UNIVERSITY „POLITEHNICA” of BUCHAREST
DOCTORAL SCHOOL OF BIOTECHNICAL SYSTEMS ENGINEERING

PHD THESIS SUMMARY

Convex optimisation for reusable demonstrators with space applications

Author: Eng. Ana-Maria NECULĂESCU
PhD supervisor: Prof. dr. eng. Ion STROE

DOCTORAL COMMITTEE

President	Prof dr. eng. Andrei CRAIFALEANU	from	University POLITEHNICA of Bucharest
PhD supervisor	Prof. dr. eng. CS I Ion STROE	from	University POLITEHNICA of Bucharest
Reviewer	CS I dr. phys. Marius Ioan PISO	from	Romanian Space Agency
Reviewer	CS I dr. eng. Cătălin NAE	from	National Institute for Aerospace Research "Elie Carafoli" - INCAS
Reviewer	Prof. dr. eng. Adrian-Mihail STOICA	from	University POLITEHNICA of Bucharest

Bucharest
2023

Table of contents

1 Introduction

The PhD thesis proposes a solution for solving the real-time trajectory optimisation problem using convex techniques for a reusable space demonstrator by developing and testing a real-time application (Software Experimental Payload – SEP). The testing and validation process for the developed application is realized using a vertical take-off and landing demonstrator, called DTV, developed by the National Institute for Aerospace Research "Elie Carafoli - INCAS.

The need and interest for the research theme is strongly related to the context of increased worldwide interest in the landing, recovery, and reuse of first stages of orbital launchers with the purpose of reducing the space mission cost and ease the access of outer space. This research topic is also of interest in the context of the efforts for the colonization of Moon and Mars for which this technology has high applicability.

The results achieved and presented in the PhD thesis demonstrate the feasibility of using the developed application that would be executed in real-time on-board space vehicles for the trajectory optimisation process.

Keywords: onboard trajectory optimisation, space demonstrators, reusability, convex techniques, automatic code generation, real-time testing, in-flight validation

1.1 Overview

The first chapter includes an introduction in the field of online optimisation of the landing trajectory for a reusable space demonstrator, presenting the current national, European and international context, the main objective of the PhD thesis, together with the current state of art in the chosen field. The second chapter presents introductive information related to the convex optimisation of vertical take-off and landing vehicles with space applications, while the third chapter describes the mathematical model for minimizing the amount of fuel required for the landing phase of the reusable space demonstrators. The mathematical formulation is presented along with the discretization method for obtaining a numerical solution.

The fourth chapter presents the architecture and testing methodology of the online optimisation application for the vehicle trajectory, an application that was developed and presented in detail in the PhD thesis. The fifth chapter includes technical details about the demonstrator that was used as in-flight testing and validation infrastructure for the SEP application developed.

Chapter six presents the implementation in the Matlab simulation environment of the mathematical model presented in detail in chapter three and the use of the integrated CVX module. A nominal case study was selected for the testing process of the developed SEP application. The seventh chapter showcases the implementation process of the mathematical model in the convex solver generator, CVXGEN. Details about CVXGEN, how to use the generated C code, and additional functions developed for using the generated code are provided in this chapter. Numerical results for the same nominal case are presented, along with a Monte Carlo analysis where numerous simulations are performed in the presence of uncertainties.

The eighth chapter presents details related to the 6DOF dynamic model for the DTV demonstrator, the integration process of the S-function in the six-degree-of-freedom DTV simulator, together with numerical results for the selected nominal mission. Chapter nine covers the real-time testing architecture of the developed SEP application, with numerical results obtained for the nominal mission being presented in detail.

The tenth chapter includes the results of the numerical simulations necessary for the preparation of an experimental flight campaign, together with the results recorded for the experimental missions in captive flight, ME#1 and ME#2. The first experimental mission had the purpose of validating the entire development process, while ME#2 represents a short lateral manoeuvre, for which four experimental flights were carried out.

The conclusions of the PhD thesis are presented in chapter eleven, emphasizing the contributions of the thesis, the results achieved, and the prospects for further development. A list of scientific papers published during the PhD Program is also presented as well as selected bibliographic notes used in the elaboration of the present work.

1.2 Context

Within this subchapter, the worldwide novelties in the field of reusable space vehicles are presented, outlining the following aspects:

- Until December 2015, launch vehicles were expandable, meaning they were used for a single launch only. The main stage and boosters that made up the space launch vehicle were discarded into the ocean, and the upper stage remained in orbit around Earth or directed on a re-entry trajectory;
- This scenario was completely changed in December 2015 by the success of the private company, Space Exploration Technologies Corporation (SpaceX) by vertically landing the first stage of the *Falcon 9* launcher [1]. The concept of reusability for space launch systems has also been demonstrated by another private company in the United States, Blue Origin, by successfully completing a suborbital flight with the *New Shepard* launcher [2], and also by the private company based in China, Deep Blue Aerospace [3];
- Europe has also taken the first steps in the development of reusable space launch systems, through the *Callisto* and *Themis* demonstrators [4];
- At regional level, in Romania, a reusable demonstrator with vertical take-off and landing capabilities (VTVL) powered by a rocket engine called *Ascent and Descent Autonomous Maneuverable Platform (ADAMP)* is in the development phase in an European project led by the National Institute for Aerospace Research "Elie Carafoli" - INCAS [5];
- The reuse of launch systems massively reduces the cost of a launch, ultimately leading to easier access to outer space. NASA estimates that a commercial launch to the International Space Station (ISS) costs 4 times less today than it did 20 years ago [6] due to technological advances.

The successful recovery and reuse of Falcon 9 launcher first stage is due to several technologies, but the most relevant in the context of this PhD thesis is the online trajectory optimisation onboard the launcher and the autonomous control during the vertical landing phase.

1.3 Thesis objectives

The PhD thesis presents the mathematical model for solving the problem of minimizing the fuel consumption of a space-application reusable demonstrator for the vertical landing phase and details the implementation of an algorithm which runs in real time on its onboard computer. The main objective of the thesis is to develop an application and propose a methodology that optimises the real-time landing trajectory for a reusable demonstrator with space applications, using a commercially licensed software, CVXGEN, that generates C code for solving convex problems. The real-time application, called SEP, is implemented, tested and validated in-flight using a vertical take-off and landing demonstrator as a test platform, based on a 1kN-class turbo-jet engine.

1.4 Current state of art

The PhD thesis approaches in an innovative way the solving process of the landing problem for retropropulsive vertical take-off and landing vehicles using convex real-time optimal control techniques, on the onboard computer of a reusable demonstrator. The problem addressed can be broken down into two main research directions:

- Optimal control problem using convex techniques for the trajectory generation of a reusable vehicle with space applications;
- Real-time algorithms for solving this problem.

1.4.1 Mathematical modelling of the convex trajectory optimisation problem

Successfully landing a space launcher represents one of the most important technological challenges for the development of trajectory and control algorithms. These challenges are due to unpredictable flight conditions, significant dispersions and uncertainties associated with other parameters, in addition to known constraints related to vehicle control and structural sizing. Considering these constraints and adding other rigorous requirements necessary to obtain a high-accuracy solution for the landing manoeuvre, the fixed-point landing problem for space vehicles becomes extremely difficult to solve.

Trajectory generation for space vehicles can be expressed as an optimal control problem, which leads to the need to solve a problem that has the following form, [7]:

$$\min_{u,p,t_f} J\left(x,u,p,t_f\right) \tag{1.1}$$

having the following constraints:

$$\dot{x}(t) = f(x,u,p,t) \tag{1.2}$$

$$\left(x(t),u(t),p,t_f\right) \in C(t), \forall t \in \left[0,t_f\right] \tag{1.3}$$

$$(x(t),p) \in X_0, \left(x\left(t_f\right),p\right) \in X_f \tag{1.4}$$

The objective function (1.1) defines the mission target, the system dynamics is modelled by differential equations that are imposed as constraints (1.2), the system states and control are imposed by (1.3), while the boundary conditions are imposed by (1.4).

Various discretization methods are known for an optimal control problem. Once the mathematical problem is discretized, numerical optimisation methods can be used to obtain a solution. At this point, solving real-time problems with applications in the space domain and beyond is the major difficulty. Depending on the mathematical formulation of the problem and the selected discretization techniques, the optimisation problem can become a nonlinear (and non-convex) optimal control problem, which is called a *nonlinear optimisation problem* (NLP) [8], [9], [10]. The big disadvantage of the NLP problem is that it requires very high computational resources and that there is no guarantee of obtaining the optimal solution. For these reasons, NLP methods are not suitable for ensuring the existence of a real-time optimal trajectory for autonomous space vehicles.

To achieve this goal, two main methods are distinguished:

- *Lossless Convexification* (LCvx) [11], [12], [13];
- *Sequential Convex Programming* (SCP) [14], [15], [16], [17].

1.4.2 Real-time algorithms for convex optimisation problems

Computing power requirements are the second consideration for onboard vehicle trajectory generation. Historically, this has been the most important factor in the development of algorithms with practical applications. In the field of real-time optimisation on a vehicle's onboard computer (*real-time embedded optimisation*) [18], [19], [20], [21], different instances of a problem must be calculated extremely fast (in the order of milliseconds or microseconds), and in most cases the result must be obtained before a strict execution deadline. This technical requirement is in contradiction with generic algorithms, which need a longer execution time and the solution of the problem is available when a certain accuracy has been achieved [22].

CVXGEN is a code generator for families of convex optimisation problems that can be reduced to *Quadratic Programming* (QP) and *Second Order Cone Programming* (SOCP). CVXGEN starts from the description of the family of convex optimisation problems and automatically generates C code. For small and medium-sized problems (with up to several hundred optimisation variables), CVXGEN generates C Code that solve these problems in a few microseconds or milliseconds, [23].

For the work performed in this PhD thesis, CVXGEN was used as code generator, for which the author received an academic license thanks to Mr. Jacob Mattingley.

2 Convex optimisation for vertical take-off and landing vehicles with space applications

In this chapter, a comparative presentation of the best-known methods for online trajectory generation onboard reusable vehicles by using convex optimisation methods is made. Convex optimisation techniques aim to minimize a ***convex objective function*** while satisfying a ***set of convex constraints***. Some theoretical elements for convex sets and convex functions used in this PhD thesis are detailed.

3 Minimizing the amount of fuel required for the landing manoeuvre

3.1 Mathematical formulation of the problem

The reference system chosen is an inertial one and related to the Earth's surface. The origin of the system is at the centre of the launch site, the positive direction of the x-axis is up, the positive direction of the y-axis is towards the East, and the positive direction of the z-axis is towards the North, how it is now illustrated in Figure 3.1. The trihedral is hereafter referred to as the "*launch pad frame*".

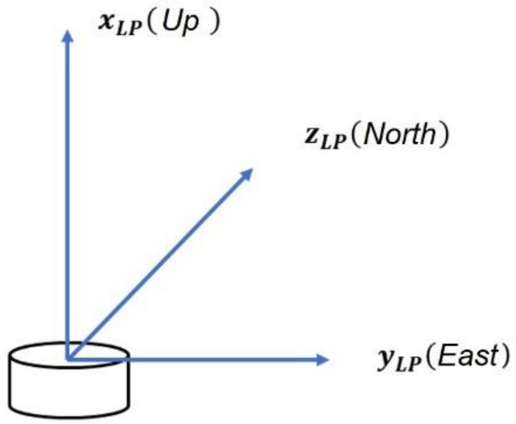


Figure 3.1 Reference system used

To generate the online trajectory onboard a reusable vehicle in such a way as to minimize the amount of fuel required in case of a precise landing at a fixed point, the problem can be formulated in the following way [24], [25] ("powered descent minimum fuel trajectory optimisation problem"):

$$\max_{t_f, T_c(\cdot)} m(t_f) = \min_{t_f, T_c(\cdot)} \int_0^{t_f} \|T_c(t)\| dt \tag{3.1}$$

subject to:

$$\ddot{r}(t) = g + \frac{T_c(t)}{m(t)} \tag{3.2}$$

$$\dot{m}(t) = -\alpha \|T_c(t)\| \tag{3.3}$$

$$0 < \rho_1 \leq \|T_c(t)\| \leq \rho_2 \tag{3.4}$$

$$\|r_d(t)\| \leq \beta r_v(t) \tag{3.5}$$

The initial conditions of problem (3.1) are:

$$m(0) = m_{wet} \tag{3.6}$$

$$m(t_f) \geq 0 \tag{3.7}$$

$$r(0) = r_0 \tag{3.8}$$

$$\dot{r}(0) = \dot{r_0} \tag{3.9}$$

and the final conditions are:

$$r(t_f) = 0 \tag{3.10}$$

$$\dot{r}(t_f) = 0 \tag{3.11}$$

To simplify the notations, the parameters α, ρ_1, ρ_2, were introduced, being defined as follows:

$$\alpha = \frac{1}{I_{sp} g_e \cos\Phi} \tag{3.12}$$

$$\rho_1 = nT_1 \cos\Phi \tag{3.13}$$

$$\rho_2 = nT_2 \cos\Phi \tag{3.14}$$

In formulating the problem, the following notations were used:

- Φ - thrusters cant angle;
- n - number of thrusters;
- I_{sp} - specific impulse;
- m_{wet} - vehicle initial mass (dry mass + fuel);
- $g \in \mathbb{R}^3$ - Earth's gravitational acceleration vector;
- T_1 - available thrust lower bound (for each thruster);
- T_2 - available thrust upper bound (for each thruster);
- $r_0 \in \mathbb{R}^3$ - vehicle initial position vector;
- $\dot{r_0} \in \mathbb{R}^3$ - vehicle initial velocity vector;
- $\beta = \tan\gamma$, with $\gamma \in \left(0, \frac{\pi}{2}\right)$ being the glide slope angle;
- $r_v \in \mathbb{R}$ - vehicle altitude;
- $r_d \in \mathbb{R}^2$ - vehicle lateral position vector („downrange position vector”) such as $r = [r_v, r_d^T]^T$.

Note that in formulation (3.1), the inequality (3.4) defines a non-convex type of constraint on the command channel. For this reason, a relaxation of the formulation (3.1) is necessary, transforming the problem now in one of convex optimisation [25]:

$$\min_{t_f, T_c(\cdot), \Gamma(\cdot)} \int_0^{t_f} \Gamma(t)\, dt \tag{3.15}$$

subject to:

$$\ddot{r}(t) = g + \frac{T_c(t)}{m(t)} \tag{3.16}$$

$$\dot{m}(t) = -\alpha\Gamma(t) \tag{3.17}$$

$$\|T_c(t)\| \leq \Gamma(t) \tag{3.18}$$

$$0 < \rho_1 \leq \Gamma(t) \leq \rho_2 \tag{3.19}$$

$$\|r_d(t)\| \leq \beta r_v(t) \tag{3.20}$$

The initial and final conditions (3.15) remain the same as the one presented earlier in (3.6) - (3.9), and in (3.10) - (3.11).

