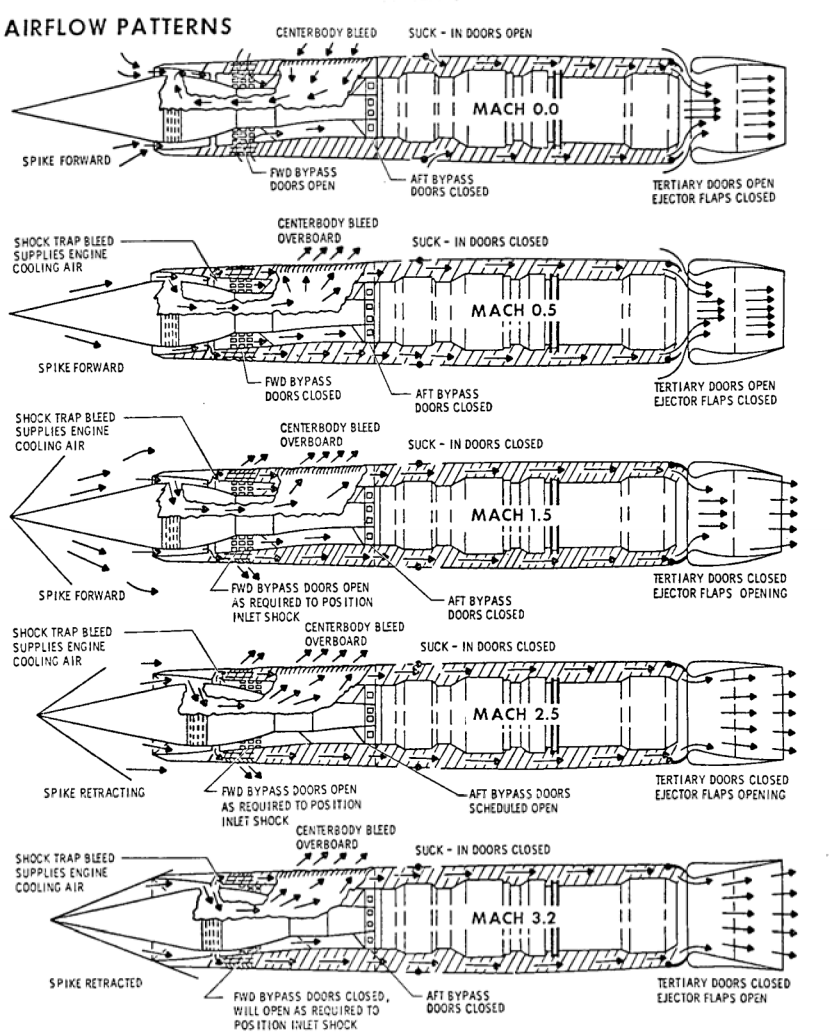


Figure 1.6: JT11D-20 (J-58) Turbo-Ramjet engine, installed on the Blackbird. Operations from 0 to cruise Mach number [6].

were operated by the pilot. At cruise speed most of the airflow bypassed the core turbojet engine and travelled directly to the post combustor. At those conditions the turbojet core provided only 17% of thrust.

Also in this case the engine presents a certain amount of further complications with respect to a conventional turbojet. Despite this the CCP solution was preferred over a CPS, due to the advantages described in section 1.3.

1.3.3 Liquid-Air Cycle Engine and pre-cooled engines

In Liquid-Air Cycle Engine (LACE) the cryogenic properties of the fuel are used to liquefy (or only to cool down) part of the air swallowed by the inlet. The procedure can be used to provide oxidant for a rocket engine subsystem or simply to reduce the temperature of the incoming flow, which is usually very high at hypersonic flight conditions.

An historical example of LACE configuration is the ScramLACE engine, proposed by the Marquardt Corporation as propulsion system for the first stage of a TSTO vehicle (see Fig. 1.4). The ScramLACE engine can be considered as the evolution of the SERJ engine also developed by the Marquardt for the same vehicle (see section 1.3.1). As in the SERJ concept, the ScramLACE engine was meant to employ liquid hydrogen and liquid Oxygen. The engine was conceived to operate in four progressive modes: liquid air cycle ejector mode; ramjet mode; scramjet mode and pure rocket mode.

An overall view of this engine is showed in Fig. 1.7. The most innovative engine's component was the air liquefaction unit consisting of a precooler and a condenser. The air liquefaction unit could be bypassed by means of a bypass door during the high speed modes. The primary rocket assembly comprised of a set of vertical two-dimensional rockets, that were also the support of the injection system for the scramjet mode. After the mixing section and diffuser another series of vertical struts was foreseen for the injection of the afterburner and subsonic ramjet mode. The nozzle throat area was controlled by means of four vertically hinged panels.

At low speed the ejector was used to provide most of the thrust, similar to the SERJ engine. In this case however the liquid oxygen needed by the rockets was not stored aboard but it was instead extracted from the atmosphere by means of the air liquefaction unit. With this strategy a substantial amount of propellant could be saved in the first phases of the flight. On the other hand the air liquefaction unit constituted a relevant inert mass during the subsequent high speed phases.

Today air liquefaction is usually avoided in CCP engines, due to the enormous amount of heat removal required by the state change and consequently the flow of hydrogen required. LACE

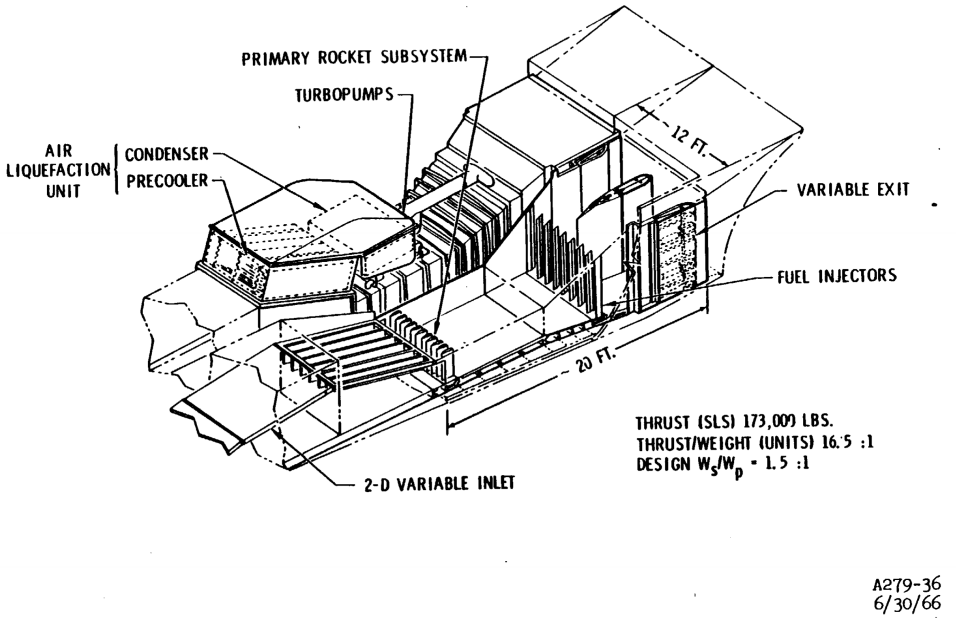


Figure 1.7: Overall view of the Marquardt ScramLACE engine[5].

like configurations are still very relevant today, but the air liquefaction is often substituted with a simple cooling, so usually the engines are dubbed pre-cooled [36]. A famous example of this approach is the Synergetic Air-Breathing Rocket Engine (SABRE) developed by Reaction Engine Limited [36, 37, 38]. A further example is given by a pre-cooled turbojet engine developed by JAXA for an hypersonic transport aircraft [34, 39].

1.4 Motivation

The development of a launch vehicle is very complex mainly due to the various disciplines and subsystems, that are all closely coupled. From the beginning of the program not only all the aspects of the design, the aerodynamic, the structure analysis, the thermal control, but also the life cycle analysis such as the reliability or the operations have to be considered together with their mutual coupling in order to avoid a disastrous outcome later on in the development.

Moreover it is vital to carefully weigh any decision taken at the beginning of the design avoiding

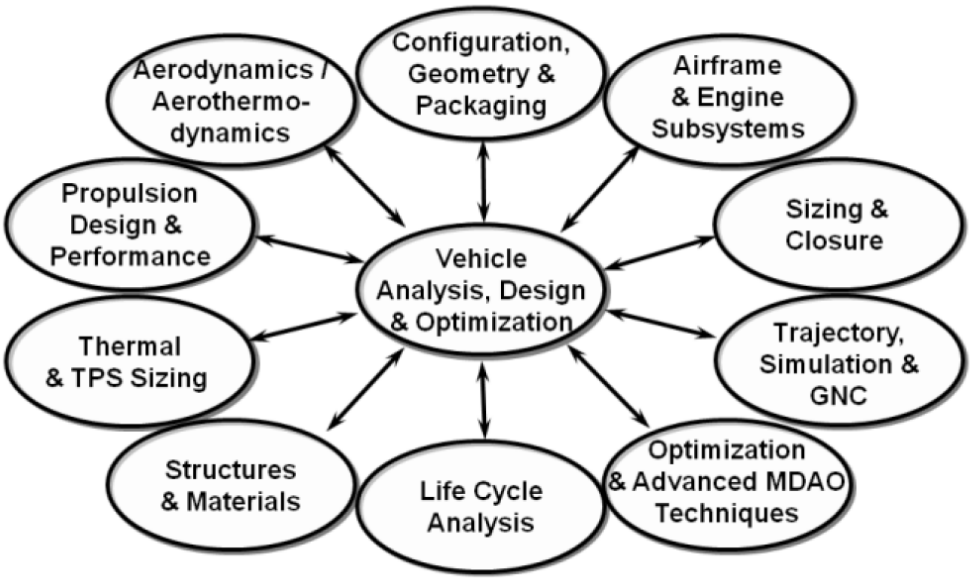


Figure 1.8: Analysis disciplines considered in Integrated Design and Engineering Analysis[7].

selecting a particular design configuration on the basis of preconceptions. The conceptual design should be based only upon very high level functional requirements, from which the vehicle configuration should be derived by means of an objective trade-off study. The trade-off process is however always influenced by the expertise of the institution or the expert performing it and often can fail to consider all the possible solutions. Evidence for this is given by the consistent amount of divergent ideas regarding what will be the best launch system design (see section 1.2.2).

In order to handle this problem an approach could consist of using multi-disciplinary design tools from the very beginning of the program. An example of this approach is given by the software Integrated Design and Engineering Analysis (IDEA) developed by the NASA for hypersonic air-breathing vehicles [7]. In IDEA the vehicle system is described by means of a modular and parametric model, over which analyses of many different disciplines can be performed (see Fig. 1.8). For each discipline different level of fidelity are available, typically the level of fidelity is increased moving forward with the design phases. All the analyses are coupled at the low fidelity level and Multi-disciplinary Design Analysis and Optimization (MDAO) can also be performed.

In [7] IDEA is being developed for a TSTO employing a TBCC engine on the first stage and a

rocket on the second stage. The propulsion analysis is performed with radical different approaches depending upon the engine configuration.

In this research a process similar to IDEA but dedicated to the propulsion subsystem is initiated, since the same considerations reported above for the vehicle system apply.

In contrast with other industrial fields, such as aviation propulsion for instance, hypersonic propulsion is a very innovative field with no well established standards. In aviation propulsion the conceptual design is indeed quite limited, since the fundamental engine design is well established. The type of engine is selected depending on the application, for example bypass turbojet is used for transonic airliner, turboprop and piston engine for slower and smaller aircraft. Moreover the industries have accumulated knowledge over time, so they do not really need to design an engine from scratch, but they rather proceed by refinements of previous engines.

In hypersonics a flying prototype does not exist yet, therefore the debate on which will be the best engine in these conditions is far from over (see section 1.2.2). Moreover, the possible engine configurations proposed for this application are diverse, and almost every time a new program is attempted a new concept is proposed (see section 1.3). The first phases of the development are therefore critical for the final success of the program. It is indeed paramount to detect any weakness in the design concept from the beginning, in order to avoid any of them jeopardizing the program at a later stage when it is too late to change.

The approach attempted in this research is aimed to provide a parsimonious modelling of the propulsion subsystem, flexible enough to investigate any possible CCP configuration. The model is called parsimonious because the complexity is kept low in order to provide a model with high level of abstraction, thus very well suited for conceptual design, when the details of the engine are not known. During the conceptual design of a propulsion system for instance, the details of the geometry and operating conditions are unlikely to be known, thus a high fidelity approach such as a CFD model is impractical not only due to its computational cost, but also due to the lack of inputs. The characterization of the system in terms for instance of the level of stage pressure ratios

or fuel-air equivalence yields more insight to the designer than if the system has to be described more fundamentally, for instance in terms of the fluid viscosity or the local turbulence intensity at some point within the engine. In a trade off analysis, with the goal of choosing between several engine conceptual design options, the accuracy of the computation is of secondary importance in comparison to the ability to perform many analyses in many different operative conditions. In this context a parsimonious model will have to produce accurate results, but even more importantly it will have to correctly predict the advantages of a design choice against another, or the differences between diverse operative conditions.

The low computational cost of the parsimonious model enables its use in design optimization loops. Beside the classical parametric optimization, we propose a configurational optimization intended to find the optimal engine design without knowing the engine conceptual design a-priori. This is a step forward with respect to the MDAO implemented in IDEA and it is something that, to my knowledge, has never been applied to engine models before (see section 5.1). Such an optimizer has the potential to perform trade-off analyses minimizing the input required by the designer and therefore minimizing the possibilities that the design could be affected by his/her preconceptions. Moreover the optimization process, not bound to any known configuration, can potentially deliver totally new engine concepts, aiding the creativity of the designer.

The configurational optimization approach is implemented at engine subsystem level due to the fact that the engine can be easily represented as a network of components, that can then be rearranged stochastically by the optimizer generating new configurations. This approach is in principle transferable also at vehicle level, even if it is more complicated. Once the methodology will be fully proved at engine subsystem, we envision to transfer it to the optimization of the whole vehicle, with the aim to let the computer answer the question: "Which is the best launch vehicle?" (see section 1.2.1).

1.5 Publications

Some of the work presented in this thesis has already been published in conference papers.

The part of chapter 2, regarding the parsimonious model implementation, has been presented at the 19th AIAA International Space Planes and Hypersonic Systems and Technologies Conference in Atlanta (USA), and published in [40]. The model has been also used and briefly described in another paper presented at the same conference [41]. Its improvements have been also published in [42] at the 20th edition of the same conference in Glasgow.

The CFD model and its validation described in chapter 3 have been presented briefly also in [40].

Part of the validations and verifications provided in chapter 4 have been presented in [40] and in [42]. The final version of the parsimonious model and its validation/verification as presented in this thesis (see chapter 2 and 4) is included in an article written for the AIAA Journal of Spacecraft and Rockets. The article went through the first review loop and the feedbacks of the reviewers have been implemented and added to both this thesis and the new version of the article.

Also the configurational optimizer and its results, described in chapter 5, have been presented for the first time in [42].

Chapter 2

Propulsion system parsimonious model

2.1 Introduction

In this research, a propulsion system model, dubbed HYbrid PRopulsion Optimizer (HyPro), which can be used in the design of future space transportation systems (i.e. SSTO or TSTO vehicles) has been developed, in order to aid the first phases of the design. In prospective the propulsion model is designed to be included in a more comprehensive vehicle system model in order to carry out vehicle system analyses and also Multi-disciplinary Design Optimization (MDO). Given current thinking on the most viable forms of future propulsion systems for these types of vehicles, the model is targeted especially toward the analysis of combined cycle configurations such as TBCC or RBCC (see section 1.3). The model is designed to be flexible enough however to model almost any possible engine configuration, from a simple rocket or ramjet through to an almost arbitrary combination of propulsive components.

Many examples of simplified models for the analysis of propulsion systems have been described in the published literature – see for instance the reviews by Tran [43] and Bradford [44]. The source

code of such models is rarely available or open to public scrutiny and individual models are often so specialised that they apply only to one or at best a few propulsion concepts. This is hardly an ideal situation particularly given that, in this field, many assumptions are always necessary to produce a tractable analysis, yet results are very sensitive to the underlying assumptions that are embedded within the models.

Given the linear topology of most aerospace propulsion systems, a common approach to modelling their physics is to employ a one-dimensional fluid dynamics-based solver to analyse the variation in bulk thermodynamic properties, such as the temperature and pressure, along the entire length of the engine flow path. The calculation would typically start from the engine intake and would march sequentially downstream towards the propulsive nozzle exit. This approach usually leads to a relatively detailed description of the physical phenomena occurring within the engine, but the method is limited in terms of the effects that it can capture naturally, or at least without some form of ad-hoc intervention.

Examples within the literature of this one-dimensional approach to propulsion systems modeling include O'Brien's very detailed one-dimensional analysis of a scramjet engine [45], where finite rate chemistry is included in the combustion modeling. Birzer presents a similar analysis but contrasts the effects on engine performance of assuming the combustion to either be mixing-controlled or chemical rate-limited [46]. Another one-dimensional scramjet model is presented by Doolan [47], where particular effort is devoted to modeling accurately the mixing of the fuel and air within the engine. A slightly different approach, but again applying a one-dimensional solver to a scramjet configuration, is described by Vanyai [48]. In this analysis the flow is considered to be inviscid and in chemical equilibrium. The inviscid assumption enables a suitable engine geometry to be determined after imposing a number of constraints on the combustion process such as the maximum pressure, or indeed the requirement that the combustion process take place at constant pressure, constant temperature or constant Mach number.

A somewhat more detailed approach to characterising propulsion system performance can be

obtained by interfacing a sequence of 3D, 2D and 1D models along the engine flow path. If the complexity of the model is matched appropriately to the complexity of the physics in the various parts of the engine, this approach can be made to be very attractive in terms of its ability to balance computational efficiency against the accuracy of performance predictions. Such a multi-fidelity approach is embodied in the SRGULL code developed at NASA Langley Research Center (see [49, 50, 51, 44]), in which a 2D Eulerian solver is used to characterise the inlet and the nozzle while a 1D solver is used to model the processes taking place within the combustion chamber. Another example of this approach is given by Gupta [52], where the engine is first divided into a series of interconnected blocks. A detailed fluid-dynamics solver is embedded within each block in order to calculate the local thermodynamic properties of the system, but considerable savings in computational time are obtained by exchanging only lumped inlet and outlet parameters between the blocks. Similarly to SRGULL, a 2D CFD model is employed for the intake, while a 1D model with finite rate chemistry is used within the combustion chamber.

Despite its advantages, this type of approach is still too slow at present to be practical within the context of preliminary design or optimization. One or more CFD computations need to be executed each time the model is called, while for instance to have practical trajectory optimization a single point run should take a fraction of second. A common way to reduce the computational cost to tractable levels for these more demanding applications is to employ the so-called "jump" solver approach, where the engine is divided up into discrete, interconnected components, and the analysis jumps in discrete steps from the beginning of each engine component to the end. A description of the system physics in terms of overall conservation principles is usually employed in order to avoid analyzing the system in more than one dimension or having to march the analysis through the component in small steps. A good example of such a strategy is embodied in the Ramjet Performance Analysis Code (RJPA) code developed at the Applied Physics Laboratory at Johns Hopkins University (see [44] and [43]). Indeed, Tian [53] describes a version of this model that can be run in real time, thus under very severe demands on its computational performance.

The inherent modularity of the jump solver approach lands itself very well to modelling those systems which have a complicated, or even variable, topology, and this inherent flexibility of the approach makes it particularly well-suited to the modelling of complex engine configurations such as those associated with the combined cycle concept mentioned earlier. The RAMSCRAM code (see [54]) is very similar to RJPA, but has been used to model complex configurations such as the ducted rocket engine. A further example of this approach is the SCCREAM code developed at Georgia Institute of Technology. This software is written in C++, and uses object oriented programming to provide a high level of modularity (see [12, 9, 55]). This model seems principally to have been used to analyse RBCC-type engines, but undoubtedly could easily be applied to a broader range of propulsive configurations. However the development of SCCREAM seems now abandoned and the source code is not available for being improved.

Other commercial jump models are also available and often used by aerospace industries. Matworks Simulink is a general purpose block based modelling suite that can be used also for propulsion models, however it is mainly conceived for controls systems. Very common in the aerospace industry, especially in the European framework, is EcosimPro and its implementation for space propulsion European Space Propulsion System Simulation (ESPSS) (see [56, 57]), that allows modelling of complex thermodynamic systems such as environment conditioning, industrial processes and propulsive systems. In contrast with SCCREAM, EcosimPro solves the system simultaneously rather then iterating sequentially from one model to another. EcosimPro however is conceived to be a general purpose tool so it needs such a more general problem formulation which, in some cases, can be more expensive in terms of computational cost.

In this chapter, the software suite, called HyPro is described in detail. HyPro is similar to SCCREAM in that it is a jump solver with the associated benefits of modularity, flexibility of configuration, and rapid computational execution. SCCREAM however was designed specifically for RBCC engine configurations, while HyPro is intended to be employed for a more general purpose, so it can be thought to be in between SCCREAM and EcosimPro. Moreover HyPro is natively

designed for engine optimization as will be explained in chapter 5.

2.2 HyPro model overview

The HYbrid PRopulsion Optimizer model is an engineering-level, modular software package for the design and analysis of advanced aircraft propulsion systems – in particular, combined cycle propulsion systems. The HyPro software is written in C++, and has a structure that takes advantage of the intrinsic modularity of the object oriented programming paradigm that is embodied within this programming language (see [58]). This modularity brings an innate flexibility to the software and allows it to be configured easily to model many different kinds of propulsion systems. Moreover, thanks to the object oriented structure of the software, it is possible to implement every module in terms of a defined set of properties and parameters that can be changed easily at any time during the execution of the model. Indeed, every engine component is characterised in terms of a dedicated C++ class and each component can be connected to any other in any possible pattern. Using advanced C++ features such as inheritance and template classes, different physical models can be combined easily to generate more complex modelling without the need to develop dedicated C++ classes (see Fig. 2.3). This maximizes the code re-utilization, facilitates the code readability and maintainability and improves the flexibility of the software. Despite the presence of all these complicated syntactical features however, C++ is well known to be very computationally efficient, since all the effort of interpreting the complex code is done at compile time. For all the aforementioned reasons, HyPro is particularly well-suited to be used within optimization studies and also as a sub-model within a more comprehensive computational representation of the entire vehicle system (see [41]).

The modules are connected between each other by means of nodes, each node contains a set of physical quantities that are then actually shared by contiguous modules. A node represents the thermo-kinetic state at a certain station within the engine, thus it contains all the information required to define the thermodynamic state (for example the composition, pressure and temperature

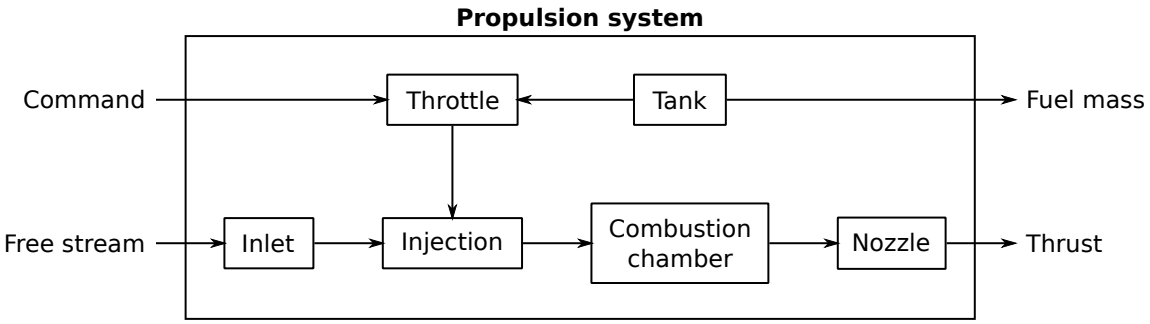


Figure 2.1: Example of the modular structure of the propulsion model for a Scramjet/Ramjet.

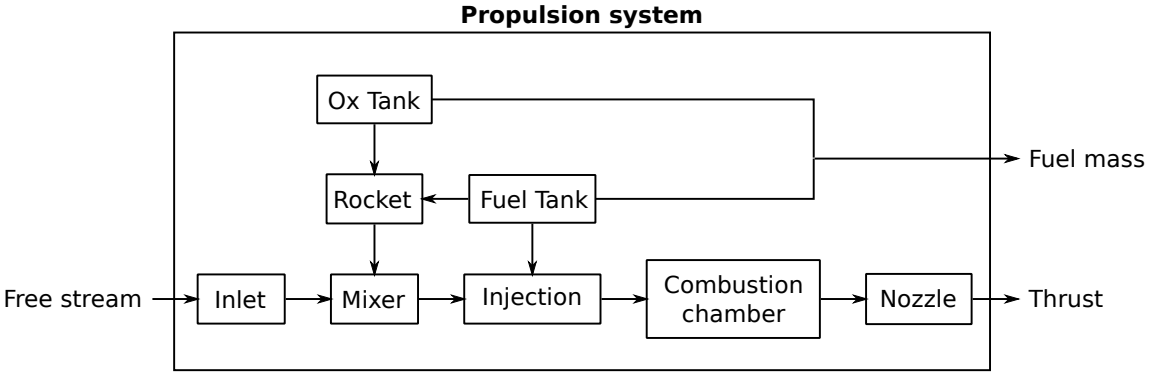


Figure 2.2: Example of the modular structure of the propulsion model for a RBCC engine.

for a mixture of ideal gases), the speed of the flow and the area of the section.

In this version of HyPro the only state law available is the ideal gas law, that can be used safely as soon as the pressure is far lower than the critical point. In the applications presented in this dissertation that is always the case, since mostly only air-breathing engines are investigated, for which the maximum operating pressure is never very far from the atmospheric pressure. For rocket propulsion however, the operating conditions can be supercritical, thus in the future different state law will need to be implemented if a detailed investigation of this engines is required.

Every module does not see directly the other modules it is connected to, but it only stores a pointer to the nodes connected to it, that in turn are connected with other modules. Each module is connected to at least two nodes, an input node hereafter called N_1 and an output node called N_2. Within the module C++ class a 'calculate' method is defined, this method will be called to evaluate the conditions at node N_2 knowing those at node N_1.

In Fig. 2.1, for instance, a schematic of the model, structured in order to represent a scramjet/ramjet engine, is depicted. The modules can easily be changed and re-arranged to represent alternative engine configurations. Combined cycle engines can be modelled by collating and interconnecting all the required modules, then switching on/off certain modules according to a predefined schedule of operation. In Fig. 2.2 an example of the model configured to represent a RBCC propulsion system is illustrated.

In Fig. 2.3 the structure of the HyPro code in terms of inheritance diagram is depicted. As can be seen a single base class called *propModule* is inherited by all the classes that model the different propulsive modules. In the base class *propModule* only basic aspects are defined, such as the presence of the input/output nodes and some other utilities common to all the modules. The actual implementation of the physics is then performed at the specific classes that inherit directly or indirectly from the *propModule*. In Fig. 2.3 beside the inheritance, it is also shown how the template classes are implemented. As an example, the *Combustion<T>* is a template class that can inherit directly from the *balanceMach* class (see section 2.3.2) or from the *Friction* class. In this way the user can decide either to add or not friction to the combustion chamber (see sections 2.3.3 and 2.9).

A class of particular importance in Fig. 2.3 is called *Collection*. This class behaves like any other propulsive model as seen from the outside, however inside it contains a list of sub-modules that together can model a complex propulsive component. The inlet modules are examples of *Collection* classes, since they inherit from the *Collection* and they contain a list of sub-modules. All the *Collection* classes contain also a list of nodes that are connected to the sub-modules allowing them to communicate between each other. The aforementioned nodes cannot be connected to modules outside the collection, therefore they are called internal nodes.

All the other classes appearing in Fig. 2.3 will be described in the following sections in terms of the physical model implemented within them.

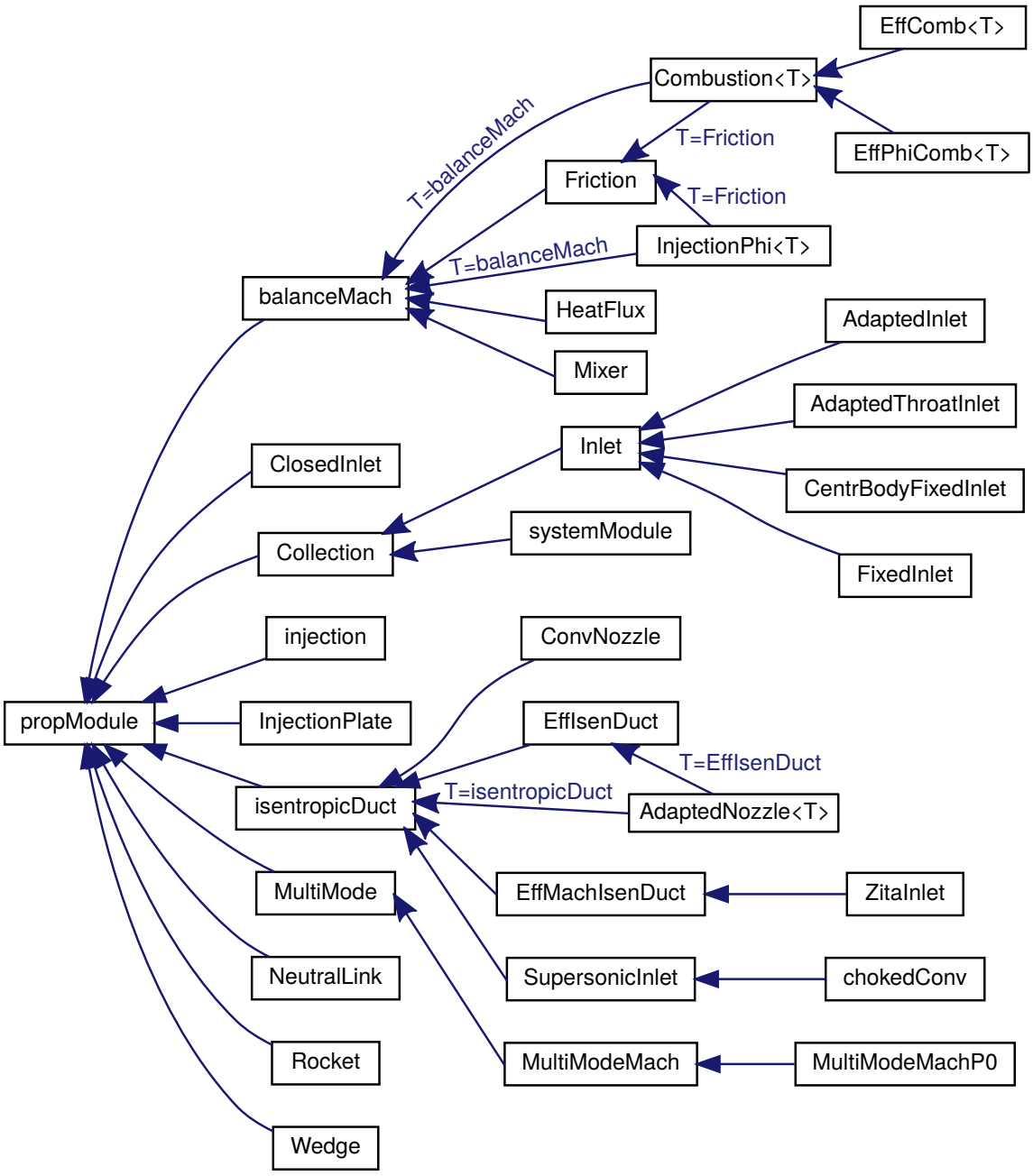


Figure 2.3: Inheritance diagram of the HyPro model classes.

2.3 Gas dynamics models

Before describing each individual propulsive module in detail, it is appropriate to summarize the general gas dynamic models that are used as the basis for the physical modelling of the flow within all of the engine components. The gas dynamic modules that have so far been implemented are the following: isentropic one-dimensional compression/expansion[59], Mach based one dimensional balance equation solver (see 2.3.2), friction model and heat flux model.

These models are embodied as independent C++ classes within the code in order to give flexibility to the software, other propulsive modules can then use them thanks to the C++ inheritance mechanism (see classes *isentropicDuct*, *balanceMach*, *Friction* and *HeatFlux* in Fig. 2.3). The first two can be thought of general fluid dynamic solvers, the former for variable area modules (i.e. inlets, nozzles etc.), the latter whenever the considered propulsive component has a constant section area (for instance a combustion chamber) and does not need the hypothesis of isentropic flow. The last two instead can be added on top of the balance equation solver to consider the presence of friction and heat fluxes through the lateral walls of the propulsive module analysed.

In all the aforementioned models the outlet conditions are calculated by means of integral equations knowing the inlet conditions, therefore the details of the flow along the module are not considered. As anticipated in section 3.1, this strategy is typical of a step code and it is very efficient in terms of computational effort.

2.3.1 Isentropic compression/expansion

This fluid dynamics solver is used whenever the module has a variable section area, as depicted in Fig. 2.4. The main hypothesis is that the flow is isentropic and it is strictly valid when the heat flux and the friction on the lateral walls are negligible.

In this version of HyPro the well known analytical solution of isentropic flow (see [59]) is not used because it is valid only in case of calorically perfect gases (i.e. those that have a constant ratio of specific heats γ). An iterative procedure, based on the bisection algorithm [60], is instead

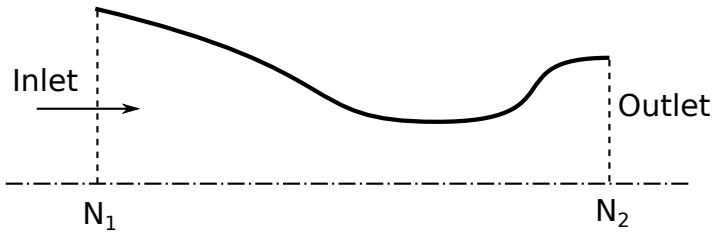


Figure 2.4: Definition of stations for a one-dimensional duct with variable cross-sectional area.

employed. The Mach number at the outlet of the duct is used as the iteration variable and all the other thermodynamic conditions at the outlet node (i.e. node N_2 in Fig. 2.4) are calculated assuming the flow to be isenthalpic and isentropic. After a tentative value for the Mach number M_2 is chosen, the static temperature at node N_2 is calculated from the energy equation

$$h_2 + \frac{M_2^2 a_2^2}{2} = h_1 + \frac{M_1^2 a_1^2}{2}. \tag{2.1}$$

In this equation, only h_2 and a_2 are initially unknown, but both are functions of the temperature T_2, which can then be determined through an inner iteration loop. The static pressure at node N_2 is then calculated from the entropy equation [61]

$$S^o(T_2) - Rln\frac{p_2}{p^o} = S^o(T_1) - Rln\frac{p_1}{p^o}. \tag{2.2}$$

The pressure p_2 is the only unknown in this equation once the inlet conditions are specified. It should be noted that Eq. 2.2 is only valid for gases with constant composition, and this requires the assumption that the chemical composition of the working fluid is frozen in all the modules that use this analysis (i.e. the 'inlet', 'nozzle', 'diffuser', etc., as presently implemented in HyPro). This assumption is acceptable almost always in 'inlet' and 'nozzle' modules, with the only exception being nozzle's convergent, where in some cases the reactions started in combustion chamber can continue up to the nozzle throat. In this version of HyPro however, this phenomena is not taken into account since the combustion is assumed to occur completely in the combustion chamber (see

section 2.9), so the error linked with frozen composition assumption in the nozzle is somewhat balanced by the aforementioned assumption in the combustion chamber.

Once all the conditions at node N_2 are calculated, the mass flow rate is compared against that of the inlet (node N_1), thus the iteration is stopped if the difference is smaller than a predefined tolerance.

In order to make possible generalization in case of realistic non isentropic and non isenthalpic conditions, the possibility to add a total pressure drop $\frac{p_{02}}{p_{01}}$ and a total temperature drop $\frac{T_{02}}{T_{01}}$ is also included. In this case a new intermediate node is generated modifying the stagnation conditions of node N_1 to account for these efficiencies, as a consequence however the mass flow rate of this intermediate node is different than that of node N_1. The aforementioned iterative procedure is therefore initiated from the new modified conditions in order to match the mass flow rate of the node $N1$.

2.3.2 Balance equation solver

In contrast with the solver described in section 2.3.1, the balance equation solver is used whenever the module cross section area is constant. In this case the pressure acting on the lateral walls has always a null component along the module axis, therefore the momentum balance equation can be easily written without knowing the pressure evolution across the module. The hypothesis of isentropic flow is thus not necessary to close the problem. The following equation can indeed be written:

$$\begin{cases} \rho_1 U_1 - \Delta G = \rho_2 U_2 \\ p_1 + \rho_1 U_1^2 - \Delta I = p_2 + \rho_2 U_2^2 \\ \frac{G_1}{G_2}\left(h_1 + \frac{U_1^2}{2} - \Delta H\right) = h_2 + \frac{U_2^2}{2} \end{cases} \tag{2.3}$$

they are, respectively, the balance of mass, momentum and energy along a duct that has constant section area. In these equations, ΔG, ΔI and ΔH represent respectively the flux of mass, momentum and energy across the walls of the duct. The solver attempts to determine the conditions at

node N_2 (i.e. at the end of the duct) given the conditions at node N_1 (i.e. at the beginning of the duct) and these lateral fluxes. In order to simplify the solution process, the system of equations is rewritten in the following form:

$$\begin{cases} \rho_1 U_1 - \Delta G = \rho_2 U_2 \\ p_1 + \rho_1 U_1^2 - \Delta I = \left(1 + \gamma M_2^2\right) p_2 \\ \dfrac{G_1}{G_2}\left(h_1 + \dfrac{U_1^2}{2} - \Delta H\right) = h_2 + \dfrac{\gamma M_2^2 R T_2}{2} \end{cases} \tag{2.4}$$

It is important to note that the only assumption needed to support the validity of Eq. 2.4 is that the gas be ideal in its behaviour; the ratio of specific heats γ can thus be modelled as a function of temperature. The solution process starts by using a tentative value for M_2 and then proceeds through the solution of each element of Eq. 2.4 in turn. First the energy equation is solved in terms of the static temperature T_2 by means of a root search algorithm, then the momentum equation is used to determine the static pressure p_2 and finally the the mass balance equation is used to check whether the tentative value assumed for M_2 is correct or not. The whole procedure is then iterated by means of a root search algorithm until the desired level of convergence is reached.

This solver is more general than the isentropic one (see section 2.3.1), but it cannot be used when the geometry of the module is different than a cylinder. This solver can be employed also when the chemical composition changes across the module simply assigning the composition at node N_2 before the solution loop starts. Once the composition is changed the gas at node N_2 behaves as an ideal gas, but with different constants, and so the Eq. 2.4 are still valid. Any complex phenomena happening at the lateral wall of the module can be also modelled simply adding the right flux related to it, the only limitations is that the flux has to be known a priori, so it has to be independent from the conditions at node N_2 (see for instance section 2.3.3 and section 2.3.4).

2.3.3 Friction model

The first model that can be added on top of the balance solver (see section 2.3.2) is the friction model. It is used to consider the presence of fluid friction on the module lateral walls.

The momentum flux due to friction is calculated after assuming a constant friction coefficient C_f along the length of the module.

$$\begin{aligned} \Delta I &= \tau \frac{A_W}{A_1} \\ \text{where} \quad \tau &= \frac{\rho_1 U_1^2}{2} C_f \end{aligned} \tag{2.5}$$

Since the variation of the dynamic pressure along the duct is not constant, Eq. 2.5 is only approximate. It does however yield a good compromise between accuracy and computational effort (see section 4.2.1), moreover its simplicity allows the possibility of coupling with other models that uses the balance equation solver, such as the heat fluxes, the combustion, the injection and so on.

2.3.4 Heat flux model

Similarly with the friction model, the heat flux model is used whenever the presence of heat flux across the module lateral walls need to be taken into account. The heat exchange with the walls is calculated using a similar procedure as per the friction, where St is the Stanton number, defined as the heat flux normalized by a reference heat flux proportional to the difference between adiabatic enthalpy and wall enthalpy [62].

$$\begin{aligned} \Delta H &= \frac{\dot{q}}{\dot{m}_1} \frac{A_W}{A_1} \\ \text{where} \quad \dot{q} &= \rho_1 U_1 \left(h_1 + \frac{U_1^2}{2} - h_W \right) St \end{aligned} \tag{2.6}$$

Also in this case the the dimensional heat flux is calculated using only the conditions at the inlet, thus neglecting the changes across the duct.

