READ 2024
RESEARCH & EDUCATION IN AIRCRAFT DESIGN
WARSAW, POLAND | 6-8 NOVEMBER 2024

USING DIRECT METHODS FOR SOLVING OPTIMAL MISSILE CONTROL PROBLEMS

Dariusz Miedziński[1] & Robert Głębocki[1]

[1]Institute of Aeronautics and Applied Mechanics, Warsaw University of Technology

Abstract

Optimization is very important part of the flight vehicles' design process. It is used to tailor the performance of the object and maximize its efficiency during flight. It is widely used in every branch of the design process, including topics in aerodynamics, propulsion, structural design, and GNC. The latter is particularly important when dealing with all kinds of missiles. The optimized guidance and control algorithms allow for fulfilling the mission objectives with high precision and accuracy. One example is the air defense systems, which need to neutralize foreign vehicles with high precision, the other one is the surface-to-surface missiles that should pinpoint the distant ground-based targets. Lowering the impact point dispersion of such missiles is a key aspect that requires optimal control algorithms development.

Various types of control and guidance methods and algorithms are used. The most widely utilized is proportional navigation with many of its variants. This method tries to keep the line-of-sight vector, which connects missile and target, fixed relative to some reference system. In recent times various optimal control methods gained more interest due to the increase in computational power, because, with the exception of simple problems they need to be solved numerically. Generally methods for solving optimal control problems can be divided into direct and indirect. The former use a general non-linear programming solvers to obtain a solution and the latter use the calculus of variations to determine the optimality conditions, like Pontryagin's Maximum Principle, formulate Hamiltonian and solve the Two Point Boundary Value Problem as a result.

In this article, the use of direct methods for solving an optimal control problem for a surface-to-surface missile will be presented. The missile examined in the paper, for simplification purposes, is controlled by four thrusters, that can apply force in both directions in two perpendicular planes of the missile. The value of the thrust of all thrusters, as a function of time, is the optimization parameter. The missile model was developed in MATLAB/Simulink environment and was validated using the flight data and then converted to C++ to increase the computation speed. The article will compare the results of optimization regarding the accuracy and time of a generally available online solver IPOPT and the solver developed by the author. Also, the influence of the use of automatic differentiation compared to the finite difference method for gradient calculations will be examined. The results showing missile accuracy with optimal control will be presented.

Keywords: missile optimal control, lateral thrusters, modeling and simulation, non-linear optimization