If a solution of the problem (3.15) is of the form $\langle t_f^*, T_c^*(\cdot), \Gamma^*(\cdot)\rangle$, then it can be considered that $\langle t_f^*, T_c^*(\cdot)\rangle$ is a solution for problem (3.1) and $\|T_c{}^*(t)\| = \rho_1$ or $\|T_c{}^*(t)\| = \rho_2$ for $t \in [0, t_f^*]$. Thus, formulation (3.15) convexifies the problematic constraint (3.4) of formulation (3.1) by using two equivalent constraints (3.18) and (3.19).

By successfully applying changes of variables in advantageous ways ($u = \frac{T_c}{m}$, $\sigma = \frac{\Gamma}{m}$ and $z = \ln m$) and using Taylor series expansions to bound the σ term (the first 3 terms are kept for the lower limit and the first 2 terms for the upper limit), the problem of minimizing the amount of fuel required for a precise landing can be rewritten again, obtaining the following final formulation [25]:

$$\min_{t_f, u(\cdot), \sigma(\cdot)} \int_0^{t_f} \sigma(t) dt \tag{3.21}$$

subject to:

$$\ddot{r}(t) = g + u(t) \tag{3.22}$$

$$\dot{z}(t) = -\alpha\sigma(t) \tag{3.23}$$

$$\|u(t)\| \leq \sigma(t) \tag{3.24}$$

$$\mu_1(t)\left[1 - \big(z(t) - z_0(t)\big) + \frac{\big(z(t) - z_0(t)\big)^2}{2}\right] \leq \sigma(t) \leq \mu_2(t)\big[1 - \big(z(t) - z_0(t)\big)\big] \tag{3.25}$$

$$z_0(t) \leq z(t) \leq \ln(m_{wet} - \alpha\rho_1 t) \tag{3.26}$$

$$\|r_d(t)\| \leq \beta r_v(t) \tag{3.27}$$

where:

$$\mu_1(t) = \rho_1 e^{-z_0(t)} \tag{3.28}$$

$$\mu_2(t) = \rho_2 e^{-z_0(t)} \tag{3.29}$$

$$z_0(t) = ln(m_{wet} - \alpha\rho_2 t) \tag{3.30}$$

The initial conditions of the problem (3.21) are:

$$z(0) = ln\, m_{wet} \tag{3.31}$$

$$r(0) = r_0 \tag{3.32}$$

$$\dot{r}(0) = \dot{r_0} \tag{3.33}$$

and the final conditions are:

$$r(t_f) = 0 \tag{3.34}$$

$$\dot{r}(t_f) = 0 \tag{3.35}$$

For a given time of flight t_f, using formulation (3.21), all the equality and inequality constraints define convex feasible regions in the state and control space, also having a convex cost (objective) function. Therefore, the problem (3.21) can be converted into a finite-dimensional convex optimisation problem, the most efficient solution method being SOCP (Second Order Cone Programming), after an appropriate discretization of it.

3.2 Discretization of the problem

In this chapter, the discretization method of the problem to be solved (3.21) is presented:

$$\underset{u_0,\ldots,u_N,\, \sigma_0,\ldots,\sigma_N,}{min} -z_n \tag{3.36}$$

subject to:

$$r_{k+1} = r_k + \frac{\Delta t}{2}(\dot{r}_k + \dot{r}_{k+1}) + \frac{\Delta t}{12}(u_k - u_{k+1}) \tag{3.37}$$

$$\dot{r}_{k+1} = \dot{r}_k + \frac{\Delta t}{2}(u_k + u_{k+1}) + g\Delta t \tag{3.38}$$

$$z_{k+1} = z_k - \frac{\alpha\Delta t}{2}(\sigma_k + \sigma_{k+1}) \tag{3.39}$$

$$\|u_k\| \leq \sigma_k \tag{3.40}$$

$$\mu_{1,k}\left[1 - (z_k - z_{0,k}) + \frac{(z_k - z_{0,k})^2}{2}\right] \leq \sigma_k \leq \mu_{2,k}[1 - (z_k - z_{0,k})] \tag{3.41}$$

$$z_{0,k} \leq z_k \leq \ln(m_{wet} - \alpha\rho_1 t) \tag{3.42}$$

$$\|r_{d_k}\| \leq \beta r_{v_k} \tag{3.43}$$

with $k = 0, 1 \dots N$, where:

$$\mu_{1,k} = \rho_1 e^{-z_{0,k}} \tag{3.44}$$

$$\mu_{2,k} = \rho_2 e^{-z_{0,k}} \tag{3.45}$$

$$z_{0,k} = \ln(m_{wet} - \alpha\rho_2 k\Delta t) \tag{3.46}$$

The initial conditions of problem (3.36) are:

$$z_0 = \ln m_{wet} \tag{3.47}$$

$$r(0) = r_0 \tag{3.48}$$

and the final conditions are:

$$t_f = N\Delta t \tag{3.49}$$

$$r_N = 0 \tag{3.50}$$

$$\dot{r}_N = 0 \tag{3.51}$$

4 Testing architecture of the real-time application (SEP)

During the PhD Program, a real-time application was developed for the onboard generation of an optimal trajectory, together with an integrated way of testing it (more details in Figure 4.1) on a demonstrator called DTV [26], which is developed by the National Institute for Aerospace Research "Elie Carafoli" - INCAS.

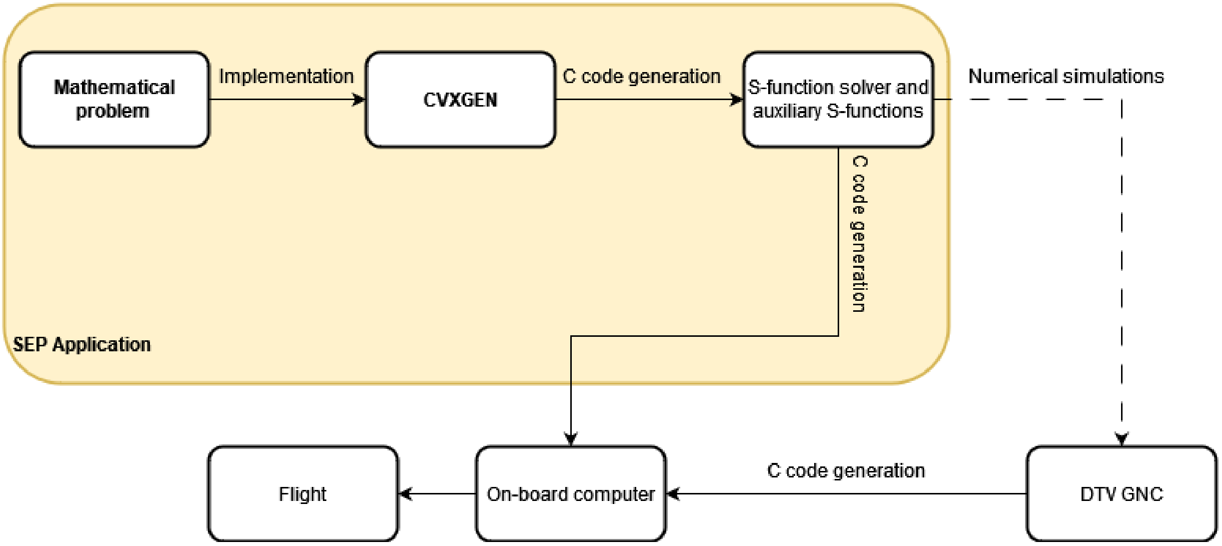


Figure 4.1 SEP application and its integration

The starting point is the mathematical model of the convex optimisation problem which minimizes the fuel consumption for the landing phase. Once the mathematical problem is written and discretized, the implementation phase in the convex solver generator for this family of mathematical problems follows. After the numerical problem has been implemented in CVXGEN, the next step is to generate code in the C programming language, resulting in several files that were further used to generate the S-function, along with other relevant additional functions. This S-function, which contains the online solver for onboard trajectory generation, is further integrated into Simulink, in the Guidance, Navigation and Control (GNC) subsystem of the demonstrator. This integration in the GNC module of the DTV vehicle allows the SEP application to be tested offline using a six degrees of freedom dynamic model of the vehicle.

DTV is a testing and validation platform for software products and hardware equipment, and INCAS has developed this functionality, to integrate experimental software in an additional GNC block named Experimental Payloads (EP) block, providing to partners the necessary interfaces that are to be taken into account for the correct integration. After integrating the S-function into the EP block of the demonstrator and passing key tests using the six-degree-of-freedom simulator, C code is automatically generated again using Matlab/Simulink functionality. C code is generated for a real-time testing generic target ("*Generic Real-Time Target*"), which is further integrated into the DTV flight software, being called every 100 milliseconds and running as a standalone application on the onboard computer, unlike the application for the GNC system which is called every 40 milliseconds.

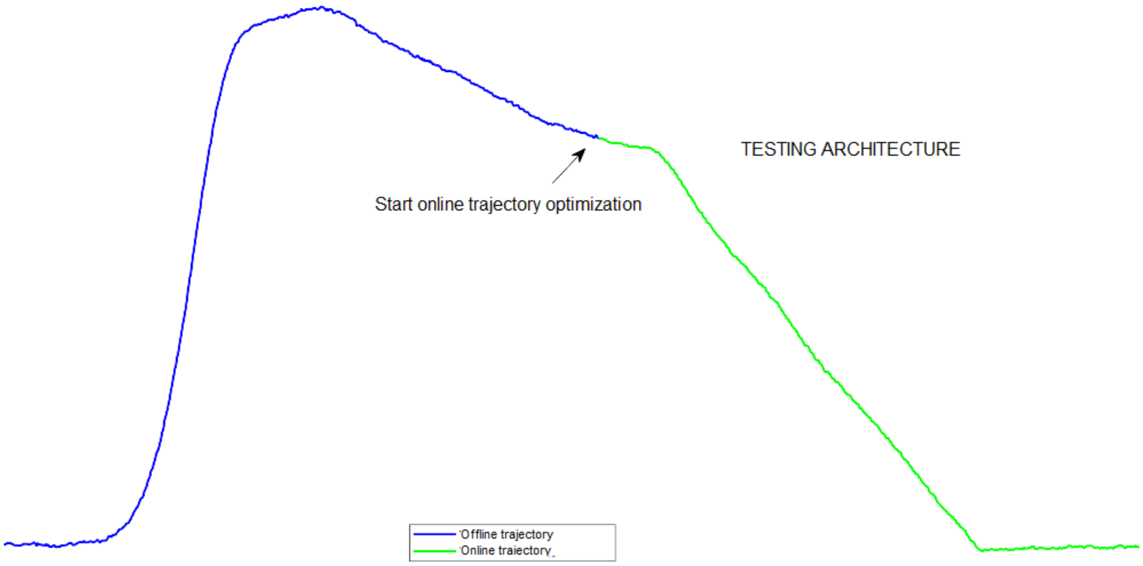


Figure 4.2 Test architecture of the online trajectory optimiser

In order to in-flight test the online trajectory optimisation application while preserving the integrity of the DTV vehicle, the following test procedure (details in Figure 4.2) has been adopted:

- DTV takes off following a predefined reference trajectory;
- After the ascent phase ends (at predefined conditions), DTV goes into hover above the take-off point;
- After 15 seconds of hover manoeuvre, the online trajectory optimisation program starts running;
- When the transition from the pre-programmed flight phase to the experimental trajectory generated online is commanded, the execution of the optimisation program starts based on the information from the navigation system by acquiring data from the GNC system (position and velocity) and generates an optimal onboard trajectory in the allocated time window of 100 milliseconds (a table containing three reference positions and three velocities for each discretization point);
- Together with the online generation of this optimal trajectory, the program sends information related to the status of the optimisation process, information determined by the means of the S-function;
- If the online optimisation process is completed successfully, the generated trajectory is sent to the vehicle as the new guidance reference;
- The demonstrator guidance is now the online optimised trajectory and the remaining part of the flight is switched to the experimental phase.

5 Presentation of the research, testing and in-flight validation infrastructure for the SEP application

In order to increase the level of maturity of the technology proposed in this PhD thesis, specifically the developed SEP application, it is necessary to test and validate it in real-time using an existing flight test infrastructure. The National Institute for Aerospace Research "Elie Carafoli" - INCAS developed DTV, a unique infrastructure in an European project financed by the European Space Agency (ESA) and with the support of the Romanian Space Agency (ROSA), within the Demonstrator for Technologies Validation project.

As is obvious from the title of the project, the DTV demonstrator (showcased in Figure 5.1) provides the opportunity for in-flight validation of both software and hardware technologies via experimental flights. The DTV is a reusable demonstrator powered by a 1kN class turbojet engine with vertical take-off and landing capabilities, being fully autonomous.

Figure 5.1 DTV test platform - configuration. Credit: INCAS/ESA

In order to carry out the experimental campaigns, INCAS management approved the use of its research infrastructure (both ground infrastructure - from INCAS Măneciu, and in-flight research infrastructure - the DTV demonstrator). Also, the management of INCAS agreed that the Flight Simulator (protected model) can be used in this PhD thesis for the necessary technical links. This allows the use of the infrastructure without having access to the implemented mathematical models (dynamic model in six degrees of freedom, control architecture and sensor’s models). INCAS has prepared a protected model of the Flight Simulator, a model which is distributed to INCAS partners for interfacing software programs developed by them in various research projects.

6 Implementation of the mathematical problem in Matlab/CVX

The mathematical problem in discrete form presented in chapter 3, was implemented in Matlab using the convex optimisation module CVX. CVX transforms Matlab into a modelling language, allowing the objective function and constraints to be specified using standard Matlab syntaxes. The implementation was studied for the nominal case selected in the PhD thesis.

The implementation of the mathematical model for minimizing the amount of fuel in the CVX module in Matlab and its solution using the SeDumi solver was necessary to verify the existence of a numerical solution, and after performing the numerical simulations, the required computational time was 2.16 seconds, which means the implementation **is not suitable** for real-time applications. Thus, the decision to implement the mathematical model in a C code generator, CVXGEN, to transform the mathematical problem into a solver that would allow solving the class of problems studied in a computational time of the order of milliseconds.