2.4 Intake

In this section the intake models available in HyPro are described. For hypersonic propulsion the intake behaviour is complex and its performances in general drive the performances of the whole engine. For this reason many different intake designs have been proposed, often characterized by movable parts and variable geometry. In this version of HyPro two main different intake model have been implemented: a convergent divergent intake with variable geometry and a fixed geometry intake with central body. Other possibilities have not been implemented so far, because they were out of the scope of this research. In principle however there are no difficulties in adding further models. Moreover, the starting of the intake has not been investigated in this research, therefore the intake is considered always started and not affected by any unsteady phenomenon (see for instance [63]). However, non adaptation problems are instead taken into account for both fixed geometry and adaptable variable geometry intakes.

Intakes are modelled as a collection of modules (see Fig. 2.3), due to their complexity and the need to consider also intermediate nodes between the free stream conditions and the end of the inlet. Therefore beside the inlet and outlet node, the intake module has also few internal nodes not connected with any other engine component. This strategy is used to differentiate between the free stream and the actual inlet of the intake, which can experience different fluid dynamic conditions, or to have access to the inlet throat properties, fundamental for the global behaviour of the intake.

2.4.1 Convergent divergent intake

The convergent divergent intake is an ideal intake that can convert a supersonic free stream into a subsonic flow isentropically. Although very attractive in theory, isentropic (or at least shock free) compression of supersonic flow is very hard to have in practice. At the designed flight Mach number, a normal shock would occur in front of the intake, only after a sensible over speed the shock will be ingested and positioned in the divergent with a much lower pressure drop. At this point the flight speed can be reduced again to the design value in which the shock approaches

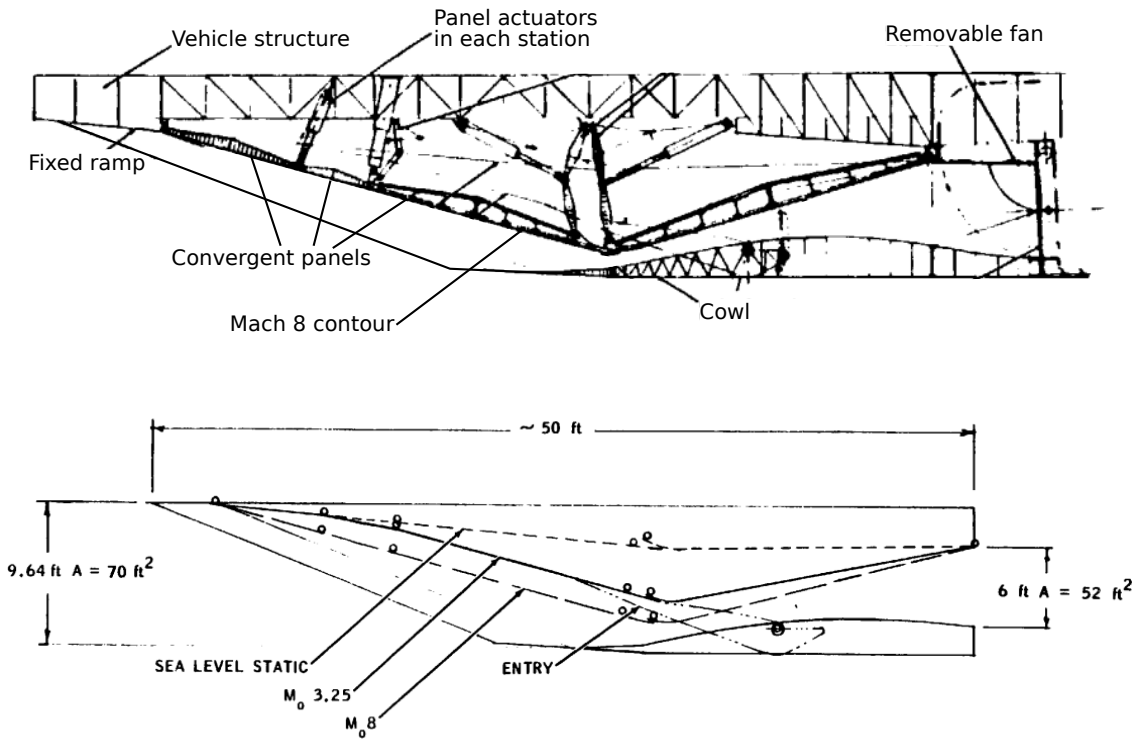


Figure 2.5: Example of variable geometry convergent divergent intake: Marquardt SERJ proposed design (see [8]).

the throat where ideally, it disappears. This process is called inlet starting (see [64]) and is often not practical. For this reason such an intake configuration is actually used only if equipped with variable geometry devices, that can change the intake throat section area to start the inlet. An example of such a solution is given by the design proposed by the Marquardt for RBCC engine dubbed SERJ. The intake of Fig. 2.5 is a 2D convergent divergent intake equipped with a set of actuators that can change both the inlet section area and the throat area. It can be seen that the mechanism is very complex, but beside the change of the aforementioned areas, it also allows the fine tuning of the convergent profile. Despite the variable geometry, the intake is far from being isentropic or even simply shock free, several oblique shocks will take place in the convergent section in correspondence of the junctions of the movable panels. The variable geometry however, gives the possibility to adapt the slopes of each panel in a way to minimize the global pressure drop across these shocks, so it is the best approximation of an ideal isentropic intake.

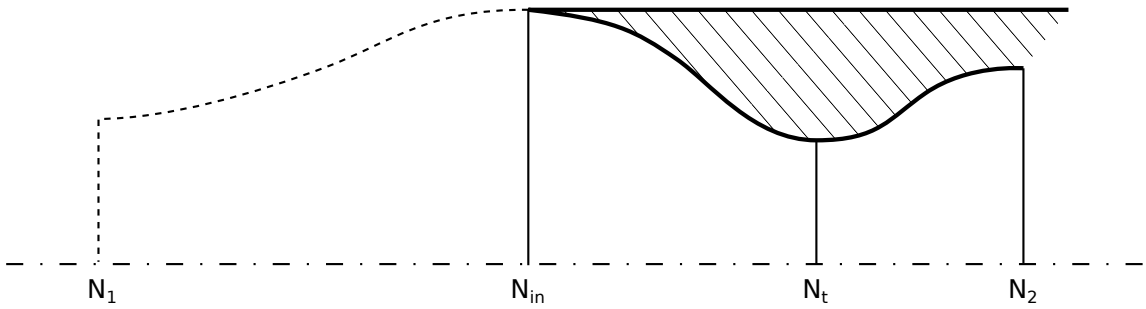


Figure 2.6: Definition of stations for the convergent divergent intake model.

In HyPro this intake model considers the four nodes depicted in Fig. 2.6. Despite the full variable geometry assumption of this intake model, the possibility of having non adapted intake is also accounted for. In the real world the variable geometry capabilities are limited within a predefined range. In HyPro this is specified in terms of section areas, due to geometrical and mechanical constraints. Therefore when the operating conditions would require the overtaking of these limits, a non-adapted behaviour of the intake is considered. The conditions at node N_1 are assumed to be different from those at node N_{in}, since an external flow path will be present to adapt the free stream conditions to the engine operating conditions. The external flow path is assumed to be isentropic if the external flow is subsonic, whereas if the external flow is supersonic then a normal shock is assumed to be present in order to slow down the flow to subsonic speed and allow the adaptation with a consequent pressure drop. Whenever the intake is not fully adapted, an intake drag is calculated by means of Eq. 2.21, as explained in section 2.11.

The remainder of the flow within the intake is calculated by considering a total pressure drop across the convergent part of the duct and assuming the flow in the divergent to be isentropic (see 2.3.1). The pressure drop is given by MIL Spec E-5007D[12] semi-empirical relation (see Fig. 2.7) or it can be prescribed by means of a user defined lookup table.

2.4.2 Central body intake

A very common intake design for supersonic engines consists of a central body, often conical, followed by a simple divergent section (see Fig. 2.8). In this case the free stream flow is decelerated

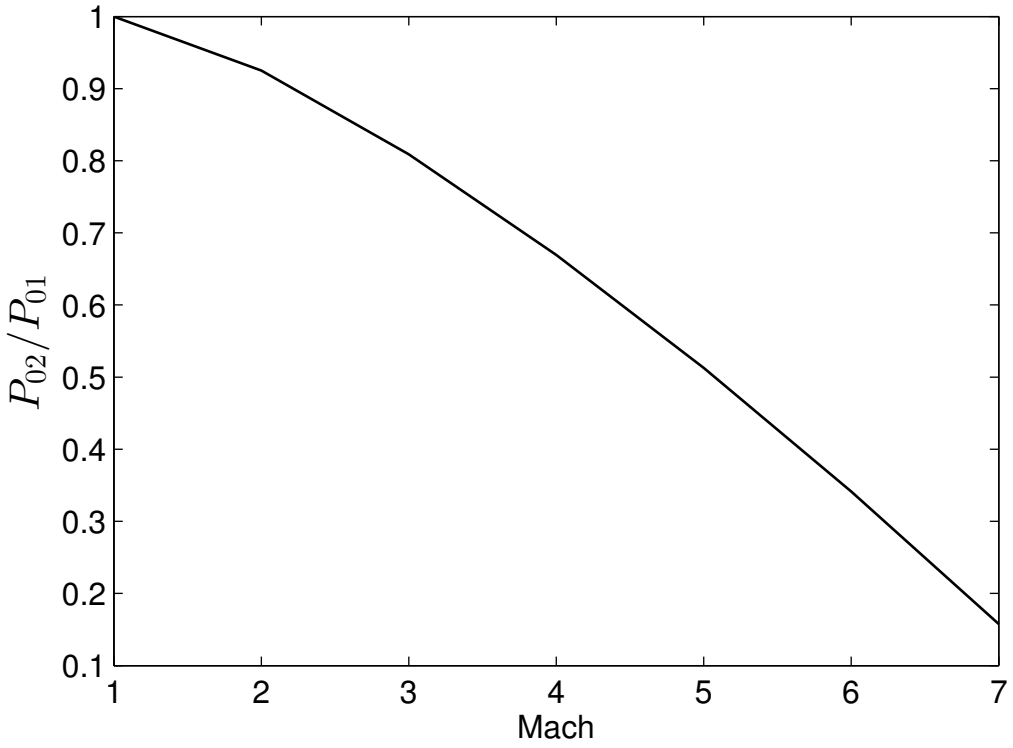


Figure 2.7: Intake total pressure drop as a function of Mach number.[9]

to subsonic speed by a set of shocks, rather than a convergent divergent duct with sonic throat. This solution is less efficient than the convergent divergent one, but it works usually better with fixed geometry. It does not require a starting procedure, since the flow is slowed to subsonic speed always by means of a final normal shock wave, which is weaker than the normal detached shock occurring in front of a convergent divergent intake.

In the simplest case two shocks will occur. An oblique shock attached to the central body leading edge followed by a normal shock, whose position changes depending on the operating conditions of the engine (see Fig. 2.9). Other more complex, but more efficient, designs might have multiple oblique shocks on the central body, in order to increase the efficiency of the compression. The intake is said to operate in critical conditions if the normal shock is placed at the inlet of the divergent section (see Fig. 2.9a), the intake operates in supercritical conditions if the normal shock moves further downstream into the divergent section (see Fig. 2.9b) and in subcritical conditions when the shock is detached ahead of the divergent section (see Fig. 2.9c). In supercritical conditions the mass flow rate ingested by the intake is constant no matter where the terminal shock wave is positioned in the divergent section, since it is only determined by the supersonic flow on the central body that cannot be influenced by what happen in the downstream divergent section. The more the shock moves downstream the more the supersonic flow expands in the divergent section leading to stronger shock and consequently to lower total pressure at the intake outlet (see vertical lines in Fig. 2.10). In subcritical conditions, the normal shock detaches from the divergent section producing a subsonic bubble just before the inlet. In the subsonic regime the information can travel upstream so the flow can be spilled out of the inlet as much as required by the engine. On the other hand the total pressure stays almost constant (see horizontal lines in Fig. 2.10). Increasing the flight Mach number increase both the critical mass flow and critical stagnation pressure as depicted in Fig. 2.10. At a certain flight condition the operations of the intake are determined by the performance, in terms of mass flow rate and total pressure at the intake outlet, required by the rest of the engine. The engine usually has a choked throat at a certain point in the flow path

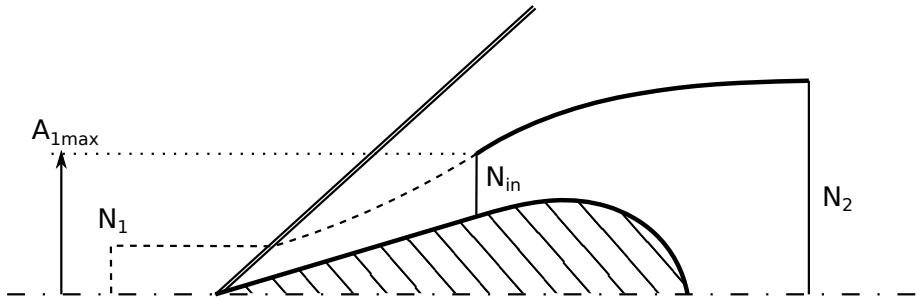
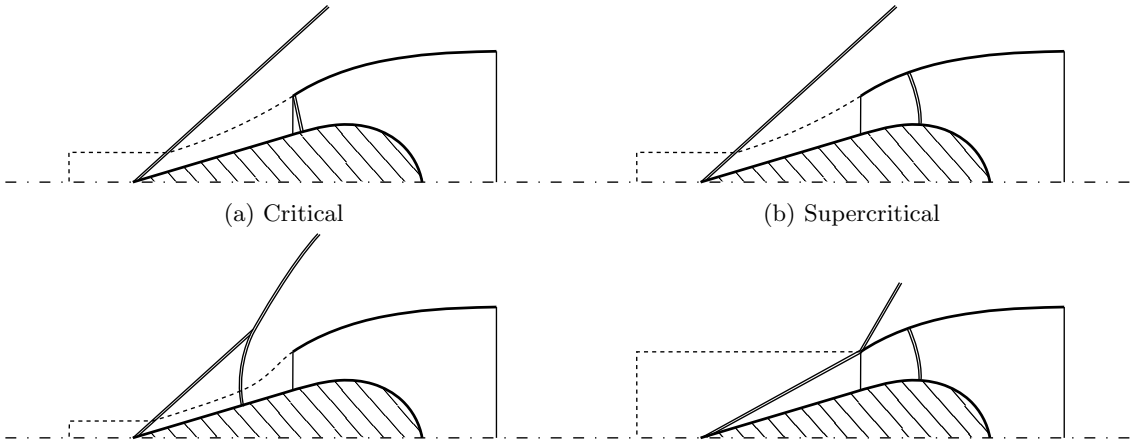


Figure 2.8: Definition of stations for the central body intake model.



(a) Critical (b) Supercritical

(c) Subcritical (d) Adapted

Figure 2.9: Possible operative conditions for central body intakes.

(typically at the nozzle throat), therefore the bigger the mass flow the bigger the total pressure will need to be. In Fig. 2.10 the operating conditions are identified by the intersections between the aforementioned engine requirement curve and the intake performance at a certain flight Mach number. As can be seen, in acceleration usually the operations pass from subcritical to supercritical. However this is not always the case since the performance requested by the engine depends also on other parameters such as, for example, the fuel injected.

In supersonic conditions the capture area of a central body intake cannot be higher than the maximum area defined in Fig. 2.8, because the information cannot travel upstream in a supersonic flow. The intake is said to be adapted if the oblique shock wave impinges exactly on the tip of the cowl, as depicted in Fig. 2.9d, and as a consequence the capture area reaches its maximum value. For lower Mach numbers the oblique shock slope will be bigger and the capture area will be lower

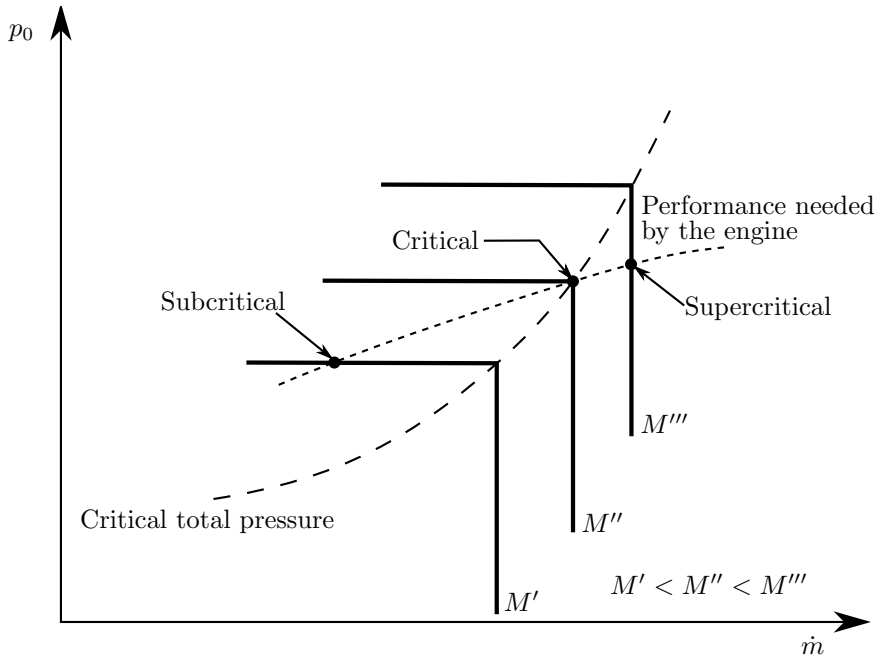


Figure 2.10: Example of central body intake behaviour. Total pressure at the intake outlet as function of ingested mass flow rate plotted at 3 different flight Mach. An example of typical performances needed by the engine downstream of the intake is also plotted along with consequent operating points (see [10]).

than the maximum, see for instance Fig. 2.9b, while for higher Mach number the shock will be closer to the central body and it will reflect inside the divergent, this however will not change the capture area that will therefore stay at its maximum value.

In HyPro the flow over the centre body can be calculated knowing the coefficient of critical flow (ζ) defined as follows:

$$\zeta = \frac{A_1}{A_{1Max}} \tag{2.7}$$

where A_{1Max} is the maximum possible capture area (see Fig. 2.8). The conditions at node N_{in} are then calculated assuming isentropic flow between N_1 and N_{in}, with the possibility for the user to add a pressure drop to take into account the non isentropic effects such as the shock itself. Both the coefficient of critical flow and the pressure drop can be defined by the user as a function of M_1 by means of a lookup table.

In the case that ζ is not known, the centre body flow can be calculated using the oblique shock theory. In this case the conditions at node N_{in} are first calculated knowing the centre cone/wedge semi-angle and then the area A_1 is calculated enforcing the mass balance between nodes N_1 and N_{in}.

In subcritical conditions the capture area A_1 is lowered with respect to the one previously calculated, then the conditions in N_{in} are recalculated applying first a normal shock and then an isentropic compression. In supercritical conditions, a pressure drop in the divergent section is added to account for the floating normal shock. In both cases the additional degree of freedom (A_1 in subcritical and the pressure drop in supercritical) will be determined by the conditions downstream of the intake, hence it is used as control parameter during the system iteration loop, as explained in section 2.11.

As for the previous case of the convergent divergent intake, the additional drag has to be calculated using Eq. 2.21, that in the case of central body intake needs to be changed as follows to account for the presence of the central body.

$$D = \dot{m}_{in} \left(U_{in} - U_1\right) + \left(p_{in} - p_1\right) A_{in} + \left(p_{CB} - p_1\right) \left(A_{1max} - A_{in}\right) \tag{2.8}$$

where the p_{CB} is the average pressure on the central body. Applying a momentum balance at the external flow tube (see Fig. 2.9d), it can be demonstrated that in adapted conditions the intake additional drag is null, as for the convergent divergent intake. Since the lateral walls of the flow tube in Fig. 2.9d are parallel to the engine axis, the momentum there does not participate to the momentum balance along the thrust axis, therefore the momentum flow at the free stream (inlet of the flow tube) is equal to the sum of the momentum flow at node N_{in} and the force acting on the central body and thus the drag as calculated by Eq. 2.8 is zero. Moreover with the same logic it is possible to conclude that the drag is also null for flight Mach numbers higher than the one in correspondence of adapted conditions. In all other cases the intake's additional drag is

calculated employing some simplifying assumptions to calculate the pressure on the central body. In supercritical conditions it is assumed to be equal to the pressure at the inlet p_{in}. This assumption is strictly true only in the case of 2D central body. In subcritical conditions it is assumed to be equal to the value before the normal shock, an assumption that does not take into account the presence of the subsonic bubble at the top of the central body.

2.5 Fan and Compressors

A simple model for fans and compressors is also provided. The model takes as input the total pressure ratio R_p and considers the compression to be isentropic. The stagnation conditions at the end of the compressor are thus

$$p_{02} = R_p p_{01}, \quad \rho_{02} = R_p^{\frac{1}{\gamma}} \rho_{01}, \quad T_{02} = \frac{1}{R} \frac{p_{02}}{\rho_{02}} \tag{2.9}$$

The static conditions and the Mach number at the end of the compressor are then calculated by imposing a mass balance across the compressor. This balance equation is solved iteratively using the Mach number as the tentative variable.

In the present version of HyPro this model is very basic, no link between compressor and turbine is provided to make sure that the energy consumed by the compressor is produced in turbine. The model is actually only used in the case of small fans, whose energy input/output can be neglected in the frame of the engine thermodynamic loop.

2.6 Mixer

The mixer module allows the mixing of two flows to be modelled. It is therefore defined as a module with an additional internal node N_3 which can be joined to any other type of module within the HyPro toolbox. This module is typically used when the momentum contribution of the flow out

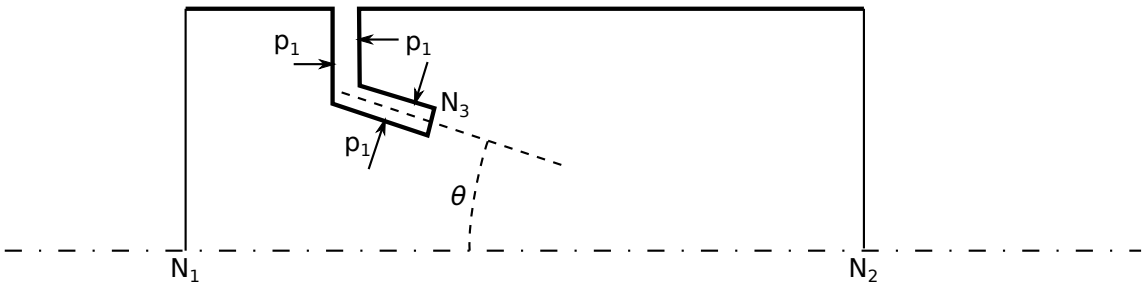


Figure 2.11: Definition of stations for the mixer module.

of the module attached to the internal node cannot be neglected. A typical example of where the mixer module would be used is in a model of an ejector system, where a rocket plume mixes with the airflow coming from the inlet (see for instance 4.4 and 4.5).

The mixer module uses the balance equation solver described in Section 2.3.2 to obtain the flow properties at node N_2 given the flow properties at nodes N_1 and N_3. To satisfy the assumption of cylindrical lateral walls embodied within the balance equation solver, the area of the duct at node N_1 must of course be made equal to the area of the duct at node N_2. The fluxes of mass, momentum and energy into the duct are calculated as follows:

$$\begin{cases} \Delta G = -\dfrac{A_3}{A_1}\rho_3 U_3 \\ \Delta I = \eta \left[\Delta G U_3 - \dfrac{A_3}{A_1}\left(p_3 - p_1\right) \right] \cos\theta \\ \Delta H = \Delta G \left(\dfrac{U_3^2}{2} + h_3 \right) \end{cases} \tag{2.10}$$

where η is the mixer efficiency and θ is the angle at which the flow is injected into the mixer at its internal node N_3 (see Fig. 2.11). Eq. 2.10 is derived under the hypothesis that the pressure surrounding the injected flow, and any pipe injecting it, is equal to the pressure at node N_1 (see Fig. 2.11); this can be assured if the mixer module is defined so that the inlet to the module (node N_1) is close to the point of injection.

The aforementioned assumption is weak whenever the cross-sectional area over which the fluid is injected (i.e. the area at node N_3) is large compared to the total area of the mixer duct (i.e.

the area of the duct at node N_2). Where necessary, this problem can be overcome by modifying slightly the formulation presented above. If it is assumed that the two mixing flows start by being parallel (i.e. $\theta = 0$), then nodes N_1 and N_3 (see Fig. 2.12) can be made coincident and only the flow downstream of the injection point needs to be modelled. This assumption of course requires the slightly more restrictive constraint that the area of the duct at N_2 be equal to to the sum of the duct areas at N_1 and N_3 so that the assumption of cylindrical lateral surfaces remains valid. In the resultant 'parallel flow mixer' module, the expressions for the mass, momentum and energy fluxes into the duct can thus be written as

$$\begin{aligned} \Delta G &= -\frac{A_3}{A_2}\rho_3 U_3 \\ \Delta I &= \eta \left[\Delta G U_3 - \frac{A_3}{A_2} p_3 \right] \\ \Delta H &= \Delta G \left(\frac{U_3^2}{2} + h_3 \right). \end{aligned} \tag{2.11}$$

In Eq. 2.11 it can be noted that the back pressure acting on any component attached to node N_3 (e.g. the pressure around the connector pipe shown in Fig. 2.12) does not appear in the momentum flux ΔI, since such a component is external to the control volume used by the module. Its influence on the engine flow path however remains, only it needs to be taken into account in the module upstream node N_1. In most cases it will be a duct with variable cross section area (see section 2.3.1) and an efficiency sufficient to model the presence of the obstacle (e.g. the pipe in Fig. 2.12).

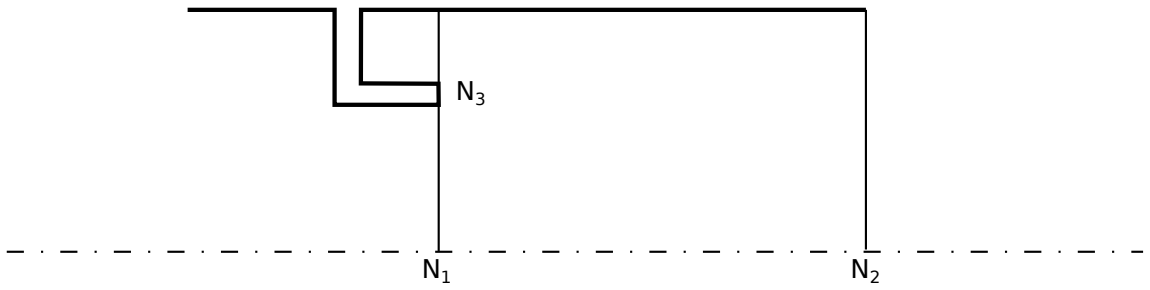


Figure 2.12: Definition of stations for the parallel flow mixer module.

The two mixer modules described above are both available in HyPro, and the user can easily

select the one most convenient depending on the application, thanks to the object oriented structure of the code.

2.7 Injection

The injection module takes care of the mixing between the injected fuel and the main air stream and does not consider the possibility of any reaction at the point of injection. In HyPro the injection modelling is separated from the combustion chamber in order to improve the flexibility of the code. The mixture of fuel and oxidant, depending on the engine design, can experience several transformations between the fuel injection point and the combustion.

In the first model of injection, the addition of the fuel mass to the flow is accounted for simply by increasing the pressure at the exit to the module over that at the entrance according to

$$p_2 = \frac{p_1}{1 - X_{fuel}} \tag{2.12}$$

where X_{fuel} is the molar fraction of the fuel. In this case the temperature and velocity are considered constant along the length of the module. Although this assumption is not physically valid it was useful during validation of the model as will be explained in section 4.3.

The second model is more accurate and makes use of the solver described in section 2.3.2. The module takes as input the equivalence ratio φ together with the fuel temperature and composition and calculates the fluxes needed in Eq. 2.3 according to

$$\begin{cases} \Delta G = -(\varphi \Phi) \dfrac{W_f}{W_1} \left(X_{ox}\right)_1 \rho_1 U_1 \\ \Delta I = 0 \\ \Delta H = \Delta G h_f \end{cases} \tag{2.13}$$

where the subscript f indicates the conditions of the fuel being injected, X_{ox} is the molar frac-

tion of the oxidant, W the molecular weight and Φ the molar ratio between fuel and oxidant in stoichiometric conditions. The principal assumption made in the derivation of Eq. 2.13 is that the momentum of the injected fuel is negligible compared to that of the main flow. Since the model in HyPro is one-dimensional the other intrinsic assumption made is the fuel being evenly mixed with the oxidant at the end of the injection module (i.e. node N_2), therefore any effect of poor mixing will need to be modelled in the combustion chamber module by means of a reduced combustion efficiency (see section 2.9).

Any loss of momentum due to other phenomena linked with injection, such as droplet breakup and mixing, can be easily included adding a dedicated momentum loss model similar to the friction model described in section 2.3.3.

2.7.1 Injection plate

The injection plate module is dedicated to the modelling of injection for liquid propellant rockets. In this case the module does not need the incoming flow with which the fuel is mixed, but it provides both the fuel and the oxidant. Typically a rocket engine is created by combining a combustion chamber module and a nozzle module with the injection plate module as shown in Fig. 2.13 (where injection takes place between nodes N_1 and N_2). The module requires the composition of the working fluid, the total mass flow rate, the Mach number, and the static temperature at node N_2 to be specified externally.

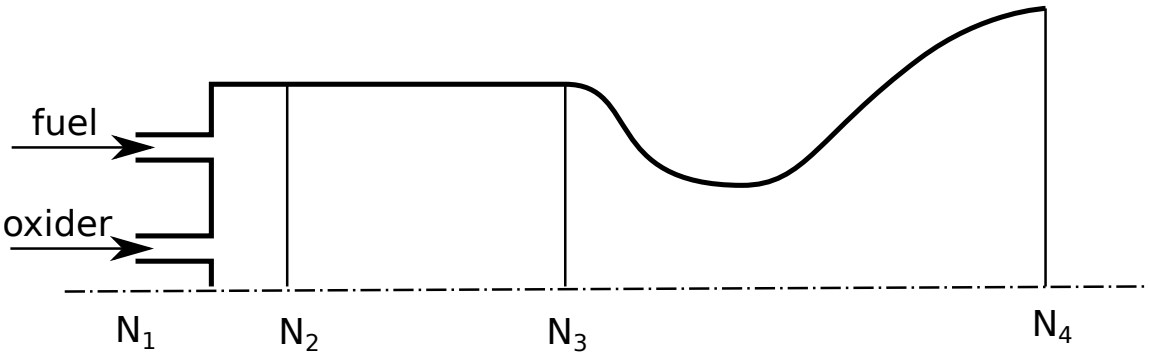


Figure 2.13: Definition of stations for the rocket injection module (as implemented in a typical rocket engine model).

The pressure at node N_2 is then calculated with the following:

$$\begin{aligned} p_2 &= \rho_2 R T_2 \\ \text{where} \quad \rho_2 &= \frac{\dot{m}}{M_2 a_2 A_2} \end{aligned} \tag{2.14}$$

Note that the Mach number at node N_2 is set as an input because, in a rocket it depends on the area ratio between the combustion chamber and the nozzle throat, which usually is constant. In the case that the rocket needs to throttle down due to choking downstream, the mass flow rate can be reduced.

2.8 Rocket

A simple rocket model is provided in case there is no need to describe the rocket engine behaviour in detail. For instance when the rocket is a subsystem of a more complex engine. The rocket engine is modelled by means of its global characteristics (i.e. total propellant mass flow rate $\dot{m}$, nozzle exit mach number M_2 and the specific impulse in void Isp) that therefore have to be provided by the user. Moreover the user needs also to define the chemical composition of the nozzle exhaust, while the composition of the propellants is not needed since the combustion is not modelled.

In the rocket module the input node N_1 is not used, thus it does not need to be defined. The output node (i.e. N_2) of the module is placed at the nozzle exit and its flow characteristics are calculated as follows. First the exit chamber temperature is calculated starting from the specific impulse, then the nozzle exit static temperature is calculated assuming that the expansion to Mach M_2 is isentropic. Knowing the temperature and the input Mach number M_2 is then possible to calculate the flow speed at nozzle exit. Eventually the missing thermodynamic variable is calculated imposing that the mass flow rate at the nozzle exit is equal to the total propellant mass flow rate provided by the user.

2.9 Combustion Chamber

The combustion chamber is probably the most complex part of the engine system to model. In the scramjet case for instance, the interaction between several distinct physical phenomena all need to be accounted for simultaneously, rendering the modularization process particularly difficult. In the case of supersonic combustion, heat exchange and friction are important processes that need to be taken into account. Thanks to the Object Oriented Programming (OOP) structure of HyPro however, this task becomes simple. The solver described in section 2.3.2 is employed once the composition of the gas at the end of the combustion chamber has been calculated assuming complete combustion (as assumed also in [12] for instance). The friction and the heat flux on the lateral walls of the combustion chamber can be then easily added using the models described in sections 2.3.3 and 2.3.4.

The reaction chemical mechanism can be modelled by means of a single reaction, that involve any number of chemical species. This reaction can be defined in the combustion model by means of its stoichiometric coefficients at products side and reactants side, as in the following:

$$\sum_{i=1}^{N_{spec}} cR_i SP_i \rightarrow \sum_{i=1}^{N_{spec}} cP_i SP_i \tag{2.15}$$

where cR and cP are the aforementioned stoichiometric coefficients, SP_i is the i-th specie name and N_{spec} is the total number of chemical species considered in the model. The composition at the outlet of the combustion module is therefore calculated by means of the following:

$$(X_i)_2 = (X_i)_1 - \frac{cR_i}{cR_{lim}} (X_{lim})_1 + \frac{cP_i}{cR_{lim}} (X_{lim})_1 \quad \forall i \in [1, N_{spec}] \tag{2.16}$$

where *lim* identify the chemical specie, that due to its limited quantity defines the amount of reacting matter.

The possibility of incomplete combustion is accounted for by assuming a combustion efficiency,

thus reducing the enthalpy of the flow downstream of the combustor by means of the following:

$$\Delta H = (1 - \eta) \left[h \left(T_{ref}, X_1 \right) - h \left(T_{ref}, X_2 \right) \right] G_1 \tag{2.17}$$

where the reference temperature T_{ref} is 300K. In Eq. 2.17 the mass flow is considered constant across the combustion chamber (this assumption should always be true unless mass addition occurs in combustion chamber, as happens in case of film cooling for instance) and the term in square brackets represents the energy release of the complete reaction. The value of η can be set constant or variable with the equivalent ratio φ by means of a user's defined lookup table.

2.10 Nozzle

The nozzle can be either with fixed geometry or with variable geometry. In this case however, the interaction with the external flow is not as complex as for the inlet, since the nozzle exit is always sonic or supersonic, thus the external flow cannot influence the functioning of the nozzle.

In the case of variable geometry and convergent-divergent nozzles, the model is composed of a single HyPro module based on the isentropic solver (see section 2.3.1). In the isentropic solver the flow is considered choked in the nozzle throat that therefore is implicitly assumed variable in size. The thermodynamic conditions at the nozzle exit (i.e. node N_2) are first calculated assuming that the nozzle is adapted, so the pressure p_2 is assumed equal to the ambient pressure. Then the nozzle exit area A_2 is calculated imposing the mass balance and if it is greater than the limit set by the user, then the area is set at its maximum and the isentropic solver is used to recalculate the non-adapted conditions at node N_2.

If the nozzle is a simple convergent with a variable throat area, then similarly with the previous model the module is derived by the isentropic solver. In this case the conditions at node N_2 are first calculated assuming sonic exit (i.e. the nozzle is assumed to be always under-expanded, since adapted conditions with subsonic exit are usually only obtained in turbojet engines at low throttle

conditions[64]) and then the area A_2 is again calculated imposing the mass balance. If the area required is higher than the maximum value set by the user, then an 'over-choked' signal is sent by the nozzle module to be properly handled by the system iteration loop, as explained in section 2.11.

Fixed geometry nozzles can be easily modelled as simple ducts, using the isentropic solver without modifications. Convergent-divergent nozzles will have to be modelled with two isentropic duct, one for the convergent section and the other for the divergent section, in order to avoid the solver to model implicitly a nozzle with variable throat area.

All the aforementioned models consider the flow along the nozzle to have frozen composition, common practice assumption for nozzles (see [12, 9]). As for all the modules based on the isentropic solver, a constant pressure drop can be set to model viscous dissipation within the nozzle.

2.11 System

The system module takes care of the integration, interconnection and communication between the various sub-modules. Moreover it manages the interface with the rest of the vehicle system, so that this in turn can treat the propulsion system as a self-contained 'black box'. The system module solves sequentially for the states associated with the various modules contained within it, respecting a predefined order of solution of the various modules. In Fig. 2.14 a typical example of a propulsion system is represented, where the arrows indicate the flow of information through the calculation. Usually the calculation proceeds from upstream to downstream, but when choking occurs within the system then information has to be propagated in the opposite direction to let the upstream flow path adapt to its presence.

Typically in HyPro choking is detected in a module when it is impossible to find a solution for the fluid dynamic equations, due to the fact that the mass flow rate at input node N_1 is too high to pass through node N_2 even in sonic conditions (i.e. at fixed stagnation conditions and area the highest mass flow rate is reached at $M = 1$). This phenomenon can happen due to a number of reasons depending on the module nature. For example simply the area A_2 can be smaller than A_1

or the the stagnation conditions of node N_2 can change due to friction, combustion etc. In any of the above cases, the conditions at node N_1 need to be 'reduced' until a fluid dynamic solution is possible in correspondence with $M_2 = 1$.

This choking loop is handled in HyPro by means of a dedicated C++ pointer dubbed here 'choking feedback'. The module detecting the choking is the source of the choking feedback, while the module pointed by it is the destination. The destination of the feedback is the module that needs to change something in its functioning parameters in order to reach sonic conditions at the outlet of the source module. In HyPro the user can assign to each module a choking feedback variable pointing to any destination upstream, depending on the desired engine behaviour. In Fig. 2.14 for instance, whenever choking is detected in the mixer module the intake model has to adapt, perhaps spilling out some air, while if choking is detected in the combustion chamber then the injection process has to reduce the amount of fuel being injected into the engine. In Fig. 2.14 the nozzle does not have any choking feedback because the engines employs an adaptable nozzle that can change the throat area. The actual action taken by the feedback destination depends on its module type. The implementation of this action is therefore located in the class definition of the destination module, so that it can be easily changed depending on its nature. The action can for instance be of passive nature, like that of a fixed geometry inlet, or it can represent an active feedback, like the reduction of the injected fuel or the adaptation of the inlet areas.

From the system module standpoint this choking loop is always handled in the same way, but with the aforementioned structure it is possible to model many different kind of feedbacks and engine configurations. In Fig. 2.15 the flow chart of the system iteration algorithm is depicted. As can be seen the modules are calculated in sequence and whenever a choking is detected an inner loop takes place until sonic conditions are found at the end of the feedback source module. The feedback loop is executed by means of the false position root finding algorithm (see [60]), this algorithm has been chosen, because the input parameter (i.e. the parameter being changed in the feedback destination module) can be bounded easily at the start of iteration loop, avoiding

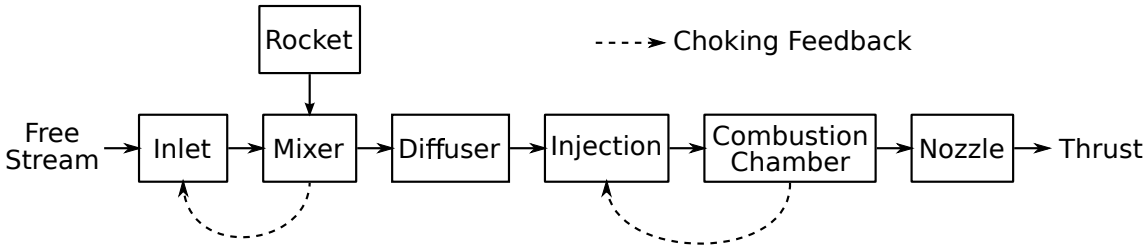


Figure 2.14: Information flow through the propulsion system model.

solutions in correspondence of non-physical values. In HyPro is also possible to consider the choking condition not exactly placed at $M = 1$ in order to take into account of viscous effects and also in order to improve the stability of the choking loop. Often the choking condition is displaced by 0.1, so that it is $M = 0.9$ in subsonic flow and $M = 1.1$ in supersonic, in order to model the effect of viscosity on the choking limit; this shift value however can be changed by the user depending on the problem studied.