1. Introduction

Designing guidance and control algorithms is a very challenging task. Such algorithms should be robust, to work in the dynamic environment, and stable to ensure the safety of the system. This is especially important for various kinds of missiles due to the nature of the tasks they need to perform. There are two major variants of missiles' control systems, aerodynamic control and gas-dynamic control. The former one utilize deflection of lifting surfaces that can be located in the front, middle or aft of the missile. The latter uses thrust vectoring of the main motor or the lateral thrusters, that are located on the circumference of the missile. The missile examined in this article is controlled by the lateral thrusters. There are many guidance and control algorithms used in modern missiles, that use pulse thrusters as a control system. One of them is the reference trajectory following [1, 2], where the thrusters are fired to keep the missile on the predefined path. The error between the missile position and the reference trajectory is calculated in terms of a magnitude and direction. The authors in [3] also added an impact point prediction to the control logic. There is a difficulty associated with this approach as a reference trajectory is a function of time, which will be generally different during the flight and is hard to account for when calculating the trajectory error. Proportional navigation, and many of its variants, are also widely used, [4, 5] present a comparison between few of them. Here only the target point is taken into consideration when calculating the control commands. However, as only the direction of control force can be changed and not its magnitude, due to the characteristics of the impulse thrusters, the resulting impact point error can be significant. There are also works that investigate the possibility to use optimal control for pulse jet controlled missiles. The authors in [6] used a model predictive control combined with a linear projectile theory to directly control the impact point. Optimal control methods can generally be divided into direct and indirect. Surveys describing various methods can be found in [7, 8]. One type of methods that belong to the class of direct methods for solving optimal control problems are pseudospectral methods. In those, the state and control variables are discretized using globally orthogonal interpolating polynomials like Jacobi polynomials [9], Chebyshev polynomials [10, 11] or Legendre polynomials [12], or quadrature discretization like Gauss-Lobatto quadrature [13]. The problem is thus transformed to the discrete non-linear programming (NLP) problem and then solved using any NLP optimization solver. A direct collocation [14, 15, 16] is another method for the discretization of the optimal control problem. The indirect methods use the calculus of variations to formulate the necessary optimality conditions, like Pontryagin's Maximum Principle. This conditions are in form of a Two-Point Boundry Value Problem (TPBVP). They consist of two non-linear ordinary differential equations for the state and costate and a minimization (or maximization) condition imposed on the Hamiltonian of the system. They can then be solved using for example variations of shooting methods [17, 18, 19]. This method tries to find unknown parameters, like initial values of the costates, integrating state and costate equation forward in time, and satisfying necessary conditions at final time transforming the problem into non-linear root finding problem. In this article a direct method for solving an optimal control problem for a missile using lateral control thrusters will be described. The aim of the control is to ensure, that the missile will directly impact a predefined ground point. A piece-wise constant control approximation is assumed and the trajectory is calculated using fourth order Runge-Kutta integration scheme. A comparison between a finite difference approximation for the cost function gradient and an automatic differentiation approach will be discussed. The rest of the article is structured as follows. Section 2. describes the mathematical model of the missile, section 3. presents the method and solvers used. Section 4. gives the description of the problem, in section 5. results of the optimization and the discussion is presented, and the article ends with section 6. with the conclusions and future work.

2. Mathematical model of the missile

The model of the missile examined in this article is shown in Figure 1. It is a generic surface-to-surface projectile having a length-to-diameter (L/D) ratio of around 16. It is controlled by a set of solid rocket motors located on the missile's circumference ahead of the center of gravity. The missile's range is about 9 kilometers, reaching a maximum speed of around Mach 1.5. In order to provide the capability of steering the missile in any direction, it rotates around its longitudinal axis during the flight. It is achieved by mounting the fins with a slight cant angle.

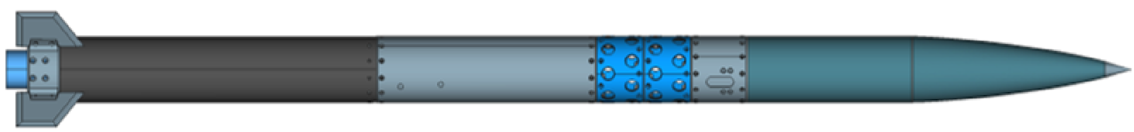

Figure 1 – Missile used as the base for the model development

The missile mathematical model was first developed in MATLAB/Simulink R2023a. The diagram of the model is presented in Figure 2. The missile is modeled as a rigid body with 6 degrees of freedom and varying inertial characteristics. The loads from propulsion system, aerodynamics, gravity and control are calculated and transferred to the equations of motion and integrated to obtain the full state vector consisting of linear and angular velocities, missile's trajectory and attitude angles. In addition, models for a standard atmosphere and launcher rail are also included. The model was validated using the data collected from several test launches, which were used to tailor the missile aerodynamic database. The details about the mathematical model can be found in [20].