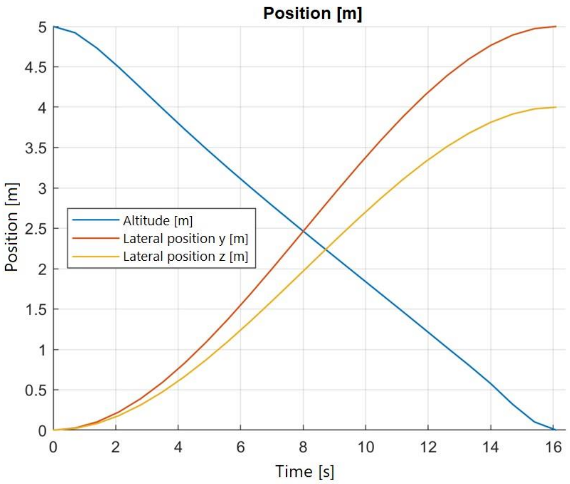


Figure 6.1 DTV position vs. time, nominal case

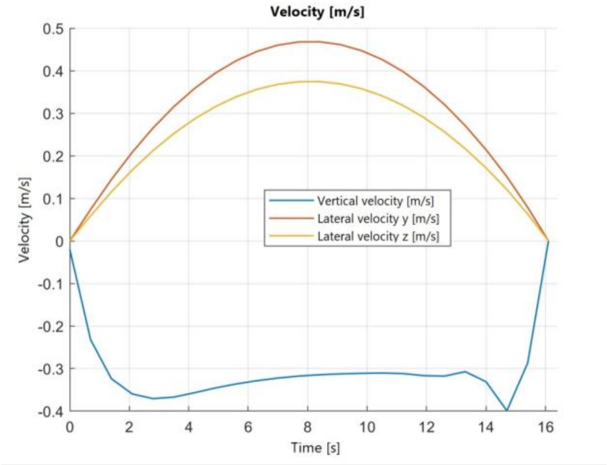


Figure 6.2 DTV velocity vs. time, nominal case

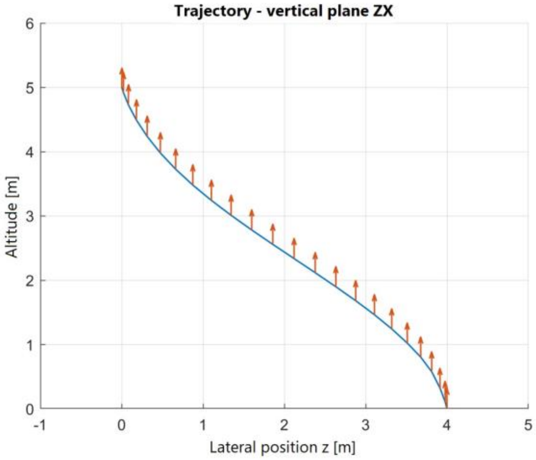


Figure 6.3 ZX Trajectory vs. time, nominal case

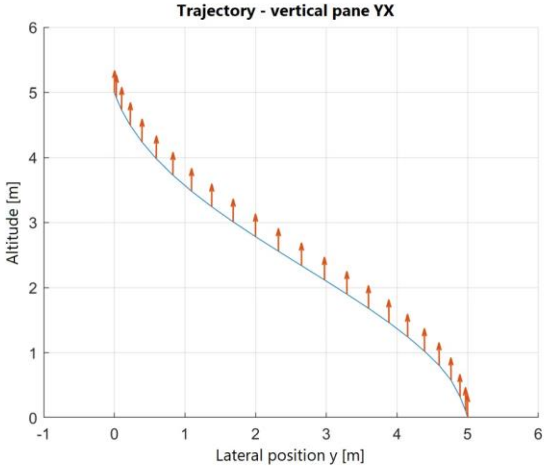


Figure 6.4 YX Trajectory vs. time, nominal case

7 Implementation of the mathematical problem in the CVXGEN code generator

7.1 Description and use of the CVXGEN package

The mathematical problem described in the chapter 3, minimization of the amount of fuel required for a safe landing, was implemented in CVXGEN [27]. CVXGEN generates C code for convex optimisation problems using an online interface and does not require the installation of any additional programs. This code generator turns a convex mathematical problem into an extremely fast solver.

To use CVXGEN, the following steps are required:

- Step 1. Mathematical modelling of the problem and its transformation into a convex problem;
- Step 2. Automatic generation of a solver dedicated to the class of problems modelled in the previous step (step 1);
- Step 3. Obtaining a solution extremely quickly.

7.2 Description of the generated code

The mathematical problem described in the chapter 3 was implemented in CVXGEN, where the dimensions of the problem, parameters, optimisation variables, objective function are set and the constraints are imposed. In CVXGEN, no numerical values are imposed for parameters or optimisation variables (only their sizes), which is why the generated code can be used to obtain an optimal solution onboard by reading the input data in real time and outputting as a result, a valid trajectory based on those readings.

After the problem is implemented in CVXGEN, C code is then generated, CVXGEN further generating five primary files in the C programming language. The algorithm to solve the mathematical problem is provided in the "*solver.c*" file which contains the main function "*solve*" and the basic routines. The KKT matrix factorization and problem are solved by functions in "*ldl.c*", while "*matrix_support.c*" contains C code for matrix and vector operations. Data structures and function prototypes are set in "*solver.h*" and "*testsolver.c*" contains an example for testing the generated code. In "*util.c*" there are additional functions for testing, while "*Makefile*" is required for automatic compilation.

7.3 Use of the code generated by CVXGEN

Although CVXGEN offers the option of individually testing the generated code, in practical engineering applications this is usually done after integration into more complex code packages. For the current case, it is necessary to integrate the generated C code into the flight application of the DTV demonstrator ("flight software") to ensure the possibility of testing the SEP application in real time. In this thesis, the SEP application has been integrated into the flight software as a standalone application and is called every 100 milliseconds. On the onboard computer, the GNC application is called every 40 milliseconds.

7.4 Auxiliary functions

This subchapter presents the auxiliary functions required for the integration of the CVXGEN generated files.

7.5 S-function generation and offline testing

S-functions represent functions of a system which provide very important mechanisms for extending Simulink capabilities. The implementation of an algorithm written in the C programming language can be encapsulated in an S-function and then used in the Matlab/Simulink development environment. S-functions are highly used in industry because they can integrate external C code to perform offline simulations and represent diagrams from which C code can be generated using an automatic C code generator (for example Simulink Coder). Matlab's "Legacy Tool" functionality was used to generate the S-function containing the C code generated by CVXGEN and the additional required auxiliary functions.

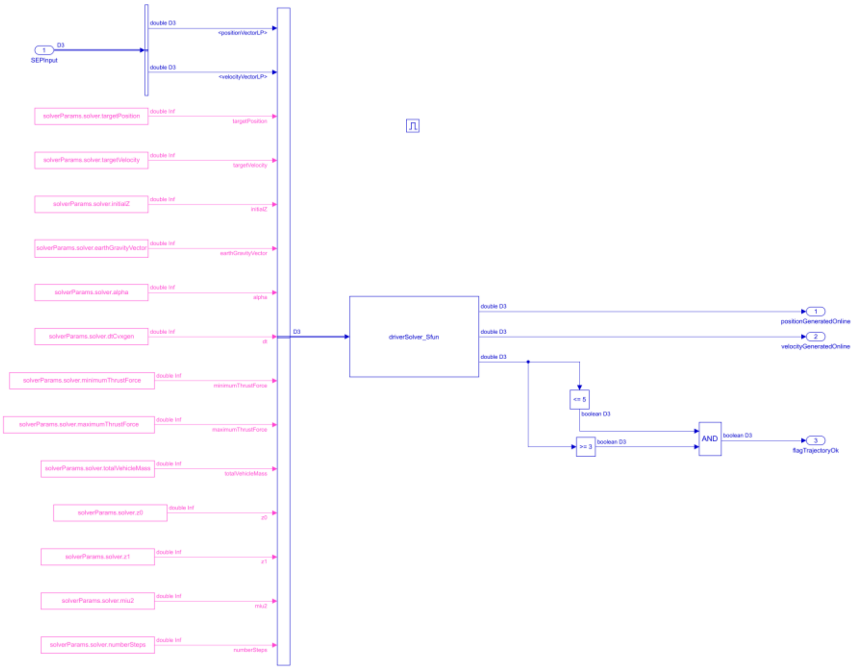


Figure 7.1 Testing the S-function generated in Simulink

The output data of the S-function, which contains the solver that will run on the onboard computer, are the following: *Position generated online*: a matrix of 3 rows and 24 columns; the number of rows represents the position on each of the 3 axes, while the number of columns is the preselected number of steps; *Online generated velocity*: a matrix of 3 rows and 24 columns; the number of rows represents the velocity on each of the 3 axes, while the number of columns is the preselected number of steps; *Minimum value of the objective function*.

In order to visualize the trajectory and plot graphs as a function of time, an interpolation of the optimal values calculated by the convex solver is required. The processing of the raw output data was also done in Simulink.

7.6 Nominal mission

For testing the generated code, more specifically the developed real-time application containing the online DTV demonstrator trajectory optimisation solver, the same nominal mission was selected as the one presented in the chapter 6. The complete list of parameters is presented in Table 7.1 and Table 7.2.

Table 7.1 Fixed parameters numerical values, nominal case

Parameter	**Numerical value**
$\boldsymbol{m_i}$ [Kg]	64.5
$\boldsymbol{g_0}\left[\boldsymbol{\frac{m}{s^2}}\right]$	9.08665
$\boldsymbol{\alpha}\left[\frac{\mathbf{kg}}{\boldsymbol{s}}\right]$	0.00043
$\Delta \boldsymbol{t}\ [\boldsymbol{s}]$	0.7
$\boldsymbol{T_1}$ [N]	600
$\boldsymbol{T_2}$ [N]	750
$\boldsymbol{z_0}[-]$	4.1058
$\boldsymbol{z_1}[-]$	4.0967
$\boldsymbol{\mu_2}$	10.47
$\boldsymbol{N}$	24

Table 7.2 Initial and final conditions, nominal case

Parameter	**Trajectory initial conditions**	**Trajectory final conditions**
$\boldsymbol{r_x}$ [m]	5	0
$\boldsymbol{r_y}$ [m]	0	5
$\boldsymbol{r_z}$ [m]	0	4
$\boldsymbol{V_x}$ [m]	0	0
$\boldsymbol{V_y}$ [m]	0	0
$\boldsymbol{V_z}$ [m]	0	0

Using these input data and performing a numerical simulation in Matlab / Simulink, the numerical results presented in the following figures are obtained, thus testing the functionality of the S-function encapsulating the convex solver for the landing trajectory optimisation for DTV.

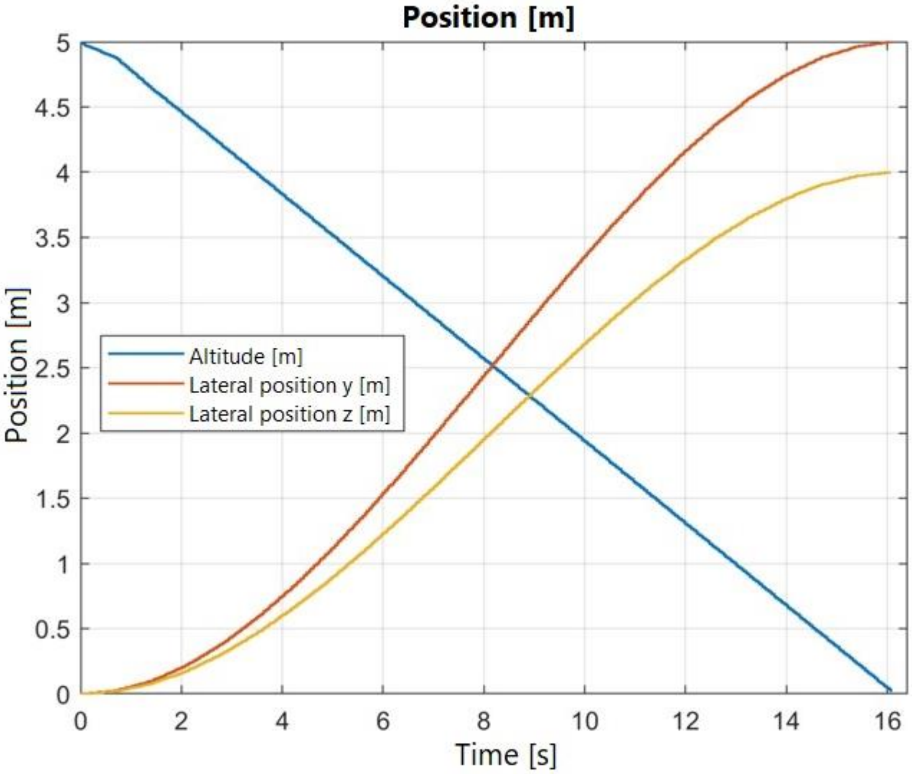


Figure 7.2 DTV position vs. time, nominal case

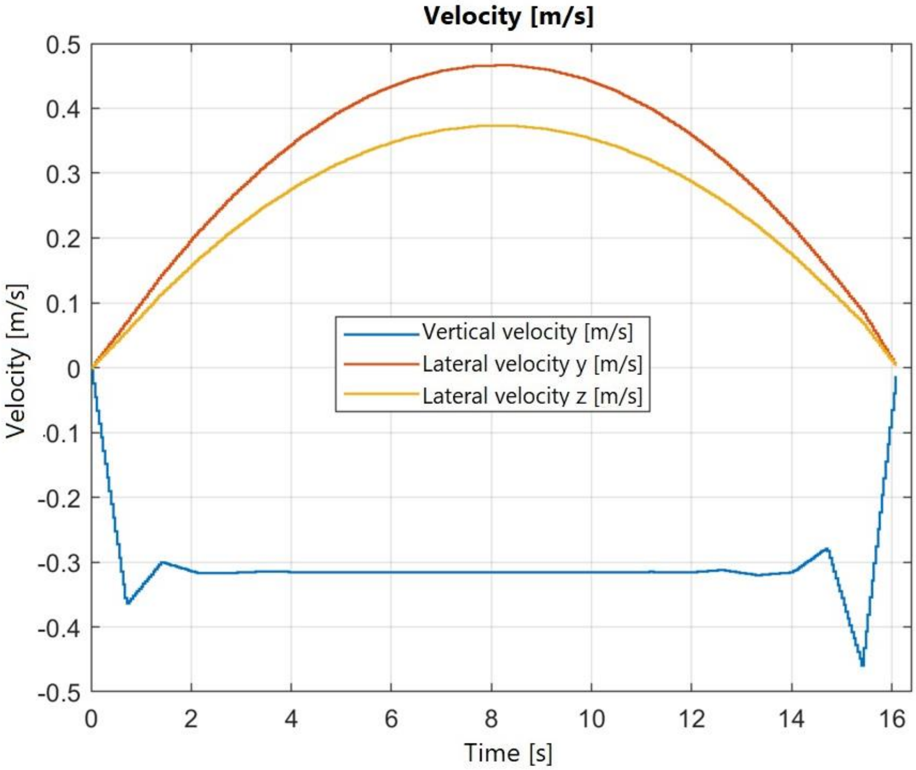


Figure 7.3 DTV velocity vs. time, nominal case

Figure 7.2 presents the DTV position on the three axes as a function of time. It can be observed that the initial and final conditions for the trajectory are respected. The experimental mission has an initial altitude of 5 m and the lateral positions are 0 on both axes. The final condition for the positions is also respected for each of the three axes.