The global performances of the engine, such as the thrust or the specific impulse, can be calculated after the engine solving loop is executed and therefore the thermo-kinetic states are known in every node of the model. In Fig. 2.16 a sketch of a generic engine is presented, the system module is interfaced with the external world by means of two nodes as for any other propulsive module. The input node N_1 represents the free stream conditions, while the output node is placed at the exit of the nozzle. The thrust is calculated by considering the balance of momentum across the control volume outlined in Fig. 2.16, therefore it is given by the following:

$$F = \dot{m}_2 U_2 + (p_2 - p_1) A_2 - \dot{m}_{in} U_{in} - (p_{in} - p_1) A_{in} \tag{2.18}$$

where the thrust F is defined as the net force acting on the engine walls, calculated assuming that the external flow is steady at ambient pressure:

$$F = \iint_{\ell} \left(\overline{\overline{\tau}} \cdot \hat{n} - p_1 \hat{n} \right) dA \tag{2.19}$$

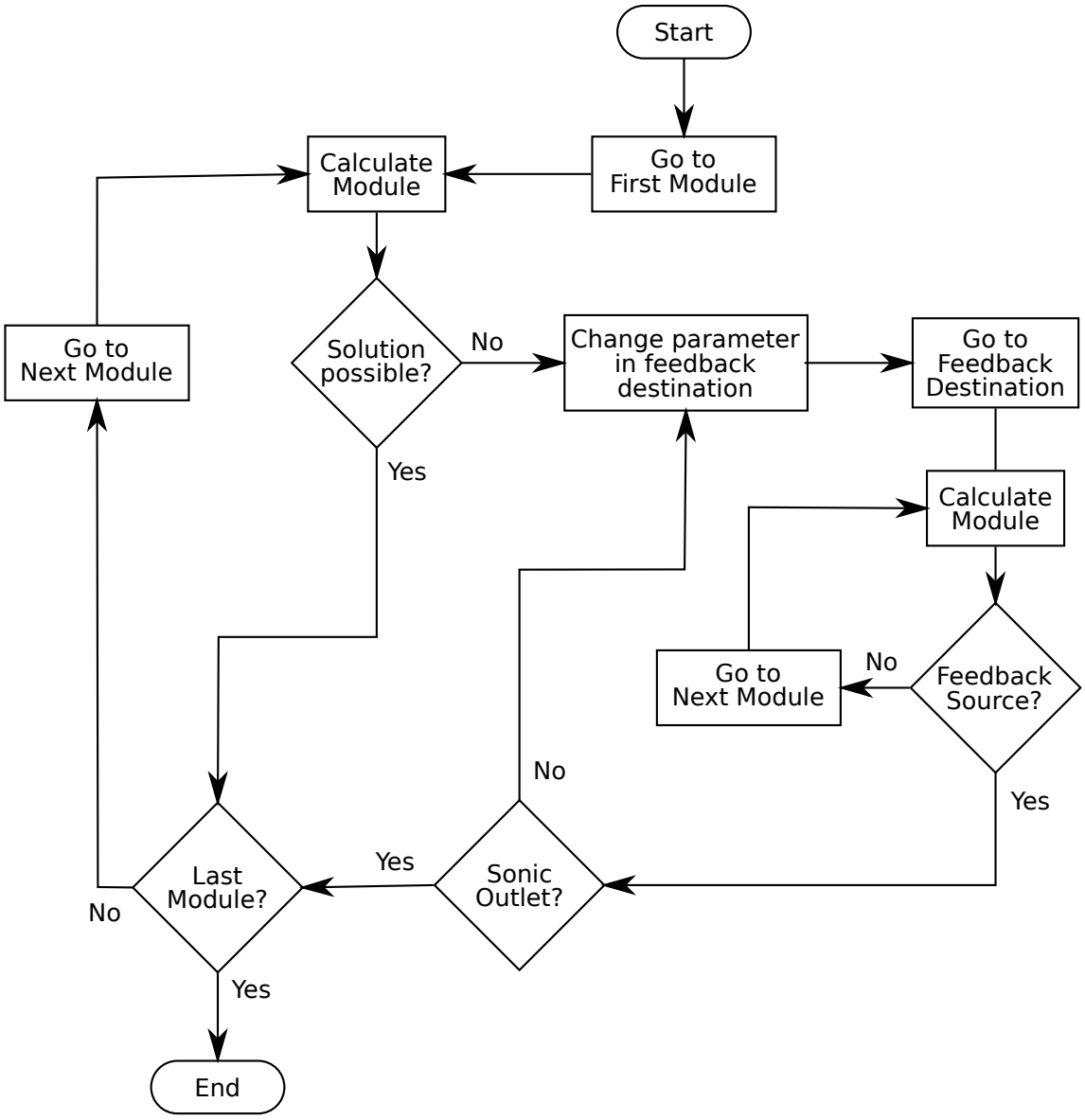


Figure 2.15: System iteration flow chart.

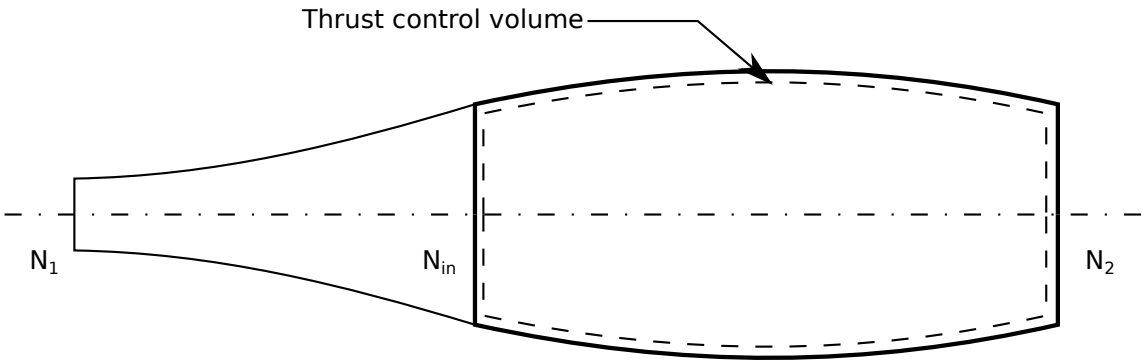


Figure 2.16: Conventions and control volume for the calculation of the engine thrust.

where ℓ is the lateral surface of the control volume in Fig. 2.16 and $\hat{n}$ is the local normal unity vector. Eq. 2.18 however is usually not preferred in the propulsion industry. It would also require the system module to be linked to a node different than the free stream one, seems in general the conditions at the inlet of the intake (i.e. node N_{in}) are not equal to the free stream ones. A much more common formulation is the following (see [65]):

$$F = \dot{m}_2 U_2 - \dot{m}_1 U_1 + (p_2 - p_1) A_2 - D \tag{2.20}$$

where D is the intake additional drag, whose value can be obtained comparing Eq. 2.20 with Eq. 2.18 and observing that $\dot{m}_1 = \dot{m}_{in}$:

$$D = \dot{m}_{in} \left(U_{in} - U_1\right) + A_{in} \left(p_{in} - p_1\right) \tag{2.21}$$

In HyPro the intake drag calculation is carried out by the intake module, since the detail of the implementation can be different depending on the model (see section 2.4).

In Eq. 2.19 the assumption of steady external flow is obviously not strictly true, even at fixed point the suction from the intake and the presence of the exhaust plume sensitively perturb the pressure field around the engine. This choice however is merely a convention used to decouple the thrust and the aerodynamic forces, defining precisely the boundary between the engine subsystem

and the vehicle system. At present the effect of the flow induced by the engine on the vehicle's external aerodynamics is not considered since it is out with the scope of the research.

Chapter 3

CFD model

3.1 Introduction

In addition to its validation against experimental data and other CFD analyses available in the literature, a dedicated CFD campaign has been performed in order to correlate the HyPro software against more physically complete representations of the fluid dynamics within the engine. Of particular importance has been the development of a dedicated CFD model to augment and expand upon the data-sets that are currently available for the model's correlation and verification. The aim has been to create a robust and reliable CFD model that can be used easily to verify the model's performance under particular conditions that might not be fully described in the literature. In addition, as they continue to be developed, the propulsion modules implemented in HyPro will continually need to be validated and correlated, and the best approach would appear to be to compare the predictions of the model to the results of dedicated CFD analyses at various pertinent operating conditions to achieve this objective.

To this end, the open source CFD software OpenFOAM® has been extended and developed to contain a number of algorithms that are appropriate to modelling high-speed propulsion systems. Open source has been preferred to a commercial general purpose CFD code in order to

have full control of the implementation. This choice has many cons in the short term, since the OpenFOAM® software is poorly documented, thus it is very difficult to master. Moreover it lack many physical models, usually available with a simple click in commercial general purpose codes. In the long term however, the open source choice pays off with an unbeatable flexibility and with a much better awareness of the CFD modelling, that is usually much appreciated in the research framework.

Among the open source CFD software available OpenFOAM® has been selected because it is one of the most used examples, it has a very big community of users and it is designed to be general purpose. Thanks to the big community behind it, the poor documentation provided officially is compensated by much non-official information easily accessible on line. Moreover the large community forces the developers to maintain it constantly and its development is unlikely to stop any time soon.

3.2 Numerical implementation

Among all the solvers available in OpenFOAM® , in this research the *rhoCentralFoam* solver developed by Greenshields (see [66]) has been used as a starting point. This solver is density based so it is very well suited to model high speed flows, where compressibility effects cannot be neglected, which is frequently true for propulsive applications. The governing equations for mass momentum and energy are the followings:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \underline{\nabla} \cdot (\rho \underline{u}) = 0 \\ \dfrac{\partial (\rho \underline{u})}{\partial t} + \underline{\nabla} \cdot (\rho \underline{u}\underline{u}) + \underline{\nabla} p + \underline{\nabla} \cdot \underline{\underline{\tau}} = 0 \\ \dfrac{\partial (\rho E)}{\partial t} + \underline{\nabla} \cdot (\rho E \underline{u}) + \underline{\nabla} \cdot (p \underline{u}) + \underline{\nabla} \cdot \left(\underline{\underline{\tau}} \cdot \underline{u}\right) + \underline{\nabla} \cdot \underline{\dot{q}} = 0 \end{cases} \tag{3.1}$$

where $E = e + \frac{u^2}{2}$ is the total specific energy. Since OpenFOAM® employs the finite volume technique, Eq. 3.1 are solved considering their integral form applied to every computational cell.

$$\begin{cases} \dfrac{\partial}{\partial t}\displaystyle\int_{\Omega} \rho dV + \int_{\partial\Omega} (\rho \underline{u}) \cdot \hat{n} dA = 0 \\ \dfrac{\partial}{\partial t}\displaystyle\int_{\Omega} (\rho \underline{u})\, dV + \int_{\partial\Omega} (\rho \underline{\underline{u}} \cdot \hat{n} + p\hat{n})\, dA + \int_{\partial\Omega} \underline{\underline{\tau}} \cdot \hat{n} dA = 0 \\ \dfrac{\partial}{\partial t}\displaystyle\int_{\Omega} (\rho E)\, dV + \int_{\partial\Omega} (\rho E + p)\, \underline{u} \cdot \hat{n} dA + \int_{\partial\Omega} \left(\underline{\underline{\tau}} \cdot \underline{u} + \underline{\dot{q}}\right) \cdot \hat{n} dA = 0 \end{cases} \tag{3.2}$$

where Ω is the computational domain of the CFD cell and $\partial\Omega$ is its boundary. Then discretization is applied to Eq. 3.2 assuming that the value of any quantity at the cell centre is equal to the average on the cell volume.

The discrete governing equations are solved sequentially segregated from each other, since in OpenFOAM® a coupled algorithm was not available at the time of *rhoCentralFoam* development. The time iteration is split in two steps. In the first only the inviscid part of Eq. 3.2 is solved and an intermediate value of solution fields is found, then in the second step the solution is updated adding the viscous terms. The two step strategy is selected in order to use different numerical methods for viscid and inviscid terms. The inviscid equations are solved with an explicit method while for the viscous terms an implicit method is used to ensure stability.

The advantage of the explicit solver used for the inviscid terms is to enable easy implementation of non linear expressions for the convective fluxes (see section 3.3). On the other hand the disadvantage is the conditioned stability linked implicit solver, to ensure stability the time step has to be such that the Courant number is lower than one everywhere in the computational domain (see [67]). This condition often leads to very small time steps and, as a consequence, a large number of iterations are needed to reach the convergence. To mitigate the aforementioned problem, in OpenFOAM® it is possible to use the local time stepping technique, that modulates the time step depending on the local Courant number, so that the solution can proceed faster where the mesh is coarser. In this case of course the time step has not any physical meaning, and the obtained

solution will be valid only at the steady state.

3.3 Convective fluxes schemes

Before proceeding with the description of the flux scheme, it is worth to define some nomenclature and conventions used in OpenFOAM® and in general for finite volume unstructured solvers. In OpenFOAM® the field data are stored usually at the cell centres (see point P and N in Fig. 3.1). Each cell face is either shared by two cells, in this case it is called internal face, or it is part of the domain boundary and thus it is connected only to one cell. In OpenFOAM® each internal cell face is conventionally owned by one of the two cells it connects (see cell P in Fig. 3.1), while the other cell is called neighbour (see cell N in Fig. 3.1). The assignment of cell face ownership is done at mesh construction time and from the solver point of view is irrelevant; however in an unstructured mesh it is fundamental to define the cell normal, always directed from owner to neighbour.

The convective term of Eq. 3.2 in finite volume CFD is discretised as follows:

$$\int_{\partial\Omega} \psi \underline{u} \cdot \hat{n} dA \approx \sum_{i=1}^{N_{faces}} k_i \psi_i \underline{u}_i \cdot \hat{n}_i A_i = \sum_{i=1}^{N_{faces}} k_i \psi_i \phi_i \tag{3.3}$$

where ψ is the generic extensive variable per unit volume (namely any variable proportional to the quantity of fluid i.e. density ρ, one of the three momentum components $\rho\underline{u}$ or the total energy ρE), ϕ is the volumetric flux and k is 1 if the face is owned by the cell Ω and -1 otherwise (see [66]). Since the field values in OpenFOAM® are commonly stored at the cell centres, the fluxes at each cell face have to be calculated based on the field value at centres of cells on both side of the face (see Fig. 3.1). In incompressible solvers a simple interpolation with a certain degree of upwinding (i.e. the interpolation weights are in favour of the cell upstream the volumetric flow) is usually sufficient to guarantee stability. In compressible flow however the propagation waves move in general in all directions and the definition of upwind itself becomes more difficult. Usually the fluxes calculation in compressible flow involve the solution of the Riemann problem taking place at

the boundary between the two contiguous cells.

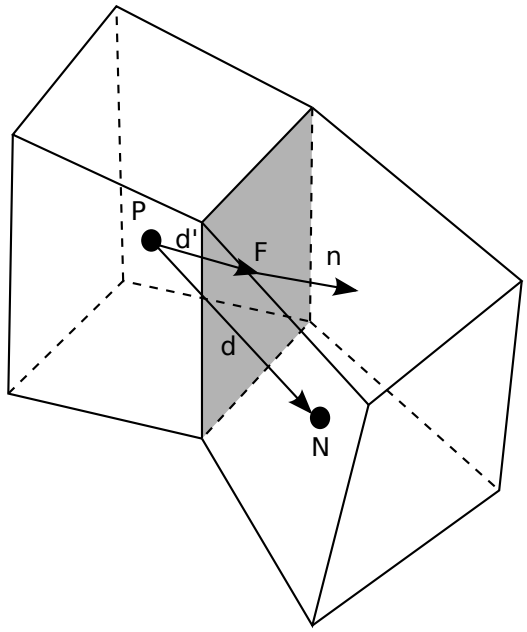


Figure 3.1: Definition of nomenclature for cell face interpolation.

3.3.1 The Kurganov scheme

The *rhoCentralFoam* solver calculates the convective fluxes on the cell faces by means of a numerical scheme developed by Kurganov and al. in [68] accordingly to the following:

$$\phi_i \psi_i = \frac{c^N \phi^N \psi^N + c^P \phi^P \psi^P}{c^N + c^P} + \frac{c^N c^P}{c^N + c^P} \left(\psi^P - \psi^N \right) \tag{3.4}$$

where c^N and c^P are the maximum wave propagation volumetric flows toward respectively the neighbour and the owner cell, and are given by the following:

$$\begin{aligned} c^P &= max(a^P A_i - \phi^P, a^N A_i - \phi^N, 0) \\ c^N &= max(a^P A_i + \phi^P, a^N A_i + \phi^N, 0) \end{aligned} \tag{3.5}$$

where a is the speed of sound. In Eq. 3.4 and Eq. 3.5 all the other superscripts P and N indicate quantities interpolated to the cell face from respectively the owner and neighbour cell side.

The interpolation is performed starting to the field values stored at owner and neighbour cells by means of the following:

$$\begin{aligned} \psi^P &= \beta C_{lin}(\psi_N - \psi_P) + \psi_P \\ \psi^N &= \beta\left(1 - C_{lin}\right)\left(\psi_P - \psi_N\right) + \psi_N \end{aligned} \tag{3.6}$$

where C_{lin} is the coefficient of linear interpolation given by:

$$C_{lin} = \frac{\underline{d}' \cdot \hat{n}}{\underline{d} \cdot \hat{n}} \tag{3.7}$$

and β is a limiter function (see [67]), that modulates among second order central linear interpolation ($\beta = 1$), downwind ($\beta = 2$) and first order upwind ($\beta = 0$) interpolation, where in this case the upwind direction (indicated with either P or N) is chosen arbitrarily a priori, since the wave propagation is accounted for directly in Eq. 3.4.

For β many different choices are available in OpenFOAM® , but in the frame of this reasearch it has been limited to VanLeer and MinMod limiters (see [67]). The VanLeer limiter function is defined as follows:

$$\beta(r) = \begin{cases} 0, & \text{if } r \leq 0 \\ \dfrac{2r}{1+r}, & \text{if } r > 0 \end{cases} \tag{3.8}$$

while the MinMod is:

$$\beta(r) = \begin{cases} 0, & \text{if } r \leq 0 \\ r, & \text{if } 0 < r \leq 1 \\ 1, & \text{if } r > 1 \end{cases} \tag{3.9}$$

It is possible to demonstrate that both the aforementioned limiters are Total Variation Diminishing (TVD), thus ensuring stability of the solution. Here the term TVD refers to numeric schemes that ensure the total variation (a measure of the degree of oscillation of a discrete function) of the solution diminishes from one iteration to the following [67]. In Eq. 3.8 and Eq. 3.9 the parameter

r is the consecutive difference ratio, defined as the ratio between the upstream variation over the local variation of the interpolated quantity. Whereas such a definition is trivial in case of structured meshes, in unstructured meshes, where the upstream cell is not clearly identified, the calculation of r takes the form:

$$r^{P|N} = 2\frac{(\underline{\nabla}\psi)_{P|N} \cdot \underline{d}}{\psi_N - \psi_P} - 1 \tag{3.10}$$

where again the superscripts P and N denote the upwind cell.

It is worth noticing that the Kurganov flux scheme is convenient because it does not require the solution of the Riemann problem at each cell face. The Eq. 3.4 is indeed obtained in [68] doing some conservative assumption on the propagation wave speeds, thus including the property of the Riemann problem in a very simple formulation. The Kurganov scheme therefore has the advantage to be simple to implement and relatively light in terms of computational costs. The assumptions made however lead to an artificial viscosity term explicitly added to the numerical scheme. The last term of Eq. 3.4 can be thought of as a viscous term since it tends to smooth out the differences across the cells, because the difference $\psi^P - \psi^N$ has the same sign of the inter-cell difference $\psi_P - \psi_N$. The artificial viscosity, although beneficial in terms of stability, can be too high in some applications.

3.3.2 The WAS scheme with exact Riemann solver

In order to avoid the biases of the Kurganov flux scheme (see section 3.3.1), the *rhoCentralFoam* solver has been integrated with other flux schemes based on the solution of the Riemann problem. A new more general compressible solver has been generated and called simply *rhoFoam* to underline the fact that the solver does not necessary rely on the Kurganov central scheme. In this new solver the flux schemes' implementation has been moved from the solver main source file to the OpenFOAM® kernel in order to produce a general implementation that can then be called also from other completely different solvers. This new code structure should facilitate the future development of density based solvers in OpenFOAM® , that for now are quite limited in capabilities.

In contrast with the Kurganov approach, these flux schemes rely on the solution of the Riemann problem taking place at each cell face. In this case the exact Riemann solver described by Toro in [67] is employed, it solves the one-dimensional Riemann problem by means of an iterative procedure based on simple algebraical equations. This solver is only strictly applicable for the Euler equations (i.e. Eq. 3.1 without viscous terms) with ideal gas model, it is however implemented into a dedicated C++ class so that in future it could be easily substituted by other more general solvers. The benefit of the aforementioned solver, beside its simplicity of implementation, is the capability to provide an exact solution of the Riemann problem, thus resolving all the wave propagation speeds. This feature is very useful in presence of complex fluid patterns with multiple shock interaction.

The Riemann problem is generated considering the flow field constant on each computational cell at a certain time step n, therefore generating a discontinuity in correspondence of each cell face. The evolution of the discontinuity in time can therefore be described by the solution of a one dimensional Riemann problem at each cell face, where the spatial coordinate is the local coordinate of the cell face (i.e. normal to the face and having origin to the face centre). In the case of Euler equations the solution will be of the form sketched in Fig. 3.2, four zones are delimited by three waves, the two waves separating the undisturbed conditions from the central zone (thick lines in Fig. 3.2) can be a shock wave or a train of compression waves, while the central wave, that separates the central zone in two, is always a contact discontinuity. The contact discontinuity, in contrast with a shock wave, does not have any pressure jump across it but only a temperature difference between the two sides of the wave. The solution of the Riemann problem is also self similar along lines passing by the origin of the plane $x-t$, therefore the solution for a generic quantity ψ can be expressed as $\psi = \psi\left(\frac{x}{t}\right)$.

The simplest approach to derive a flux scheme from the Riemann problem solution was firstly proposed by Goudonov (see [67]) and consists of integrating between two consecutive time steps (i.e. between t^n and t^{n+1}) the integral form of the Euler equations (i.e. Eq. 3.2 without the viscous terms). Recalling Eq. 3.3 the Euler equation for the generic conserved variable ψ can be then

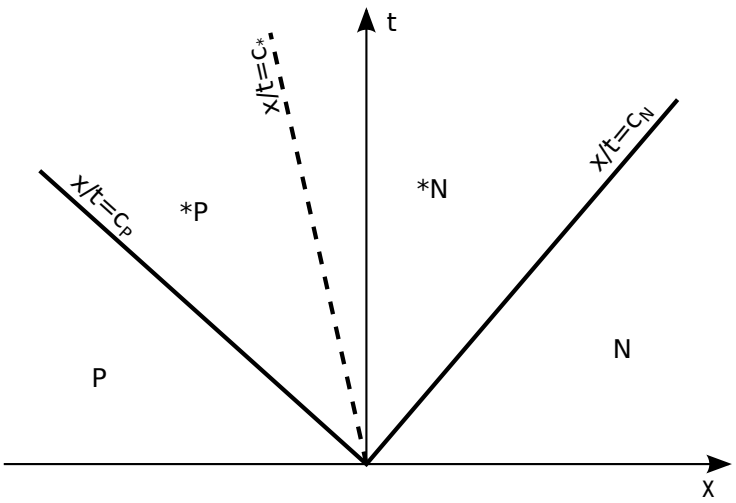


Figure 3.2: Sketch of typical solution of the Riemann problem for Euler equations.

written:

$$\psi_P^{n+1} = \psi_P^n + \left(t^{n+1} - t^n\right) \sum_{i=1}^{N_{faces}} k_i \psi_i(0) \phi_i(0) \tag{3.11}$$

where $\psi_i(0)\phi_i(0)$ comes from the Riemann problem solution evaluated on the cell face, thus when $\frac{x}{t} = 0$. Eq. 3.11 is only valid when the waves departing from the cell face do not cross the other cell boundaries, therefore the derived scheme cannot be used if the selected time step does not match the following requirement:

$$\Delta t = t^{n+1} - t^n \leq min\left[2\frac{x_P}{c_P}, 2\frac{x_N}{c_N}\right] \tag{3.12}$$

where $x_P = -\underline{d}' \cdot \hat{n}$, $x_N = (\underline{d} - \underline{d}') \cdot \hat{n}$ (see Fig. 3.1) and $c_{P|N}$ are the propagation speeds of the most external waves (see Fig. 3.2). It can be demonstrated that the scheme of Eq. 3.11 is first order accurate (see [67]), thus it is usually undesirable for practical applications. This Goudonov first order scheme however has been implemented in *rhoFoam* since it is very stable, therefore it is very convenient during initialization of steady state simulations.

A second order accurate scheme for the convective fluxes can be derived using the Weighted Average States (WAS) approach, where the fluid dynamic quantities on the cell faces can be calculated

integrating the solution of the Riemann problem along the spatial dimension x at the intermediate time step:

$$\psi_i = \frac{1}{x_N - x_P} \int_{x_P}^{x_N} \psi \left(\frac{2x}{t^{n+1} - t^n} \right) dx \tag{3.13}$$

Since the solution of the Riemann problem $\psi\left(\frac{x}{t}\right)$ is a piecewise constant function, the Eq. 3.13 can be easily written as an algebraic sum:

$$\psi_i = \frac{1}{2}\left(\psi_P + \psi_N\right) - \frac{1}{2} \sum_{i=[P,*,N]} Co_i \Delta\psi_i \tag{3.14}$$

where Co_i is the Courant number related to any of the three propagation waves:

$$Co_i = \frac{t^{n+1} - t^n}{x_N - x_P} c_i \tag{3.15}$$

and $\Delta\psi_i$ is the jump of the general variable ψ across the propagation wave:

$$\begin{aligned} \Delta\psi_P &= \psi_{*P} - \psi_P \\ \Delta\psi_* &= \psi_{*N} - \psi_{*P} \\ \Delta\psi_N &= \psi_N - \psi_{*N} \end{aligned} \tag{3.16}$$

The scheme of Eq. 3.14 can however produce unwanted oscillations around discontinuities in the flow fields, as stated by the Goudonov theorem (see [67]). To overcome this limitation the WAS scheme can be made TVD using a limiter function as done for the Kurganov scheme in section 3.3.1. Therefore the Eq. 3.14 becomes:

$$\psi_i = \frac{1}{2}\left(\psi_P + \psi_N\right) - \frac{1}{2} \sum_{i=[P,*,N]} w_i \Delta\psi_i \tag{3.17}$$

where w is the new TVD weight defined as follows (see [67]):

$$w_i = sign(Co_i)\,[1 - (1 - |Co_i|)\beta(r_i)] \tag{3.18}$$

The consecutive differences ratio r in this case is the ratio between the upstream and the local delta defined in Eq. 3.16, rather than the simple difference between consecutive cells. In mathematical terms it can be defined as follows:

$$r_i = \begin{cases} \dfrac{\Delta\psi_i^P}{\Delta\psi_i} = 1 - \dfrac{\underline{d}\cdot(\underline{\nabla}\psi_i)_P}{\Delta\psi_i} & \text{if } c_i > 0 \\ \dfrac{\Delta\psi_i^N}{\Delta\psi_i} = 1 + \dfrac{\underline{d}\cdot(\underline{\nabla}\psi_i)_N}{\Delta\psi_i} & \text{if } c_i \leq 0 \end{cases} \tag{3.19}$$

where the upstream direction is defined with respect to the propagation of the wave considered. As well as for the previous flux scheme here the identification of upstream face and its Riemann solver is not trivial, since the generally mesh is not structured. Similarly to Eq. 3.10 in Eq. 3.19 the upstream value for $\Delta\psi_i$ has been calculated using the gradient evaluated at the cell centre.

The WAS flux scheme with TVD limiter described in Eq. 3.17 is preferable to the Kurganov one due its much lower numerical diffusivity, the main disadvantage however is its complexity, resulting in a much longer iteration time. It will be shown however that for some applications, where the interaction between shocks and viscosity has to be well captured, the accuracy provided by the WAS with respect to the Kurganov scheme is a necessity.

3.4 Validation

In order to verify its predictions, our CFD approach has been used to model the scramjet testbed that was studied by Lorrain[11] during the development of the SCRAMSPACE I propulsion system. This test case has been selected because it involves the complex fluid dynamics typical of hypersonic engines and because the CFD model of Lorrain was in turn validated against shock tunnel tests (see [69]), showing a very good agreement with experiment.

Lorrain's testbed is a channel with constant width, a two-slope intake, a constant area combustion chamber and a single-slope nozzle (see Fig. 3.3 and [11]). The testbed was operated in a shock tunnel using air and hydrogen as fuel. The fuel was injected into the intake in order to facilitate pre-mixing prior to combustion, and to take advantage of the concept of 'radical farming' for ignition of the fuel-air mixture (see [11, 70, 71]).

In a scramjet engine, due to the high speed inside the combustion chamber, it is very challenging to limit the required length of the engine, especially when the temperature is also limited at the combustion chamber inlet. The 'radical farming' occurs when the fuel is injected in the inlet and it partially mixes with the air before the combustion chamber. The fuel/air mixture than ignites in the combustion chamber even when the average temperature at the inlet is lower than the auto ignition condition. This phenomenon takes place thanks to the presence of a shock waves' train in the combustion chamber, that produces a series of hot spots where locally the conditions allow the mixture to auto-ignite. In the first hotspot some very reactive radicals are created without the release of heat and the consequent increase in pressure, then after the second hotspot the actual combustion process takes place.

As done in [11], the gas model is ideal but with variable specific heat depending on temperature. The specific heat law is given by a polynomial function[72]. The gas mixture is modelled as a single component, whose properties result from the weighted average of the properties of its components. The chemical species considered are the Nitrogen and the Oxygen of the air and the Hydrogen fuel, the other components of the air are neglected due to their limited amount (see Tab. 3.2). The transport properties (i.e. viscosity and thermal conductivity) are calculated by means of the Sutherland law [73]. In all cases described below the flow is considered to be fully turbulent, and the turbulence is modelled using the SST $k-\omega$ model with standard wall functions [74]. Among the numerical schemes available in *rhoFoam* (see section 3.3) here the WAS has been preferred to the Kurganov flux scheme, due to its lower diffusivity.

As in Lorrain's work[11], our CFD analysis was performed first in a fuel-off condition (see

Table 3.2) in order to gather data for use in the subsequent fuel-on analysis. The intake inlet conditions are listed in Table 3.1. The results of this preparatory run are illustrated in the pressure contour plot shown in Fig. 3.4. The fuel-on CFD calculation is performed only for that part of the duct downstream of the intake (see Fig. 3.5). For equivalence with Lorrain's analysis, the pressure, density and velocity fields produced during the fuel-off computation were interpolated onto the combustion chamber inlet and then applied as boundary conditions for the fuel-on case. Moreover, the pressure downstream of the intake was increased compared to that of the fuel-off case, using Eq. 2.12 and assuming the flow to be evenly mixed at the outlet of the intake[11], in order to take into account the increased mass flow due to the fuel that was injected into the intake.

		Intake Inlet
U	[m/s]	2830
p	[Pa]	4100
T	[K]	370

Table 3.1: Flow conditions at the inlet of the intake.[11]

Species	Fuel-Off	Fuel-On
H_2	0.0	0.250
O_2	0.209	0.157
N_2	0.791	0.593

Table 3.2: Mixture composition in terms of molar fraction.

A contour plot of the pressure in the fuel-on case, assuming frozen conditions, is depicted in Fig. 3.5. In Fig. 3.5 a streamline close to the combustor walls is also plotted, the pressure over this streamline has been then used to compare against the data reported by Lorrain in [11]. As can be seen by comparing the data presented in Fig. 3.6, the results from our CFD model compare very well with Lorrain's calculations. Slight differences between the two calculations are evident, however, in the region of the nozzle. Despite in [11] a full reacting case is also available, here only this frozen computation has been completed, since a combustion model development has been considered out of scope of the present research. In Fig. 3.6 the results obtained employing the Kurganov flux scheme

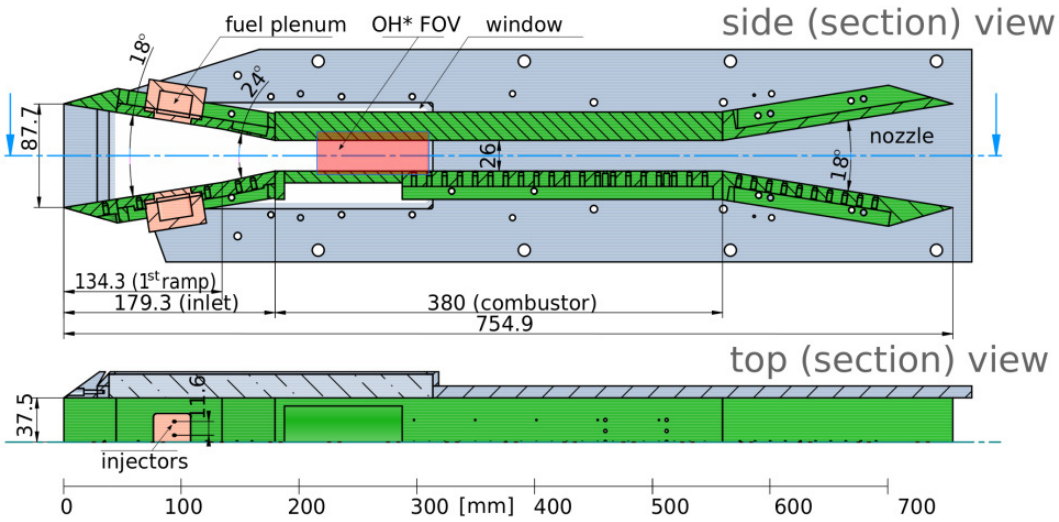


Figure 3.3: Geometry of the SCRAMSPACE I test bed, see [11]

have been also plotted for completeness. As expected the Kurganov scheme does not deliver the same degree of accuracy obtained with the WAS scheme. The increased diffusivity of the Kurganov scheme leads to the wrong prediction of the shocks interaction with the boundary layer and to the consequent delay of the pressure peaks observed on the streamline.

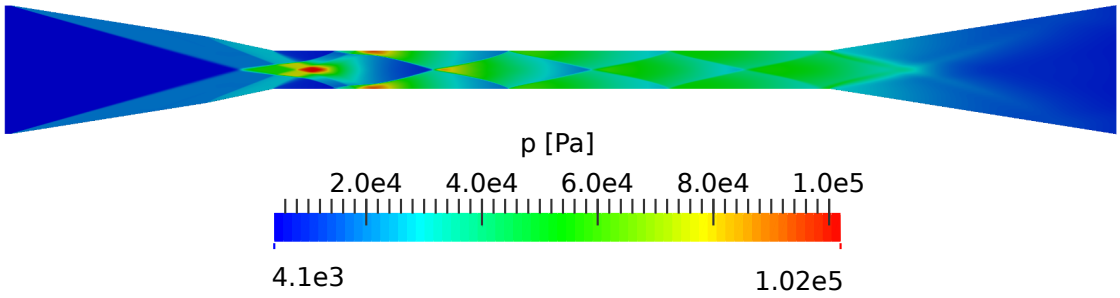


Figure 3.4: RhoFoam CFD results for Lorrain's geometry in fuel-off conditions, showing pressure contours [Pa] within the engine duct.

To conclude, this test case proves the validity of the CFD model described in this chapter, in the case of internal high speed fluid dynamic. The fact that the test case was very challenging with multiple shocks interactions and also shock boundary layer interactions, gives high confidence to the usage of this CFD approach for the other applications of this research, that are surely less complicated. It must be also pointed that the aforementioned test case can be considered a proper

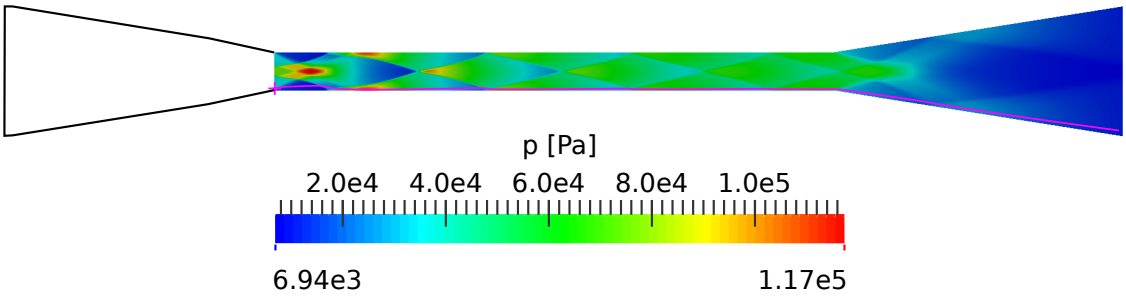


Figure 3.5: RhoFoam CFD results for Lorrain's geometry in fuel-on conditions, with frozen chemistry, showing pressure contours [Pa] within the engine duct.

validation, since the reference CFD model carried out by Lorrain in [11] was then validated against shock tunnel data in [69]. The validation performed by Lorrain was accomplished including in the model also the three dimensional effects of the fuel injection, while the model used as reference here is two dimensional. The physical and numerical model was however not changed and can therefore be considered validated. Moreover a 2D test case is better suited as first validation step, especially considering that all the CFDs required in this research are 2D.

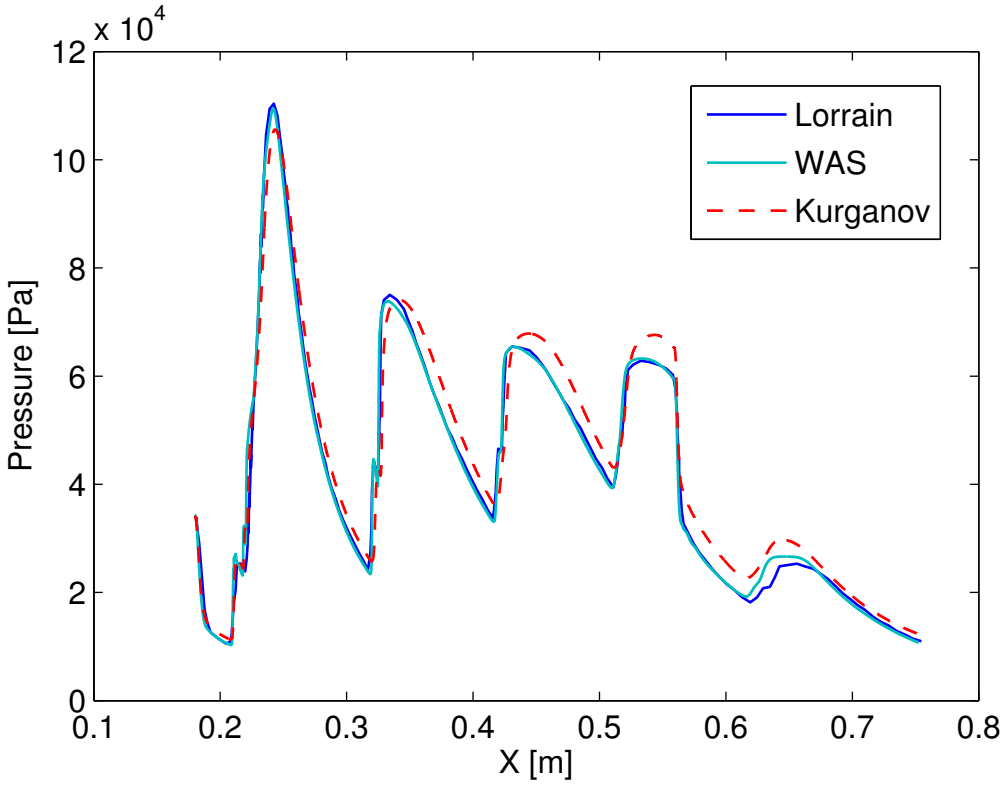


Figure 3.6: Comparison between rhoFoam predictions and Lorrain's[11] calculations. The pressure profile is calculated along a pathline close to the combustion chamber wall (see Fig. 3.5).