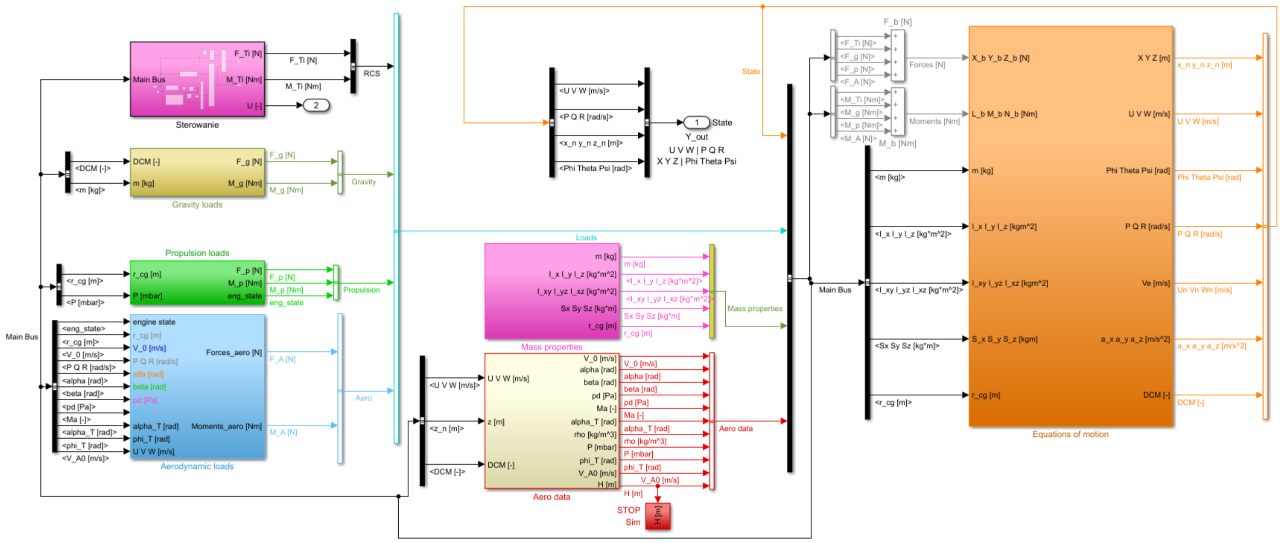

Figure 2 – Missile's mathematical model schematic

The model was then converted to C++ using the automatic code generation capabilities of the MATLAB environment and adjusted properly to allow for the use of an automatic differentiation package. The mathematical model of the missile can be represented as a set of nonlinear ordinary differential equations of the form:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}) \tag{1}$$

where t is time, $\mathbf{x}$ is the state vector, and $\mathbf{u}$ is a control vector. It is assumed that the missile is controlled only in the descending phase of the flight, after reaching the apogee (vertex). In order to perform the optimal control calculations, the missile control model had to be appropriately designed. For simplification purposes, it was assumed that the missile can be controlled by two control parameters, acting in two perpendicular planes of the missile, that are able to provide a thrust in both directions. The equation for the control thrust force expressed in the Body frame of the missile is given as:

$$\mathbf{F}_T = \begin{bmatrix} 0 & \cos(\Phi)u_2 + \sin(\Phi)u_1 & -\sin(\Phi)u_2 + \cos(\Phi)u_1 \end{bmatrix}^T \tag{2}$$

where Φ is the missile roll angle, and $u_i, i = 1, 2$ are the control parameters. That prepared model was then used inside an optimization solver which will be described in the following sections. A sample trajectory of the missile without control is presented in Figure 3. The charts presenting the trajectory components and the linear velocity components as a function of time are presented in Figure 4. The missile was fired in the north direction with the launcher elevation angle of 45 degrees. It flew around 8700 meters reaching an apogee of about 2500 meters. The oscillations visible on the lateral velocities of the missile are the result of its rotation about the longitudinal axis and very low stability margin.