Figure 7.3 presents the velocity of the demonstrator on the three axes as a function of time. It can also be seen that the initial and final conditions for the imposed velocities are met. Another observation is related to the maximum vehicle velocity, which do not exceed 0.5 m/s. This ensures that the selected nominal trajectory is a feasible one for the real capabilities of the DTV. In Figure 7.4 one can see the optimal 3D trajectory generated for the nominal case.

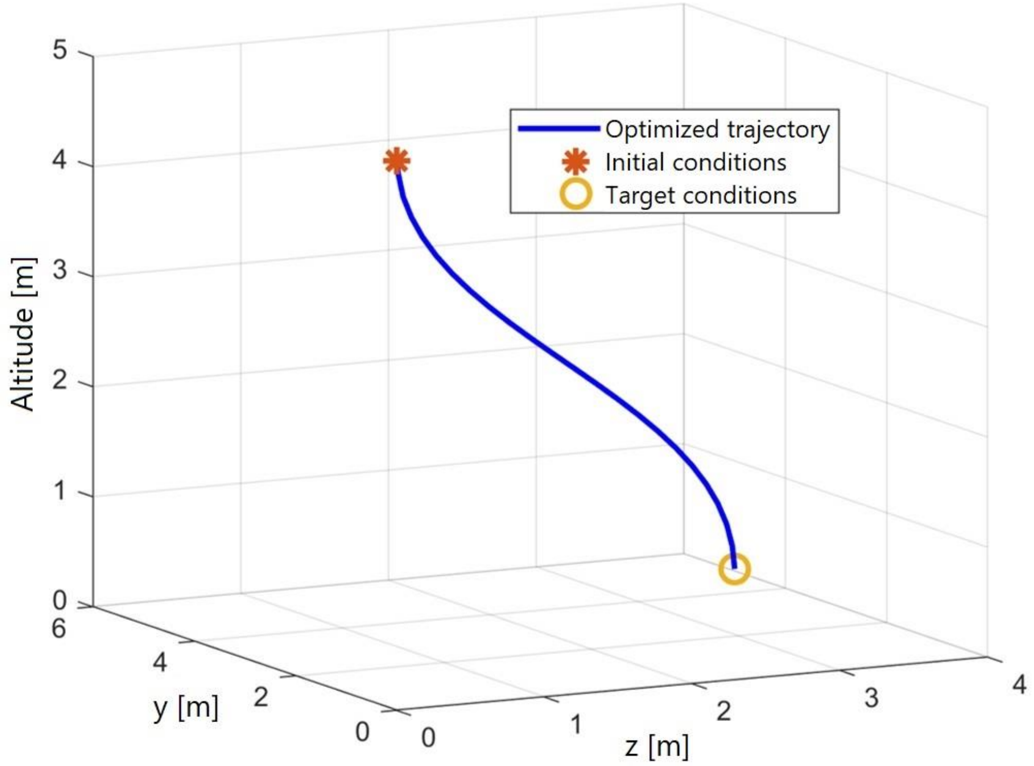


Figure 7.4 3D trajectory, nominal case

The numerical results presented in this chapter can also be compared with the ones presented in chapter 6, obtained by implementing the mathematical problem in Matlab, using the CVX module. Although the mathematical problem is the same and the input data are identical for the nominal mission, the test modality is different:

- For the numerical simulations using CVX, the mathematical problem is defined in Matlab and solved with the SeDumi solver (which is written in C code) and used in the Matlab environment via .mex files.
- For the numerical simulations using CVXGEN, the code is written in the C programming language, which makes it suitable for running in real time on a demonstrator onboard computer.

7.7 Monte Carlo analysis

To test the functionality of the developed and implemented real-time application, a Monte Carlo analysis in the presence of initial positions and velocities uncertainties was performed. The results obtained based on 1000 Monte Carlo simulations demonstrate that the convex solver always finds an optimal and feasible solution for the DTV demonstrator under the conditions considered.

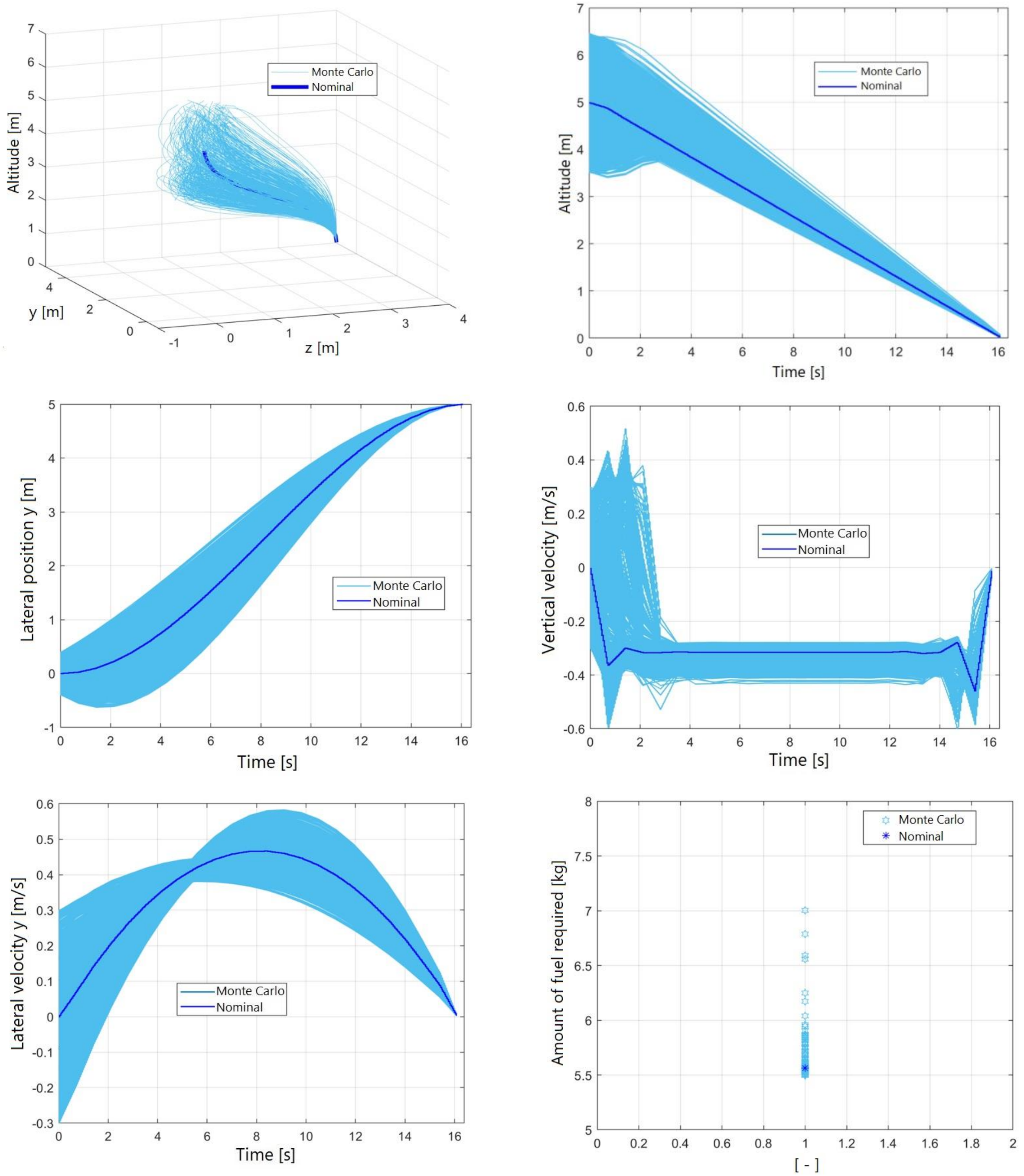


Figure 7.5 Monte Carlo simulations - results

8 Simulations using the six-degree-of-freedom (6DOF) dynamic model

8.1 Mathematical modelling of the DTV demonstrator

The Newtonian mechanics formulation is used to write the mathematical equations of the 6DOF dynamic model specific to the DTV demonstrator (variable mass body), by applying the general theorems of momentum (to describe the translational motion of the centre of mass) and angular momentum (to describe the movement around the centre of mass). Next, the dynamic model for the DTV vehicle is presented based on the general model detailed in [28].

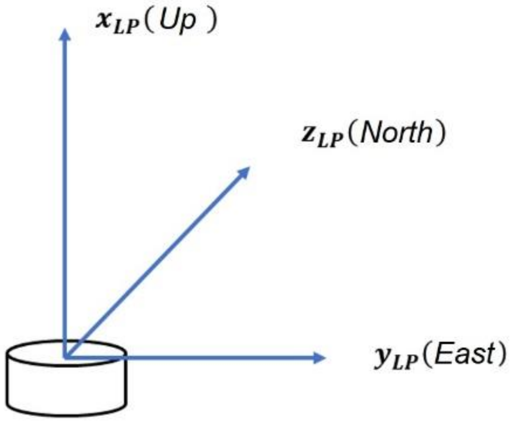


Figure 8.1 Inertial frame - launch pad frame

To describe the translation of the centre of mass, the *launch pad frame* (LP) is used (presented in Figure 8.1). To describe the rotational movement around the centre of mass, the reference system linked to the body (B - “*body frame”*) is used, being shown in Figure 8.2. The origin of the body system is in the vehicle centre of mass.

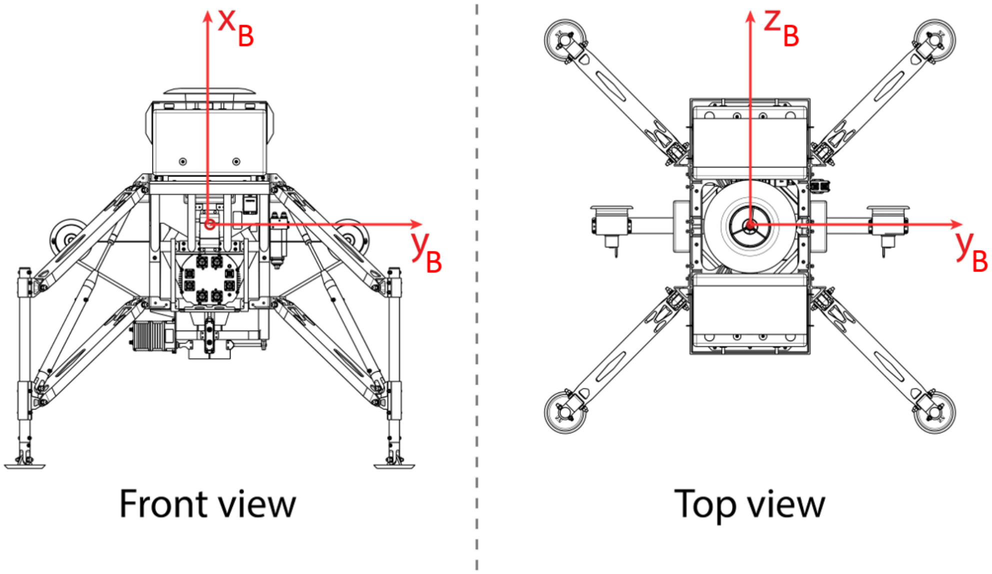


Figure 8.2 Body-fixed reference frame - DTV. Credit: INCAS

The translational equations (kinematic and dynamic), in vectorial form, are the following:

$$\begin{gathered}\dot{\boldsymbol{r}} = \boldsymbol{v} \\ \boldsymbol{m}\dot{\boldsymbol{v}} = \boldsymbol{N}\end{gathered} \tag{8.1}$$

where:

- $\boldsymbol{r}$ is the cartesian position vector of the vehicle centre of mass with respect to the LP frame origin, expressed in the LP frame;
- $\boldsymbol{v}$ is the cartesian velocity vector of the vehicle centre of mass with respect to the LP frame origin, expressed in the LP frame;
- $\boldsymbol{m}$ is the mass of the vehicle (instantaneous);
- $\boldsymbol{N}$ is the applied force, being composed of the force generated by the main engine thrust ($\boldsymbol{F_T}$), the force generated by the vane system ($\boldsymbol{F_V}$), the force generated by the fan system ($\boldsymbol{F_{DF}}$), the force due to gravity ($\boldsymbol{F_g}$) and a disturbance force ($\boldsymbol{F_d}$):

$$\boldsymbol{N} = \boldsymbol{F_T} + \boldsymbol{F_V} + \boldsymbol{F_{DF}} + \boldsymbol{F_g} + \boldsymbol{F_d} \tag{8.2}$$

The force generated by the main engine thrust ($\boldsymbol{F_T}$) is computed using:

$$\boldsymbol{F}_T = \boldsymbol{A}_{LP\_B} \begin{Bmatrix} T \\ 0 \\ 0 \end{Bmatrix} \tag{8.3}$$

where T is the main engine thrust (details in Table 7.1) and $\boldsymbol{A}_{LP\_B}$ is a rotation matrix from B to LP frame which has the following form:

$$A_{LP\_B} = \begin{bmatrix} cos\psi\ cos\theta & cos\psi\ sin\theta\ sin\phi - sin\psi\ ccos\phi & sin\psi\ sin\phi + cos\psi\ sin\theta\ cos\phi \\ sin\psi\ cos\theta & cos\psi\ cos\phi + sin\psi\ sin\theta\ sin\phi & sin\psi\ sin\theta\ cos\phi - cos\psi\ sin\phi \\ -sin\ \theta & cos\theta\ sin\phi & cos\theta\ cos\phi \end{bmatrix} \tag{8.4}$$

The force generated by the vane system ($\boldsymbol{F_V}$) is computed using the following relation:

$$\boldsymbol{F_V} = \boldsymbol{A}_{LP\_B} \begin{Bmatrix} 0 \\ F_{Vy} \\ F_{Vz} \end{Bmatrix} \tag{8.5}$$

where F_{Vy} and F_{Vz} are the forces generated by each of the two DTV vanes. Note that F_{Vy} and F_{Vz} can be either positive or negative, depending on the vane deflection. The sign convention is shown in Figure 8.3.