Chapter 4

Validation and verification of the parsimonious model

4.1 Introduction

In order to provide a validation/verification of its predictions, the HyPro software has been used to model several test cases for which data are available from experimental sources, other CFD analyses or the predictions of other simplified models. The complete list of the test cases described in this section is reported in the test matrix in Tab. 4.1.

The HyPro code has been verified or validated at different levels of comprehensiveness, from a single module to the whole engine system in its entire operational range. In this chapter, the words 'Validation' and 'Verification' refer to the assessment of accuracy of the code by comparison against respectively experimental data and outcome of other computational models. This nomenclature is the most adopted today, for instance described in [75].

The validation has been performed only in two test cases, due to the lack of experimental data and because it is difficult to compare the results of a one dimensional code against wind tunnel

[1] Supercharged Ejector RamJet

Name	Section	Flight Regime	Type	level
Fanno flow	4.2.1	any	Verification	Friction model
Rayleigh flow	4.2.2	any	Verification	Combustion model
SCRAMSPACE I	4.3	Hypersonic	Validation	system at single operation point
Hyperion	4.4	Subsonic/ Supersonic	Verification	Combined cycle system wide operational range
Marquardt SERJ[1]	4.5	Subsonic	Verification Tuning	Mixer model
STX	4.6	Supersonic	Validation	System along flight trajectory

Table 4.1: Test matrix

test, where data are usually available on the wall and are not representative of the average across the engine section area (see for instance section 4.3).

The SCRAMSPACE I case involves a shock tunnel test, but only the CFD reconstruction of the test is used for comparison. Even if the CFD is a computational model, the comparison is still considered a validation because the CFD model was in turn validated against test data. In contrast with the test however, the CFD can be used to retrieve section averaged data, that are directly comparable against HyPro.

A much better validation test case is instead the STX one, because it provides global experimental performances data over a large set of operational conditions. The global data such as the thrust or the specific impulse are better means of comparison, because they are the final target of the HyPro model and they result directly from averaged properties. The availability of data across a wide operational range on the other hand is a guarantee that the model is not just well fit to a single operative condition. An engine model can indeed contain many uncertain parameters, such as the combustion efficiency, the inlet pressure drop or the friction in any module of the engine, that can be tuned until the engine actual performances are reproduced. Once the tuning is performed at a certain operational point however, there is no guarantee that the accuracy is also good outside of the tuning point. A test case such as the STX is therefore important to verify that the engine

model can fit the data with a single set of efficiencies parameters.

All the other test cases reported in Tab. 4.1 cannot considered validations since they use data coming from other computational models. These comparisons are however useful to give a qualitative assessment on the behaviour of the code and are very useful to evaluate single components of the code that would be cumbersome to validate against tests. The comparison against Fanno and Rayleigh flows is used for instance only at component level, but is very useful to demonstrate the general correctness of some models in HyPro (see section 2.3), which are widely used as basic building blocks of any complex engine. The Hyperion test case is used to prove, to the best of our knowledge, that HyPro can be used also for very complex Combined Cycle engines, for which experimental data are not available in the literature. In the end the Marquardt test case shows how high fidelity models (i.e. CFD in this case) can be used to verify and tune the simplified HyPro model. In this case the CFD data are used not only to qualitatively verify the behaviour of the model, but also to fine tune a parameter of the model (the mixing efficiency defined in section 2.6) that is usually affected by high uncertainty.

4.2 Verification against analytical models

4.2.1 Fanno flow

In this section HyPro predictions will be compared against the well known Fanno flow (see [65, 59]), that in case of constant γ gas has an analytical solution. The Fanno flow is a one dimensional flow characterised by a constant mass flow (i.e. constant duct section area and no mass addiction in the duct) and constant total enthalpy (i.e. no heat exchange) but with a non constant impulse due to the presence of friction. In the Fanno analytical model the friction is modelled by means of a constant friction coefficient, so this test case is ideal to verify the validity of the HyPro friction model that also considers a constant friction coefficient (see section 2.3.3).

In Fig. 4.1 the nomenclature employed in this test case is sketched, together with the definition

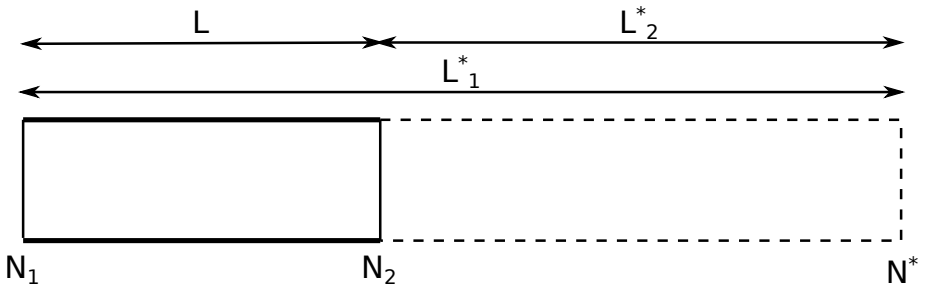


Figure 4.1: Stations definition for the Fanno flow model.

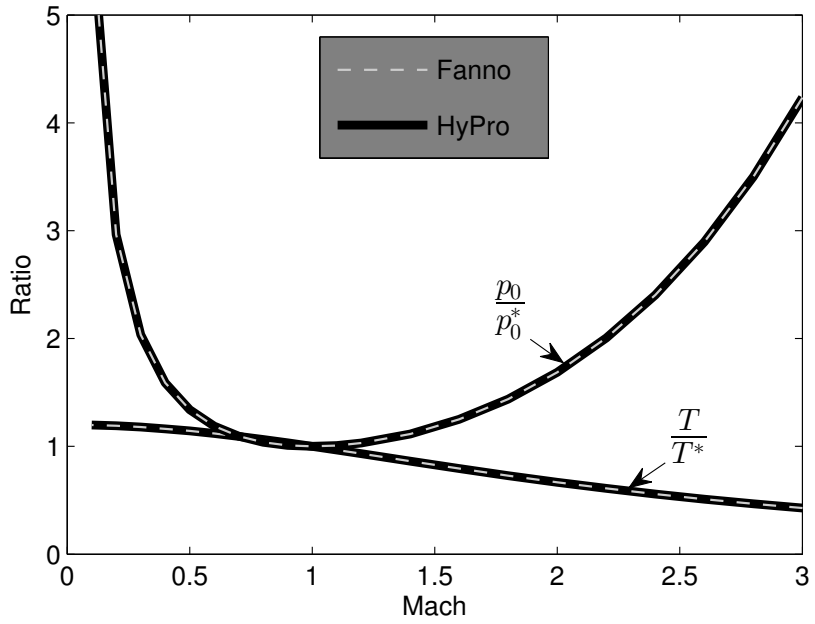


Figure 4.2: Comparison between Fanno analytical solution and HyPro: Static temperature ratio and stagnation pressure ratio.

of the lengths used in the common Fanno flow case. The starred conditions, used in the analytical model to facilitate the calculations, indicates sonic conditions reached at the outlet of an imaginary longer duct. In this comparison the HyPro friction model is iteratively used to find the sonic length L^* as a function of the Mach number in order to have a direct comparison with the analytical formulas of the Fanno flow. The gas model in this case is assumed to have a constant γ ($\gamma = 1.4$) to eliminate the differences due to real gas effects.

In Fig. 4.2 the results of this comparison are showed in terms of thermodynamic properties' ratios between inlet and sonic section, typically used in Fanno flow. It is evident that the two

results are actually identical, demonstrating the validity of the thermo-kinetic model implemented in HyPro. However the missing relation needed to close the problem of a generic non-sonic Fanno duct is plotted in Fig. 4.3, it is the normalized sonic length as function of inlet Mach number. As can be observed in this case there is a sensible difference between the Fanno analytical model and HyPro. Given a duct with fixed length, the aforementioned difference causes an error in the HyPro prediction of the Mach number at the outlet of the duct M_2, that in turn causes error in all the other thermo-kinetic properties. The difference visible in Fig. 4.3 is due to the fact that the momentum flow in HyPro (see Eq. 2.5) is calculated based only on the dynamic pressure at the inlet of the duct (node N_1 in Fig. 4.1), while in the analytical Fanno formulation it is correctly integrated considering the variation of the dynamic pressure along the duct. This difference however, is not as big in practice, because it is very unlikely to have a duct as long as the sonic length L^*. In Fig. 4.3 the error on M_2 is also plotted as a function of inlet Mach number M_1, unity duct aspect ratio $\frac{L}{D} = 1$ and friction coefficient $C_f = 0.01$, that are all common values for engine components. The error plotted in Fig. 4.3 of course works just as a guideline and it is always smaller that 0.1, therefore in most cases it is not an actual problem.

Despite the aforementioned difference with the Fanno analytical solution, the strategy implemented in HyPro is preferable to the former because it is more general and more flexible. Fanno analytical formulas are rigorously valid only for constant γ gases and cannot be coupled with anything else, such as heat exchanges, mass addictions or gas composition changes, while as seen in chapter 2 the HyPro friction treatment can be coupled with many other models very easily. In principle there is not any difficulty in using a Fanno analytical model in HyPro, but due to the flexibility needs explained in section 3.1 and the small improvement in accuracy the simpler method explained in section 2.3.3 is almost always preferable.

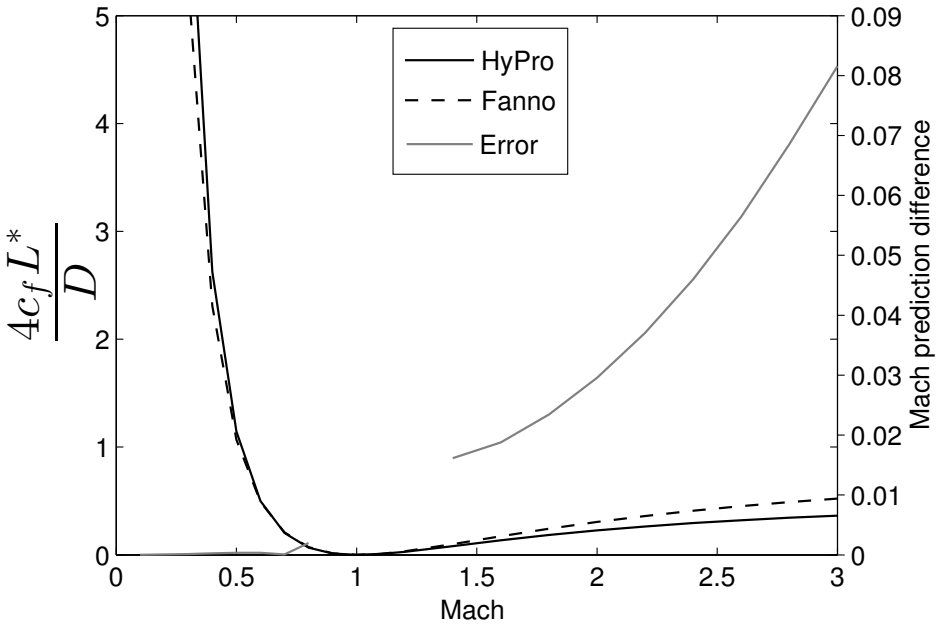


Figure 4.3: Comparison between Fanno analytical solution and HyPro: critical length and consequent error on M_2 in case of $\frac{L}{D} = 1$ and $C_f = 0.01$.

4.2.2 Rayleigh flow

Similarly with section 4.2.1 here HyPro has been compared against an analytical solution of the Rayleigh flow. As it is well known, Rayleigh flow is a one dimensional flow characterised by a constant mass flow a constant impulse (i.e. no presence of friction with lateral walls) and a variable total enthalpy due to the presence of a heat source. Usually Rayleigh flow is employed to model the engine combustion chamber, where the source of heating comes from the combustion process. The analytical solution of the Rayleigh flow is strictly valid with the hypothesis of fixed composition gas with constant γ, this hypothesis has not been considered in the HyPro model because in HyPro there is no combustion without change of composition. A closer comparison would have been actually possible using the HyPro heat flux model (see section 2.3.4), but it would have not added any value to the comparison showed in section 4.2.1.

In Fig. 4.4 the thermo-kinetical properties' ratios between the inlet and sonic station (here the same nomenclature of the Fanno flow, explained in Fig. 4.1, applies) are shown. As can be seen, all

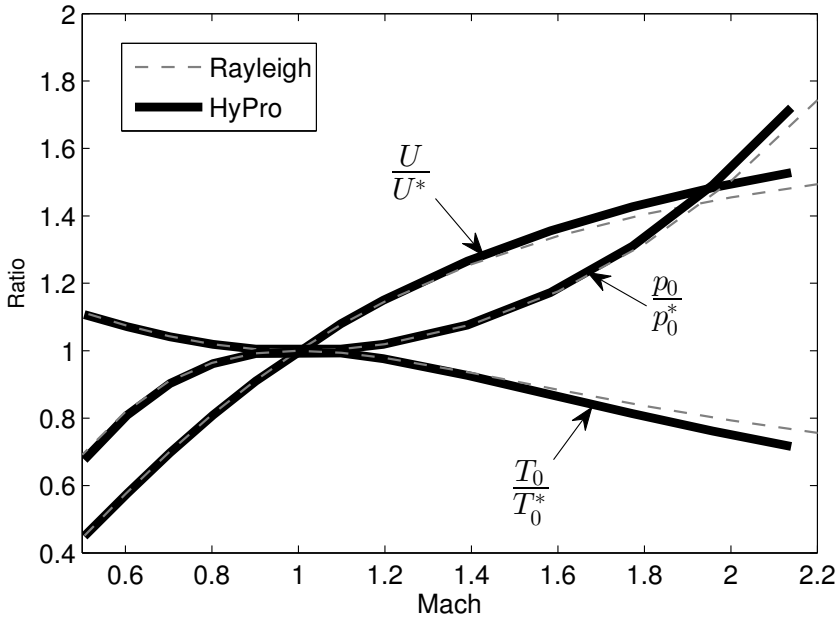


Figure 4.4: Comparison between Rayleigh analytical solution and HyPro: velocity ratio, stagnation temperature ratio and stagnation pressure ratio.

three ratios show only small differences between the HyPro model and the Rayleigh one, in any case the main behaviour is the same. It is however difficult to get any quantitative comparison since the Rayleigh analytical model cannot be considered the closest to reality, due to the aforementioned limitations. Nevertheless this comparison is useful to qualitatively verify the HyPro combustion chamber model, since Fig. 4.4 demonstrates how it behaves similarly to the Rayleigh model, that is universally accepted as a good simplified model of the combustion chamber [76].

4.3 SCRAMSPACE I test case

In this section HyPro will be tested for a whole engine system in order to verify its capabilities when applied to a scramjet engine configuration. The test case selected is the scramjet testbed that was studied by Lorrain [11] during the development of the SCRAMSPACE I propulsion system and it has been used in this research also to validate our CFD model in section 3.4.

The aforementioned test bed was analysed in [11] by means of CFD analyses in reacting and

frozen chemistry hypothesis at the inlet conditions reported in Tab. 3.1. In [11] the CFD data are available in terms of pressure over a streamline close to the combustion chamber wall (see Fig. 3.5). As explained in section 4.1, this type of data are not ideal for comparison with HyPro due to their two dimensional nature as opposed to averaged values used in HyPro. However the model used for the validation of the CFD solver *rhoFoam* (see section 3.4), which replicated the Lorrain's data in frozen conditions, can be employed to retrieve averaged data, thus helping to bridge the gap between the Lorrain CFD and the HyPro model.

Here an HyPro model equivalent to this reference CFD analysis has been constructed for comparison (see Fig. 4.5). The injection point has been placed downstream of the intake, and it is assumed that the pre-mixing of the fuel and air is completed entirely within the intake itself. This assumption is consistent with the procedure used by Lorrain to evaluate the fluid dynamic conditions at the inlet of the combustion chamber (see section 3.4 for details). The radical farming process is modelled by adding a pre-combustion module before the combustion chamber itself. The pre-combustion process is modelled using the friction model described in section 2.3.3 with $C_f = 0.0035$. The pre-combustion module takes into account the fact that, in a scramjet that employs the radical farming strategy, combustion does not occur at the inlet of the combustion chamber, but only downstream of the first shock reflection on the walls of the duct (see [70, 11, 71]). The combustion module assumes a combustion efficiency of 40% with wall friction (see 2.9) assuming again that $C_f = 0.0035$. The friction coefficient value has been calculated averaging the wall friction data of the *rhoFoam* validation model (see Fig. 4.6).

A comparison between HyPro predictions and the CFD calculations performed by Lorrain [11] is illustrated in Fig. 4.7 in both reacting and frozen chemistry hypothesis. The path line, over which the Lorrain's results are calculated, is almost straight and horizontal, so in Fig. 4.7 the difference between curvilinear abscissa and axial coordinate have been neglected. The HyPro results can be seen to be in good agreement with those produced by the CFD analysis, once it is realised that the CFD results are local in a two dimensional domain, while the HyPro data, coming as it does

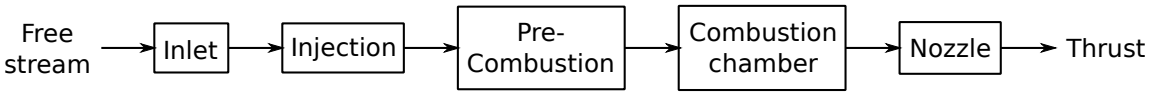


Figure 4.5: HyPro model structure used to reconstruct Lorrain's scramjet testbed.

from a quasi one-dimensional model, represents better the averaged flow quantities at each axial station along the engine duct. Indeed, a better comparison between the two models results if the pressure calculated in Lorrain's CFD [11] is averaged across several engine stations, as showed in Fig. 4.7 for the frozen chemistry case, where detailed CFD data were available from the *rhoFoam* validation test case.

The agreement between HyPro and the CFD data in the frozen case is very good, with only a small error (8% relative error on the pressure at the combustion chamber outlet) justifiable with the arguments presented in section 4.2.1. For the reacting case a direct comparison is difficult due to the absence of averaged data, but some qualitative conclusions can be drawn extrapolating the difference between averaged and path line values from the frozen test case to the reacting one. In the frozen case the average value at the end of the combustion chamber is slightly lower than the pressure peak of the corresponding path line data; similarly in reacting conditions HyPro pressure at the end of combustion chamber is also a bit lower than the peak of the CFD. The same comparison can be done at the end of the nozzle, where the average value is almost equal to the path line value facilitating the whole operation. The fact that in reacting conditions the HyPro pressure at the end of the nozzle matches the CFD pressure, strengthens the validity of the assumptions done in the combustion model.

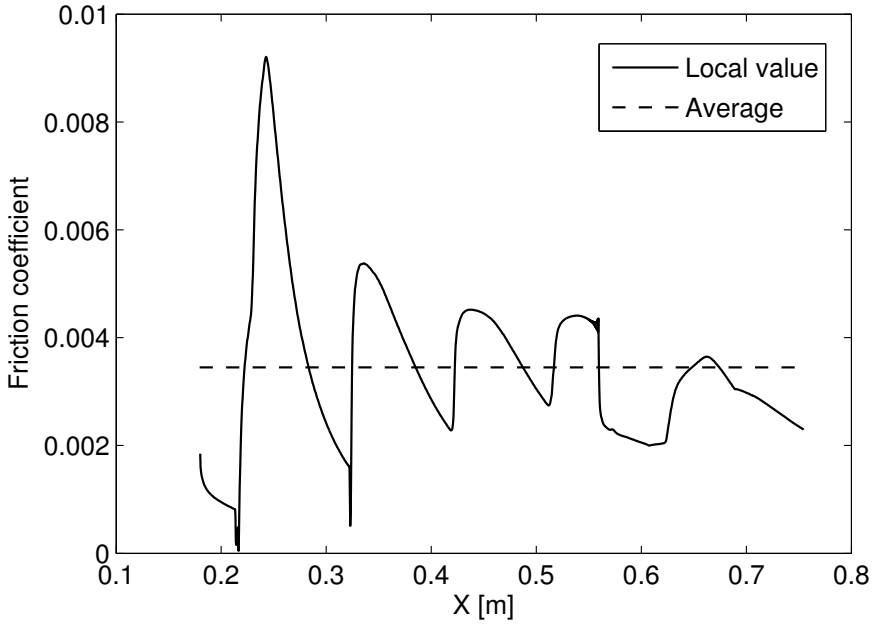


Figure 4.6: Friction coefficient resulting from the CFD reconstruction of the Lorrain frozen chemistry test case (see section 3.4).

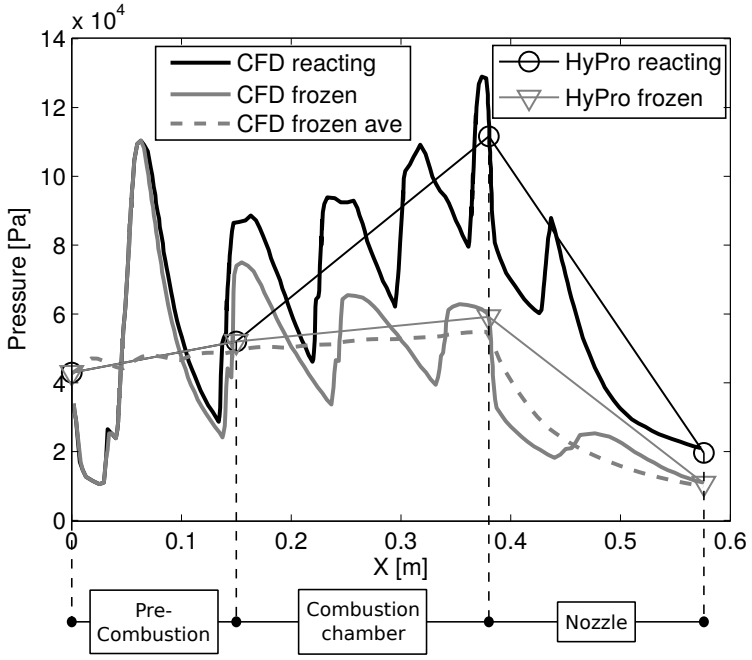


Figure 4.7: Comparison between the axial pressure profile predicted by HyPro with combustion, and results from Lorrain's CFD calculations[11].

4.4 Hyperion test case

A verification case is presented in order to compare the predictions of HyPro against those of a similar software package called SCCREAM [12, 9, 55]. This software was used to evaluate the performance of the ejector ramjet engine that was proposed for Hyperion, a launch vehicle conceived by the Aerospace Systems Design Laboratory at Georgia Tech [12]. This propulsive system was designed to operate in several operational modes: at low speed, from stationary to low supersonic (i.e. Mach 3), it operates in ejector ramjet mode; then, at higher speed, it can switch off the rocket primary and operates as a pure ramjet; eventually at even higher speed or at high altitude when the ramjet is not effective, the rocket primary can be turned on again to operate in pure rocket mode. The purpose of this test case is to verify the performances of HyPro against a typical RBCC configuration, thus to show how the flexibility of the modular approach can be employed for combined cycle engines.

4.4.1 Ejector ramjet mode

The HyPro model representing the Hyperion propulsion system (and reproducing the work of Olds [12]) in ejector ramjet mode is depicted in Fig. 4.8.

The intake is modelled as fully adaptable, i.e. with the possibility that both throat and inlet have a section area that can be varied with the operational condition of the engine (see section 2.4.1). An isentropic diffuser is added after the intake module in order to be able to model a pinch point area (at the end of the intake module) that is smaller than the mixer section area. The injection of the rocket plume is assumed to occur along the axis of the engine (see 2.6). The rocket is not modelled in any detail, however: its nozzle exit conditions are fed directly into the input

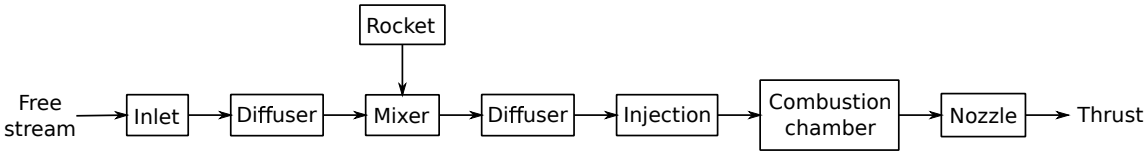


Figure 4.8: HyPro model of the Hyperion ejector ramjet engine.

node N_3 of the mixer module (see first mixer model in section 2.6). After the flow passes through a second isentropic diffuser, injection takes place assuming a constant φ followed by combustion with constant efficiency (see 2.7 and 2.9). The equivalence ratio φ within the injection module can be reduced if choking is detected downstream of the injection point (see 2.11). A full description of the distribution of nodal areas along the axis of the engine is given in Table 4.4, bearing in mind that both the inlet throat and intake as well as the nozzle exit have a variable area.

In order to better reproduce the results of SCCREAM the efficiency of the combustion process, the mixer efficiency and the diffuser total pressure drop have been tuned in order to correlate with additional data provided by Olds and al. in [12] (see Tab. 4.3 and Tab. 4.2).

Mixer efficiency	Diffuser tot. pressure drop	Combustion efficiency
0.9	0.92	0.8

Table 4.2: Correlated values of the efficiencies in the HyPro model of Hyperion

		Intake	Pinch Point	Chamber End	Nozzle Exit
Area	SCCREAM	-	0.77	2.09	1.61
m^2	HyPro	0.84	0.77	2.09	1.62
Mach	SCCREAM	0.5	0.57	-	1.51
	HyPro	0.5	0.57	0.41	1.51
Velocity	SCCREAM	170	194	-	1612
$\frac{m}{s}$	HyPro	170	194	488	1621
Total pressure	SCCREAM	122078	122078	350371	350371
Pa	HyPro	120172	120172	347500	348667
Total temperature	SCCREAM	302.6	302.6	3080.4	3080.4
K	HyPro	302.5	302.5	3070.8	3081.9

Table 4.3: Detailed output of SCCREAM and correlated HyPro at sea level and Mach 0.5

A comparison between HyPro and SCCREAM predictions[12] of thrust and specific impulse for the Hyperion engine when operating in ejector mode over a range of altitudes and Mach numbers is shown in Figs. 4.9 and 4.10. Despite, or indeed because of, the tuning of the several efficiencies in the HyPro model, the thrust predicted by HyPro matches reasonably well with that predicted

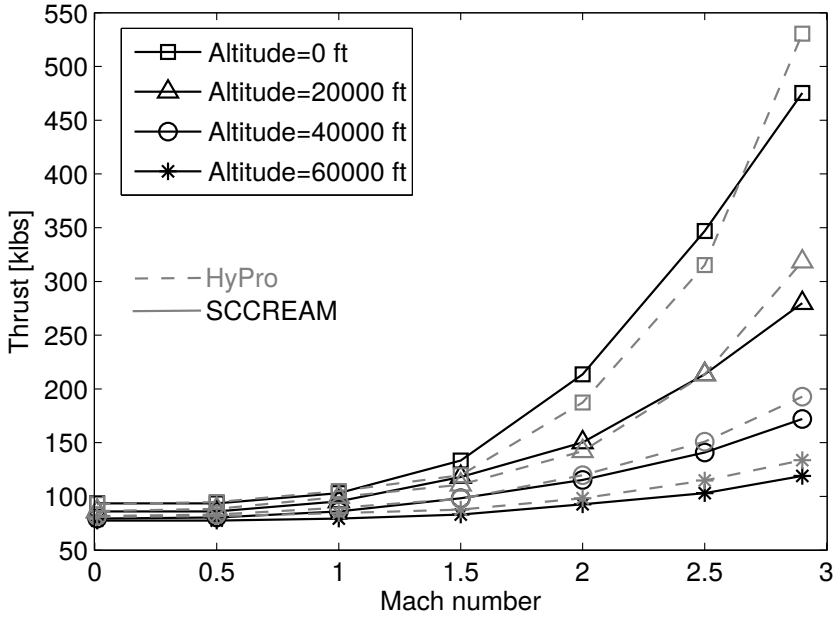


Figure 4.9: Comparison between HyPro and SCCREAM predictions of the thrust of the Hyperion engine in ejector mode[12].

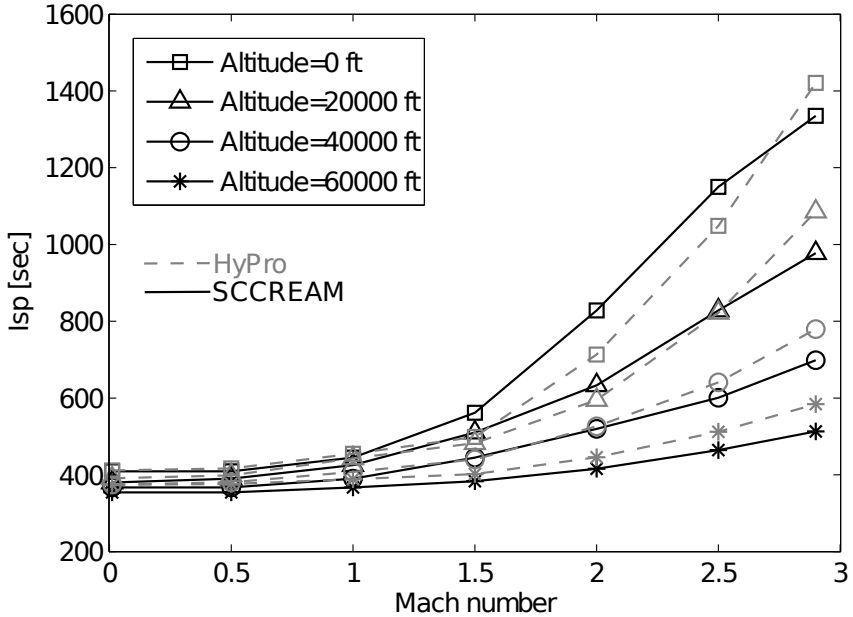


Figure 4.10: Comparison between HyPro and SCCREAM predictions of the specific impulse of the Hyperion engine in ejector mode[12].

	Area $[m^2]$	
Node name	Min	Max
Pre-Intake	0	∞
Intake	0	2.5084
Throat	0	0.6271
Pinch Point	0.7655	0.7655
Primary	1.0452	1.0452
Mixer End	1.0452	1.0452
Rocket Outlet	0.2796	0.2796
End Diffuser	2.0903	2.0903
Injection	2.0903	2.0903
End Chamber	2.0903	2.0903
Nozzle	0	8.8258

Table 4.4: Nodal areas within the HyPro model of the Hyperion engine.

by SCCREAM, at least in trend. Comparison of the predictions by the two models of the specific impulse of the Hyperion engine leads to similar conclusions. These differences however, can be considered lower than the uncertainty applicable to SCCREAM, as demonstrated later on this section.

Further insight into the predictive performance of the HyPro model can be gleaned from the plots shown in Fig. 4.11, Fig. 4.12 and Fig. 4.13, where the results for operation of the Hyperion engine at Mach 0 and at Mach 2.5 are contrasted both at sea level and at an altitude of about 20 km. The intake is predicted to be fully opened at Mach 0, although, since the capture area tends to infinity the intake will not have adapted fully to the external flow under these conditions. On the other hand a much smaller intake area is sufficient at Mach 2.5 to guarantee that the inlet is adapted to the external flow. As shown in Fig. 4.12, the working pressure within the engine changes dramatically in passing from Mach 0 to Mach 2.5 at sea level. It is worth noting however that the pressure at the end of the mixer is the same for all operating conditions with the exception of Mach 2.5 at sea level; this is because the secondary flow is always much smaller than the primary flow (i.e. the flow in the rocket exhaust plume), with the exception of at this particular flight condition

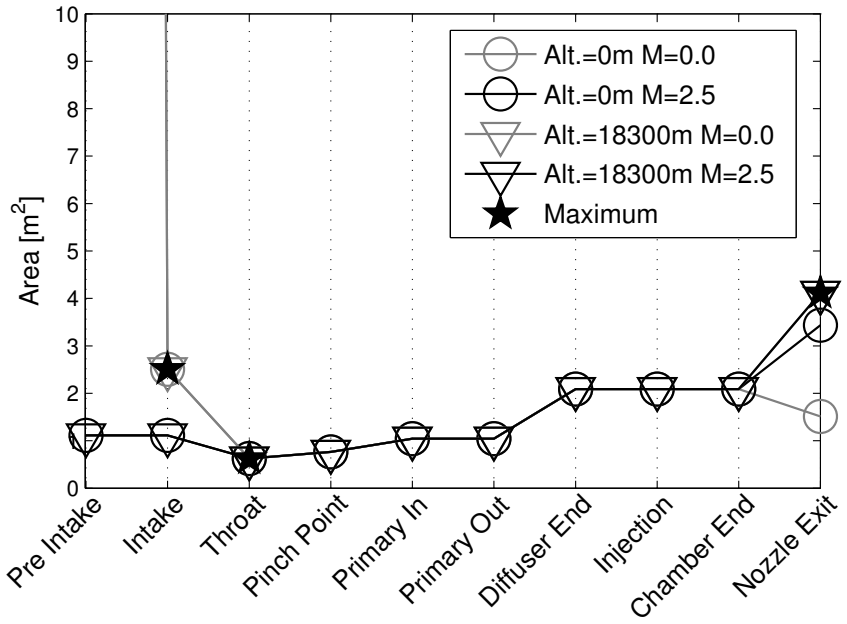


Figure 4.11: HyPro predictions of the variation of section area along the length of the Hyperion engine when operated in ejector mode.

where the dynamic pressure of the free stream is much greater. Finally, Fig. 4.13 illustrates the conditions under which the flow becomes choked within the engine. In interpreting this diagram it is important to realise that the flow is considered to be choked within the model if the Mach number approaches 0.9 rather than unity – this choice gives stability to the code and accounts for viscous effects that in the real system would reduce the available area for the flow under choked conditions.

4.4.2 Pure ramjet mode

In order to demonstrate how HyPro can be used for modelling of multi mode engines, the performances of Hyperion in pure ramjet mode has also been analysed. As for the ejector mode, the intake of the engine is modelled as fully adaptable. For the ramjet mode the maximum throat area is increased[12] compared to when the engine is in ejector mode. The Mixer module is substituted by a constant area duct with imposed total pressure drop, in order to model the presence of the

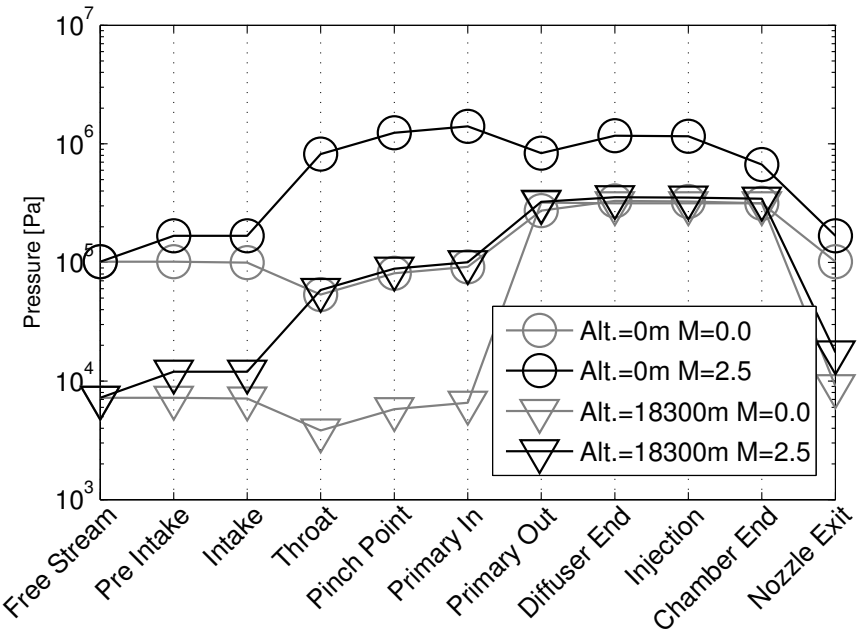


Figure 4.12: HyPro predictions of the variation of static pressure along the length of the Hyperion engine when operated in ejector mode.

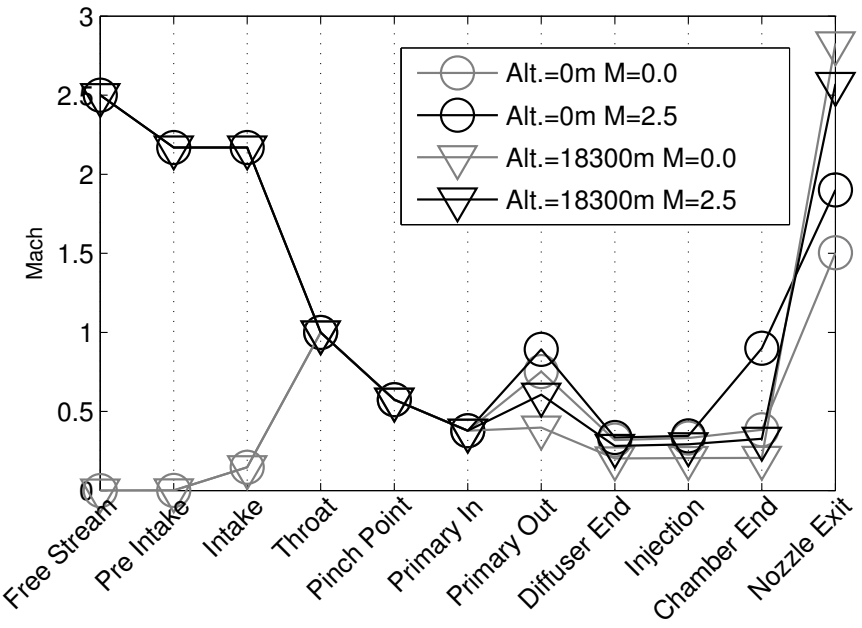


Figure 4.13: HyPro predictions of the variation of Mach number along the length of the Hyperion engine when operated in ejector mode.

primary, which in this case is only an obstacle. All the remaining modules are then modelled as in the Ejector ramjet mode. The value of this additional parameter, the total pressure drop across the mixer, has been set to 27.5% in order to correlate with detailed data along the axis of the engine[12] (see Tab. 4.5).

		Intake	Pinch Point	Chamber End	Nozzle Exit
Area	SCCREAM	2.51	0.77	2.09	4.12
m^2	HyPro	2.51	0.77	2.09	4.12
Mach	SCCREAM	3.02	0.63	0.90	2.08
	HyPro	3.02	0.63	0.90	2.10
Velocity	SCCREAM	999	330	758	1484
$\frac{m}{s}$	HyPro	996	329	754	1468
Total pressure	SCCREAM	692466	539351	312438	312438
Pa	HyPro	694173	559365	315272	315272
Total temperature	SCCREAM	747.2	747.2	1914.4	1914.4
K	HyPro	745.6	745.6	1903.6	1903.6

Table 4.5: Detailed output of SCCREAM[12] and correlated HyPro at an altitude of 56000 feet and Mach 3.5

In Fig. 4.14 and Fig. 4.15 the performances of the Hyperion engine in ramjet mode in terms of respectively thrust coefficient and the specific impulse are plotted for both HyPro and SCCREAM models. Here the thrust coefficient defined in Eq. 4.1 is preferred to the dimensional thrust because in ramjet mode it allows easy comparison between different engines.

$$Ct = \frac{F}{\frac{1}{2}\rho_\infty U_\infty^2 A_1} \tag{4.1}$$

In Eq. 4.1 A_1 is the maximum inlet area of the engine.