3. Direct optimization method

A direct optimization method requires a formulation of the NLP problem of the form:

$$\min_{x \in \mathbb{R}^n} f(x) \tag{3}$$

$$h_i(x) = 0, i \in E$$

$$h_j(x) \leq 0, j \in I$$

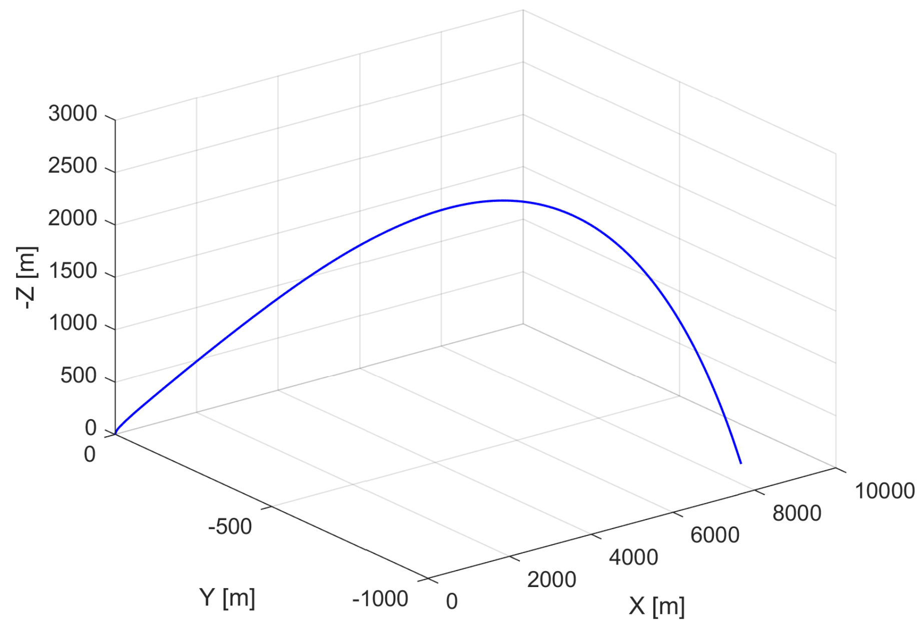

Figure 3 – Missile's trajectory in North-East-Up coordinate system

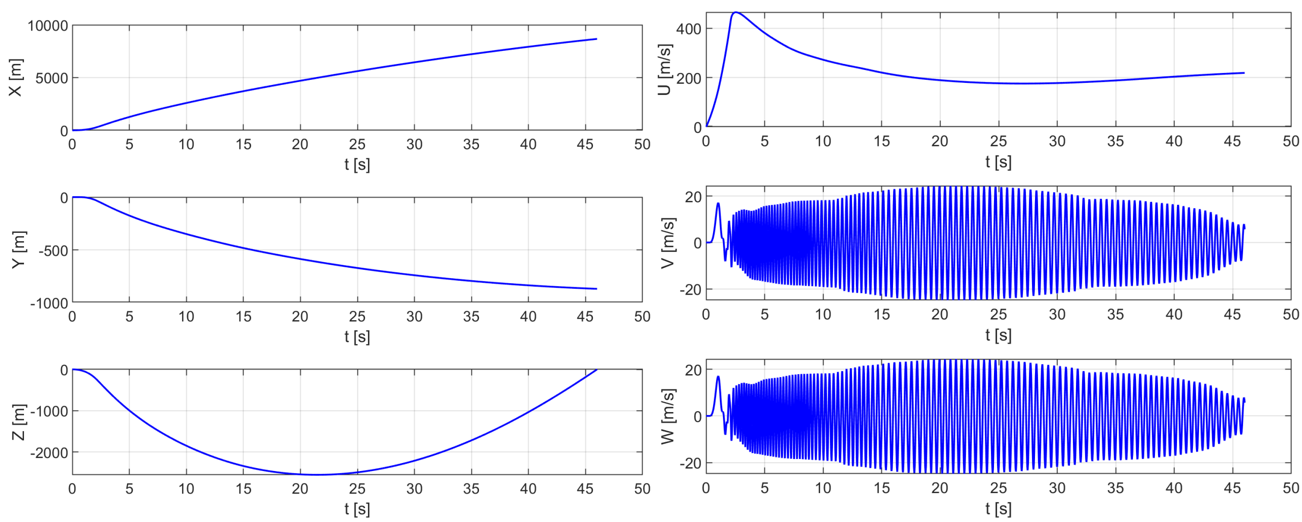

Figure 4 – Missile's trajectory components in North-East-Down frame (left) and linear velocity components in Body frame (right)