The force generated by the duct-fan system ($\boldsymbol{F}_{DF}$) is computed using:

$$\boldsymbol{F}_{DF} = \boldsymbol{A}_{LP\_B} \begin{Bmatrix} 0 \\ 0 \\ F_{DF\_left} + F_{DF\_right} \end{Bmatrix} \tag{8.6}$$

Note that F_{DF_left} și F_{DF_right} are always positive, as per the sign convention shown in Figure 8.4.

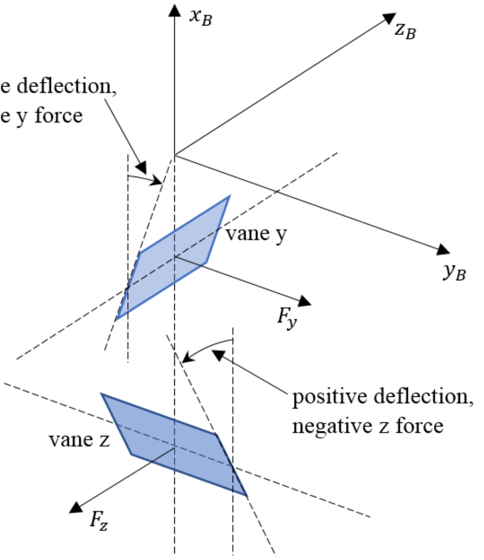


Figure 8.3 Sign convention, vane system

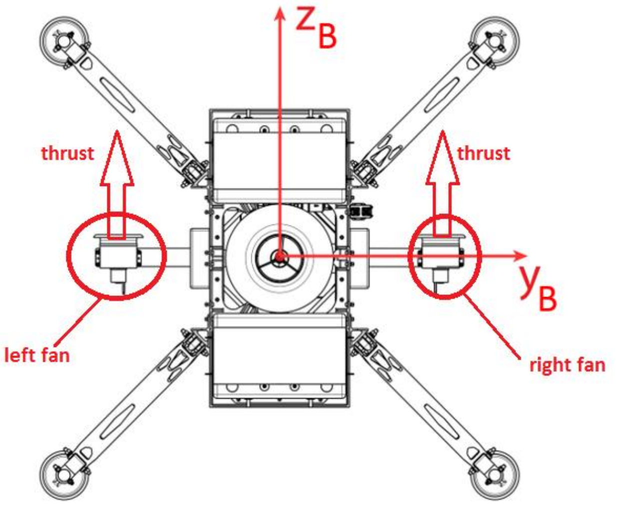


Figure 8.4 Sign convention, ducted fans (used for roll control). Credit: INCAS

The force due to gravity ($\boldsymbol{F_g}$) is computed using:

$$\boldsymbol{F}_g = \begin{Bmatrix} -mg \\ 0 \\ 0 \end{Bmatrix} \tag{8.7}$$

where g is the standard gravitational acceleration.

For conservative and control system design reasons, a disturbance force is also used ($\boldsymbol{F_d}$).

The rotational equations of motions of the DTV vehicle (kinematic and dynamic), in vectorial form, are the following

$$\begin{gathered} \dot{\phi} = p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \\ \dot{\theta} = q\cos\phi - r\sin\phi \\ \dot{\psi} = q\frac{\sin\phi}{\cos\theta} + r\frac{\cos\phi}{\cos\theta} \\ I\dot{\omega} + \omega \times (I\omega) = M_T + M_V + M_{DF} + M_r + M_d \end{gathered} \tag{8.8}$$

where:

- ϕ, θ and ψ are the roll, pitch and yaw angles (Euler angles ZYX);
- p, q and r are the components of angular velocity $\boldsymbol{\omega}$ in the body-fixed reference frame (B);
- $\boldsymbol{I}$ the inertia matrix with respect to the centre of mass and body frame axes.
- $\boldsymbol{M_T}$ is the moment produced by the main engine thrust $\boldsymbol{F}_T$**,** described in (8.3**)**;
- $\boldsymbol{M_V}$ is the moment produced by the force generated by the vane system $\boldsymbol{F_V}$, described in (8.5);
- $\boldsymbol{M_{DF}}$ is the moment produced by the fan system $\boldsymbol{F}_{DF}$, described in (8.6);

- $\boldsymbol{M}_r$ is the moment produced by the rotor of the main engine ($\boldsymbol{I}_r$ is the inertia matrix of the rotor relative to the rotor centre of mass, while $\boldsymbol{\omega}_r$ is the angular velocity of the rotor), described by the following relation:

$$\boldsymbol{M}_r = -\boldsymbol{I}_r\dot{\boldsymbol{\omega}}_r - \boldsymbol{I}_r(\boldsymbol{\omega}\times\boldsymbol{\omega}_r) - \boldsymbol{\omega}_r\times(\boldsymbol{I}_r\boldsymbol{\omega}_r) - \boldsymbol{\omega}_r\times(\boldsymbol{I}_r\boldsymbol{\omega}) - \boldsymbol{\omega}\times(\boldsymbol{I}_r\boldsymbol{\omega}_r) \qquad (8.9)$$

- $\boldsymbol{M}_d$ is the disturbance moment

8.2 S-Function integration into the DTV Vehicle Flight Simulator

The flight simulator developed by INCAS for DTV can accept experimental computing programs. To test and validate in-flight an experimental application, it is necessary to integrate it into the flight simulator for closed-loop testing using the six-degree-of-freedom models. The "SEP" (*Software Experimental Payload*) block in Figure 8.5 contains the online convex trajectory optimisation program developed in this PhD thesis, together with the required additional functions developed. As highlighted in Figure 8.5, the SEP block is coloured blue, which in this case represents that its execution is occurring every 100 milliseconds. Additional implementation details are provided in Figure 8.6 and Figure 8.7.

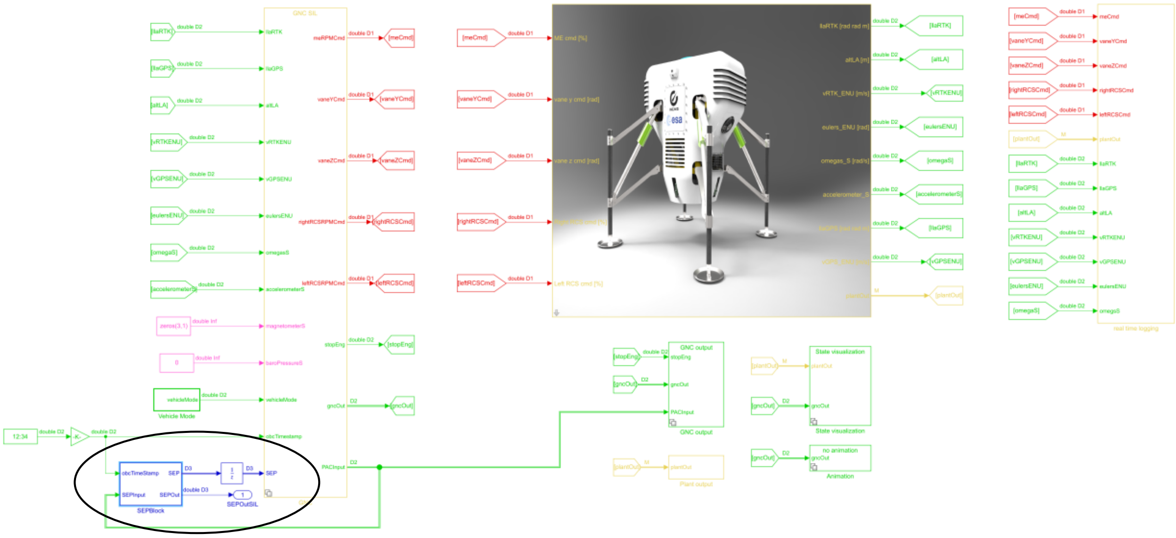


Figure 8.5 DTV Integration of the online solver into the DTV Flight Simulator

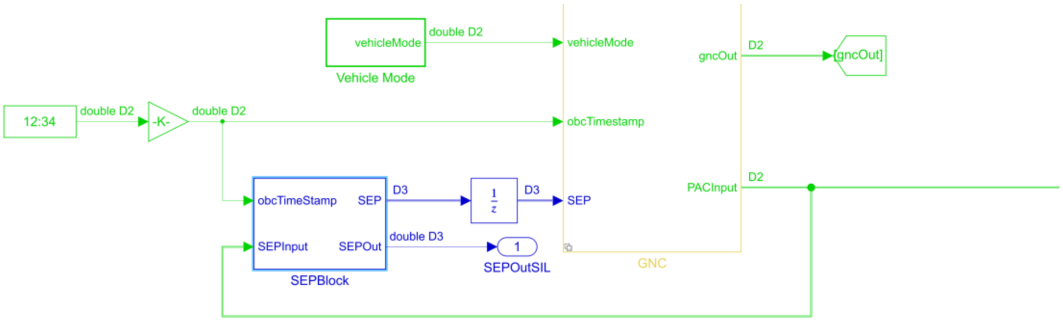


Figure 8.6 DTV Integration of the online solver into the DTV Flight Simulator (detailed)

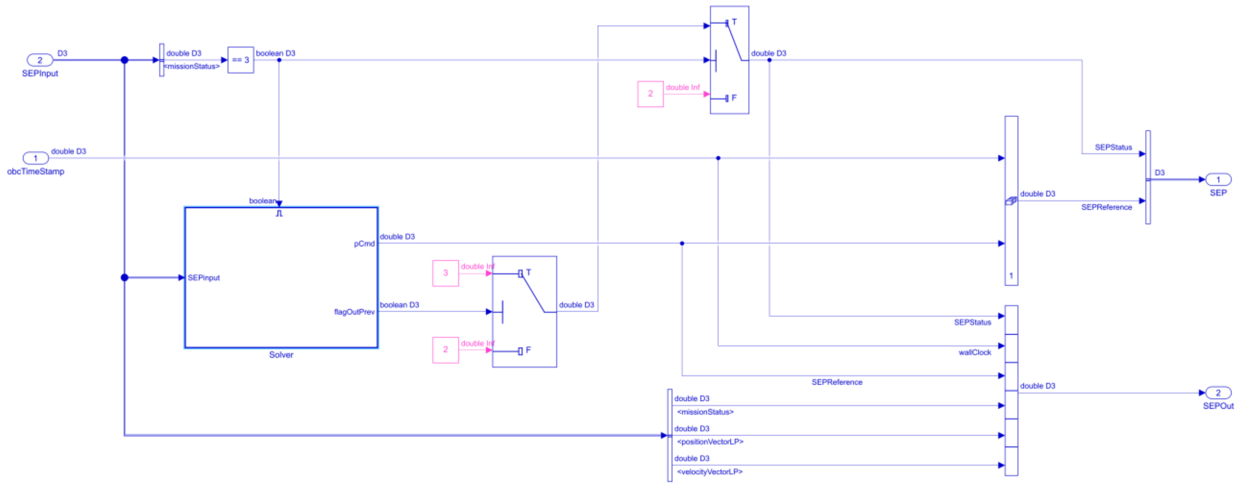


Figure 8.7 Contents of the SEP Block - Simulink

The output data from the SEP block is as follows:

- ***SEPStatus***: represents the signal that determines whether the trajectory sent to the onboard computer is the reference trajectory or the experimental trajectory. DTV's flight simulator contains a program that decides, based on several critical parameters, whether the online calculated trajectory is valid before sending it to the vehicle;
- ***SEPReference***: contains the trajectory computed online (Positions and velocities on the 3 axes).

8.3 Offline numerical results

This subchapter presents the results obtained using the flight simulator (protected model) for DTV, which is called closed-loop testing. The DTV begins its mission with a predefined reference trajectory: ascent to an altitude of 5 meters, then it goes to hover flight phase, and after 15 seconds the real-time application developed in this PhD thesis starts to compute a trajectory onboard. At the beginning, the application asks for real-time in-flight DTV position and velocities data from the GNC application and uses it as a starting point. The target position that the DTV must reach is 0.65 m above the ground (the position of the centre of mass relative to the ground), 5 meters in the y direction and 4 meters in the z direction. The predefined parameters for the real-time application are those presented in subchapter 7.6.

The complete mission for DTV is shown in Figure 8.8, where the part of the flight in which the reference trajectory is generated onboard is highlighted.

Figure 8.9 shows the online generated trajectory onboard the DTV. The altitude from which the online calculation starts (5 meters) can be observed, as well as the final lateral positions, thus showing compliance with the constraints imposed on the developed application. The velocities on the 3 axes are illustrated in Figure 8.10 and it can be seen that they have values below 0.5 m/s, thus respecting the constraints related to the maximum flight velocity. The attitude of the vehicle is shown in Figure 8.11, and the angular velocities of roll, pitch and yaw are shown in Figure 8.12.