4.4.3 Rationale of the difference between HyPro and SCCREAM

The discrepancies between the predictions of the two models are thought to most likely result from the presence of a number of unresolved differences between the parameters, and their variation with

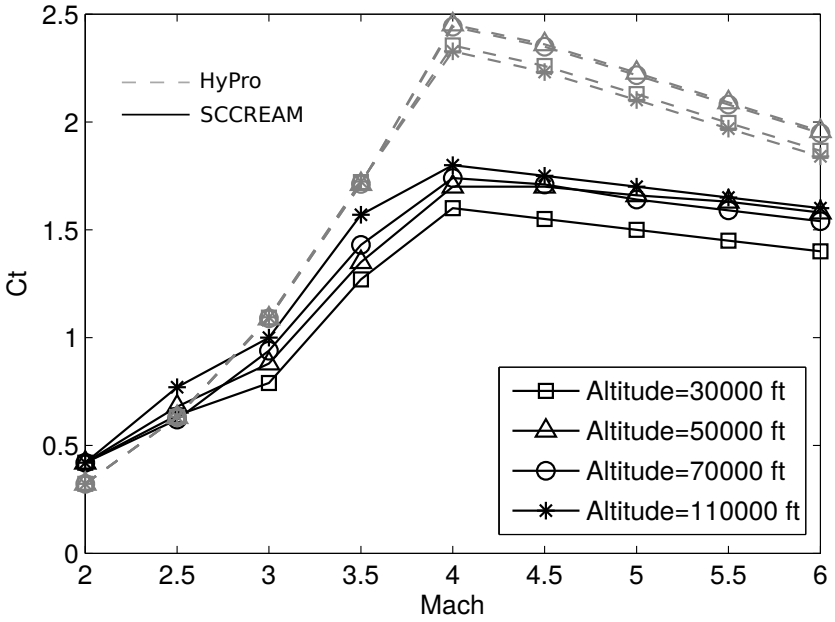


Figure 4.14: Comparison between HyPro and SCCREAM predictions of the thrust of the Hyperion engine in pure ramjet mode[12].

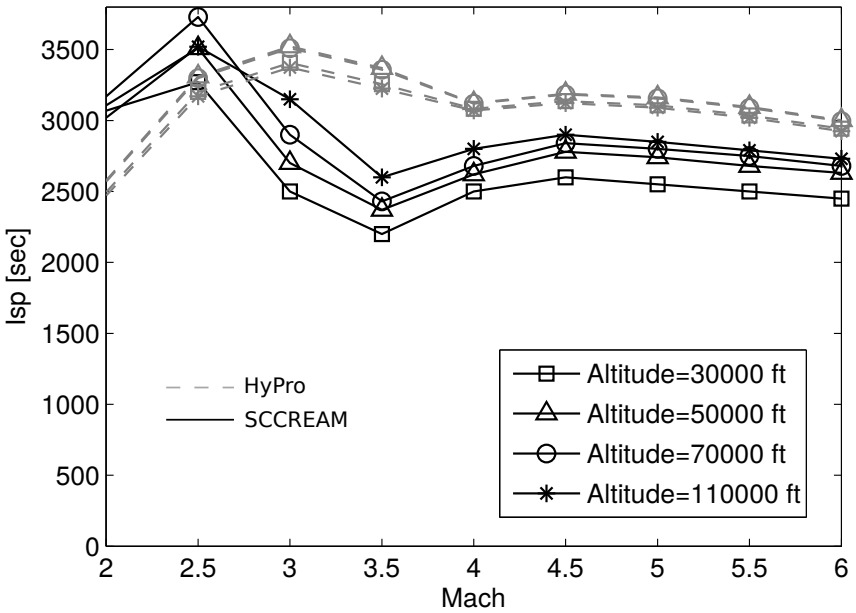


Figure 4.15: Comparison between HyPro and SCCREAM predictions of the specific impulse of the Hyperion engine in pure ramjet mode[12].

altitude, which is assumed by the two models. Given that full details of the SCCREAM calculation and the specifics of the model are not available in the open literature, the observed discrepancies in the predictions of the two models are unlikely to be resolved any time soon. However looking at the verification process provided for SCCREAM, it is possible to see how these observed differences compare with the uncertainty applicable to SCCREAM.

In Fig. 4.16 and Fig. 4.17 the verification provided by Olds[12] has been amended with the performances calculated by HyPro in the same conditions. In this verification the performance of Hyperion along a trajectory at fixed maximum dynamic pressure (see Fig. 4.18) is compared against other engine data available in literature. Two engines conceived by the Marquardt in Ejector RamJet (ERJ) and Ejector ScramJet (ESJ) configuration and one studied by the Astronautics Corporation for the U.S. Air Force in ESJ mode.[12]

As can be seen in Fig. 4.16 in ejector mode (the transition between ejector and ramjet mode has been set at Mach 2.7) the distance between the prediction of HyPro and SCCREAM is much smaller than the difference between SCCREAM and the reference data, thus suggesting that the errors observed in Fig. 4.10 are within the uncertainty interval of the SCCREAM model. The same thing cannot be said for the thrust because no such a verification is provided for SCCREAM, even though it is reasonable that the same considerations apply.

Regarding the ramjet mode, the differences between HyPro and SCCREAM observable in Fig. 4.16 and Fig. 4.17 are bigger, however also in this case they are at least comparable with the uncertainty of SCCREAM, or more precisely to the dispersion of the data.

Therefore it can be stated that HyPro performs reasonably well for this application, providing results comparable with SCCREAM and surely not less accurate than it. However this verification is mainly shown to demonstrate how HyPro can be applied to a RBCC test case and a proper validation against more reliable data will be attempted in the following sections of this paper.

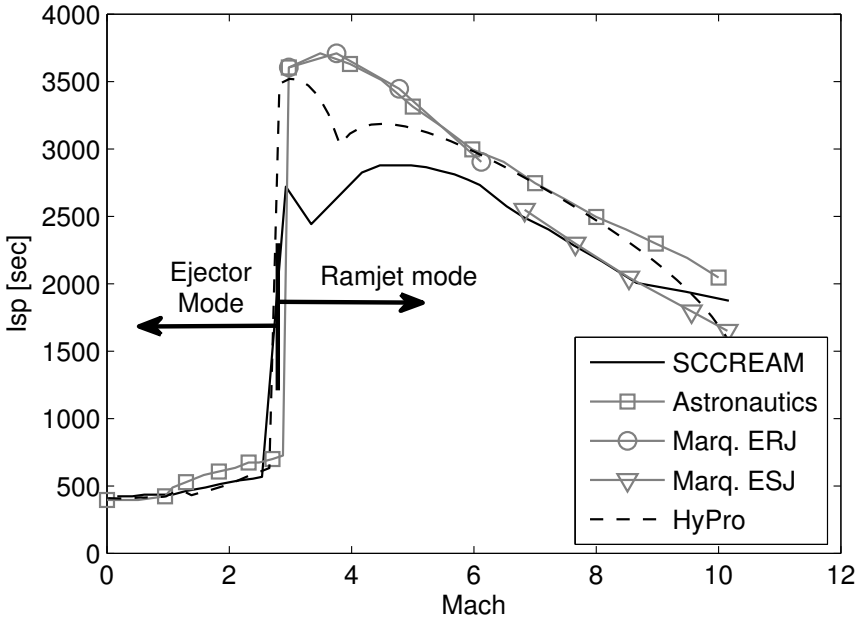


Figure 4.16: Comparison between specific impulse data of Hyperion, evaluated by SCCREAM and HyPro, and other datasets available in literature.[12].

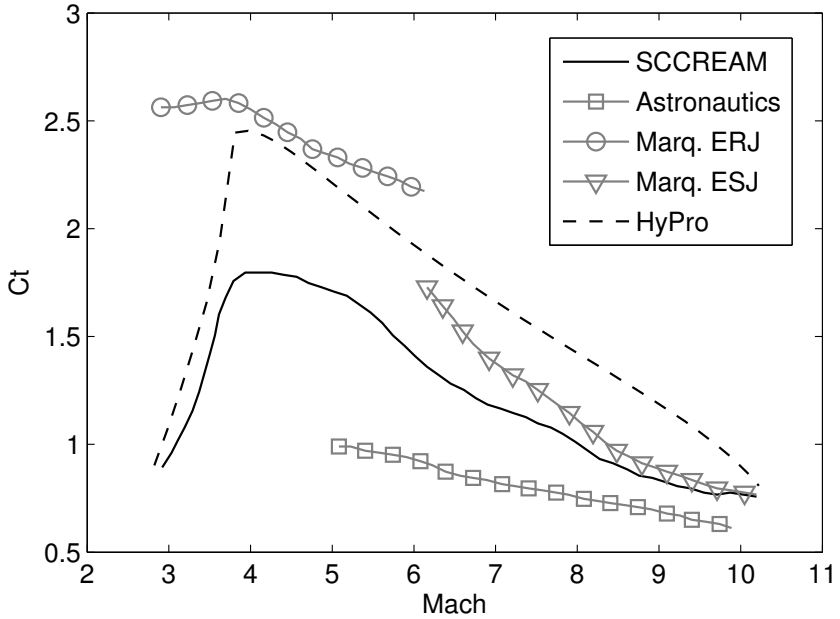


Figure 4.17: Comparison between thrust coefficient data of Hyperion, evaluated by SCCREAM and HyPro, and other datasets available in literature.[12].

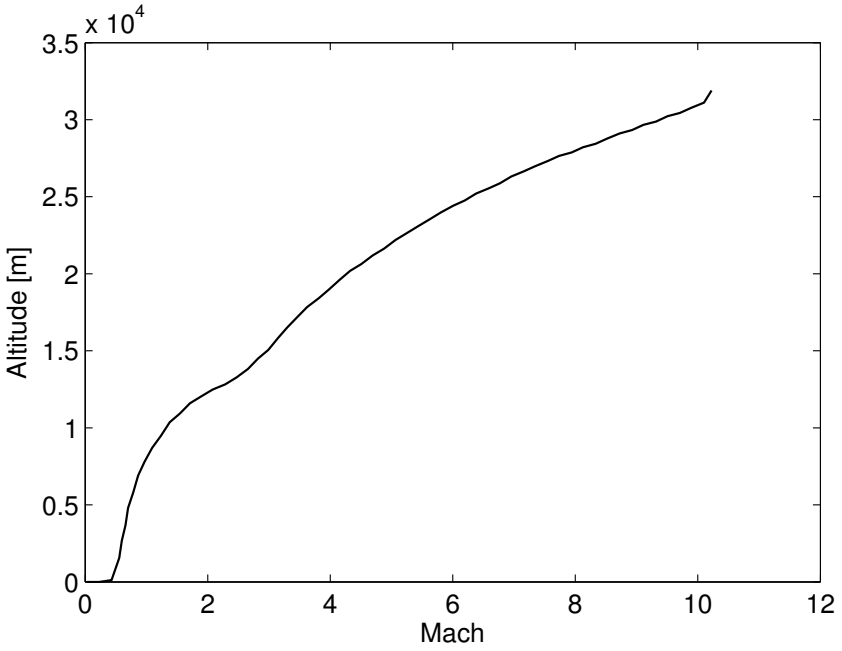


Figure 4.18: Hyperion ascent trajectory[12].

4.5 Marquart SERJ test case

The performance of ejector type engines seems to be heavily influenced by the efficiency of the ejector (see 4.4.1). In this section, the results of a CFD analysis of the internal flow through the SERJ engine that was proposed by the Marquardt Corporation in the 1960s (see [4, 5, 8] and section 1.3.1) are compared to the predictions of the HyPro model configured as shown in Fig. 4.19.

The CFD tool *rhoFoam* described in chapter 3 has been employed to model the mixer of the SERJ engine. The working gas is assumed to be ideal but is subject to variable specific heat depending on temperature, as specified using a pre-defined polynomial function[72]. The flow is considered to be fully turbulent, and the turbulence is modelled using the SST $k-\omega$ model with

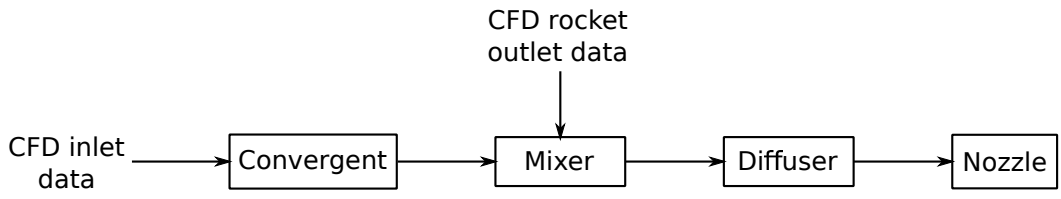


Figure 4.19: System diagram for the Marquardt SERJ mixer as replicated in HyPro.

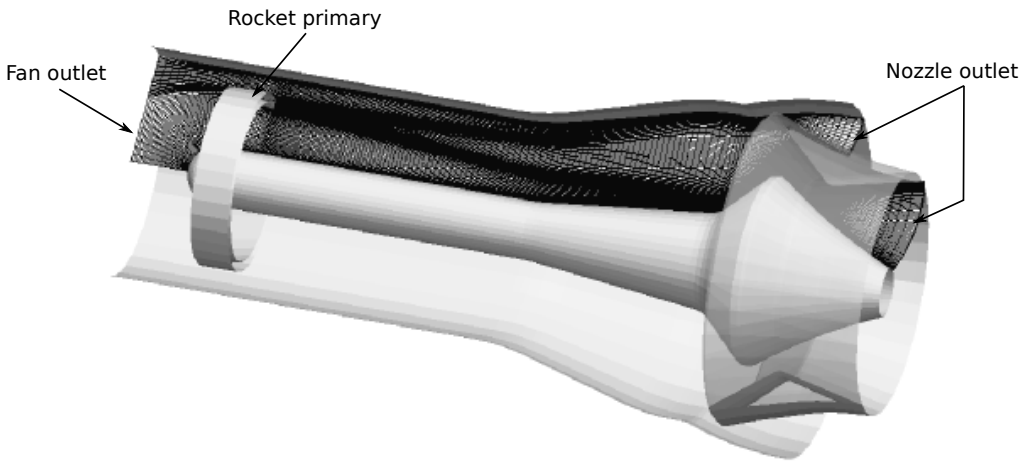


Figure 4.20: CFD model discretization of the Marquardt SERJ mixer.

standard wall functions.

The CFD analysis presented here evaluates only the performance of the ejector-mixer section of the Marquardt engine: many aspects of this system's rather complex configuration are thus neglected. As illustrated in Fig. 4.20, the CFD calculation domain starts downstream of the fan and extends up to the nozzle exit. The flow through the ejector-mixer is considered to be axially symmetric throughout. The flow medium is assumed to be air with constant composition, thus combustion and the mixing of gas species are not considered in the results presented here. The post-combustor is also not considered to be active in this case. On the other hand, the primary rocket is considered to be active, but under the simplification that its working fluid, and hence the composition of the plume that it generates, is air. It is important to note that this assumption does not compromise the final purpose of the analysis, since mixing within the system is affected primarily by the momentum of the flow in the plume and only secondarily by its composition.

The boundary conditions for the calculation are selected to reproduce the working conditions experienced by the engine at sea level and at Mach 0.75 [8]. The total mass flow rate and the stagnation temperature are assigned at the inlet (see Tab. 4.6), while the other quantities are

calculated by the solver. The pressure, temperature and velocity at the rocket outlet are directly specified once calculated from the known values of the total mass flow rate through the rocket nozzle and the rocket chamber pressure and temperature (see Table 4.6). Since the post-combustor is not included in this model, the throat area of the nozzle is reduced in order to allow the engine to operate at the same mass flow rate and pressure of the actual configuration. The nozzle plug is positioned appropriately along the axis of the engine in order to achieve the correct throat area (see Fig. 4.20). All walls are assumed to be adiabatic, and, as such, the results presented here do not account for the influence of any thermal losses upon the performance of the engine.

		Rocket Outlet
U	[m/s]	2623
p	[Pa]	40094
T	[K]	1062.9
$\dot{m}$	[kg/s]	177.8
p_0	[Pa]	$1.034 \cdot 10^7$
T_0	[K]	3675
		Inlet
$\dot{m}$	[kg/s]	743.4
T_0	[K]	620.2

Table 4.6: Marquardt SERJ boundary conditions[8]

The pressure and velocity contours inside the engine duct, calculated using *rhoFoam*, are shown in Figs. 4.22 and 4.23. Fig. 4.23 shows in addition the flow path lines within the engine. The flow within the duct is driven by the shear interaction between the rocket plume and the remainder of the fluid in the system, and this entrainment mechanism is responsible for many of the observed features of the flow. Indeed, a predominant feature of the flow within the duct is the large separation bubble near the end of the mixer: the presence of this phenomenon leads to a very low mixer efficiency. It appears that the adverse pressure gradient that is produced by the addition of the flow through the rocket ejector is primarily responsible for the separation of the boundary layer on the inner wall of the duct and thus the production of the separation bubble. This interpretation is supported by

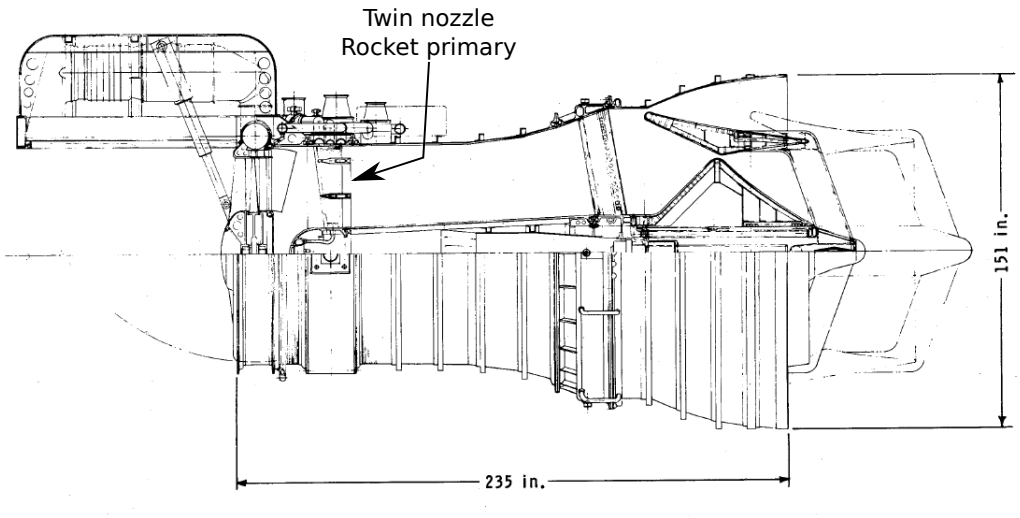


Figure 4.21: Marquardt SERJ class 3 configuration [4].

the observation that, because of the distance of the inner wall from the rocket plume, the kinetic energy of the flow there is very low – indeed, separation does not occur on the upper wall of the duct because it is much closer to the plume. It is interesting to speculate on how this behaviour, if indeed it would have been a characteristic of the full-scale engine, could have been avoided by re-design. Ostensibly a different design of the ejector with enhanced momentum transfer to the central part of the engine might have been sufficient to eliminate the separation bubble. There is evidence to suggest that a twin-nozzle ejector system was also considered by Marquardt during the evolution of their design (see Fig. 4.21). In present context, however, the occurrence of these interesting secondary flows serves admirably to exercise and explore the capabilities of HyPro – the presence of these complex adverse conditions yields an opportunity to demonstrate the flexibility and robustness of the software.

An interesting contour plot, perhaps allowing the origins of separation phenomena within the ducts of engine configurations such as this to be understood a little better, is shown in Fig. 4.24. In this figure, the temperature contours within the duct, together with the flow path lines of the solution obtained using the second-order WAS scheme (see section 3.3.2), are compared against the solution that is obtained using the Goudonov first order scheme (used only at the beginning of

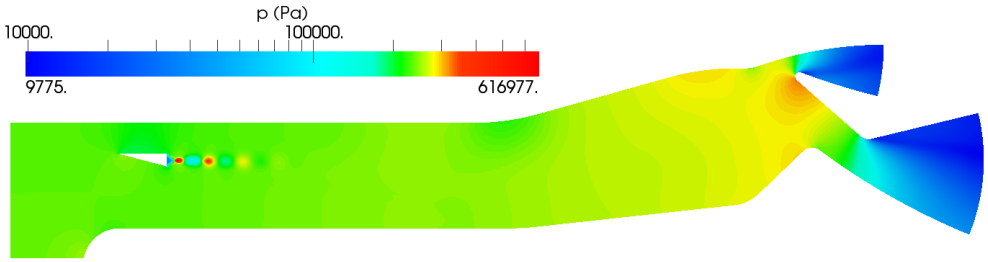


Figure 4.22: CFD results for the Marquardt SERJ mixer: static pressure contours.

the computation to facilitate the convergence). It appears that, because of the higher diffusivity of the first order scheme compared to the second order WAS flux scheme, more kinetic energy is transferred from the rocket jet to the boundary layer at the inner wall of the duct and the separation is thus prevented. In Figs. 4.25 and 4.26 the pressure and Mach number for both the first- and the second-order solutions are plotted along the engine axis (after averaging across each axial station of the duct). These figures show conclusively how the presence of the separation bubble acts to reduce the performance of the ejector.

As a final confirmation of the performance of the CFD approach, in particular regarding the insensitivity of its predictions to the discretization of the computational domain, results obtained using two different mesh densities (Mesh A with 21,232 cells, and Mesh B, with 84,928 cells obtained by subdividing the cells of Mesh A) are also compared in Figs. 4.25 and 4.26. Although small differences between the two calculations are present, these differences are much smaller than the differences between the solutions with and without the separation bubble. The level of mesh insensitivity demonstrated in these figures is entirely adequate in the present context, particularly given the magnitude of some of the other likely sources of error within the approximation to the engine geometry and the treatment of the composition of its internal flow.

In order to compare the aforementioned results against HyPro, the CFD boundary conditions are fed as direct inputs into the HyPro model. In the HyPro implementation of the SERJ mixer, the mixer module itself is preceded by an isentropic convergent duct to account for the change in

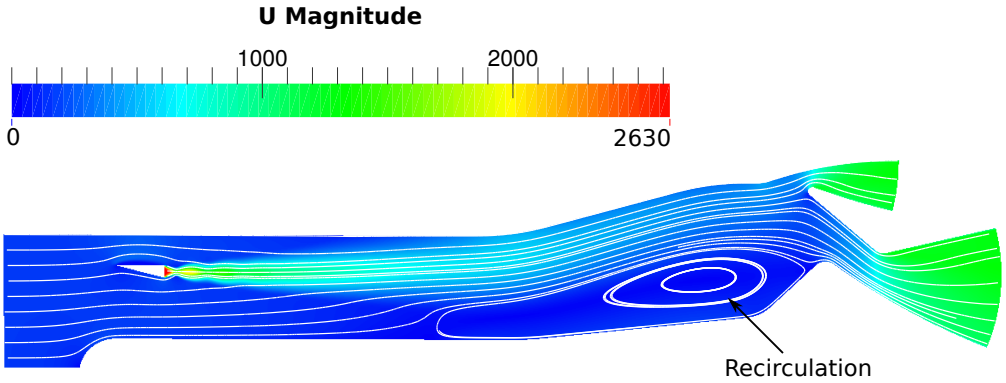


Figure 4.23: CFD results for the Marquardt SERJ mixer: velocity contours and flow path lines.

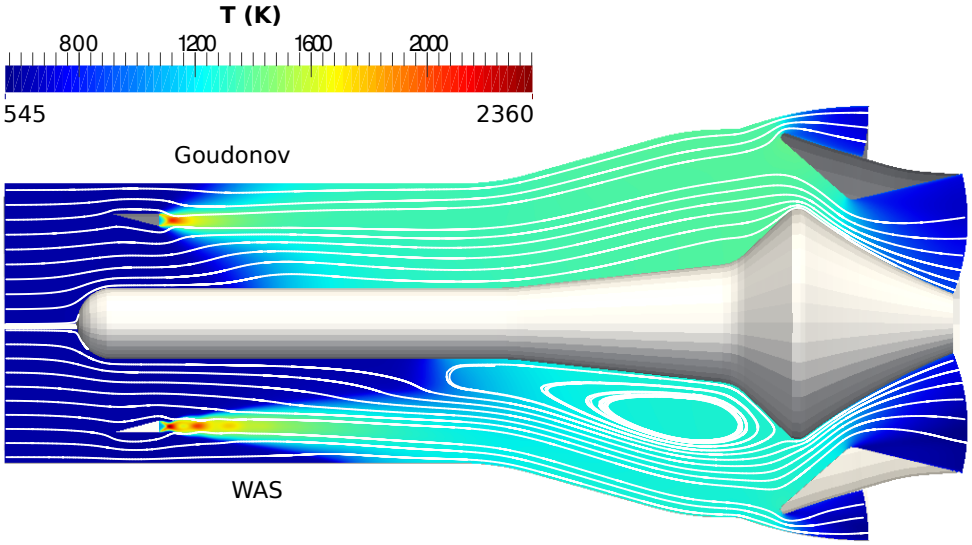


Figure 4.24: CFD results for the Marquardt SERJ mixer: static temperature contours and flow path lines. Top half: First order Godunov flux scheme. Bottom half: Second order WAS flux scheme.

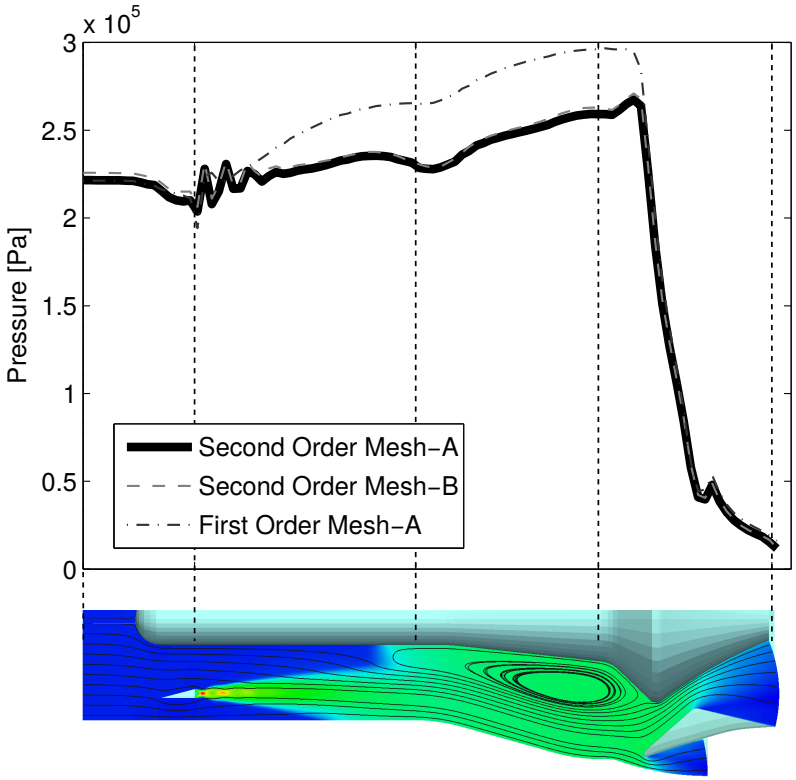


Figure 4.25: CFD results for the Marquardt SERJ mixer: section-averaged static pressure along the engine axis.

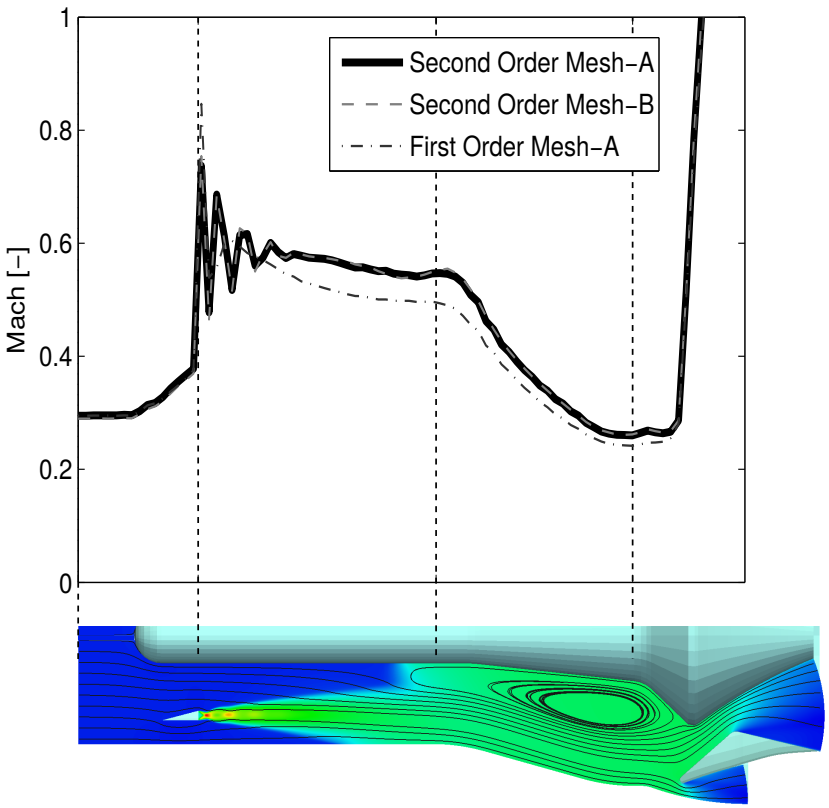


Figure 4.26: CFD results for the Marquardt SERJ mixer: section-averaged Mach number along the engine axis.

cross section area between the section after the fan and the primary section. Another isentropic duct downstream of the mixer module simulates the diffuser of the engine, and the flow through the nozzle is also considered to be isentropic. Thus, in the analysis presented here, only the mixer is considered to have an efficiency lower than unity. For consistency with the CFD analysis, the post-combustor is not modelled even though the capacity to do so exists within the HyPro toolbox.

Fig. 4.27 shows a comparison between the results of our CFD analysis and the predictions of the HyPro model. The CFD data are averaged at each section along the axis of the engine in order to yield results that are directly compatible with the one dimensional nature of the HyPro predictions. As can be seen, the CFD analysis and the HyPro predictions are in very close agreement all the way up to the mixer pinch point, while some small discrepancies are present within the mixer itself. In particular, the results of the first order CFD calculation are in very good agreement with the predictions of the HyPro model when no account is taken of any inefficiency within the duct (i.e. when the HyPro mixer efficiency is set equal to unity, see section 2.6); the HyPro results showing just a slight overestimation of performance due to the unrealistic assumption of a perfect mixer. The second order CFD analysis yields a much lower predicted performance for the engine – mainly, as postulated above, due to the strong separation that is predicted to occur within the mixer. Good correlation between the two models can be restored however if the mixer efficiency within the HyPro model is tuned to have a value of 70%, as shown in Fig. 4.27. By tuning this parameter alone it is possible to obtain very good correlation between the CFD analysis and the reduced-order HyPro model along the entire length of the flow path within the engine. This lends significant confidence to the robustness of the HyPro model to capture the performance of complex, hybrid propulsion systems even in the presence of extreme and non-optimal conditions, such as this large-scale separation, within the flow path of the engine. In this way high fidelity analysis, such as CFD, can be used to generate a very precise parsimonious model, that can be later used for system design and optimization purposes.

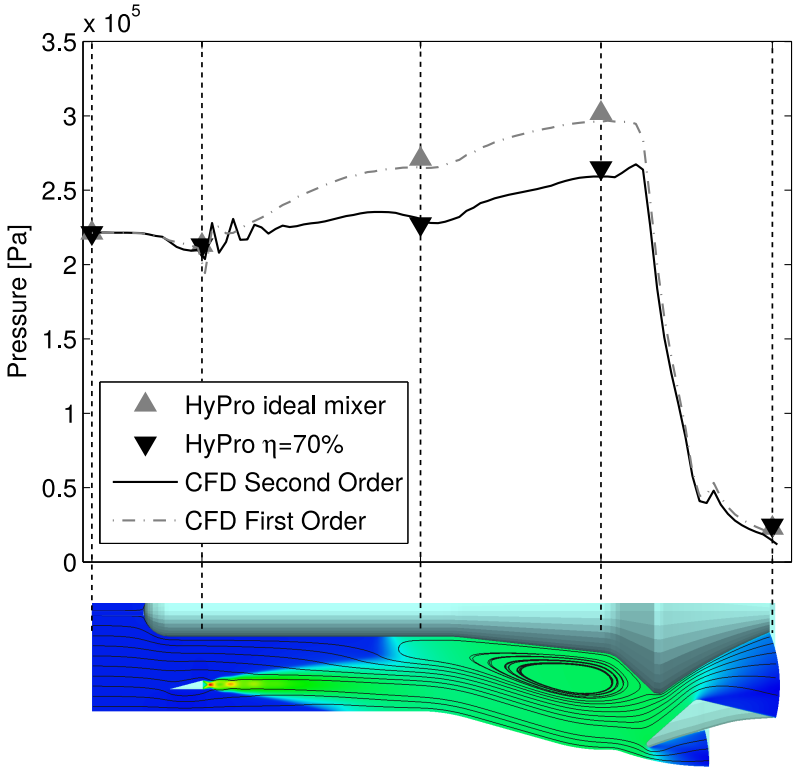


Figure 4.27: SERJ Marquardt Mixer, comparison CFD vs HyPro: section averaged pressure plot along engine axis.

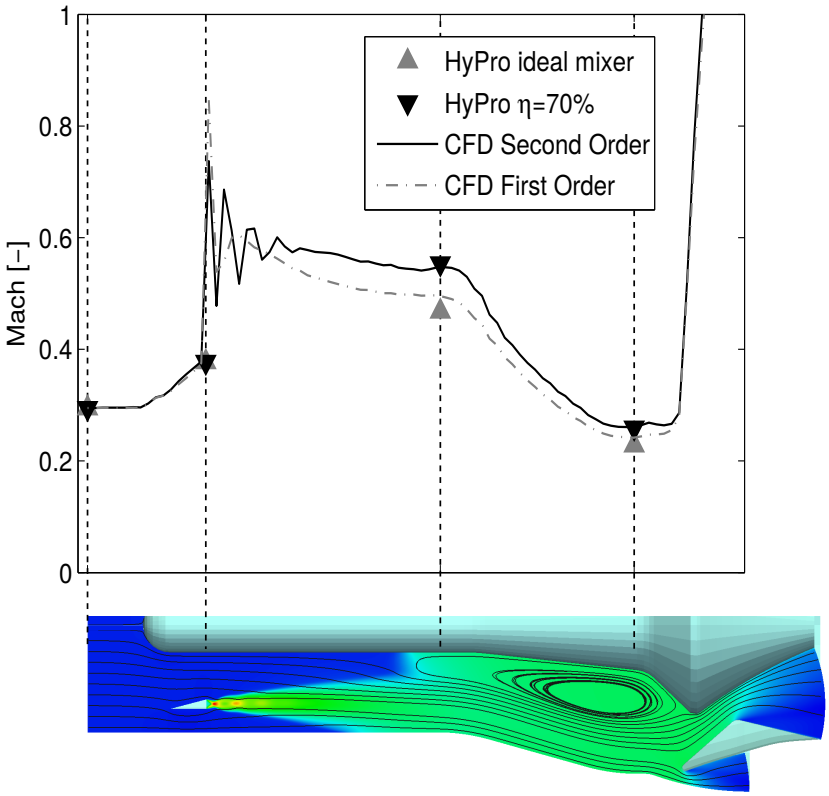


Figure 4.28: SERJ Marquardt Mixer, comparison CFD vs HyPro: section averaged Mach number plot along engine axis.

4.6 Stataltex ramjet missile

In this section HyPro will be validated against flight data available in literature for a ramjet configuration (see [13]). In the early 60s the ONERA conducted a series of flight tests within a program called Stataltex (STX). The tested ramjet was accelerated up to $1000\,\frac{m}{s}$ by a Stromboli booster. After separation from the booster the STX performed an unguided self-powered flight up to almost Mach 5. In this campaign ten launches were fired, but only three of them were completely successful. In this paper these three flight tests have been replicated with a dedicated HyPro model and the results in terms of thrust coefficient and specific impulse have been compared against the data available in [13].

In Fig. 4.29 the geometry of the STX engine is depicted together with the correspondent model build in HyPro. Both inlet and nozzle have fixed geometry, with the inlet designed to be adapted at Mach 3.5 and to work always in the supercritical regime (see [13]). The inlet forebody modelling is performed using the coefficient of critical flow (see section 2.4.2) available in [13] as a function of free stream Mach number (see Fig. 4.30). The diffuser is assumed isentropic while all the engine internal friction is concentrated in the injection module. This choice was made since in [13] the internal friction is defined by means of a single drag coefficient based on the dynamic pressure and area of node 2 (i.e. the input node of the injection module, see Fig. 4.29). This drag coefficient is defined as in the following and it is reported to be almost four:

$$C_g = \frac{X_g}{\frac{1}{2}\rho_2 U_2^2 A_2} \simeq 4 \tag{4.2}$$

The coefficient defined in Eq. 4.2 is then translated into friction coefficient C_f assuming a unitary length of the injection module for convenience. The injection is performed always at stoichiometric composition unless the maximum allowable fuel mass flow rate is reached (see Tab. 4.7). All the following modules are then assumed to have no friction again, but the nozzle friction has been accounted for using an efficiency coefficient equal to 0.96 that is multiplied by the nozzle exit

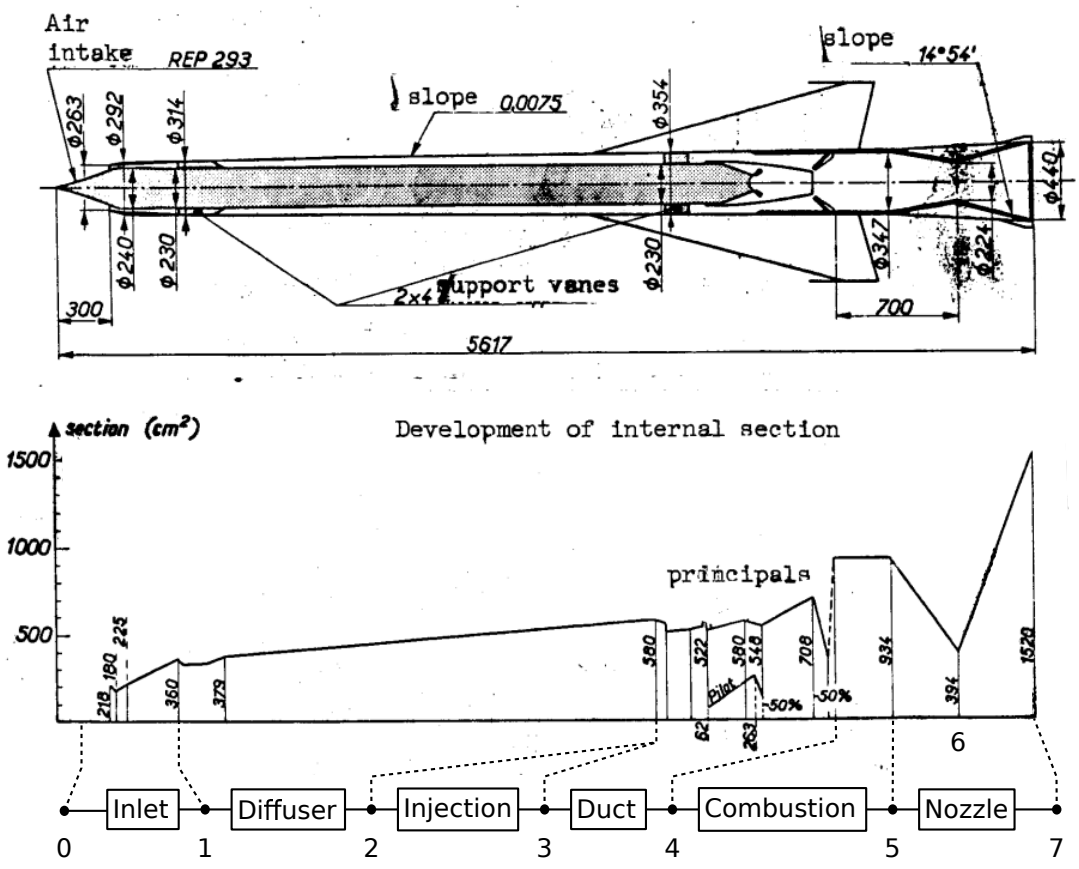


Figure 4.29: STX geometry and correspondent HyPro model. [13]

momentum flow, in coherence with [13]. Regarding the combustion chamber, the reaction has been modelled as follows:

$$4\,C_{12}H_{23} + 71\,O_2 \rightarrow 48\,CO_2 + 46\,H_2O \tag{4.3}$$

where the average formula of JET-A fuel has been considered (see [72]). As explained in section 2.9, the reaction is considered complete, but an efficiency is later applied to the heat of reaction. The efficiency is calculated as a function of the mixture composition φ as stated in [13] and plotted in Fig. 4.31.