where $f(x)$ is the cost function, $h_i(x)$ are the equality constraints, and $h_j(x)$ are the inequality constraints, which then can be solved by a non-linear programming solver. In this work two solvers are used, the open-source solver IPOPT and a solver developed by the author. IPOPT [21] is an open-source package written in C++ used to solve large-scale non-linear programming problems with constraints that can be formulated as in (3). It implements an interior-point method with line-search filter method for step selection. The details of the implemented algorithm can be found in [22]. Based on the book by Nocedal and Wright [23] an unconstrained optimization solver was developed by the author, which will be addressed here as OptimizeUnc. It is based on the Trust Region (TR) approach for the step selection and Quasi-Newton approximations for the Hessian matrix. In the TR method the next step is selected as a minimizer of a quadratic subproblem of the form:

$$\min_{p \in \mathbb{R}^n} m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p \tag{4}$$

$$||p|| \leq \Delta_k$$

where Δ_k is a TR radius. The model from equation 4 is a quadratic approximation of the cost function in the vicinity of the current optimization point. The TR radius is adjusted appropriately based on the set of rules, to ensure that the model is adequately accurate. The Hessian B_k is calculated using a BFGS update formula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k s_k} \tag{5}$$

where

$$s_k = x_{k+1} - x_k \tag{6}$$

$$y_k = \nabla f_{k+1} - \nabla f_k \tag{7}$$

This subproblem is solved using a dogleg method following the description by Nocedal and Wright. The step length is then calculated as

$$p_k = H_k \nabla f_k \tag{8}$$

using a Quasi-Newton BFGS formula to directly update an inverse of the Hessian using the equation:

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \tag{9}$$

$$\rho_k = \frac{1}{y_k^T s_k}$$

The developed solver is fast and accurate as will be presented in the following sections. Any gradient based optimization solver requires the computation of the cost function gradient, and, if used, the jacobian of the constraints. This can be accomplished using finite difference approximations, like forward or central differences, but the resulting gradient or jacobian can have a significant error resulting from a finite step size used for calculations. Another approach is to use an automatic differentiation. This approach can compute fast and exact, up to machine precision, derivatives of any degree. One of the online available packages for automatic differentiation that can be used in C++ programs is ADOL-C [24, 25]. It is a tool that uses operator overloading features of C++ that allows to compute the derivatives in forms of gradients, jacobians, hessians end other matrix-vector products that are needed in optimization calculations. In this article both the finite difference and automatic gradient will be used and compared in terms of time required for their calculation and the accuracy of the results of the optimization.

4. Optimization problem formulation

The aim of the optimization is to find a control function $\mathbf{u}(t)$ that will cause the missile to hit a predefined ground-based target. The cost function is chosen as the difference between the target point and the last point of the missile trajectory as shown in equation:

$$\min_{\mathbf{u}(t)} J(\mathbf{u}(t)) = (\mathbf{x}_{POS}^{\mathbf{u}} - \mathbf{x}_{CMD})^2 \tag{10}$$

where $\mathbf{x}_{POS}$ is the missile's last position and $\mathbf{x}_{CMD}$ is the target position. Superscript $\mathbf{u}$ means that the trajectory depends implicitly on the control during the flight. The missile's trajectory is calculated using Runge-Kutta fourth order integration scheme. The final time T_f of the integration is not known in advance as the integration should finish when the missile hits the ground. This time depends on the control which changes in every iteration of the optimization. A usual approach in such situations is to transform the integration time span from $0 - T_f$ to $0 - 1$ and to add the final time to optimization parameters. To reduce the dimensions of the problem the control vector was assumed to be piecewise constant during the flight. It was divided into 50 equal subintervals and the value of the control in each of them is being optimized. Therefore the NLP problem has $2 \cdot 50 + 1$ optimization parameters, 50 control values acting in two perpendicular planes of the missile plus a final time of the integration, giving the form:

$$\min_{\mathbf{u}_k, T_f} J(\mathbf{u}_k, T_f) = (\mathbf{x}_{POS}^{\mathbf{u}, T_f} - \mathbf{x}_{CMD})^2 \tag{11}$$