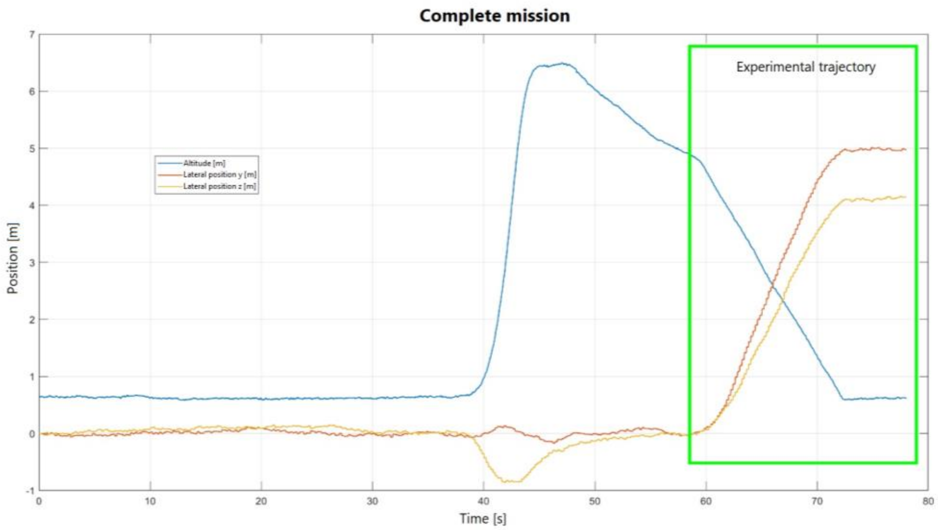


Figure 8.8 6DOF simulation, complete mission

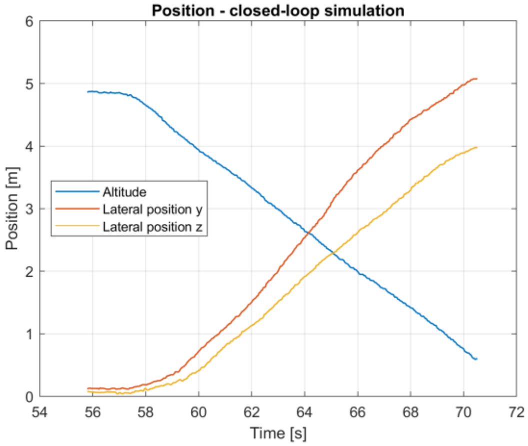


Figure 8.9 Position - online generated trajectory, closed-loop simulation

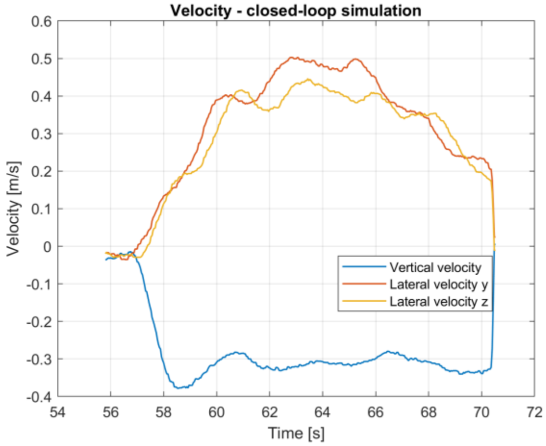


Figure 8.10 Velocity - online generated trajectory, closed-loop simulation

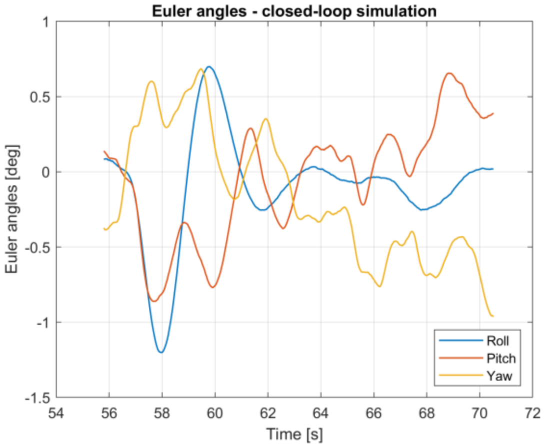


Figure 8.11 DTV attitude - online generated trajectory, closed-loop simulation

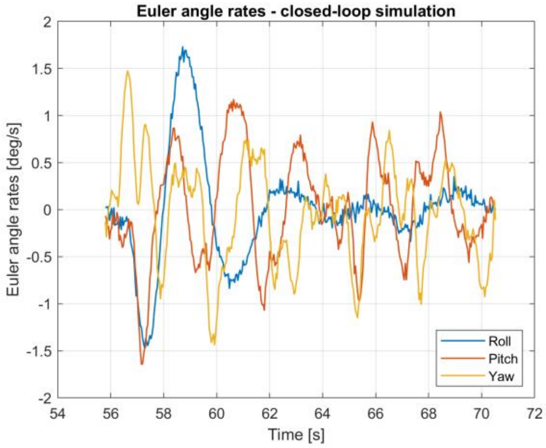


Figure 8.12 Angular velocities - online generated trajectory, closed-loop simulation

9 Testing the SEP application using the Real Time Target Machine

9.1 Developed real-time application test architecture

To successfully test the developed real-time application (which contains the convex solver), it is necessary to first integrate it into the DTV flight software. The operating system running on the onboard computer within DTV is Linux Real Time. The flight software is the main layer and the one that "decides" the execution order of the processes. The GNC application is integrated within the flight software, and the exchange of signals between the GNC and the developed SEP application is ensured by this flight software by means of predefined signals.

Figure 9.1 illustrates the equipment required for real-time testing of the developed SEP application. Using programs written in the Matlab/Simulink development environment, code is automatically generated for a real-time test application. The real-time test system, SpeedGoat, is specifically designed for "Simulink Real-Time" [29], which is the reasoning behind selecting this test modality. The SpeedGoat runs DTV's six-degree-of-freedom dynamic model and sends signals simulating the nominal mission in real time. These signals are further sent to the onboard computer running the GNC system and the vehicle's flight application as two separate processes. All these hardware devices are placed in a local LAN (local area network) so that signals and files can be exchanged between them.

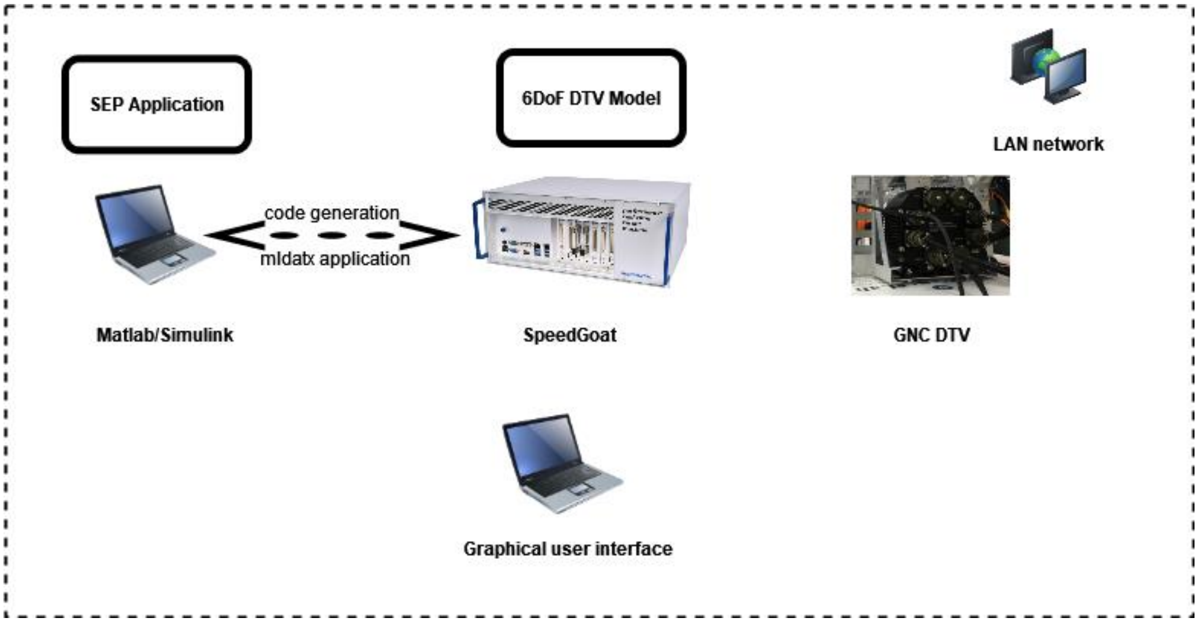


Figure 9.1 Real-time testing procedure and hardware

9.2 S-functions required to integrate the real-time application into the flight software

The SEP block, described in detail in subchapter 8.2, is tested in real time using the procedure described in the same subchapter, where the simulated signals are replaced by S-functions that link the SEP application to the flight software. These signals are used to ensure the exchange of data between the flight software, the SEP application and the GNC application.

9.3 Results of the nominal mission real-time (HIL) simulation

The real-time simulation results using the developed application are shown in Figure 9.2 and Figure 9.3. These are forwarded via S-functions to the demonstrator's GNC and become the new position and velocity references for the DTV. The optimal trajectory is shown in Figure 9.2, and the flight velocities calculated onboard are presented in Figure 9.3. It can be seen that all final constraints imposed as application parameters are respected.

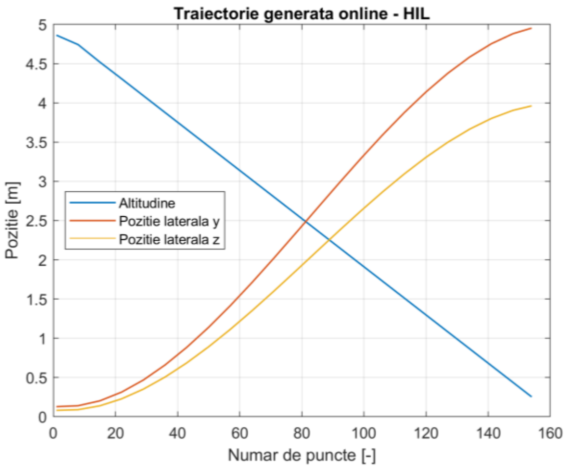


Figure 9.2 Optimal online trajectory - HIL simulation

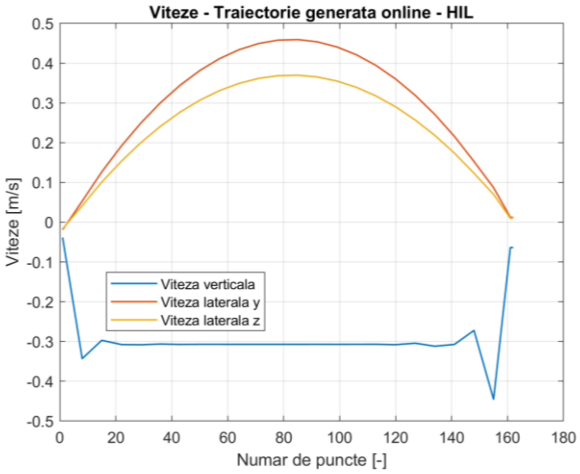
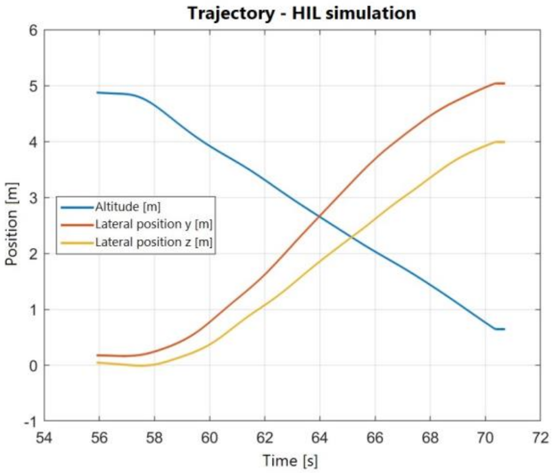


Figure 9.3 Online computed velocities - HIL simulation

The real-time simulated positions and velocities (in six degrees of freedom) are shown in Figure 9.4 and Figure 9.5 and can be seen to be similar to those calculated by the flight computer application, which means that the DTV can follow the online optimised trajectory with a small error in real time. The Euler angles and angular velocities resulting from the six-degree-of-freedom simulations can be viewed in Figure 9.6 and Figure 9.7. Since the commanded velocities are low, the required attitude angles are also low.



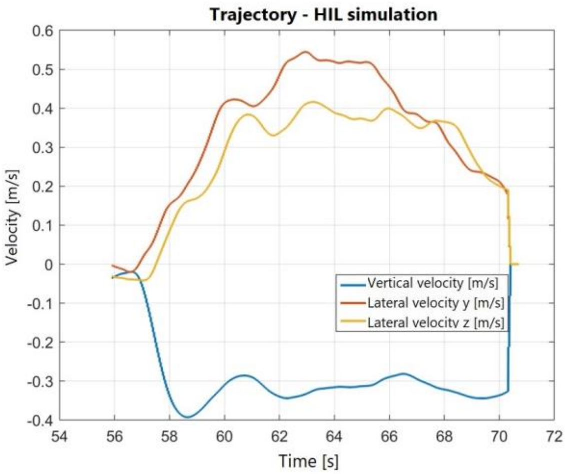


Figure 9.4 Trajectory - HIL simulation

Figure 9.5 DTV velocities – HIL simulation

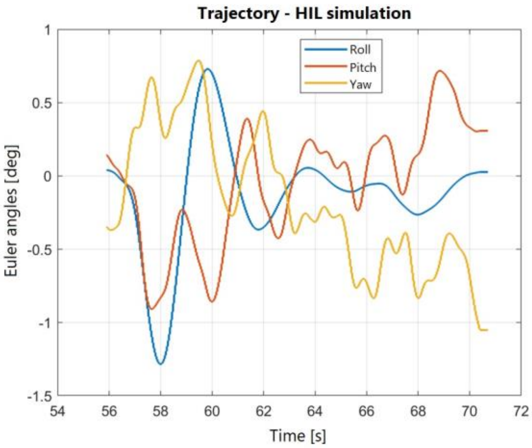


Figure 9.6 DTV attitude - HIL simulation

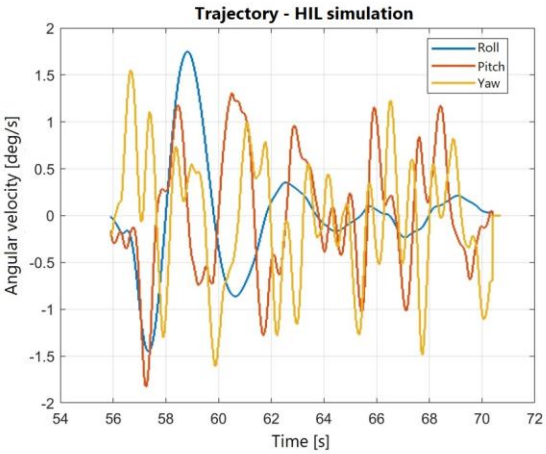


Figure 9.7 Angular velocities - HIL simulation

The test results of the developed application under real-time conditions confirm the possibility of using the methodology and algorithms developed in this PhD work onboard reusable demonstrators with space applications. The execution time of the developed SEP application on the onboard computer of the real-time DTV demonstrator is about 10 milliseconds, which confirms the feasibility of its use during a real mission.

10 Testing and validating the application under tether flight conditions

For the in-flight validation of the developed application, an experimental test campaign was carried out at the INCAS Măneciu in Prahova. Five experimental flights were successfully carried out and the results are presented in this chapter. Before each experimental flight, numerical simulations were performed using the six-degree-of-freedom flight simulator to ensure their feasibility. In Figure 10.1, a comparison between offline simulations and experimental results using the developed SEP application.