The three firings considered in this validation were performed by very similar engines, with only few differences listed in Tab. 4.7 and accounted for in the HyPro model. As can be observed the

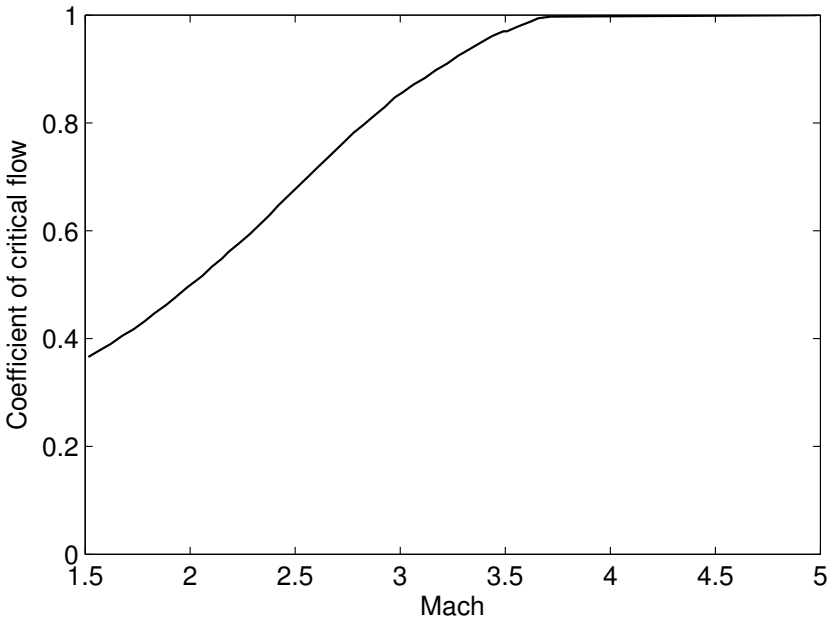


Figure 4.30: Coefficient of critical flow of the STX inlet. [13]

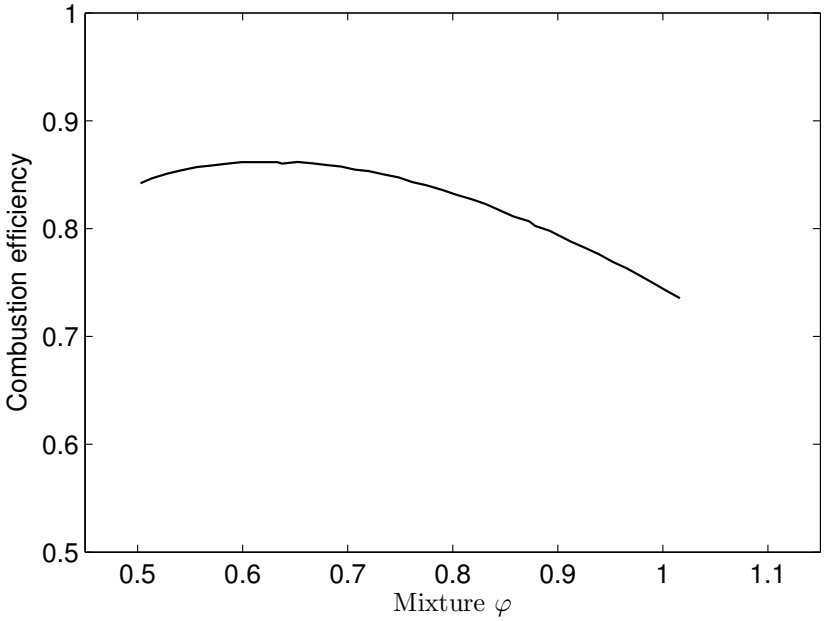


Figure 4.31: Combustion efficiency for STX engine. [13]

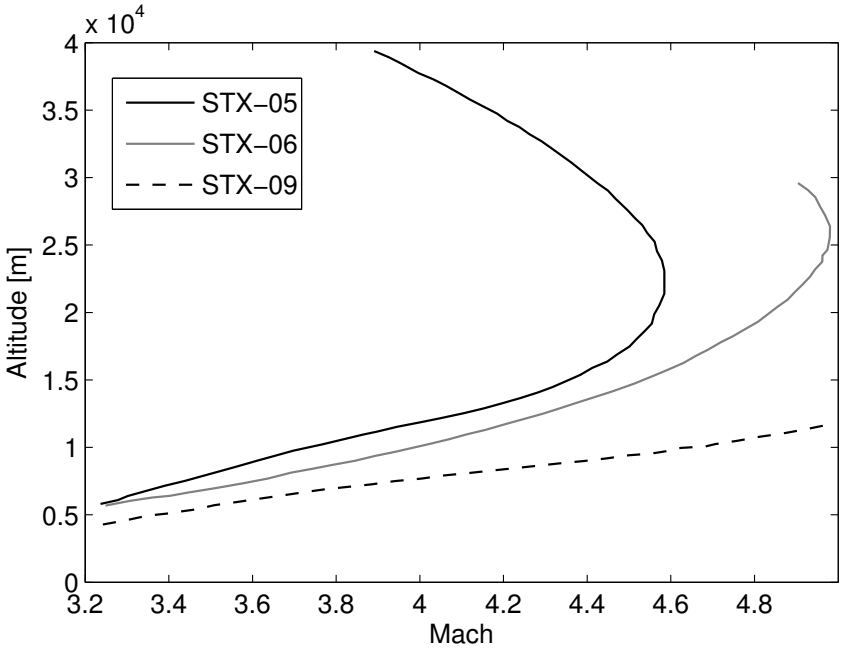


Figure 4.32: Trajectories of STX firings. [13]

only differences between engines were the maximum allowable fuel mass flow rate and the nozzle design.

STX #	Max $\dot{m}_f$ $\frac{kg}{s}$	φ	A_6 m^2	A_7 m^2
5	1.10	1	415	1308
6	1.20	1	394	1576
9	1.30	1	394	1576

Table 4.7: Characteristics of STX missiles. [13]

The trajectories of the firings, in terms of Mach number versus altitude are plotted in Fig. 4.32, in this work the full free stream conditions have then been calculated using the standard atmosphere model, since in [13] the atmosphere data measured at the time of firing are not reported.

In Fig. 4.33 and Fig. 4.34 respectively the thrust coefficent and the specific impulse of both the HyPro model and the flight measurements are plotted. Then in Fig. 4.35 and Fig. 4.36 the relative errors of the HyPro model calculated comparing against the flight measurements are plotted versus

the flight Mach number. As can be observed, the relative error on the prediction of specific impulse is always lower than 15%, while for the thrust the error increases up to 30% if the two outsiders at 40 and 90% are not considered. This behaviour can be explained with the uncertainty linked with the atmosphere, that can varies sensitively with respect to the standard one. Some evidence for this can been given by Fig. 4.37 where the thrust relative error is plotted against the altitude. As can be observed, the high errors are all concentrated at low altitude (i.e. below 11km) and at high altitude (i.e. above 22km), while in the central zone the error stays always below 10%. It is also evident that the two outsider points at 40 and 90% are both at very high altitude, where, as also stated in [13], the thrust is low and the measurement errors have a great impact. Fig. 4.37 could indeed be explained with the hypothesis that the atmosphere experienced during the STX campaign was close to the standard atmosphere in the layer between 11 and 21km (interestingly enough where the standard atmosphere foresees constant temperature) while differing substantially from it above and below, thus causing a drop in the HyPro predictions error just in correspondence of that layer. This proof however, is not confirmed due to the lack of further data. In any case the uncertainty here outlined is acceptable given the field of application for which the HyPro is intended.

Regarding the general behaviour of specific impulse with Mach number (see Fig. 4.34), the sharp corner visible for run STX-05 and STX-06 is explained by the fact that the fuel injection moves from being limited by the maximum mass flow rate (see Tab. 4.7) to being limited by the stoichiometric composition (i.e. $\varphi = 1$). The former occurs in the first part of the trajectory increasing the sensitivity of performances upon the atmospheric properties. The specific thrust and the specific impulse of a ramjet are indeed strongly affected by the φ, that, in case of limited fuel flow, depends upon the mass flow rate passing through the inlet, and thus on the free stream conditions. This observation could also explain why the thrust error increases so much at low altitudes, especially observing in Fig. 4.38 that the fuel mass flow rate is always limited by the maximum value below 11km.

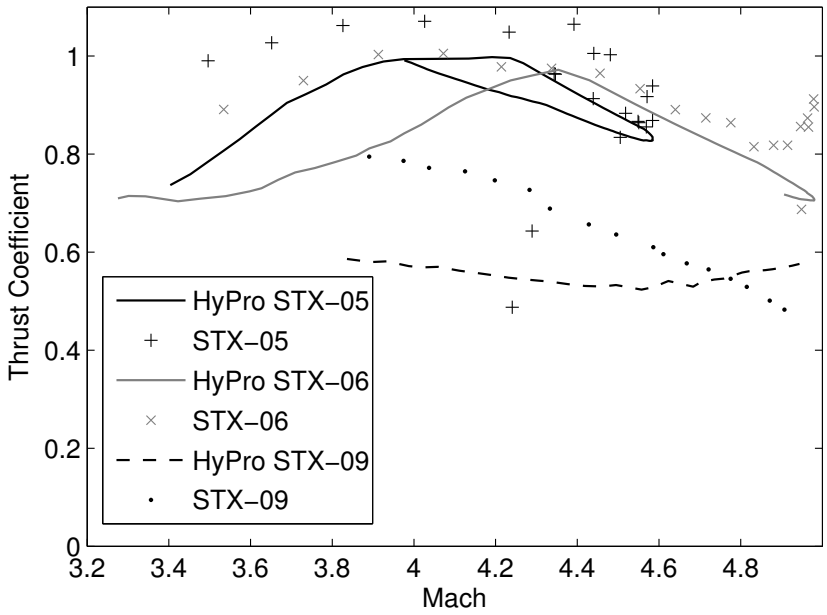


Figure 4.33: Specific thrust comparison between STX flight data and HyPro model. [13]

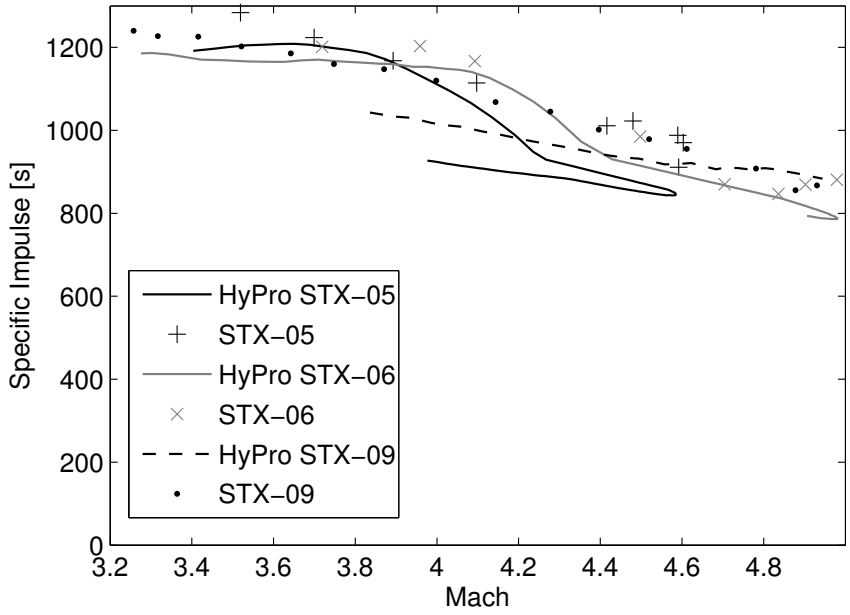


Figure 4.34: Specific impulse comparison between STX flight data and HyPro model. [13]

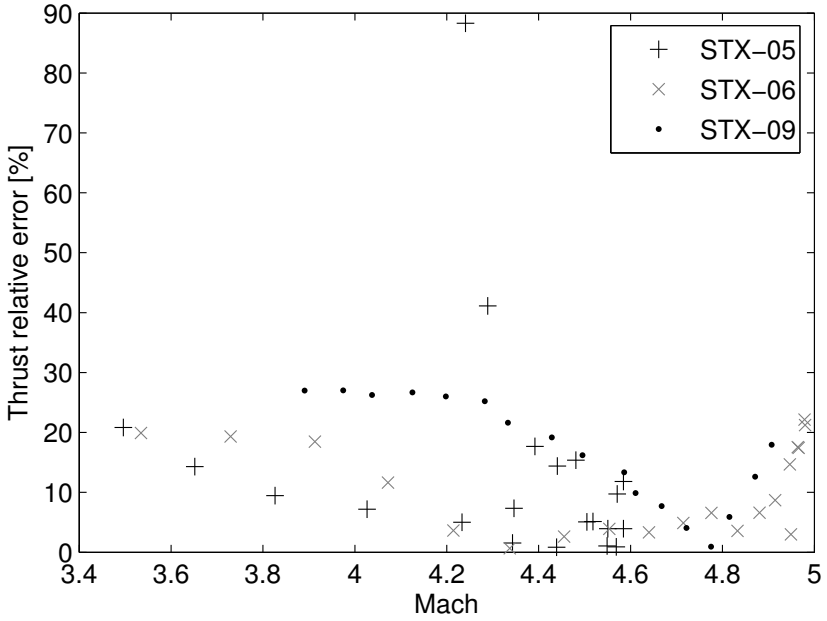


Figure 4.35: Thrust relative error of HyPro model.

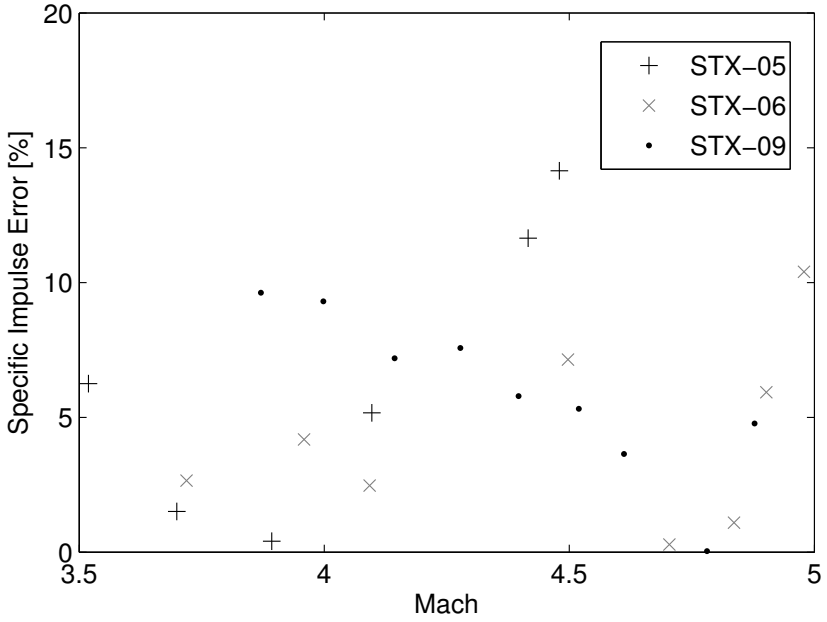


Figure 4.36: Specific impulse relative error of HyPro model.

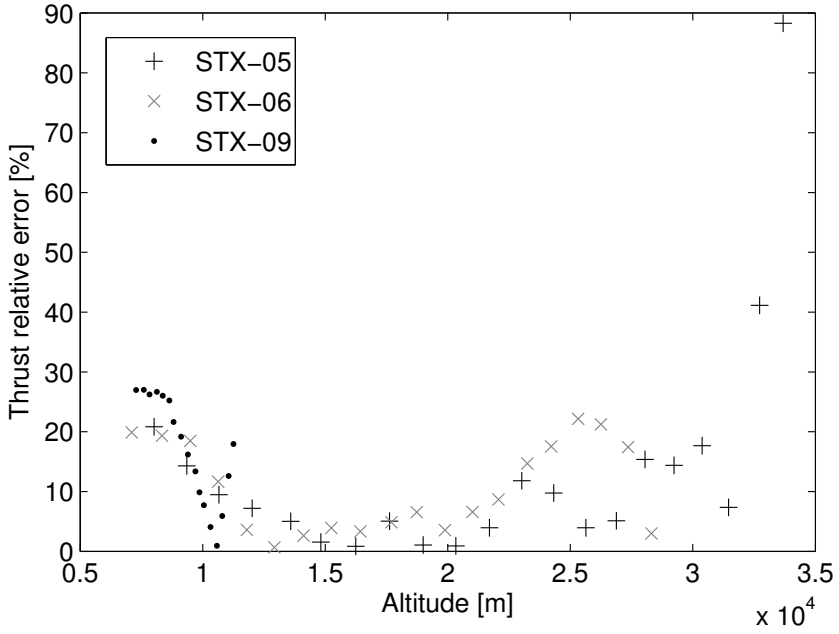


Figure 4.37: Thrust relative error of HyPro model plotted vs altitude.

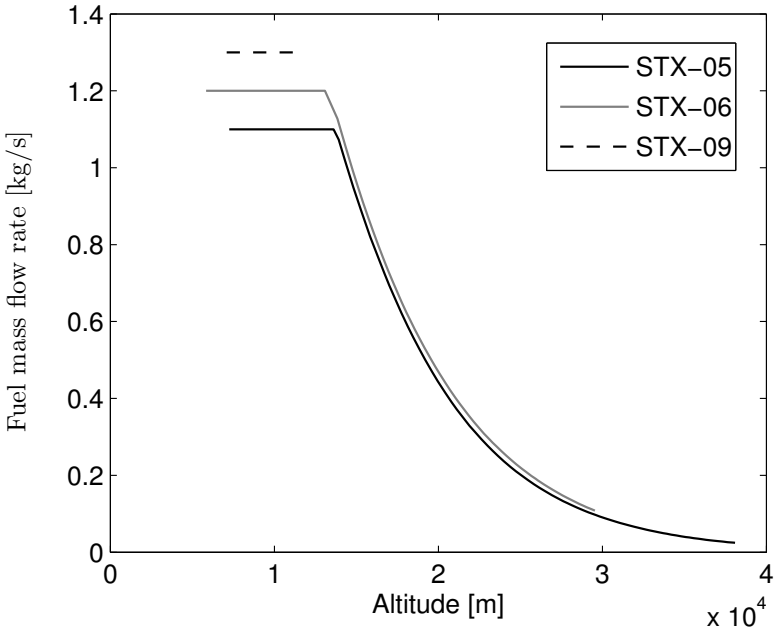


Figure 4.38: STX fuel mass flow rate predicted by HyPro plotted vs altitude

Chapter 5

Engine Configurational Optimization

5.1 Introduction

Design optimization is becoming a common practice in engineering, mainly thanks to the advance of computer technology. The capability of executing complex calculations in a very short time has enabled the usage of Computer Aided Engineering (CAE) not only as a verification tool when the design is frozen, but also as a decision making tool. CAE analyses are used from simple trade off studies based on a single aspect of the design, to complex MDO that systematically choose the optimum solution considering many different aspects of the problem simultaneously.

In aerospace engineering and especially in the field of hypersonic propulsion the need for optimization is particularly strong, given the challenge of the final task. Many examples of propulsion system optimization are indeed present in the recent literature. Jahingir et al. in [77] and Pastrone et al. in [78] for instance performed optimization of a RBCC engine using evolutionary techniques, while in [79] an example of optimization of TBCC engine can be found. In [80] an example of optimization of a single parts of an engine, the geometry of the ejector, can also be found.

Often in hypersonic propulsion, especially when applied to space access vehicles, the engine optimization is performed at the same time as vehicle optimization. Bayley et al. [81] for instance performed a full vehicle optimization of a classical solid propellant rocket. While more recently Gong et al. [23] completed a full vehicle and trajectory optimization of a suborbital vehicle powered by an RBCC engine. In this case the full potential of cutting edge High Performance Computing (HPC) is exploited to include into the optimization also high fidelity tip to tail CFD analyses of the vehicle.

In all the previous examples however, the engine conceptual configuration is defined and fixed, while the optimizer can change the design by tweaking a set of parameters. This limitation can seem non relevant for many applications, such as an airliner's engine, where the conceptual design is fixed and well established. Choosing the engine conceptual configuration is instead much more important in the case of hypersonic vehicles, for which an established design is yet to be defined and the debate on which will be the best choice is very open.

An attempt to automatically select the best conceptual engine design is given by Steele in his PhD work [82]. In [82] a pool of predefined engine configurations is considered and then the optimizer acts on the number of engines installed aboard, their type and the sequence in which they are used. In his literature review Steele also noticed that other attempts to do that have been only trade-off analysis between different solutions. Brock et al. [83] performed several studies for a TSTO vehicle considering different propulsion solutions, such as RBCC, TBCC and simple rockets, in order to determine which one would have been the best choice. In [84] similarly to [82] the study considered first a pool of different engine types and then the performance of the space access vehicle was evaluated changing the combination of engine type, the sequence of their usage, the fuel option and also the possibility to take off horizontally or vertically. In all these studies however the configuration of the propulsion system had to be defined a-priori, the optimization procedure (where present) is only able to choose between a pool of engines with fixed design.

In this chapter instead a novel engine optimization procedure will be attempted, the engine

conceptual configuration will not need to be defined a-priori, but it will be the actual result of the optimization process. The modularity and flexibility of HyPro (see chapter 2) is employed here by a configurational optimizer to create randomly generated engine configurations later optimized by means of an Evolutionary Algorithm (EA). The scope of the optimizer is therefore to define how the HyPro modules are interconnected, thus potentially being able not only to choose between TBCC or RBCC (as done in previously cited examples) but also to propose new engine concepts.

In contrast with the examples of engine optimization cited above, here the optimization will only be attempted at a single operation point, since the optimizer is not mature enough to be used for a full mission optimization. The optimization at a single point is however considered by the author an important milestone to be developed before attempting a full mission study. The results presented in this chapter are very promising in both quality and computational resources requirement, leaving confidence on the feasibility of full mission configurational optimization.

5.2 Genetic Programming

5.2.1 Generalities

The problem of optimisating the structural layout or configuration of the engine for its particular operating conditions is complex. This is especially true given the potentially very large number of permutations that need to be considered, the fact that very little may be known about the likely structure of the ideal engine, and that the designer may wish to introduce very little information *a priori* regarding the likely characteristics of the system for fear of contaminating, with his preconceptions, the eventual solution that is reached.

These kind of problems can be handled very efficiently though, using a computational approach known as Genetic Programming (GP)[85, 14, 16]. GP is an evolutionary algorithm-based methodology, inspired by biological evolution, that can be applied to the problem at hand to find engines with configurations that are particularly well-adapted to the operating conditions in which they

are expected to function.

In the engineering context, GP was initially applied to the optimisation of the internal structure of computer programs[14]. It has subsequently been applied successfully to various other engineering problems, where the structure of the design needed to be optimized. In many of these cases, similarly to HyPro (see chapter 2), a block diagram approach is used to model the physics of the problem and its structure is then optimized with GP. Examples of this approach can be found for dynamical systems [86, 87], control systems development[88], electronic circuit design[89] and also to various problems in aerospace engineering such as the design of an antenna for the NASA Space Technology 5 spacecraft[90, 15].

The GP approach is a specialisation of a more general formalism known as the Genetic Algorithm (GA). In the GA a population of individuals, each representing a candidate solution to the problem being analysed, is evolved through several generations. The survivors of this process are taken to be those that are best adapted to the conditions in which the population finds itself. Each individual is uniquely identified by the sequence of 'genes' that make up its 'chromosome' – for instance, in GA, the chromosome might simply be comprised of the list of parameter values that uniquely define the particular individual.

In GA, but also in GP, the main evolutionary procedure is described in the flow chart in Fig. 5.1. Before starting the evolutionary process the population at the first generation is generated by means of a random procedure, that can differ depending on the problem to be studied. The population at generation i is then produced firstly by selecting a subset of the population at generation $i-1$, called breeding population, that is best adapted to their environment (as ranked according to a problem-dependent 'fitness function') and then applying to this subset a sequence of operators that are inspired by the process of biological breeding. The most commonly used operators include 'cross-over', where the chromosomes of two individuals (parents) are combined to generate one or more (child) individuals, and 'mutation', during which random changes are made to parts of one individual's chromosome.

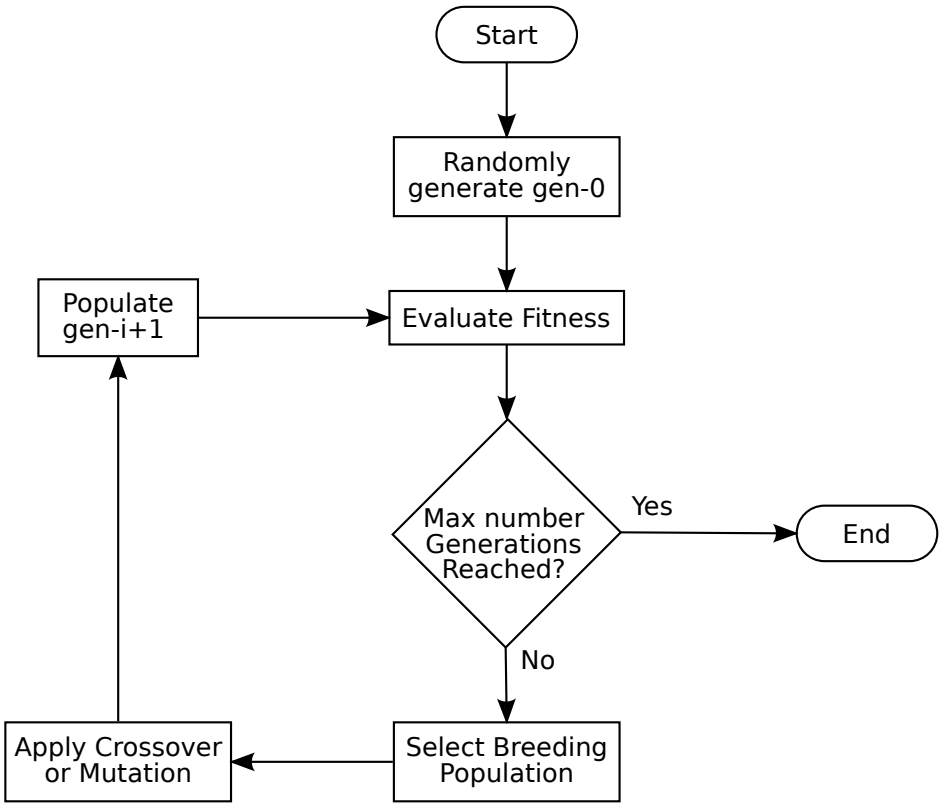


Figure 5.1: Flow chart of the main iteration procedure in GP.

The main difference between GA and GP is that, in GA, the chromosome has a linear structure and a fixed length, whereas in GP the chromosome can have variable length and a tree-shaped structure [85] (see, for example, Fig. 5.2). Each gene in any GP chromosome can have one parent gene and one or more child genes, the gene on top of the three not having any parent is called root gene, the genes not having any child are instead called terminals, while all the other intermediate genes are called functions. The gene behaviour, that is how a gene processes its arguments (i.e. its child), is determined by its value. In Fig. 5.2, for instance, the 'summation operator' gene (the symbol '+' in this case is the value of the gene) at the left-hand side of the three, represented by the symbol '+', has two children (the summands 'x' and '5') and one parent (the operator 'max', that operates on the result of the summation). Because of the structural flexibility of its chromosomes, GP, in contrast with GA, can be used to solve optimization problems without requiring the user to specify the structure of the solution in advance[14] (see Fig. 5.2). This key feature of the

methodology enables the structure of the solution to the problem to evolve and thus potentially to change during the course of the calculation.

The crossover operator in GP operates in analogy with the crossover happening in nature, different sequences of genes of the two parents are mixed to generate the chromosome of the offspring (see Fig. 5.3). First a crossover point is randomly chosen in both parents in order to select a sub-tree, then the offspring is generated as a copy of the first parent, where the selected sub-tree is replaced with the sub-tree selected on the other parent. The mutation operator instead operates on a single individual in order to generate a different individual with a chromosome slightly mutated (see Fig. 5.4). In GP, similarly with the crossover operator, a mutation point is first randomly selected, then a new individual is created exchanging the sub-tree selected by the mutation point with a randomly generated sub-tree. Another operator, called reproduction, is simply defined to copy an individual to the population of the next generation.

The choice between usage of the aforementioned operators is done probabilistically and the probability of application of each operator is defined by the user. Usually high probability has to be assigned to the crossover operator (e.g. 90%), while only a small probability is usually given to the mutation operator (e.g. 1%), the remaining probability is then assigned to the reproduction (refer to [14, 16] for details).

5.2.2 Implementation for the HyPro model

In the case of the previously cited antenna optimization [90, 15] the correspondence between GP tree and the antenna structure is clear, because, as can be seen in Fig. 5.5, the antenna design is also a tree. Other designs models such as an electric circuit, a control system with feedbacks and also the flow structure of an engine, cannot be sketched as a tree. Despite this difficulty also these problems can be optimized using GP, using a dedicated technique to map the physical model starting from the GP tree (see for instance [89, 86, 87, 88]).

Due to the similarities between the engine flow network of HyPro and an electrical circuit scheme,

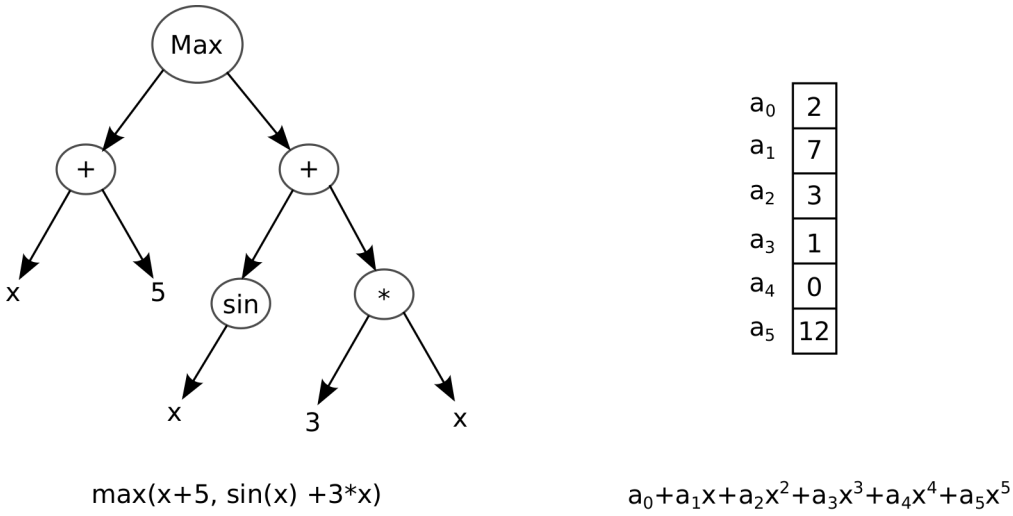


Figure 5.2: Example of a GP tree chromosome (left) compared to a linear GA chromosome (right). They both represent a mathematical expression, but GA requires its structure to be defined a priori, while GP can represent a vast variety of symbolic expression without user's intervention.

the approach used by Koza to optimise the layout of electronic circuits [89] is most appropriate for this application, but some modifications to account for certain specifics of the engine modelling problem need first to be accounted for. As anticipated above, not all engine configurations can be represented directly in terms of a tree-shaped chromosome (see for instance the structure shown in Fig. 2.14). This is particularly the case when the possibility of feedback between the modules that comprise the engine must be accommodated in the HyPro model (see section 2.11 for details). To obviate this problem, Koza introduced a method called *cellular encoding* [14], in which the GP genes represent a list of instructions that modify an initial simple prototype structure for the individual [14, 89], rather than representing directly the structure of the individual itself. In the automated design of a complex electronic circuit many different instruction gene types need to be coded for (for instance the addition of parallel components, serial components, and so on), in this work only two types of operation are defined: the addition of one specified HyPro module, and the addition of feedback between two specified modules.

The 'module addition' gene's value is the HyPro module to be added, therefore the number of child required depends on the value's type of module. For each input node of the module the

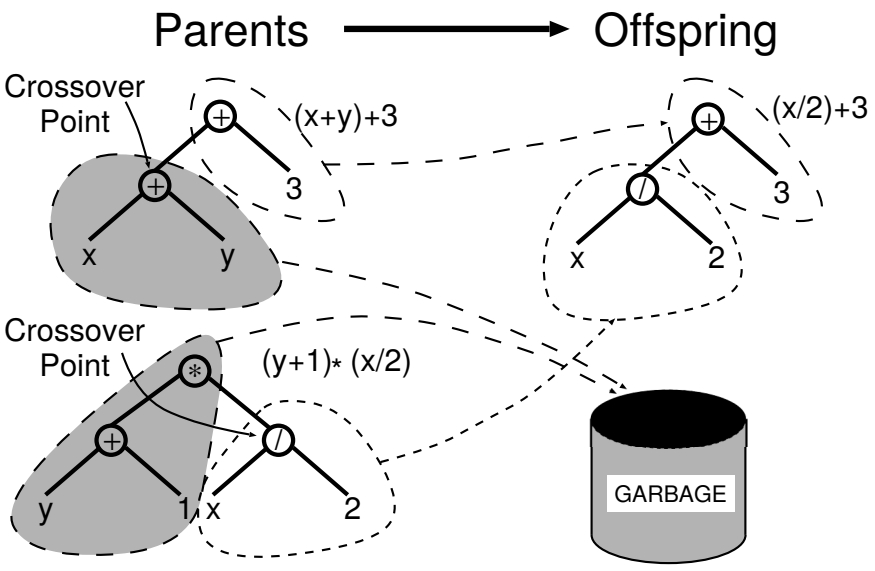


Figure 5.3: Example of a GP crossover operation. [14]

'module addition' gene will have a child of type 'module addition', that in turn will add the module connected with the aforementioned input node. In Fig. 5.6 for example, the gene with value equal to 'Mixer' has two child of type 'module addition' that provide the modules 'Rocket' and 'Inlet' connected with the 'Mixer' input nodes. If its value needs to be the source of a choking feedback (i.e. it can experience choking, see section 2.11), the 'module addition' gene has also one 'feedback definition' gene as a child.

The 'feedback definition' gene's value is an integer, that defines the feedback destination, the value indicates the number of modules to be counted upstream of the feedback source (see section 2.11 for details about the feedback structure). In Fig. 5.6 for instance, the 'feedback definition' gene 'Feed-1' (i.e. with value equal to 1) child of the gene 'Combustion Chamber' defines a feedback from the 'Combustion Chamber' module to the 'Injection' module, that is placed one position upstream. The gene 'Feed-0' instead defines a feedback loop with the same source and destination, since the destination is zero positions upstream of the source. In the case that the 'feedback definition' value would set the destination past the most upstream module, the latter is assumed as feedback destination. The aforementioned counting strategy however, is not univocal in the case where the engine flow path is a branching structure (i.e. there is at least a 'Mixer' module). In Fig. 5.6 for

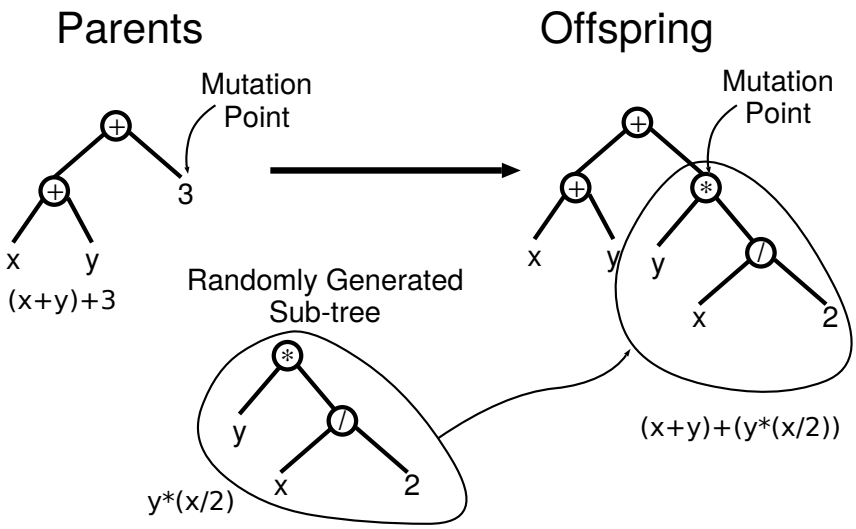


Figure 5.4: Example of a GP mutation operation. [14]

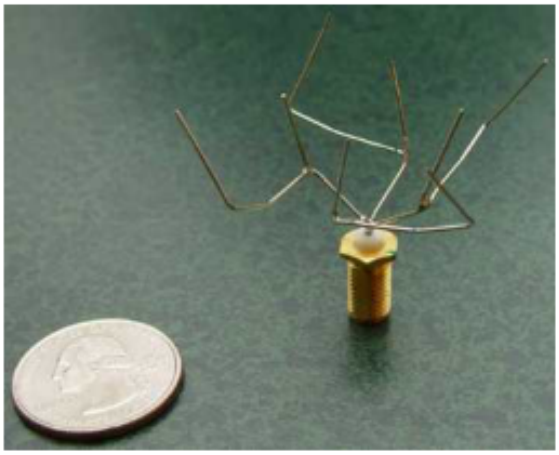


Figure 5.5: Antenna designed using GP for the NASA Space Technology 5 spacecraft. [15]

instance, when counting past the 'Mixer' module, it is not clear if the module upstream the 'Mixer' is the 'Inlet' or the 'Rocket'. This problem is solved introducing an iterator that counts by branches, such an iterator moves only on the branch linked to the node N_1 of the 'Mixer' and then, once the most upstream module of that branch is reached, it moves on the other branch starting from the module connected with node N_2, (see also the example of Fig. 5.7). That is why in Fig. 5.6 the gene 'Feed-1' child of the 'Mixer' defines a feedback pointing to the 'Inlet' and not to the 'Rocket', because the former is linked to the node N_1 of the 'Mixer'. In the end it is worth noticing that the 'feedback definition' gene requires no child, therefore it is always a terminal.

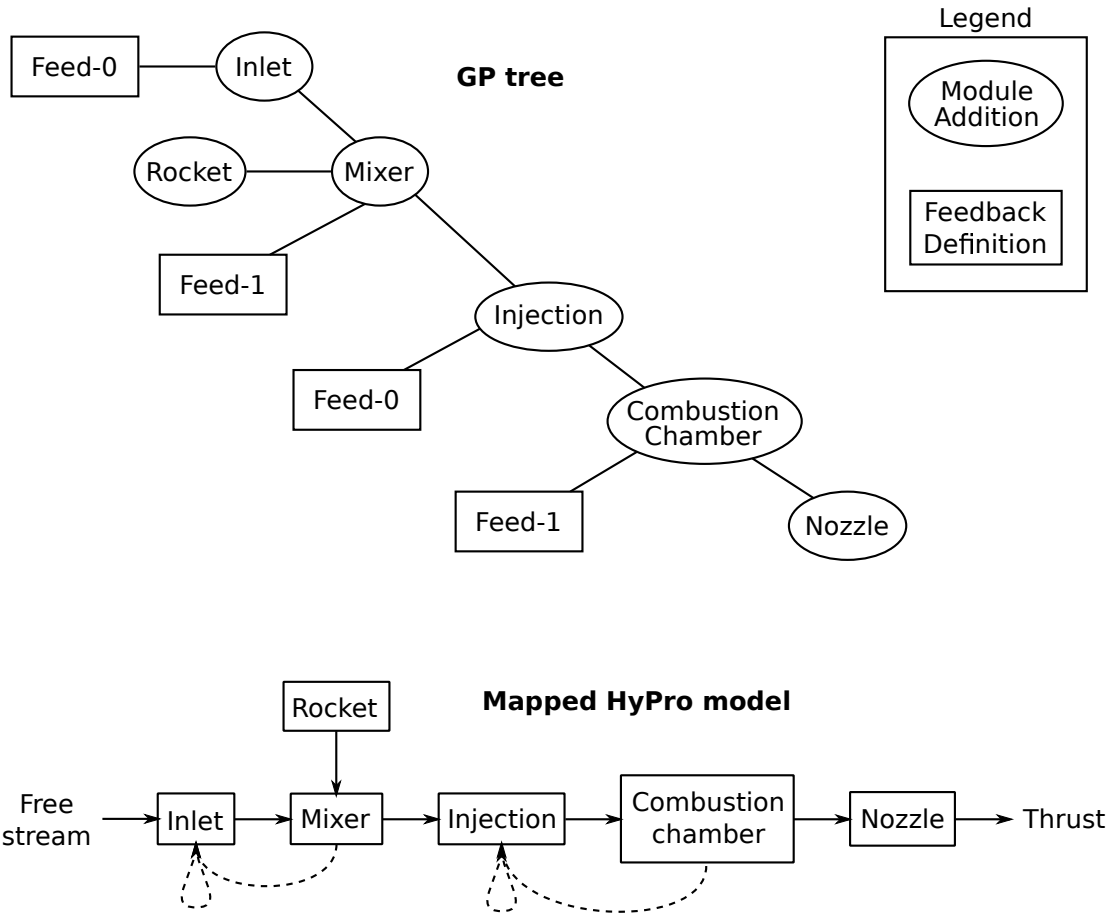


Figure 5.6: Mapping between an example GP tree chromosome (top) and its corresponding HyPro engine model (bottom).

All the gene's type rules explained above need to be obeyed also during the evolution of the population, leading to a strategy that is called *strongly typed GP* in Genetic Programming jargon [14, 91]. In *strongly typed GP* the crossover and mutation operators have to enforce the following type rules. In the crossover operator the two selected crossover points have to be of the same type, while in the mutation the randomly generated tree has to have the root gene of the same type of the mutation point.

As explained above, with *cellular encoding* the genome is different from the real structure of the individual, therefore prior to the evaluation of its fitness a computational step, called *genotype to phenotype* decoding [14], is required. In HyPro this step creates the actual engine model adding all the required modules and nodes together with their mutual connections (see Fig. 5.8). The

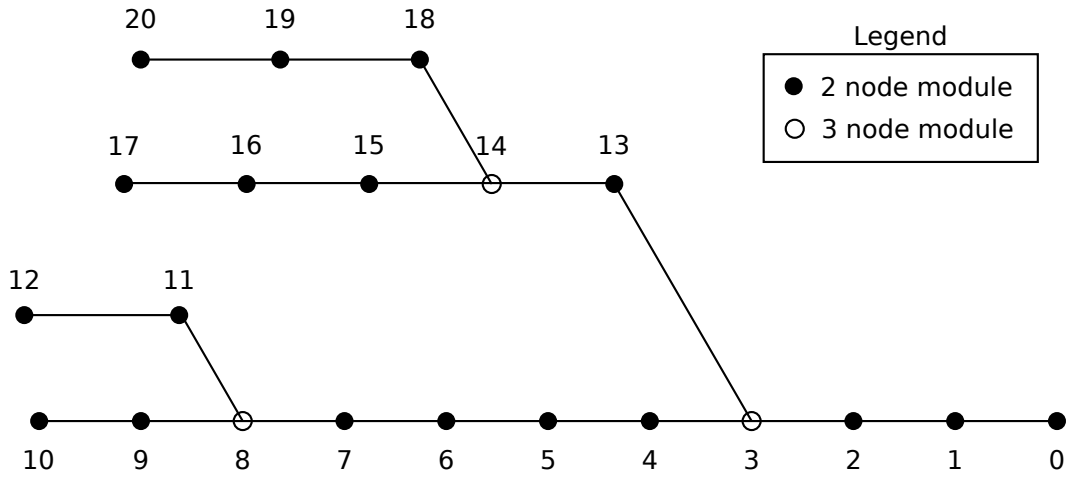


Figure 5.7: Example of upstream counting by branches in case of multi branched HyPro model. By convention the node N_1 of every 3 node module (i.e. the 'Mixer') branches straight to the left of the graph, while the node N_2 branches upward.

decoding algorithm described in Fig. 5.8 works recursively starting from the root gene, that calls the decoding routines of its child, that in turn calls the same routine of their child, until all the genes are decoded.

To close the optimization problem a fitness function needs to be defined in order to rank the individuals and consequently choose the breeding population (see section 5.2.1). In this case the fitness function chosen is the specific impulse of the engine. This choice represents the simplest possible optimization for an engine, because it does not take into account many other requirements such as the specific thrust for example. However at this stage of optimizer development it is sufficient to demonstrate the performances of the optimization procedure. In detail the fitness function *ft* is defined as follow:

$$ft = \begin{cases} max\left(0,\ \dfrac{F}{\dot{m}_f}\right), & if \text{ calculation is correct} \\ 0, & if \text{ exception is thrown} \end{cases} \tag{5.1}$$

where the calculation performed in HyPro is not successful an exception is thrown the fitness is assumed null. The handling of exceptions during the fitness evaluation is important because during the evolution is possible to generate individuals that cannot actually be run in a stable manner to produce thrust. These non-viable engines however are not discarded out-of-hand from

the population, since they might merely correspond to an intermediate stage in the evolution of an engine configuration that might prove very successful in later generations.

Finally in order to start the evolution the GP search space need to be defined. The search space consists of a pool of genes that will be used to randomly generate the population at generation zero and to randomly generate sub-trees during the mutation operation. The pool of 'feedback definition' genes simply consists of a gene for each integer number between zero and a sufficient high value chosen by the user. The pool of 'module addition' genes consists instead of a gene for each propulsive module, which the user wants to consider in the optimization. More than a module per HyPro class (see section 2.2) can be considered, simply changing the parameters that define it such as the efficiencies, the length or the section areas. The parameters of a module cannot be changed by the GP optimizer, that act only on the interconnection between modules present in the pool.

The section area of the modules defined within the initial gene pool has a complex influence on the final optimized engines, that is worth explaining in detail here. As can be seen in Fig. 5.8, each 'module addition' gene is responsible for the construction of the output node of the module added, but not of the input nodes, since they have already been added by its child. The area of the input nodes of a module thus are not controlled by the gene that actually adds it. Moreover for modules with constant area, the area of output node will also not be determined by the gene that construct it, since it has to be equal to the area of the input node. The 'Injection' of Fig. 5.6 for instance, will have a cross section area determined by the exit area of the 'Mixer' upstream. Therefore during the construction of the 'module addition' genes' pool the user needs to remember that the value of the section area of a constant area module and the area of the inlet of a variable area duct are irrelevant.

The aforementioned GP formalism is implemented in HyPro using the Genetic Programming C++ Class Library (GPC++) that is based upon the work of Koza[16, 14, 92]. The library has been integrated into HyPro after some modification to allow strongly typed GP of the form described

above.

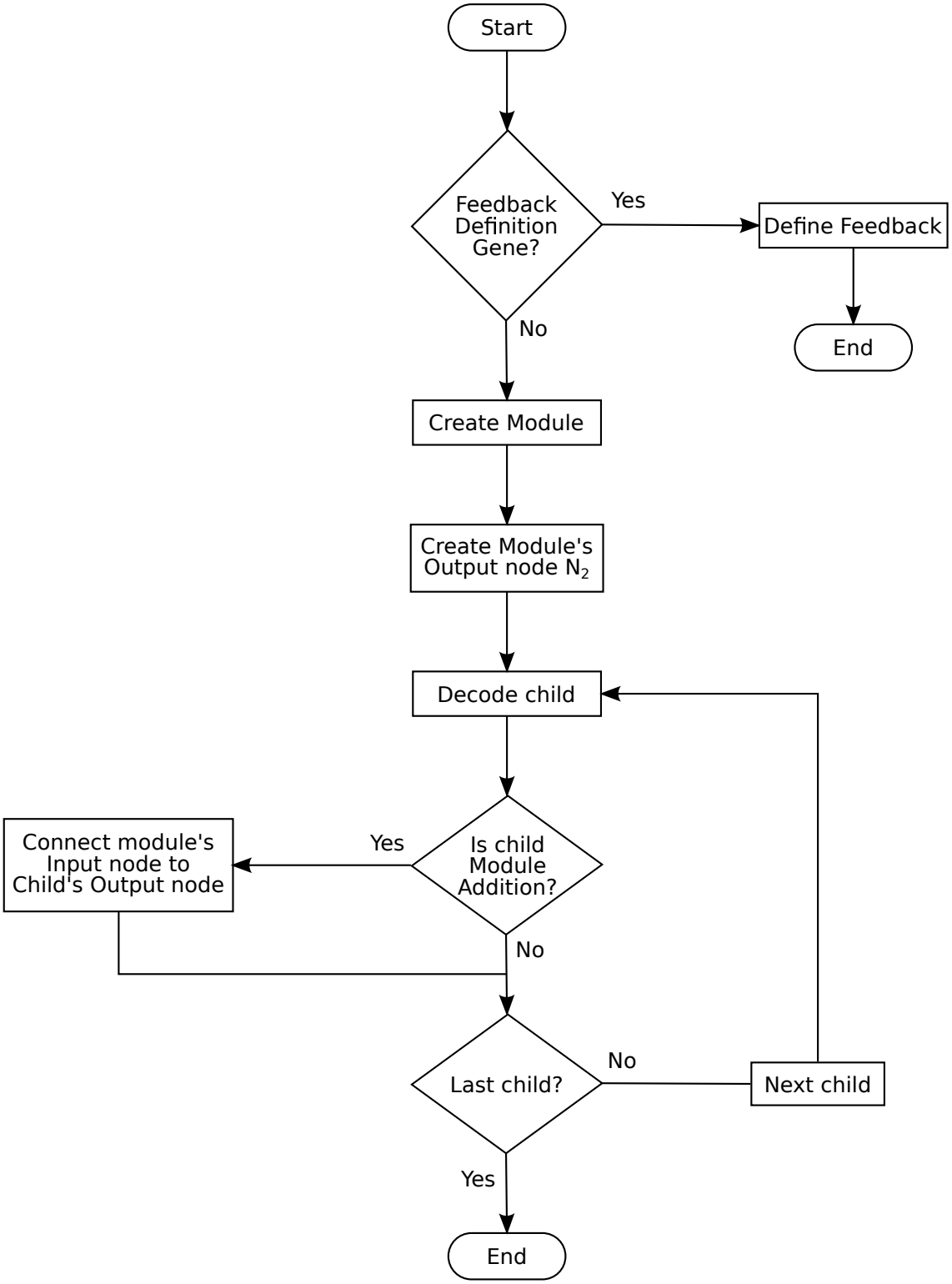


Figure 5.8: Flow chart of the GP tree decoding routine.

5.3 Results

Results are presented here showing the performance of the resultant algorithm in selecting an appropriate engine configuration for the propulsion of a typical space plane under two different representative flight conditions, one supersonic and one subsonic.

The pool of 'module addition' genes (see section 5.2.2) has been constructed starting from the modules used to represent the Hyperion engine, presented in section 4.4. For the purposes of the demonstration, a stagnation pressure drop of 2% is applied between the inlet and outlet of all 'inlet', 'diffuser' and 'nozzle' modules and a friction coefficient of 0.01 is applied within all 'injection' and 'combustor' modules.

The 'rocket' module used in section 4.4 is supplanted with the more effective 'injection plate' module described in section 2.7.1, in order to avoid the so called 'seeding effect' [14]. This problem arises when a clearly superior configuration of engine is introduced already at the first generation. Because of the resultant very high probability of its subsequent selection, its chromosome can come to dominate the population in subsequent generations. In this particular case the 'rocket' module is by itself a relatively good engine, since it can provide thrust also alone. Especially at the first generations, when the probability to find thrust generating engines is low, the 'rocket' can easily take over, shadowing any other more complex but more efficient engine. Substituting the 'rocket' with its injection, the rocket subsystem is not removed from the engine optimization, since a rocket can always be modelled by the combination of an 'injection plate' a 'combustor' and a 'nozzle'. But the seeding effect is prevented due to the decreased likelihood to generate a functioning rocket at the first generation.

The principal parameters of the optimization algorithm are listed in Table 5.1. A population of 300 engines is evolved for 20 subsequent generations. A maximum depth for the tree structure of the chromosome of each individual was imposed simply in order to limit the complexity of the configurations that might emerge through evolution from the initial population. The probability of mutation in the examples presented here was kept non-zero but small in order to increase the

diversity of the population.

Population Size	300
Number of Generations	20
Maximum Tree Depth for Creation	6
Maximum Tree Depth for Crossover	17
Selection Type	tournament
Tournament Size	20
Swap Mutation Probability	3.0%
Shrink Mutation Probability	3.0%

Table 5.1: Parameters for the GP algorithm[16].

5.3.1 Supersonic test case

As the first example of the application of this procedure, the engine that emerges from the evolution as most suited to supersonic flight conditions at Mach 2.5 at sea level (temperature 288.15 K and pressure 101.325 kPa) has the configuration shown in Fig. 5.9. It is quickly apparent that the engine has the configuration of a ramjet, albeit, most interestingly, one with several combustion chambers.

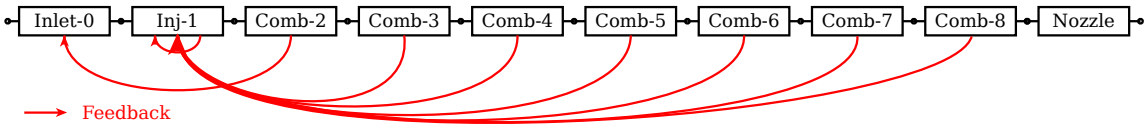


Figure 5.9: Supersonic test case: Configuration of the best engine of the last evolutionary generation.

A small amount of additional investigation reveals that the computational algorithm has simply attempted in this instance to achieve some form of solution that circumvents the restrictions and inherent limitations that were originally placed on the formulation of the problem by its programmers. At first sight, the presence of more than one combustion chamber looks rather unhelpful, particularly since, given a particular quirk of the way in which the model is defined, all the fuel is always consumed in the first combustion chamber. Moreover the friction that is associated with

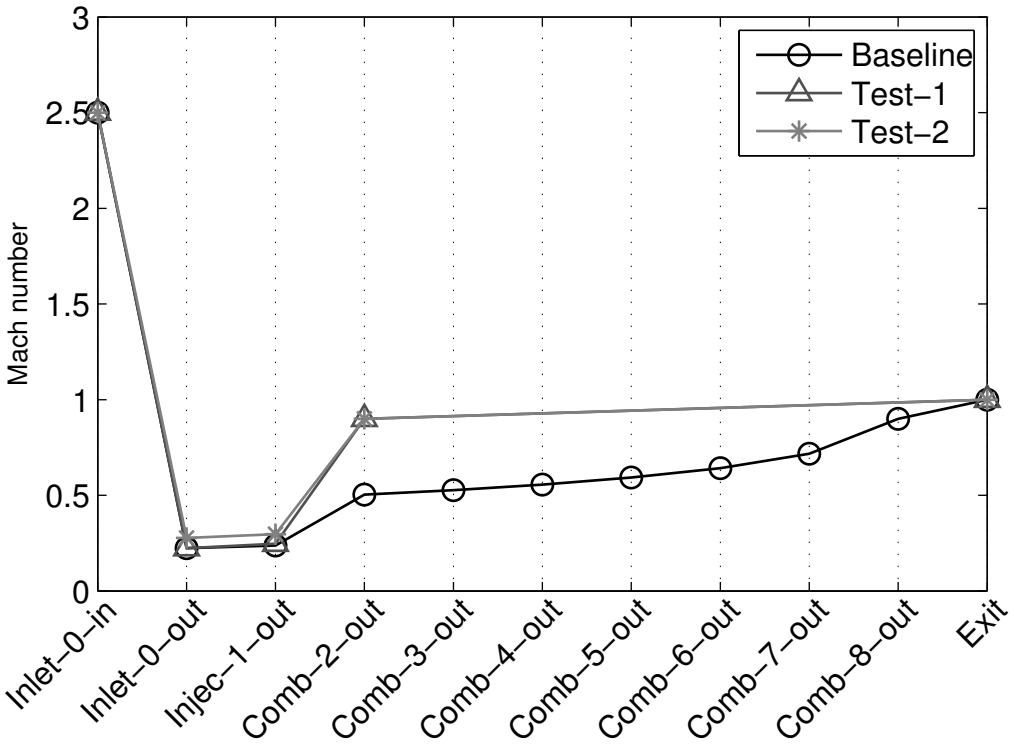


Figure 5.10: Supersonic test case: Variation of Mach number along the axis of the engine.

the additional ducting should result in a significant performance penalty to the engine compared to the equivalent engine with only one combustion chamber. The data presented in Tab. 5.2 reveals however that the fitness of the engine, having all parameters equal but with only one combustion chamber (Test-1), is slightly lower than that of the engine selected by the GP algorithm itself.

This behaviour can be explained by looking at Fig. 5.10, where the variation of Mach number along the axis of the engine is plotted. As can be seen, the Mach number always reaches the choking limit (here set conservatively to Mach 0.9 to account for unmodelled viscous effects in the engine, as described in section 2.11) at the end of the last combustion chamber, but for the Test-1 configuration the combustion chamber is much shorter than that of the baseline configuration, so the pressure drop across it is also smaller. The lower friction allows more fuel to be injected into the Test-1 configuration before the choking limit is reached, as is demonstrated in Fig. 5.12, where it is clear that the jump in mass flow rate across the injection module of the Test-1 configuration

Test name	Description	Gross Thrust [N]	Net Thrust [N]	Ram drag [N]	propellant flow rate $\dot{m}_f$ [kg/s]	Fitness $F/\dot{m}_f$ [m/s]
Baseline	Outcome of GP evolution	$8.990 \cdot 10^5$	$3.133 \cdot 10^5$	$5.857 \cdot 10^5$	8.957	$3.50 \cdot 10^4$
Test-1	Baseline with one comb. chamber	$1.099 \cdot 10^6$	$5.137 \cdot 10^5$	$5.857 \cdot 10^5$	15.61	$3.29 \cdot 10^4$
Test-2	Test-1 with $\varphi = 0.45$	$1.111 \cdot 10^6$	$3.979 \cdot 10^5$	$7.127 \cdot 10^5$	10.90	$3.65 \cdot 10^4$

Table 5.2: Supersonic test case: performance of the best adapted engine configuration and its variations.

is bigger than in the baseline case. Further evidence of this effect can be found in Fig. 5.11, where the variation of static temperature is plotted along the axis of the engine. It can be seen that the post-combustion temperature is much higher in the Test-1 configuration than in the baseline case. This is simply because the conditions in its combustion chamber are closer to being stoichiometric. To prove this assertion, further data is presented in Tab. 5.2 for when the equivalence ratio of the injection module in the simplified Test-1 configuration has been lowered to 0.45, in other words to the value that is required in the baseline condition in order to induce choked conditions at the end of the combustion chamber. The fitness of this engine (or, in real terms, its specific impulse) is higher than that of both the Test-1 and the baseline engine configurations. Interpreting these results, the process of addition of combustion chambers can be seen to have emerged during the evolution of the engine system as an indirect strategy that can be used to tune the equivalence ratio within the injection module.

Thus it can be concluded in this particular case that the GP algorithm has attempted to use the modules at its disposal to overcome the limitations of the current formulation of the method in not being able to tune directly the module parameters (see section 5.2.2), in this specific case, the equivalence ratio φ in the injection module. The resultant solution of the evolutionary algorithm thus might appear to be unnecessary complicated to human eyes, but, in the frame of the computational procedure and its constraints, it is the best solution that can be found.

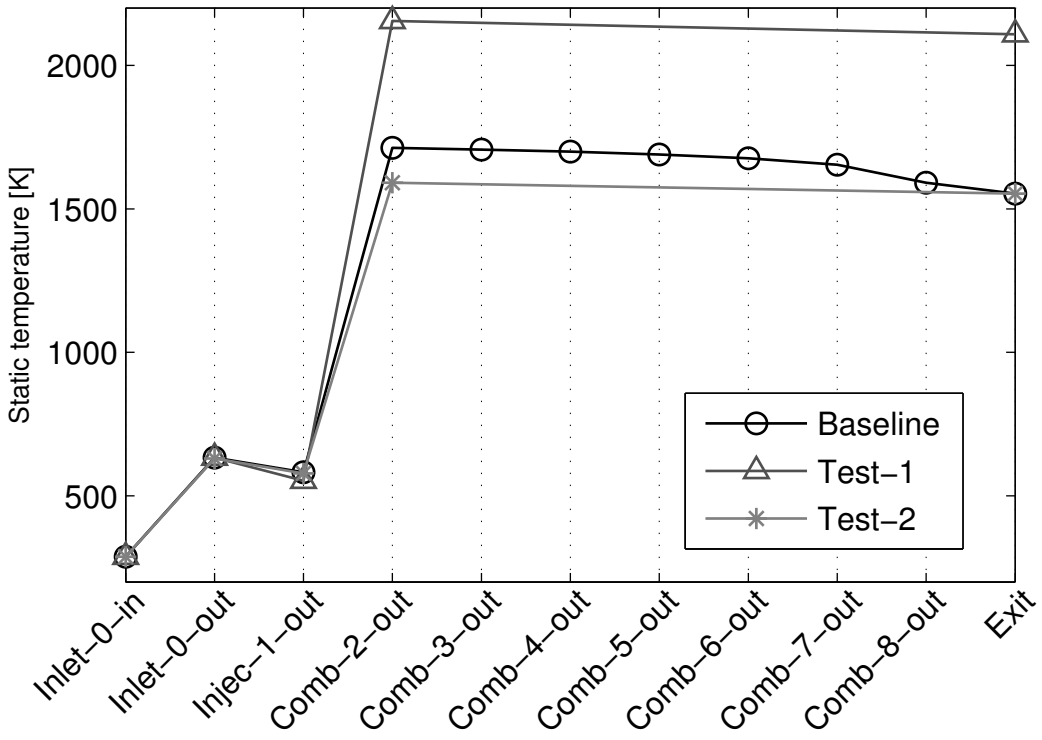


Figure 5.11: Supersonic test case: Variation of temperature along the axis of the engine.

In Fig. 5.13 the structure of the best engine at different evolutionary generations is depicted. It is worth noticing that the best engine at generation zero is completely different than the final optimized configuration, its flow path can be considered an air augmented rocket rather than a ramjet. This observation is important to demonstrate that the randomly generated first generation does not strongly influence the final result. It is also interesting to notice how the evolution moves from an air augmented rocket (generation 0) to an ejector ramjet (generation 1) and finally to a pure ramjet (generation 3). In this case only 3 generations were needed to evolve the conceptual design of the engine, indeed after generation 3 only minor changes are included, that resemble more a parameter tuning rather than a proper configuration change. The algorithm converges to the final solution already at generation 11, since the best engine configuration is always confirmed in all the subsequent generations. The population as a whole is however not constant after generation 11, since the average fitness (calculated removing from the population all the engine with negative

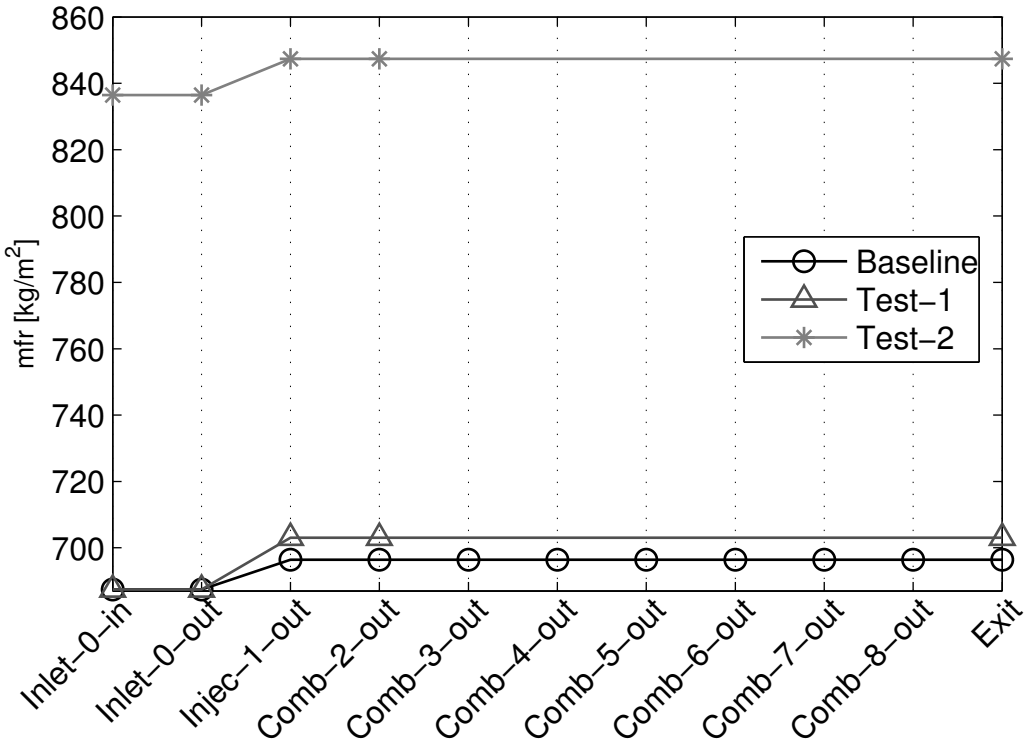


Figure 5.12: Supersonic test case: Variation of mass flow rate along the axis of the engine.

thrust or with unsuccessful calculation of performances) keeps improving after that point (see Fig. 5.14) and the number of bad engines (i.e. the engines not included in the previously mentioned average calculation) keeps decreasing, as can be seen in Fig. 5.15.

5.3.2 Subsonic test case

The configuration that emerges from the evolution when the operational conditions are changed to represent subsonic flight conditions (Mach 0.5) at sea level (temperature 288.15 K and pressure 101.325 kPa) is shown in Fig. 5.16, where the feedback loops have been removed for clarity. With a little bit of puzzling it is apparent that, in its essence, the configuration is that of an ejector ramjet, albeit with multiple inlets and a very complex internal flow path. The performance of the corresponding engine is summarised in Tab. 5.3. As in the previous case, it is easy to see that, although the procedure has produced a completely plausible and readily acceptable configuration

for the engine that is best suited to the imposed operating conditions, the algorithm has introduced additional complexity into the engine configuration in order to accommodate the inherent deficiencies in the modelling procedure, namely the inability of the optimizer to tune directly the parameters of the engine's modules. In this case the various additional inlets seem to have been introduced as a way of augmenting the flow of air trough the secondary; the human designer would have achieved the same end simply by increasing the size of the primary inlet to the engine.

Looking at the evolution history in Fig. 5.17 and Fig. 5.18, it is evident that in this case the evolution has not converged to a definitive solution, since the fitness does not reach a plateau before generation 20. The GP evolution was therefore run for an other 20 generations in order to try to find a fully converged solution, as can be observed in Fig. 5.17 and Fig. 5.18, which does not occur. The reason why an optimum solution seems not to be possible in this case, is that, in the frame of the mathematical model used in HyPro for the ERJ, the greater the secondary flow is compared with the primary the greater is the performance of the engine. In Fig. 5.19 and Fig. 5.20 it is clear how the amount of inlets (i.e. the size of the secondary) increases iteration after iteration, leading to a very complex configuration at generation 20 and to a configuration at generation 40 so complex that it is actually impossible to show here. This behaviour is after all physically reasonable if we assume that the mixing is always possible no matter how much the secondary flow is bigger than the primary, indeed the propulsive efficiency of an air-breathing engine increases if the mass of air used increases. This design optimization process indeed resembles the design trend in turbofan engines to progressively increase the bypass ratio. In order to obtain practical results, the addition of other considerations beside the pure engine performances are therefore required. A mass model would for instance take into account the penalty of increasing the size of the engine in order to increase the size of the secondary flow.

Despite these initial frustrations, the author believe that the methodology shows significant promise in being able to produce a workable means of allowing the engine with the configuration that is most appropriate to the operating conditions of the vehicle to be selected automatically, or at

least with a minimum of *a priori* intervention from the designer, and thus to introduce at least some degree of objectivity into the process of discriminating between the many different configurations that have been proposed for space plane propulsion in the past.

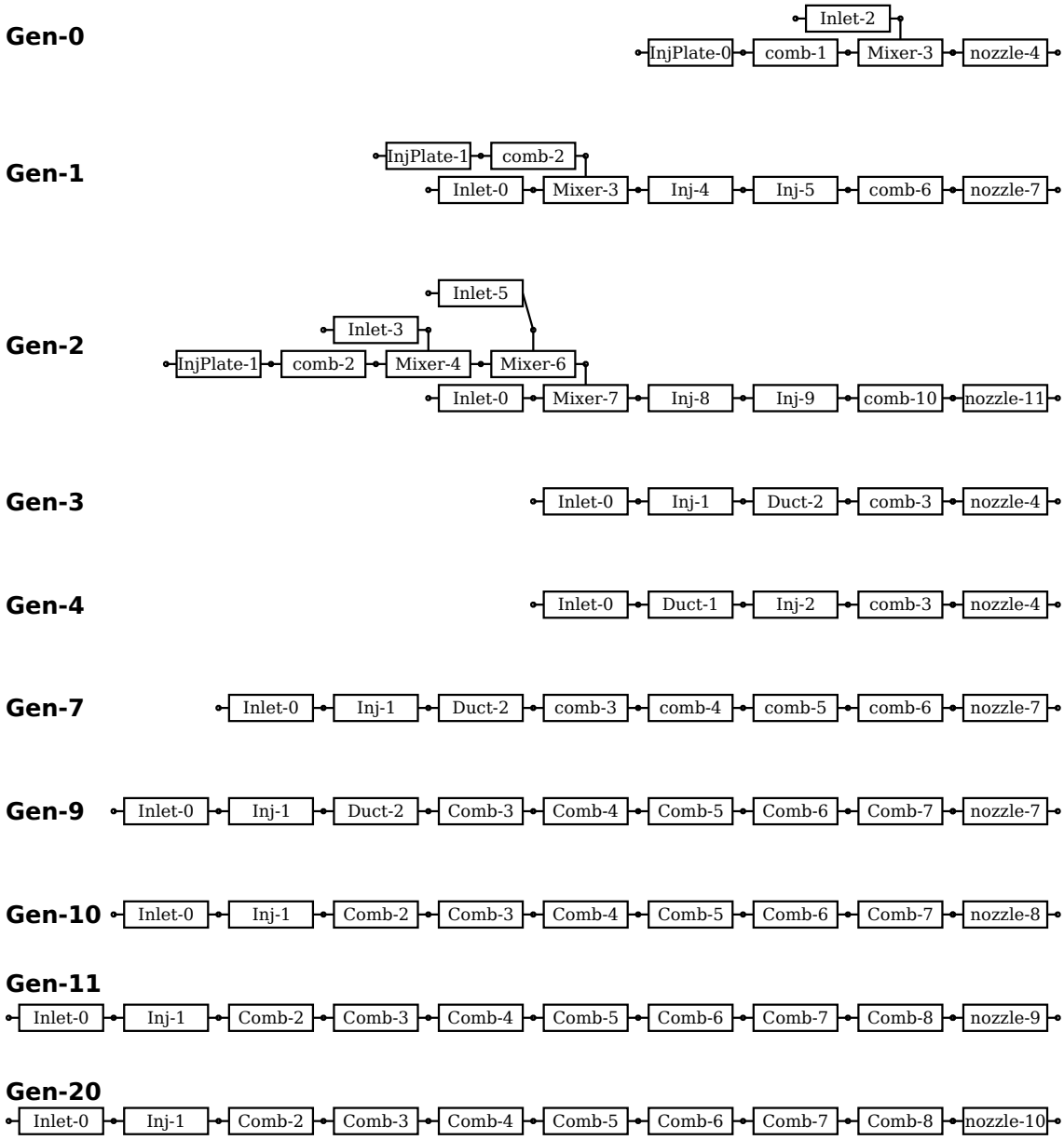


Figure 5.13: Supersonic test case: Evolution of the best engine of the population.

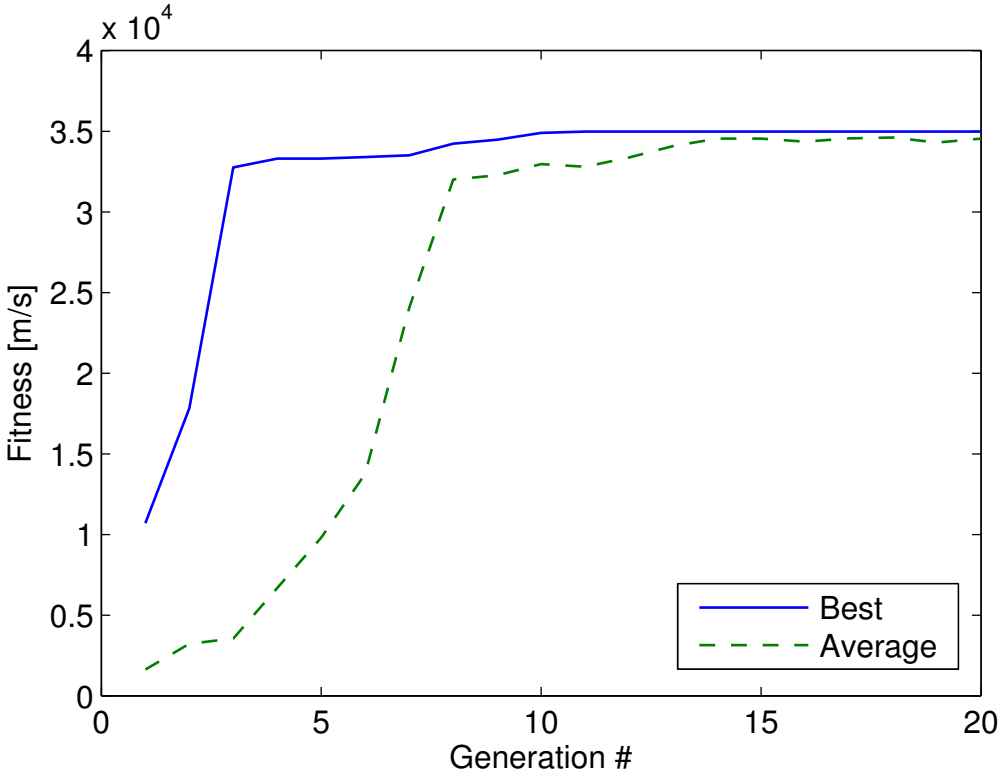


Figure 5.14: Supersonic test case: Evolution of the fitness. The fitness of the best engine and the average fitness of all engines with positive thrust are plotted.

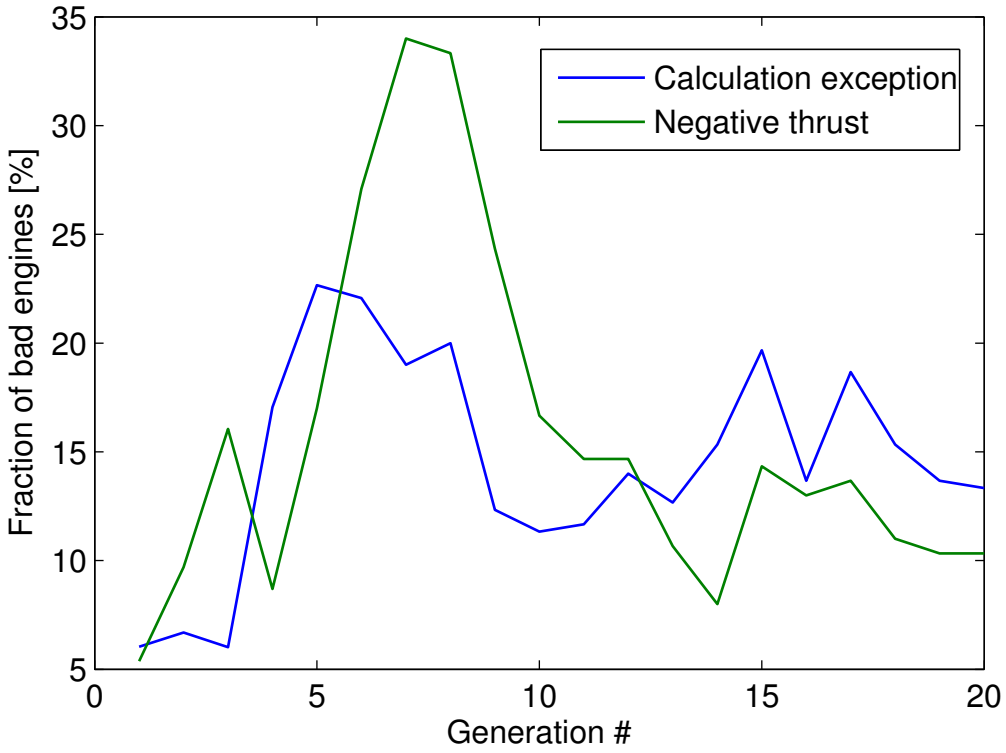


Figure 5.15: Supersonic test case: Evolution of the fraction of bad performing engines within the population. Both the fraction of engine with negative thrust and that of engine throwing an exception during the calculation of performances are plotted.

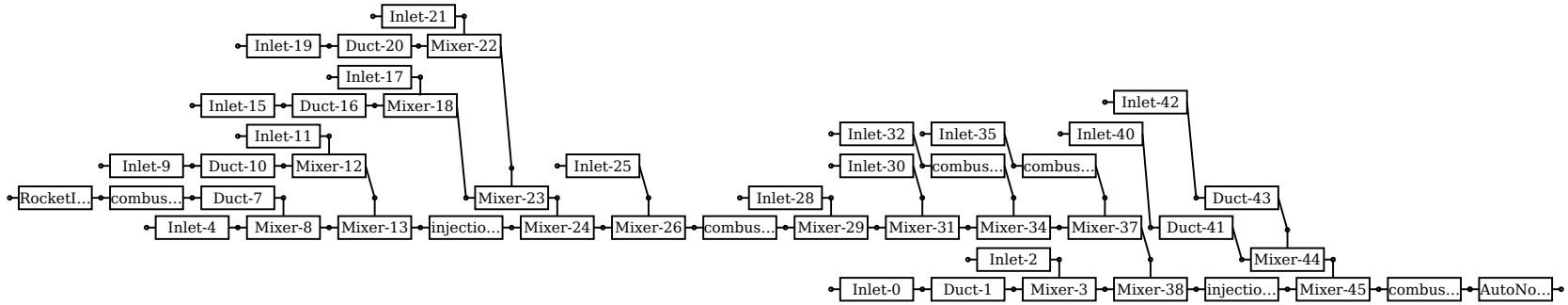


Figure 5.16: Subsonic test case: Configuration of the best engine of the last evolutionary generation.

Test name	Description	Gross Thrust [N]	Net Thrust [N]	Ram drag [N]	propellant flow rate $\dot{m}_f$ [kg/s]	Fitness $F/\dot{m}_f$ [m/s]
Baseline	Outcome of GP evolution	$4.627 \cdot 10^6$	$4.166 \cdot 10^6$	$4.610 \cdot 10^5$	166.79	$2.50 \cdot 10^4$

Table 5.3: Subsonic test case: performance of the best adapted engine configuration.