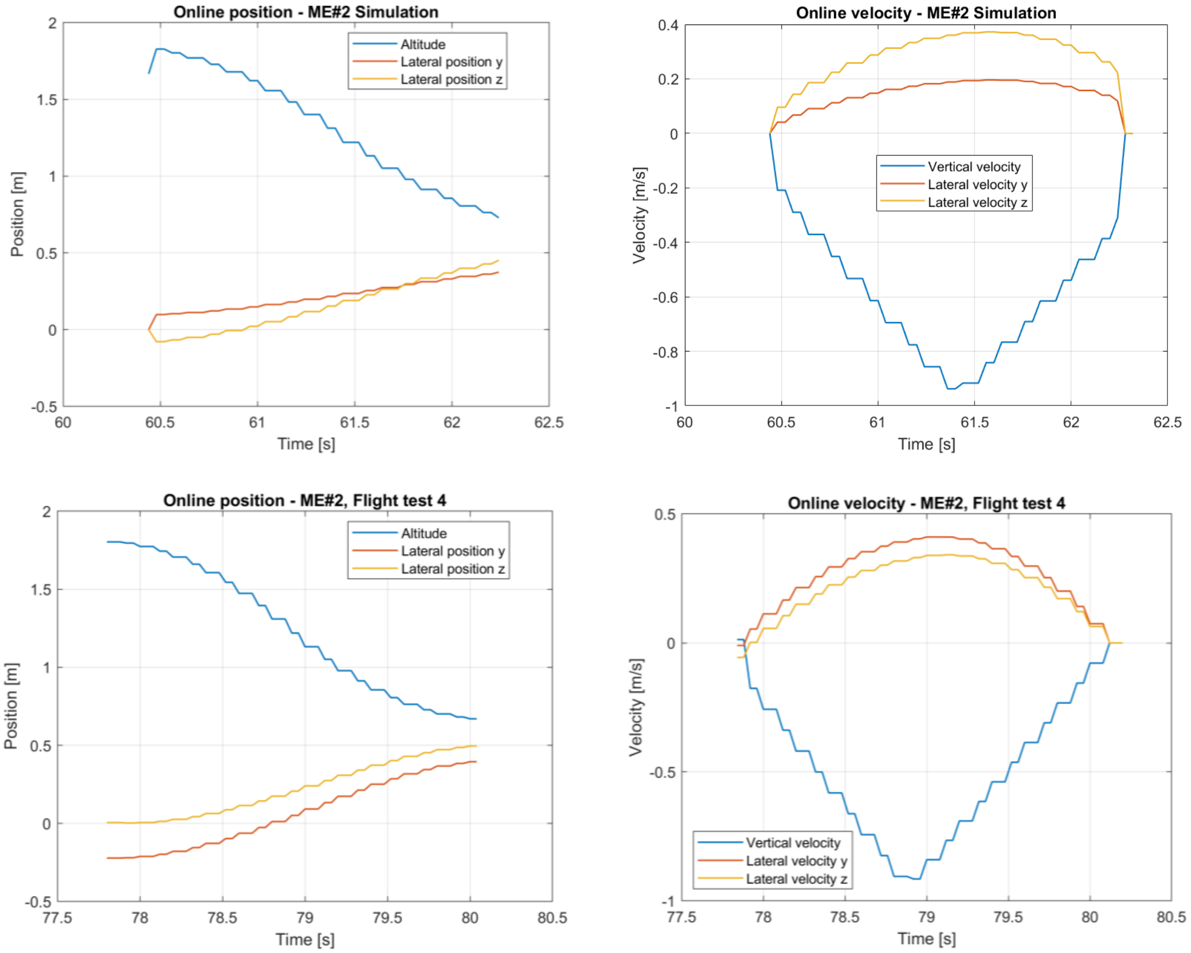


Figure 10.1 Online position and velocities (simulation and experimental flight)

The results recorded during test flight no. 4 are shown in Figure 10.2. The trajectory computed online respects the imposed constraints, and this trajectory becomes the new reference for the DTV demonstrator. A comparison between the commands and the real states of DTV is also shown and it can be seen that the demonstrator follows the on-board calculated trajectory with very good accuracy considering the limited physical flight space. Onboard commands for the vane system, thrust force and attitude angles are also shown.

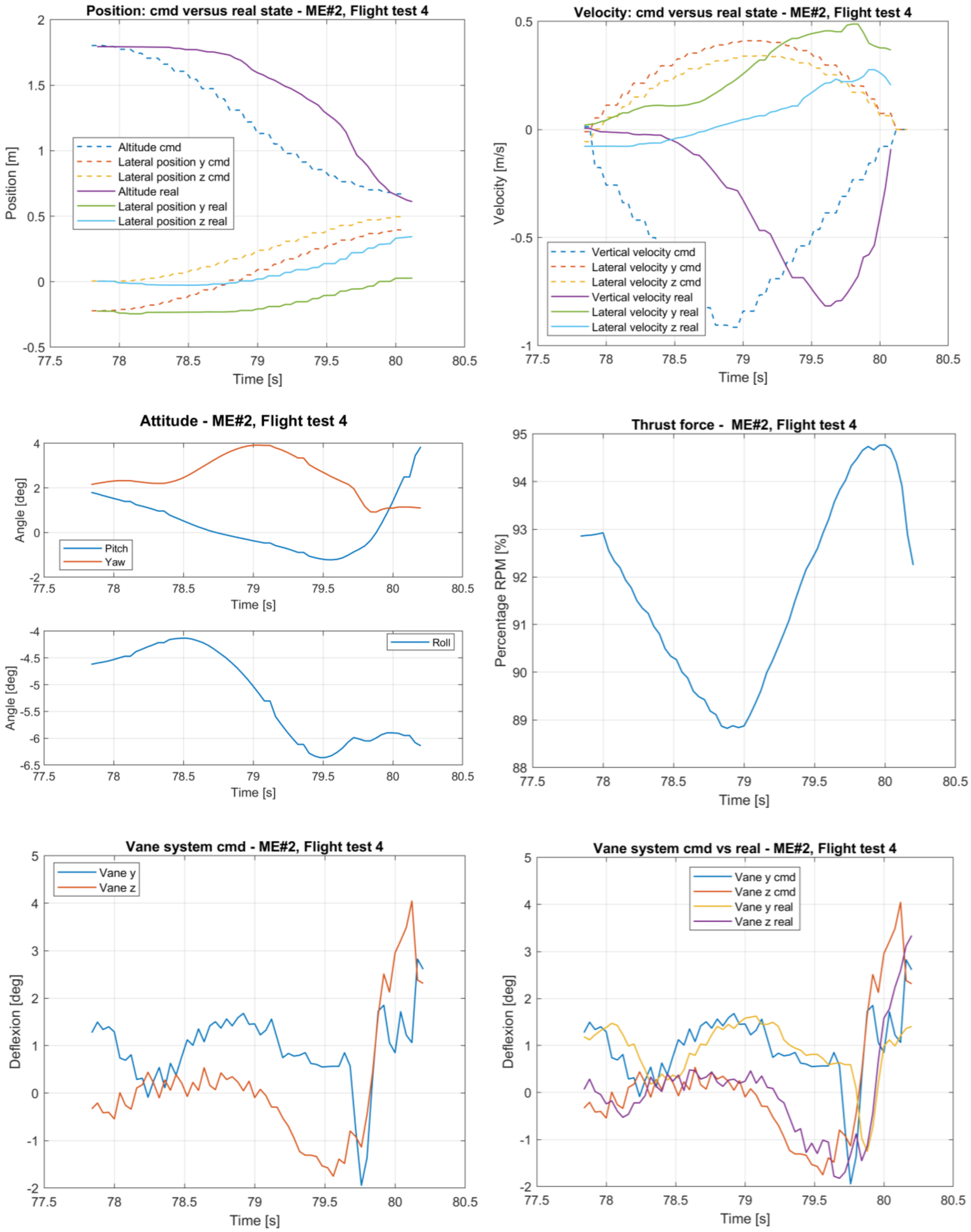


Figure 10.2 Experimental results, ME#2, test flight number 4

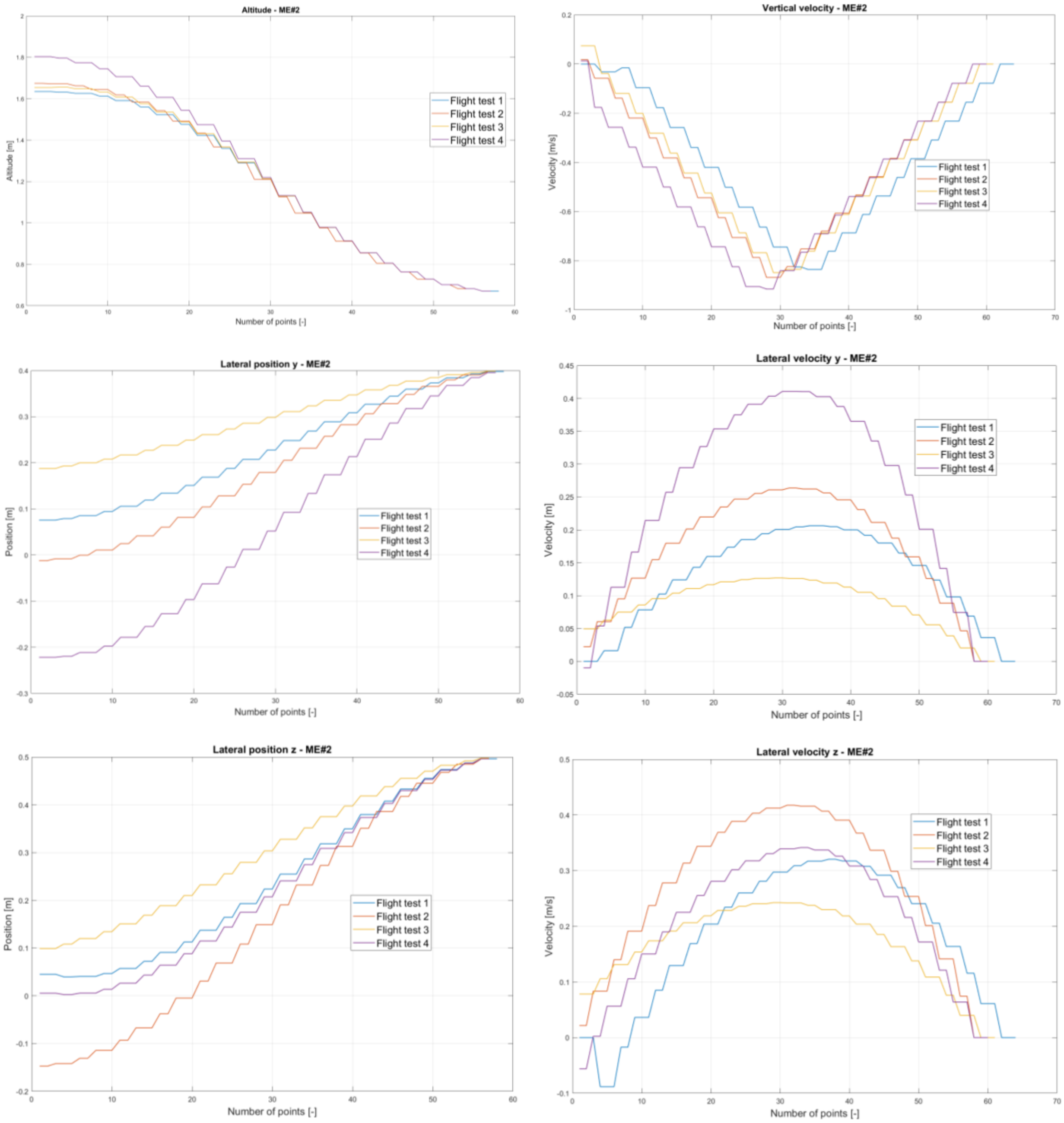


Figure 10.3 Comparative results, ME#2 experimental flights

The in-flight experimental results achieved demonstrate that the application proposed and developed within this PhD thesis can be used in real time and can solve the powered descent minimum fuel trajectory optimisation problem while running on the onboard computer of space vehicles. All test flights were successfully carried out, the developed algorithm found in each case an optimal trajectory on the demonstrator's onboard computer, regardless of the dispersion of the demonstrator's initial states.

For all the flights carried out in the experimental campaign, the execution time of the algorithm on the onboard computer for generating an optimal trajectory did not exceed 70 milliseconds. The total time is represented by the sum of the following events that are taking place in succession:

- The experimental algorithm calls, through the S-functions described in subchapter 9.2, the GNC function that runs as a separate process, and requests the data related to the real state of the demonstrator (position and velocity);
- Generation of an onboard optimal trajectory using as input data the real state of the demonstrator;
- Performing a check of the onboard generated trajectory to ensure the feasibility of the optimal solution obtained;
- Sending the optimal trajectory as a new reference for the demonstrator.
- Writing a .mat file of all parameters of interest for each execution of the application.

During the HIL test campaign, the execution time for the developed application is 10 milliseconds, while during the experimental campaign, the maximum execution time was 70 milliseconds. This difference is mostly due to the change in the way of the memory was allocated for the recording of the in-flight data. The way of memory allocation can be improved in the future, and the total execution time can be significantly reduced.

11 Conclusions

11.1 Thesis contributions

Towards the achievement of the main thesis objective, the following contributions were made:

a) The completion of the mathematical model for the convex optimisation problem of the vertical landing of a demonstrator by introducing additional constraints for the trajectory;

b) Proposal of an innovative integrated testing methodology using different programming environments, Matlab/Simulink, CVXGEN, C programming language (chapter 4);

c) Development of a computational algorithm for solving the convex optimisation problem for the landing of a reusable demonstrator using the Matlab programming environment, together with the module dedicated to convex programming, CVX (chapter 6).

d) Development of a computational algorithm for solving the convex optimisation problem for landing a reusable demonstrator and implementing it in the convex solver generator in CVXGEN, in order to streamline/optimise its execution time to ensure its use in real-time applications. The implementation methodology is explained in chapter 7, which represents a national first and is, at the same time, a starting point in the further development of technology for real-time trajectory optimisation;

e) Development of additional functions written in the C programming language for the use and integration of the code generated by CVXGEN in Matlab/Simulink, thus making important contributions to the use of CVXGEN, considering that there are very few scientific sources available;

f) Development of a Monte Carlo simulation algorithm for testing and validating the functionality of the implemented optimisation method, contributions being also made to demonstrate the reliability of the developed application through offline simulations;

g) Development of a process for the automatic generation and compilation of an S-function containing the solver generated by CVXGEN and the additional functions developed for the integration of this code in Matlab/Simulink, presented in subchapter 8.1;

h) Development of an application, called SEP, for testing the entire convex optimisation methodology in real-time for the studied problem using SpeedGoat (Real-Time Target Machine);

i) Verification and validation of the SEP application under captive flight conditions using a research and validation infrastructure, thus contributing to increasing the maturity of online spacecraft trajectory optimisation technology;

j) Demonstration of the repeatability and reliability of the SEP algorithm developed for the optimisation of the landing trajectory for vertical take-off and landing reusable demonstrators by performing experimental flights in similar conditions.

11.2 Results obtained

The research activities of the PhD thesis focused on the development of an integrated methodology and a real-time application containing a solver used in obtaining the optimal solution of the convex problem for the optimisation of the landing trajectory of a reusable retro-propulsion demonstrator with space applications. To achieve this objective, the mathematical model was derived and numerically solved using the Matlab mode, CVX/SeDumi. The mathematical model was also implemented in the C code generator, CVXGEN, encapsulated in an S-function and integrated into a six-degree-of-freedom simulation module for the reusable demonstrator called DTV. Numerical simulations were performed for a nominal mission to test the developed real-time application, along with Monte Carlo simulations in the presence of uncertainties. The developed application was tested and validated in real-time using a real-time test machine, SpeedGoat prior to flight testing to mitigate any potential risks. An experimental campaign was carried out to test and validate in-flight the application called SEP, developed in this PhD thesis.

In the first chapter, the current context of research in the field of convex optimisation, but also of the algorithms for solving the problem in real time using the onboard computers equipping the space vehicles, was presented. In the second chapter some introductory notions related to the convex optimisation technique were presented, and in the third chapter the mathematical derivation of the problem of minimizing the amount of fuel required for the vertical landing of space demonstrators and its discretization was presented.

Chapter four provided an overview of the methodology proposed and pursued in this PhD thesis, while chapter five briefly presented the reusable DTV platform used for testing and in-flight validation of the developed real-time application.

In the sixth chapter, the numerical results for a mission considered nominal for the validation of the derived mathematical model for the problem of interest were presented. The results demonstrated that the problem is a convex one, and the selected CVX module was considered suitable for solving it. Chapter seven details the numerical implementation in the commercial C code generator, CVXGEN. Monte Carlo simulations were carried out in the presence of the uncertainties of the initial conditions and the results obtained confirmed the robustness of the real-time application containing the solver, but also its applicability in real-time convex optimisation problems.

The developed real-time application was integrated into DTV's flight simulator, provided as a protected model, and the results presented in Chapter eight confirmed that the solver-optimised trajectory is operationally feasible given the constraints of the DTV demonstrator.

The results of real-time testing and validation were presented in the ninth chapter, and the results obtained from tests using the SpeedGoat system confirm that this application can be used in a real flight to optimise the trajectory of the demonstrator. The computation time required to obtain an optimal in-flight trajectory is in the order of 10 milliseconds in ideal hardware conditions, which makes it suitable for in-flight optimal trajectory generation.

The developed SEP application was tested and validated following experimental flight campaigns, and the results were presented in chapter ten. Two experimental missions were considered and a total of five flights were performed at the INCAS quarters in Măneciu, Prahova. The results of the experimental campaign demonstrate the feasibility of using the methodology proposed in this PhD thesis. The developed SEP application was successfully used in all experimental flights and provided every time an optimal trajectory in a very short time (of the order of tens of milliseconds, in real-word operational hardware) regardless of the initial conditions of the demonstrator, which demonstrates the potential of using this methodology in space missions.

11.3 Prospects for continuation and further development

This PhD thesis focused on the development of algorithms and a complete methodology integrated into the SEP application for the real-time solution of the convex optimisation problem for space reusable demonstrators.

The mathematical problem studied is the fixed-time-of-flight convex optimisation problem. This hypothesis is valid under the conditions studied in this thesis, but if it is desired to study a vehicle re-entering the Earth's atmosphere, and, in this regard, it is also necessary to optimise the flight time onboard. A further improvement is to study and implement the free flight time landing problem and optimise it onboard the vehicle. Another improvement that can be made to the mathematical model is the addition of vehicle attitude constraints, which finally leads to a mathematical problem with six degrees of freedom.

In this thesis, a commercial solver generator, CVXGEN, was used for automatic C code generation. Considering the current state of research in the field and the directions that the space industry is pursuing, it is of interest to develop a dedicated automatic code generator to solve the fixed-point landing problem, because CVXGEN or commercial alternatives deal with general convex optimisation problems, which is not always an optimal implementation option. Another future perspective of interest is the development of a complete process that provides users with a dedicated environment for the integration and further use of this generated code, together with a flexible graphical interface.

Another direction for further research and development is the use of the SEP application proposed in this work in a complex program for landing on the Moon or Mars. For high precision landing it is necessary to use a SEP type application in conjunction with onboard image processing software. By analysing images during re-entry in real time, the algorithm can select a landing site based on the constraints imposed and transmit the coordinates of the chosen target site to an application resembling the developed SEP application. The target point selected by the image processing algorithm can be sent to the that specific application, thus becoming an integrated part of the real-time landing trajectory optimisation process. Given the current space research context, it is expected that the scenario described above will become recurring practice in future space missions, and the research efforts described in this PhD thesis represent a solid starting point, demonstrating the feasibility of implementation and use the SEP application in real time.

11.4 List of publications

During the preparation of the PhD thesis, 12 scientific articles were published in collaboration with other researchers, which can be accessed in national and international journals together with a book chapter, published by Springer.

2017
T. V. Chelaru, A. I. Onel, T. P. Afilipoae, **A. M. Neculăescu**, “Mathematical Model for Microlauncher performances evaluation,” UPB Scientific Bulletin, Series D: Mechanical Engineering, vol. 79, nr. 4, pp. 49-66, 2017.
B. Oving, A. Kleef, B. Haemmerli, A. Boiron, Markus Kuhn, Ilja Müller, Ivaylo Petkov, **A.M Neculaescu**, T.P. Afilipoae and Marina Petrozzi, "Small Innovative Launcher for Europe: achievement of the H2020 project SMILE", 7th European Conference for Aeronautics and Space Sciences (Eucass); DOI: 10.13009/EUCASS20170600, 2017
M.V. Pricop, M.G. Cojocaru, C.I. Stoica, M.L. Niculescu, **A. M. Neculăescu**, A.G. Persinaru, M.Boscoianu, "Glide back booster wind tunnel model testing", AIP Conference Proceedings 1863, DOI: 10.1063/1.4992593, 2017.
2018
A. M. Neculăescu, T. P. Afilipoae, A. I. Onel, M. V. Pricop, I. Stroe, “Trajectory Optimization for Small Launchers Using a Genetic Algorithm Approach,” AIP Conference Proceedings, vol. 2046, 2018.
A. I. Onel, T. P. Afilipoae, **A. M. Neculăescu**, M. V. Pricop, “MDO approach for a two-stage microlauncher,” *INCAS Bulletin*, vol. 10, nr. 3, pp. 127-138, 2018.
T. P. Afilipoae**, A. M. Neculăescu**, A. I. Onel, M. V. Pricop, A. Marin, A. G. Perşinaru, A. M. Cişmilianu, I. C. Oncescu, A. Toader, A. Sirbi, S. Bennani, T. V. Chelaru, “Launch Vehicle - MDO in the development of a Microlauncher,” Transportation Research Procedia, vol. 29, pp. 1-11, 2018.
A. I. Onel, A. Stăvărescu, M. G. Cojocaru, M. V. Pricop, M. L. Niculescu, **A. M. Neculăescu**, T. P. Afilipoae, “Computation of the Hypersonic Heat Flux with Application to Small Launchers,” AIP Conference Proceedings, vol. 2046, 2018.
A. I. Onel, T. P. Afilipoae, **A. M. Neculăescu**, M. V. Pricop, “Drag coefficient modelling in the context of small launcher optimisation,” *INCAS Bulletin*, vol. 10, nr. 4, pp. 103-116, 2018.

2019
A. I. Onel, O. I. Popescu, **A. M. Neculăescu**, T. P. Afilipoae, T. V. Chelaru, “Liquid rocket engine performance assessment in the context of small launcher optimisation”, *INCAS Bulletin*, vol. 11, nr. 3, pp. 135-145, 2019.
A. M. Neculăescu, A. Marin, A. Toader, A. G. Persinaru, A. M. Cismilianu, M. Tudose, Camelia-Elena Munteanu, I. Popescu, H. Strauch, S. Dussy, “System Identification and Testing for a VTVL vehicle”, EUCASS 2019, DOI: 10.13009/EUCASS-925, 2019.
2021 (book chapter)
A. M. Cismilianu, I. Chirita, A. G. Persinaru, A. Marin, C. E. Munteanu, **A. M. Neculăescu**, C. Dragoman, „Re-entry vehicle structural optimization for mass minimization”, published in the book „Innovations in Mechanical Engineering”, DOI: 10.1007/978-3-030-79165-0_9, 2021.
2022
T. P. Afilipoae, **A. M. Neculăescu**, P. Simplicio, S. Bennani, H. Strauch, „Control strategies comparison and performance evaluation for a reusable VTVL platform based on a rocket engine”, EUCAS 2022, DOI: 10.13009/EUCASS2022-4480
2023
A.M. Neculăescu, A.I. Onel, C. B. Briceag, A. Toader, “Online convex optimization for a reusable vertical take-off and vertical landing demonstrator”, in the process of being published.

Bibliography (selective)

[1] Musk, E., “Space X, Falcon 9 launcher,” https://www.spacex.com/vehicles/falcon-9/.

[2] J. Foust, “Blue Origin flies six on first New Shepard suborbital flight of 2022,” https://spacenews.com/new-shepard-completes-fifth-crewed-suborbital-flight/, 2022.

[3] A. Jones, “Deep Blue Aerospace completes kilometer-level rocket launch and landing test,” https://spacenews.com/deep-blue-aerospace-completes-kilometer-level-rocket-launch-and-landing-test/, 2022.

[4] ESA Communication team, “ESA plans demonstration of a reusable rocket stage,” https://www.esa.int/Enabling_Support/Space_Engineering_Technology/ESA_plans_demonstration_of_a_reusable_rocket_stage, 2021.

[5] ESA GSTP Annual Report, “GSTP Annual Report 2020,” https://indd.adobe.com/view/fb37fecf-5ab6-4cd1-868d-a0679e7432c7, 2020.

[6] H. W. Jones, “The Recent Large Reduction in Space Launch Cost,” vol. 48th International Conference on Environmental Systems, 2018.

[7] D. Malyuta, T. P. Reynolds et all, “Convex optimization for Trajectory Generation,” https://arxiv.org/abs/2106.09125, 2021.

[8] Mordecai Avriel, Nonlinear Programming: Analysis and Methods, Dover Publications, 2003.

[9] Mokhtar S. Bazaraa, Hanif D. Sherali, C.M. Shetty, Nonlinear programming, Theory and algorithms, Third Edition, John Wiley & Sons, 2006.

[10] Andrzej Ruszczynski, Nonlinear Optimization, Princeton University Press, 2006.

[11] Behçet Açıkmeşe, Lars Blackmore, “Lossless convexification of a class of optimal control problems with non-convex control constraints,” *Automatica*, vol. 47, no. 2, pp. 341-347, 2011.

[12] Lars Blackmore, Behçet Açıkmeşe, John M. Carson III, “Lossless convexification of control constraints for a class of nonlinear optimal control problems,” *Systems & Control Letters,* vol. 61, no. 8, pp. 863-870, 2012.

[13] Behçet Açıkmese, John M. Carson III and Lars Blackmore, “Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem,” *IEEE Transactions on Control Systems Technology,* vol. 21, no. 6, 2013.

[14] Quoc Tran Dinh, “Sequential Convex Programming and Decomposition Approaches for Nonlinear Optimization,” Arenberg Doctoral School of Science, Engineering & Technology, 2012.

[15] John Duch, “Sequential Convex Programming,” Stanford University, 2018.

[16] Taylor P. Reynolds, Mehran Mesbahi, “The Crawling Phenomenon in Sequential Convex Programming,” *American Control Conference,* p. 2020.

[17] M. Szmuk, T. P. Reynolds, B. Acikmese, “Successive convexification for real-time 6-DoF powered descent guidance with state-triggered constraints,” *AIAA Journal of Guidance, Control, and Dynamics,* 2018.

[18] Meikang Qiu, Jiayin Li , Real-Time Embedded Systems: Optimization, Synthesis, and Networking, 1st Edition, CRC Press, 2011.

[19] Xue Liu, Xi Chen, Fanxin Kong, Utilization Control and Optimization of Real-Time Embedded Systems, now Publishers Inc., 2015.

[20] Fateh Boutekkouk, Bout Fat, “Real-Time Embedded Systems Scheduling Optimization: A Review on Bio-Inspired Approaches,” *International Journal of Applied Evolutionary Computation,* vol. 12, no. 1, 2021.

[21] X. Liu, “Fuel-optimal rocket landing with aerodynamic controls,” *Journal of Guidance, Control and Dynamics,* vol. 42, pp. 65-77, 2019.

[22] Yuanqi Mao, Behcet Acikmese, Pierre-Loïc Garoche, Alexandre Chapoutot, “Successive Convexification for Optimal Control with Signal Temporal Logic Specifications,” in *25th ACM International Conference on Hybrid Systems: Computation and Control (HSCC ’22),* Milan, Italy, 2022.

[23] Mattingley, J. E., “CODE GENERATION FOR EMBEDDED CONVEX OPTIMIZATION,” https://stanford.edu/~boyd/papers/pdf/code_gen_impl.pdf, 2011.

[24] B. Açıkmeşe, L. Blackmore, D. P. Scharf, A. Wolf, “Enhancements on the Convex Programming Based Powered,” in *AIAA/AAS Astrodynamics Specialist Conference, Honolulu,* Hawaii, 2008.

[25] B. Açıkmeşe, S. R. Ploen, “Convex Programming Approach to Powered Descent Guidance,” *Journal of Guidance, Control, and Dynamics,* vol. 30, no. 5, pp. 1353-1366, 2007.

[26] ESA, INCAS, “Demonstrator masters flight sequences for reusable rocket stages,” https://www.esa.int/Enabling_Support/Space_Transportation/Future_space_transportation/Demonstrator_masters_flight_sequences_for_reusable_rocket_stages, 2020.

[27] J. Mattingley, S. Boyd, “CVXGEN: Code Generation for Convex Optimization,” https://cvxgen.com/docs/index.html, 2022.

[28] I. E. Mooij, “The motion of a vehicle in a planetary atmosphere,” TU Delft, Delft, 1994.

[29] Speedgoat Products and Services, “https://www.speedgoat.com/products-services”.