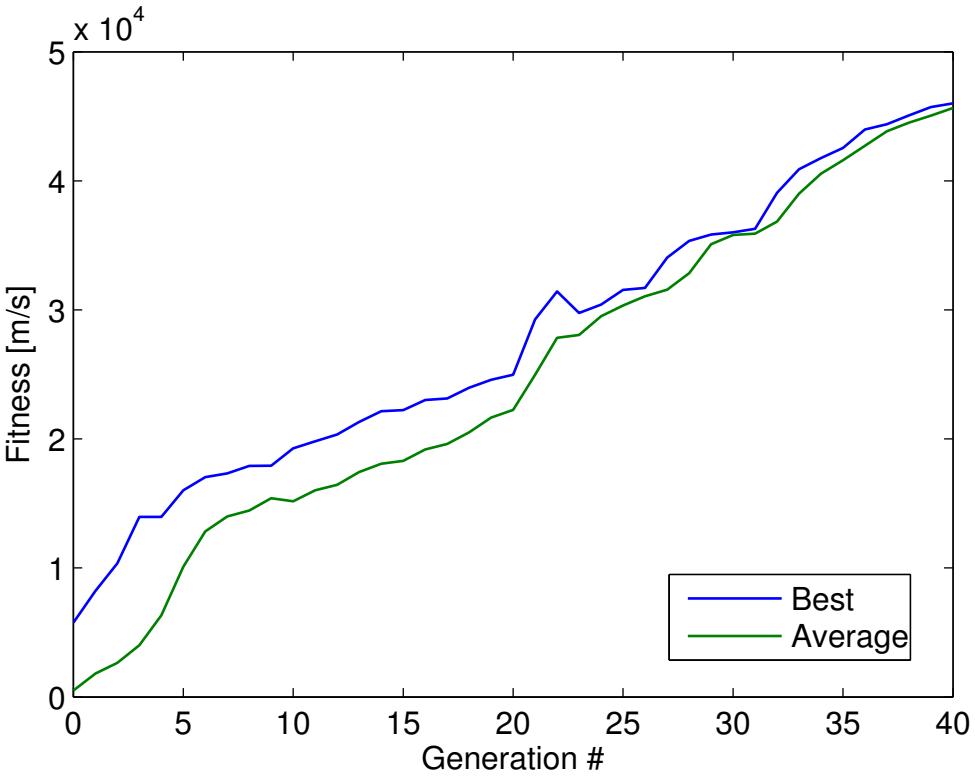


Figure 5.17: Subsonic test case: Evolution of the fitness. The fitness of the best engine and the average fitness of all engines with positive thrust are plotted.

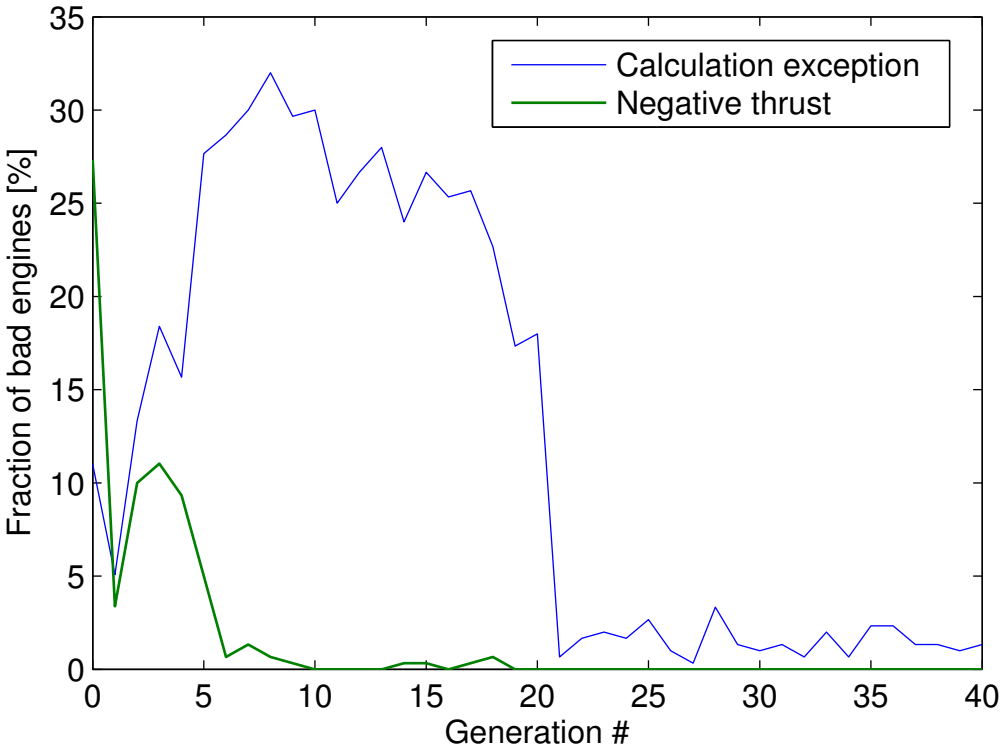


Figure 5.18: Subsonic test case: Evolution of the fraction of bad performing engines within the population. Both the fraction of engine with negative thrust and that of engine throwing an exception during the calculation of performances are plotted.

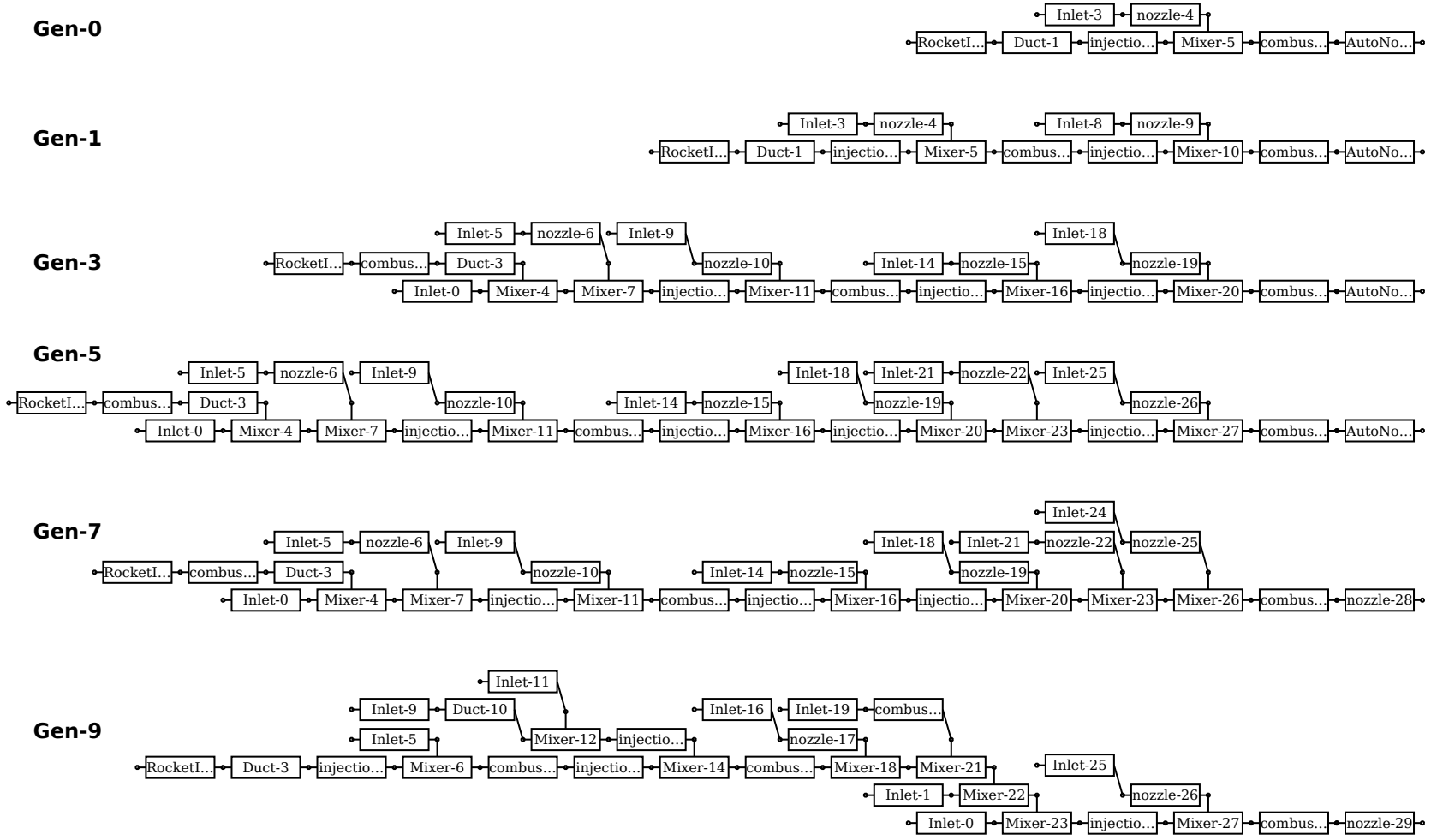


Figure 5.19: Supersonic test case: Evolution of the best engine of the population.

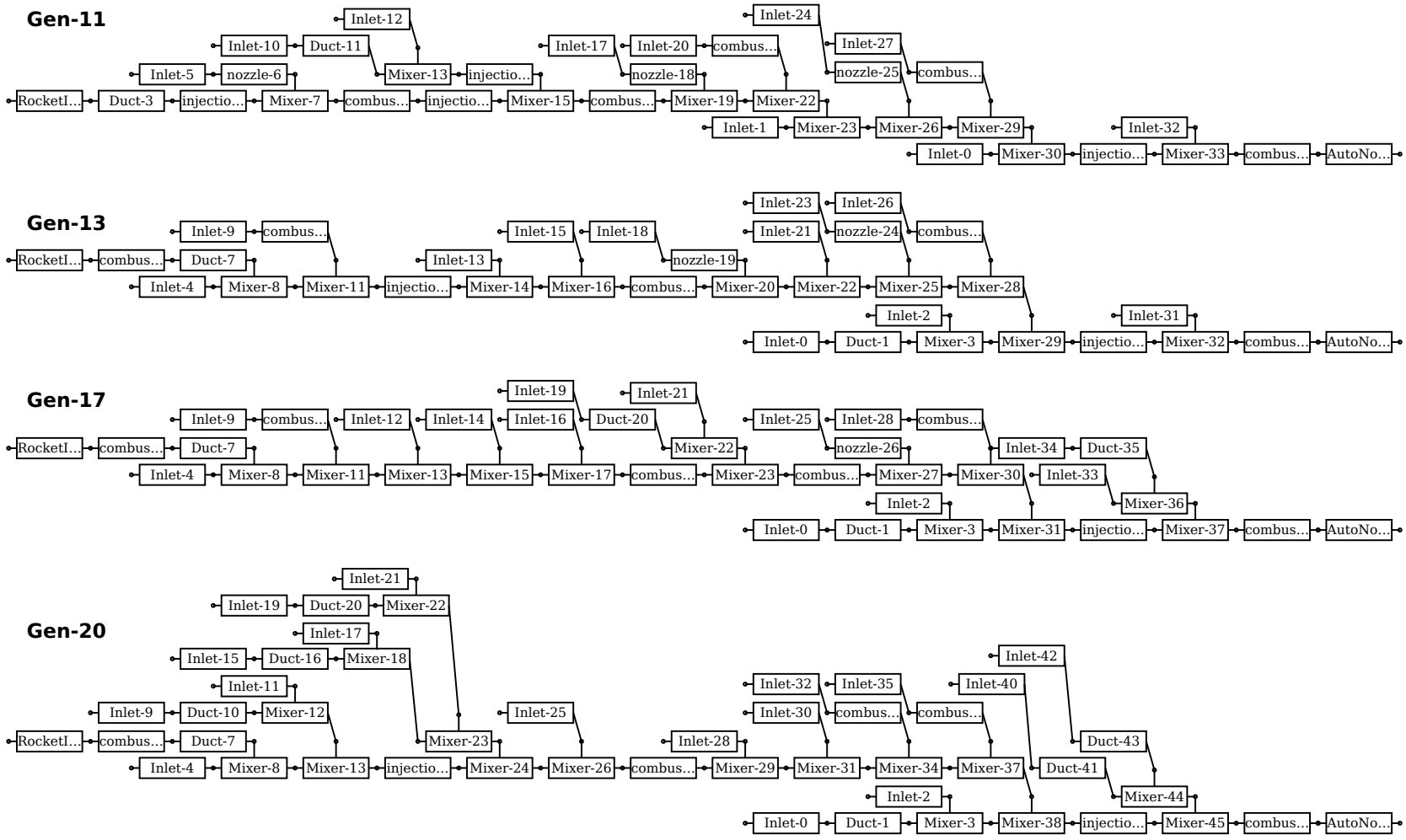


Figure 5.20: Supersonic test case: Evolution of the best engine of the population. (cont.)

Chapter 6

Conclusion

6.1 Parsimonious model

A parsimonious propulsion system modelling tool, called the HYbrid PRopulsion Optimizer, or HyPro, has been developed with a modular and parametric structure by means of the OOP native with C++. Thanks to the modular structure of the code, it is very general purpose and can therefore be used for many different propulsive system configurations. HyPro is indeed mainly conceived to be applied to CCP where several different propulsion technologies can be combined in a single engine. The engine model is for now capable of modelling RBCC engines, therefore it can model rocket engines, ramjets, scramjets and ejectors. In addition a very simple model is available for turbine and compressors, but the component matching between the two is not implemented yet.

In parallel, an in-house CFD model has been developed and used to validate HyPro. The new CFD code, called rhoFoam, has been developed in the OpenFOAM® environment, which is one of the most popular open source CFD general purpose codes. The rhoFoam solver employs the WAS flux scheme with exact Riemann solver. This approach has been successfully validated against CFD results available in literature. The selected test case concerns a scramjet test bed in frozen chemistry conditions, thus it is very representative of the complex fluidynamic conditions found in

hypersonic air-breathing engines. This reference approach used a commercial CFD code and was then validated against shock tunnel data.

The HyPro model has been extensively validated and verified against analytical models, CFD data, other simplified models and flight data. The correctness of the general behaviour of HyPro has been verified at component level by means of comparison against well known and established analytical models. At system level it has instead been verified against a similar model called SCCREAM, considering an ejector ramjet engine as test case. The verification against SCCREAM is important to demonstrate that the global behaviour of HyPro is correct also for complex configurations such as the ejector ramjet, for which is difficult to find experimental data. In the comparison against SCCREAM our model was capable of capturing the evolution of the thrust and specific impulse with altitude and Mach number, with an accuracy comparable with that of SCCREAM.

A dedicated verification has been also provided for the mixer module, since it is critical for the assessment of the global performances of ejector engines. First a CFD analysis has been conducted in order to obtain detailed data on a typical ejector configuration (i.e. the Marquardt's SERJ engine) then the CFD data have been replicated with HyPro. The mixer efficiency in the HyPro model has been tuned in order to replicate the CFD data, showing how is possible to embed results of high fidelity computation within the HyPro parsimonious model simply using a tuning parameter. This procedure could be used also for other modules improving sensitively the accuracy of the HyPro model without increasing its computational cost. It is however a procedure only possible at advanced project phases where the engine design is known in such a detail to be meaningfully studied by means of CFD analyses. At conceptual design it is more likely to use efficiency's values known by experience.

Validation for HyPro is first provided for a scramjet configuration at single operative point, comparing it against CFD data available in literature. It is then also validated against flight data available for a ballistic ramjet engine along a complete trajectory. The results in the first case are very good, but maybe less significant due to the analysis of a single operative point. More

interesting are the results in the second validation due to the wide flight range considered. In this case the results seems less accurate with a peak error of 30% for the thrust and 15% for the specific impulse, but the source of such an uncertainty remains unclear due to missing information about the atmosphere during the test. The important outcome is however that the general behaviour of the engine is correctly evaluated and the accuracy is however acceptable, considering that the code is meant to be applied mainly at conceptual design.

The HyPro model has demonstrated to be effective and accurate (at least accurate enough for conceptual design) in modelling of RBCC, ramjet, scramjet and rocket propulsion systems. Very diverse propulsion system can be modelled with the same tool thanks to its modularity. Another very important characteristic of HyPro is the short execution time required. A performance evaluation run at single operative point requires order of one hundredth of second to run, in case of complex RBCC configuration, much less for simple configuration like a rocket engine. These performances are vital when HyPro is used within optimization loops where it needs to be run thousand if not million of times.

6.2 Configurational optimization

Thanks to its fast execution it has been possible to include HyPro within an optimization loop. The optimization is however innovative because it does not optimize a set of engine's parameters as it is commonly done in literature, but it is an optimization of the engine configuration, that therefore is not fixed during the process. The optimizer uses the GP optimization algorithm in order to optimize how the engine components are interconnected, taking advantage of the modular approach used in HyPro. The user does not need to define the conceptual design of the engine prior the start of the optimization loop, since it will be the outcome of the optimization process. All the user needs to do is to set up a list of engine modules that the optimizer can use to build the engine.

This procedure has been tested in two operative conditions one supersonic and the other sub-

sonic. The modules considered for the optimization are those used to describe ejector ramjet RBCC engines, while the specific impulse is the fitness function. At supersonic conditions the optimized configuration was a simple ramjet. This result is perfectly reasonable because the ramjet is actually the most efficient solution in supersonic. Ejector ramjet indeed are foreseen only in order to provide acceleration in subsonic and low supersonic flight when the ramjet is not effective.

The actual optimized configuration, even if recognisable as a ramjet, has however some oddities that can be explained as a reaction to a limitation of the code. At this stage the optimization algorithm is only able to move and connect the predefined modules, it cannot modify the parameters that characterise the detail of each module (for instance the module size, the quantity of fuel injected and so on). The optimized configuration presents therefore some unexpected features evolved in order to modify indirectly the parameters of the modules and consequently improve the performances of the engine. In this test case in particular, the fuel injected is lowered adding more than one combustion chamber, thus improving the performances in terms of specific impulse. Although this is an imperfection that should be solved in the long term, here it can be considered good news because it demonstrates the effectiveness of the GP algorithm, able also to work around the limitation of its own implementation.

Similar conclusions can be drawn for the subsonic case, where the optimized configuration is an ejector ramjet as expected. In this case the limitation discussed above leads the optimizer to propose a solution with many inlets in an effort to increase the ratio between the secondary flow and the rocket primary. In this case the optimizer seems not to converge to a solution, since it adds more and more inlets at each generation. This behaviour can be explained by the fact that, in the frame of the modelling provided in HyPro, the efficiency of an ejector ramjet can be indefinitely improved increasing the secondary to primary ratio. This is of course not practical since the size of the engine would rise indefinitely too, but it proves again that the optimizer is working correctly. A fitness function accounting for instance for the mass of the engine also, could fix the issue.

The outcome of the optimizer can seem quite obvious in these cases, since the results are in line

with engineering practice and do not find any innovative propulsion concept. This is however due to the limited amount of propulsive modules available in this version of the code, so here the results have to be considered only as a verification of the optimization procedure. The fact that a solution is found in line with engineering practice demonstrate that the optimizer is well designed and that practical and innovative results could be obtained with extension of the modelling capabilities of HyPro. These results have to be considered as the first step toward an engine configurational optimizer that to my knowledge has not been attempted before.

6.3 Future development

As explained in the previous sections, this is only a first version of the HYbrid PRopulsion Optimizer, the full potentiality of the code can be unleashed only if further development is performed. First of all the applicability range has to be extended beyond the RBCC configurations. This task can be accomplished adding modelling for few other propulsion components such as heat exchangers, turbines and compressors. It is also possible that other iteration structures will be required beside the choking feedback, for instance in order to perform the component matching between turbine and compressor.

Beside pure performance analysis, other disciplines will have to be added in order to fulfil the objective of multidisciplinary design. The development of a component-based mass model is probably the most urgent analysis to be added, followed by structural and thermal analyses. These analyses do not have to be very complicated, they can rely on a simple database of thermo-structural thresholds. Later on, life cycle analyses could be provided in order to take into account aspects such as the maintenance and the reliability of the system.

In parallel to the improvement of HyPro modelling, the optimizer will have to be updated in order to handle the new capabilities. Even without considering the changes in the modelling, the optimizer will need improvements in several directions. First of all access to modules' parameters has to be granted to the optimizer. In GP this is possible in the frame of typed GP and it has

already been used in other areas, such as electric circuit design. Also, the addition of multi-objective optimisation is important, because it is useful when other disciplines are added to the engine model. Subsequently the optimization has to be implemented also for more than one operative condition and eventually the optimization should be performed for a whole vehicle trajectory.

The accomplishment of all the aforementioned tasks will require a considerable effort in terms of development and testing. It is however reasonable to see this as a good opportunity to deliver innovative research, since each time a new feature is added a new optimization is possible with results every time more realistic. With team work and coordination it is also possible to perform this plan in a reasonable time, with a very valuable outcome for the space plane community. This tool could generate some innovative engine configurations during its further development. With a bit of luck, or intelligence depending on the point of view, the discovered configuration could demonstrate to be both innovative and advantageous, leading to a new research branch focused on that particular new idea. There have been indeed documented cases of patents released from the outcome of GP optimization, in other engineering fields. If successful for the engine, this methodology with a bit of creativity can be also extended at vehicle level with even more interesting possibilities.

This research, from my point of view, is very interesting not just for the results presented here after the short time of a PhD studentship, but for its future possibilities. A continued development of this tool can provide a big playground for students (a few are already contributing to the project) and researchers and, if successful, could become valuable tool in the effort to reduce the cost of access to space.

References

[1] Czysz, P. A. and Bruno, C., *Future Spacecraft Propulsion Systems*, Springer Praxis Books, Springer Berlin Heidelberg, 2009.

[2] Penn, J. P. and Lindley, C. A., "RLV Design Optimization for Human Presence in Space," *The Aerospace Corporation* [online], `http://www.spacefuture.com/archive/rlv_design_optimization_for_human_presence_in_space.shtml` [retrieved 04 December 2015].

[3] Mankins, J. C., "Highly reusable space transportation: Advanced concepts and the opening of the space frontier," *Acta Astronautica*, Vol. 51, No. 10, 2002, pp. 727 – 742.

[4] Escher, W. J. and Flornes, B. J., "A Study of Composite Propulsion Systems for Advanced Launch Vehicle Applications," Vol. 1, The Marquardt Corporation, Van Nuys, CA, 1966.

[5] Escher, W. J. and Flornes, B. J., "A Study of Composite Propulsion Systems for Advanced Launch Vehicle Applications," Vol. 3, The Marquardt Corporation, Van Nuys, CA, 1966.

[6] Merlin, P. W., "Design and development of the blackbird: Challenges and lessons learned," *47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Orlando, FL, January*, 2009, pp. 5–8.

[7] Robinson, J. S. and Martin, J. G., "An Overview of NASAs Integrated Design and Engineering Analysis (IDEA) Environment," *17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, April 2011.

[8] Escher, W. J. and Flornes, B. J., "A Study of Composite Propulsion Systems for Advanced Launch Vehicle Applications," Vol. 7, The Marquardt Corporation, Van Nuys, CA, 1966.

[9] Bradford, J. and Olds, J., "Improvements and Enhancements to SCCREAM, a Conceptual RBCC Engine Analysis Tool," *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, American Institute of Aeronautics and Astronautics, Cleveland, OH, 13-15 July 1998.

[10] Billig, F., "Ramjet and Ramrocket Propulsión Systems for Missiles," *AGARD PEP lecturas Series*, Vol. 136, 1984.

[11] Lorrain, P., Brieschenk, S., Capra, B., and Boyce, R., "A Detailed Investigation of Nominally 2-D Radical-Farming Scramjet Combustion," *18th AIAA/3AF International Space Planes and Hypersonic Systems and Technologies Conference*, Tours, France, 24 - 28 September 2012.

[12] Olds, J. and Bradford, J., "SCCREAM (Simulated Combined-Cycle Rocket Engine Analysis Module): A Conceptual RBCC Engine Design Tool," *33rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, American Institute of Aeronautics and Astronautics, Seattle, WA, 6 - 9 July 1997.

[13] Marguet, R., "Study and flight tests of a mach 5 experimental ramjet," Tech. Rep. 19670008070, National Aeronautics and Space Administration, 1966.

[14] Poli, R., Langdon, W. B., and McPhee, N. F., *A field guide to genetic programming*, Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008, (With contributions by J. R. Koza).

[15] Hornby, G. S., Globus, A., Linden, D. S., and Lohn, J. D., "Automated antenna design with evolutionary algorithms," *Proc. 2006 AIAA Space Conference*, 2006, p. 8.

[16] Koza, J. R., *Genetic programming II: Automatic discovery of reusable subprograms*, The MIT Press, 55 Hayward Street, Cambridge, MA 02142 USA, 1994.

[17] Launius, R. D., "NASA AND THE DECISION TO BUILD THE SPACE SHUTTLE, 1969-72," *The Historian*, Vol. 57, No. 1, 1994, pp. 17, Last updated - 2013-02-23.

[18] Stuhlinger, E., "Why Explore Space? A 1970 Letter to a Nun in Africa," *Roger Launius's Blog* [online], `https://launiusr.wordpress.com/2012/02/08/why-explore-space-a-1970-letter-to-a-nun-in-africa/` [retrieved 05 December 2015].

[19] Moskowitz, C., "Stephen Hawking Says Humanity Won't Survive Without Leaving Earth," *SPACE.com* [online], `http://www.space.com/8924-stephen-hawking-humanity-won-survive-leaving-earth.html` [retrieved 04 December 2015].

[20] Hollingham, R., "Should we build a village on the moon?" *BBC* [online], `http://www.bbc.com/future/story/20150712-should-we-build-a-village-on-the-moon` [retrieved 05 December 2015].

[21] "Ariane 6," *European Space Agency (ESA)* [online], `http://www.esa.int/Our_Activities/Launchers/Launch_vehicles/Ariane_6` [retrieved 04 December 2015].

[22] Ueda, S., Tomioka, S., Saito, T., Tani, K., and Yoshida, M., "R&D on Hydrocarbon-fueled RBCC Engines for a TSTO Launch Vehicle," *20th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, 2015, p. 3611.

[23] Gong, C., Chen, B., and Gu, L., *Design and Optimization of RBCC Powered Suborbital Reusable Launch Vehicle*, AIAA Aviation, American Institute of Aeronautics and Astronautics, 06 2014, doi:10.2514/6.2014-2361.

[24] Olds, J., Bradford, J., Charania, A., Ledsinger, L., McCormick, D., and Sorensen, K., "Hyperion: an SSTO vision vehicle concept utilizing rocket-based combined cycle propulsion," *AIAA paper*, 1999, pp. 99–4944.

[25] Sippel, M., Schwanekamp, T., Trivailo, O., and Lentsch, A., "Progress of SpaceLiner Rocket-Powered High-Speed Concept," *64th International Astronautical Congress*, 2013.

[26] Lockwood, M. K., "Overview of conceptual design of early venturestar configurations," *AIAA paper*, Vol. 1042, 2000.

[27] Jenkins, D. R., Landis, T., and Miller, J., *American X-vehicles: An inventory, X-1 to X-50*, National Aeronautics and Space Administration, Office of External Relations, History Office, 2003.

[28] Peebles, C., *Road to Mach 10: lessons learned from the X-43A flight research program*, Amer Inst of Aeronautics &, 2008.

[29] McClinton, C. R., Rausch, V. L., Sitz, J., and Reukauf, P., "Hyper-X program status," *AIAA paper*, Vol. 828, 2001, pp. 2001.

[30] Hank, J. M., Murphy, J. S., and Mutzman, R. C., "The X-51A scramjet engine flight demonstration program," *AIAA Paper*, Vol. 2540, 2008, pp. 2008.

[31] Bolender, M. A., Dauby, B., Muse, J. A., and Adamczak, D., "HIFiRE 6: Overview and Status Update 2014," *20th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, 2015, p. 3537.

[32] Malucchi, G., Zaccagnino, E., Drocco, A., Dussy, S., and Préaud, J., "The European Re-entry Program, from IXV to ISV–GNC/Avionics Development Status and Challenges," *AIAA, Boston, USA*, 2013.

[33] Clark, C., "The History and Promise of Combined Cycle Engines for Access to Space Applications," Tech. Rep. DFRC-E-DAA-TN2127, NASA Dryden Flight Research Center, Edwards, CA, United States, June 2010.

[34] Taguchi, H., Kobayashi, H., Kojima, T., Ueno, A., Imamura, S., Hongoh, M., Harada, K., and Tsuchiya, T., "Systems Analysis on Hypersonic Airplanes using Pre-Cooled Turbojet Engine,"

17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, 2011, p. 2320.

[35] Savino, R., Russo, G., Carandente, V., and Doriano, V., "Hyplane: Challenges for Space Tourism and Business Transportation," *J Aeronaut Aerospace Eng*, Vol. 2, No. 123, 2013, pp. 2.

[36] Varvill, R. and Bond, A., "A comparison of propulsion concepts for SSTO reusable launchers," *JOURNAL-BRITISH INTERPLANETARY SOCIETY*, Vol. 56, No. 3/4, 2003, pp. 108–117.

[37] Bond, A., "Study on the application of the sabre engine for FESTIP," Tech. Rep. 950826, Reaction Engine Limited, 1997.

[38] Webber, H., Feast, S., and Bond, A., "Heat exchanger design in combined cycle engines," *Internationa Aerospace Conference*, 2008.

[39] Taguchi, H., Kobayashi, H., Kojima, T., Hongoh, M., Masaki, D., and Nishida, S., "Performance Evaluation of Hypersonic Pre-Cooled Turbojet Engine," *20th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, 2015, p. 3593.

[40] Mogavero, A., Taylor, I., and Brown, R. E., "Hybrid Propulsion Parametric and Modular Model: a novel engine analysis tool conceived for design optimization," *AIAA Aviation - 19th AIAA International Space Planes and Hypersonic Systems and Technologies Conference.*, American Institute of Aeronautics and Astronautics, 06 2014, doi:10.2514/6.2014-2787.

[41] Wuilbercq, R., Pescetelli, F., Mogavero, A., Minisci, E., and Brown, R. E., "Robust Multidisciplinary Design and Optimisation of a Reusable Launch Vehicle," *19th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, AIAA Aviation, American Institute of Aeronautics and Astronautics, 06 2014, doi:10.2514/6.2014-2363.

[42] Mogavero, A. and Brown, R., "An improved engine analysis and optimisation tool for hypersonic combined cycle engines," *AIAA Aviation - 20th AIAA International Space Planes*

and Hypersonic Systems and Technologies Conference., American Institute of Aeronautics and Astronautics, 07 2015.

[43] Tran, K., *One Dimensional Analysis Program for Scramjet and Ramjet Flowpaths*, Master's thesis, University Libraries, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2010.

[44] Bradford, J., *A technique for rapid prediction of aftbody nozzle performance for hypersonic launch vehicle design*, Ph.D. thesis, Georgia Institute of Technology, 2001.

[45] O'Brien, T., Starkey, R., and Lewis, M., "Quasi-one-dimensional high-speed engine model with finite-rate chemistry," *Journal of propulsion and power*, Vol. 17, No. 6, 2001, pp. 1366–1374.

[46] Birzer, C. H. and Doolan, C. J., "Quasi-One-Dimensional Model of Hydrogen-Fueled Scramjet Combustors," *Journal of propulsion and power*, Vol. 25, No. 6, 2009, pp. 1220–1225.

[47] Doolan, C. J. and Boyce, R., "A Quasi-one-dimensional mixing and combustion code for trajectory optimisation and design studies," *15th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, No. 2008-2603, American Institute of Aeronautics and Astronautics, 2008.

[48] Vanyai, T., Brieschenk, S., Bricalli, M. G., and Boyce, R. R., "Quasi-One-Dimensional Investigation of Combustion Processes on Scramjet Performance," *19th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, AIAA Aviation, American Institute of Aeronautics and Astronautics, 06 2014.

[49] Pinckney, S., Ferlemann, S., and Mills, G., *Program Manual for SRGULL. Version 2.0: Second Generation Engineering Model For The Prediction Of AirframeIntegrated Subsonic/Supersonic Combustion Ramjet Cycle Performance*, NASA Langley Research Center, Hampton, VA, December 2004.

[50] Corbin, C. R., *Design and analysis of a Mach 3 dual mode scramjet combustor*, Ph.D. thesis, Wright State University, 2008.

[51] Zweber, J., Kabis, H., Follett, W., and Ramabadran, N., "Towards an integrated modeling environment for hypersonic vehicle design and synthesis," *AIAA/AAAF 11th International Space Planes and Hypersonic Systems and Technologies Conference, AIAA, Orleans France*, 2002.

[52] Gupta, N. K., Gupta, B. K., Ananthkrishnan, N., Shevare, G. R., Park, I. S., and Yoon, H. G., "Integrated Modeling and Simulation of an Air-breathing Combustion System Dynamics," *AIAA Paper*, Vol. 6374, 2007.

[53] Tian, C. and Gong, G., "A scramjet engine modeling method and validation for real-time simulation," *Modelling, Identification & Control (ICMIC)*, IEEE, 24-26 June 2012, pp. 665–672.

[54] Burkardt, L. and Franciscus, L., "RAMSCRAM: A flexible ramjet/scramjet engine simulation program," Conference Paper 19900004919, NASA Lewis Research Center, Cleveland, OH, United States, Jan 1990.

[55] Bradford, J. and Olds, J., "SCCREAM v. 5: a Web-Based Airbreathing Propulsion Analysis Tool," *35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Georgia Institute of Technology, American Institute of Aeronautics and Astronautics, Los Angeles, CA, 20-24 June 1999.

[56] Jorrín, A., de Prada, C., and Cobas, P., "EcosimPro and its EL Object-Oriented Modeling Language." *EOOLT*, 2008, pp. 95–104.

[57] Villacé, V. F. and Paniagua, G., "Simulation of a Combined Cycle for High Speed Propulsion," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace*

Exposition, American Institute of Aeronautics and Astronautics, Orlando, Florida, January 2010.

[58] "Cplusplus.com," *cplusplus.com* [online], `http://www.cplusplus.com/` [retrieved 19 January 2016].

[59] Yamaguchi, H., *Engineering Fluid Mechanics*, Springer, 2008.

[60] Gomes, A., Voiculescu, I., Jorge, J., Wyvill, B., and Galbraith, C., "Root-Finding Methods," *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*, Springer London, 2009, pp. 117–140.

[61] Gordon, S. and McBride, B. J., "Computer Program for Calculation of Complex Chemical Equilibrium," *NASA Reference Publication 1311*, 1994.

[62] Cengel, Y. A., Klein, S., and Beckman, W., *Heat transfer: a practical approach*, WBC McGraw-Hill Boston, 1998.

[63] Mayer, D. W. and Paynter, G. C., "Prediction of supersonic inlet unstart caused by freestream disturbances," *AIAA journal*, Vol. 33, No. 2, 1995, pp. 266–275.

[64] Greatrix, D. R., *Powered Flight*, Springer, 2012.

[65] Carlomagno, G. M., *Elementi di Gasdinamica*, Liguori, 2009.

[66] Greenshields, C., Weller, H., Gasparini, L., and Reese, J., "Implementation of Semi-Discrete, Non-Staggered Central Schemes in a Colocated, Polyhedral, Finite Volume Framework, for High-Speed Viscous Flows," *International Journal for Numerical Methods in Fluids*, Vol. 63, No. 1, 2010, pp. 1–21.

[67] Toro, E. F., *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, Springer, 2009.

[68] Kurganov, A., Noelle, S., and Petrova, G., "Semidiscrete Central-Upwind Schemes for Hyperbolic Conservation Laws and Hamilton–Jacobi Equations," *SIAM Journal on Scientific Computing*, Vol. 23, No. 3, 2001, pp. 707–740.

[69] Lorrain, P. M. C., *Flow Structure/Chemistry Coupling in the Ignition Process in Shock-Induced-Combustion Scramjets*, Ph.D. thesis, University of Queensland, 2014.

[70] Gardner, A. D., Paull, A., and McIntyre, T. J., "Upstream Porthole Injection in a 2-D Scramjet Model," *Shock Waves*, Vol. 11, No. 5, 2002, pp. 369–375.

[71] McGuire, J., *Ignition Enhancement for Scramjet Combustion*, Ph.D. thesis, University of New South Wales, Australian Defence Force Academy, School of Aerospace and Mechanical Engineering, 2007.

[72] McBride, B. J., Gordon, S., and Reno, M. A., "Coefficients for Calculating Thermodynamic and Transport Properties of Individual Species," Tech. Rep. NASA TM-4513, National Aeronautics and Space Administration, Washington D.C., 1993.

[73] Sutherland, W., "LII. The viscosity of gases and molecular force," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Vol. 36, No. 223, 1893, pp. 507–531.

[74] Menter, F. R., "Two-equation eddy-viscosity turbulence models for engineering applications," *AIAA journal*, Vol. 32, No. 8, 1994, pp. 1598–1605.

[75] Oberkampf, W. L., Trucano, T. G., and Hirsch, C., "Verification, validation, and predictive capability in computational engineering and physics," *Applied Mechanics Reviews*, Vol. 57, No. 5, 2004, pp. 345–384.

[76] Yahya, S., *Fundamentals of Compressible Flow: SI Units with Aircraft and Rocket Propulsion*, New Age International, 2003.

[77] Jahingir, M. N. and Huque, Z., "Design optimization of rocket-based combined-cycle inlet/ejector system," *Journal of propulsion and power*, Vol. 21, No. 4, 2005, pp. 650–655.

[78] Pastrone, D. and Sentinella, M. R., "Multi-objective optimization of rocket-based combined-cycle engine performance using a hybrid evolutionary algorithm," *Journal of Propulsion and Power*, Vol. 25, No. 5, 2009, pp. 1140–1145.

[79] Clough, J. A., *Modeling and Optimization of Turbine-Based Combined-Cycle Engine Performance*, Master's thesis, University of Maryland, 2004.

[80] Takahashi, T. T. and Gibson, G., *Multi-Disciplinary Design of a Rocket Engine Thrust Augmentation Ejector for Endoatmospheric Flight*, AIAA Aviation, American Institute of Aeronautics and Astronautics, 06 2014, doi:10.2514/6.2014-3091.

[81] Bayley, D. J., Hartfield, R. J., Burkhalter, J. E., and Jenkins, R. M., "Design optimization of a space launch vehicle using a genetic algorithm," *Journal of Spacecraft and Rockets*, Vol. 45, No. 4, 2008, pp. 733–740.

[82] Steele, S. C. W., *Optimal Engine Selection and Trajectory Optimization using Genetic Algorithms for Conceptual Design Optimization of Resuable Launch Vehicles*, Ph.D. thesis, Virginia Tech, 2015.

[83] Brock, M. and Franke, M., "Two-Stage-To-Orbit Reusable Launch Vehicle Propulsion Performance Study," *40th Joint Propulsion Conference and Exhibit, AIAA Paper*, Vol. 3903, 2004, p. 2004.

[84] Hank, J. M., Franke, M. E., and Eklund, D. R., "TSTO Reusable Launch Vehicles Using Airbreathing Propulsion," *42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Vol. 4962, 2006.

[85] Willis, M., Hiden, H., Marenbach, P., McKay, B., and Montague, G., "Genetic programming: an introduction and survey of applications," *Genetic Algorithms in Engineering Systems: Innovations and Applications, 1997. GALESIA 97. Second International Conference On (Conf. Publ. No. 446)*, Sep 1997, pp. 314–319.

[86] Gray, G. J., Li, Y., Murray-Smith, D., and Sharman, K., "Structural system identification using genetic programming and a block diagram oriented simulation tool," *Electronics Letters*, Vol. 32, No. 15, 1996, pp. 1422–1424.

[87] Marenbach, P., Bettenhausen, K. D., and Freyer, S., "Signal path oriented approach for generation of dynamic process models," *Proceedings of the First Annual Conference on Genetic Programming*, MIT Press, 1996, pp. 327–332.

[88] Marko, K. A. and Hampo, R. J., "Application of genetic programming to control of vehicle systems," *Intelligent Vehicles' 92 Symposium., Proceedings of the*, IEEE, 1992, pp. 191–195.

[89] Koza, J. R., Bennett III, F. H., Andre, D., and Keane, M. A., "Automated WYWIWYG design of both the topology and component values of electrical circuits using genetic programming," *Proceedings of the First Annual Conference on Genetic Programming*, MIT Press, 1996, pp. 123–131.

[90] Lohn, J. D., Hornby, G. S., and Linden, D. S., "An evolved antenna for deployment on nasas space technology 5 mission," *Genetic Programming Theory and Practice II*, Springer, 2005, pp. 301–315.

[91] Haynes, T. D., Schoenefeld, D. A., and Wainwright, R. L., "Type inheritance in strongly typed genetic programming," *Advances in genetic programming*, Vol. 2, No. 2, 1996, pp. 359–376.

[92] Fraser, A. and Weinbrenner, T., "Genetic Programming C++ Class Library," [online], `http://www0.cs.ucl.ac.uk/staff/W.Langdon/ftp/weinbenner/gp.html` [retrieved 08 April 2015].