The cost function gradient was calculated in two ways. For the finite difference approximation, the forward differences were used given by the equation:

$$\nabla_u J = \frac{J(u+h) - J(u)}{h} \tag{12}$$

where $h = \sqrt{ME}$, where ME is the machine precision on the order of $1e^{-16}$. In this case, the cost function had to be evaluated 101 times to get the full gradient. For the automatic differentiation a traceless forward mode of ADOL-C was used. From the programming point of view it requires marking the dependent and independent variables before calculating the cost function value and afterwards the gradient is already available. Although it requires only one function evaluation the computational cost is also affected by the overloaded operations required by ADOL-C.

5. Results and Discussion

The procedure for calculating the optimal control values was as follows. First, a full model was run to obtain the reference trajectory, and the coordinates of the impact point were saved as desired target. Then, the initial attitude of the missile launcher was disturbed by few degrees, which served a purpose of accounting for the initial uncertainty of the model. The trajectory was calculated until the apogee, where the state of the missile was saved. As the control was only realized in the descend phase of the flight, to calculate the cost function only the part of the flight, starting from the state saved in apogee, was integrated. Three cases of the initial attitude disturbance were calculated:

- Case 1: 2 degrees in initial azimuth angle
- Case 2: 2 degrees in initial elevation angle
- Case 3: 2 degrees in both azimuth and elevation

Calculations were performed on Intel(R) Core(TM) i5-1135G7 with 16GB RAM. The IPOPT solver was used with its default settings, maximum number of iterations was bounded on 50. Tables 1 - 3, give the results of the cases 1, 2 and 3 respectively. Both solvers were run using finite difference (FD) and automatic (AD) gradients. Number of iteration, number of cost function evaluations, first order optimality condition (gradient norm) and execution time were compared.

Table 1 – Results of the case 1

Solver	Final cost	Iter	F-count	$\|\|\nabla f(x)\|\|$	Time [s]
IPOPT (FD)	$5.199e-10$	50*	1160	$4.36e-5$	–
IPOPT (AD)	$4.193e-16$	16	30	$2.96e-9$	27.383
OptimizeUnc (FD)	$6.542e-12$	17	1855	$2.099e-3$	3.23
OptimizeUnc (AD)	$1.054e-21$	14	17	$9.607e-8$	13.609

Table 2 – Results of the case 2

Solver	Final cost	Iter	F-count	$\|\|\nabla f(x)\|\|$	Time
IPOPT (FD)	$2.846e-10$	50*	910	$5.62e-5$	–
IPOPT (AD)	$2.941e-17$	13	34	$1.40e-9$	31.443
OptimizeUnc (FD)	$7.537e-12$	20	2165	$2.491e-3$	3.691
OptimizeUnc (AD)	$6.680e-20$	21	26	$9.267e-7$	21.896

Table 3 – Results of the case 3

Solver	Final cost	Iter	F-count	$\|\|\nabla f(x)\|\|$	Time
IPOPT (FD)	$2.379e-8$	50*	1215	$6.97e-5$	–
IPOPT (AD)	$1.382e-19$	14	42	$7.63e-11$	39.463
OptimizeUnc (FD)	$8.395e-12$	19	2063	$2.59e-3$	4.096
OptimizeUnc (AD)	$5.353e-22$	19	24	$9.035e-8$	19.621

IPOPT solver was unable to converge to a solution using finite difference gradient in any case. It reached maximum number of iterations limit, due to the fact that the errors in finite difference gradient approximations did not allow it to fulfill the default convergence criteria. For the AD case it found a solution in around 15 iterations. The OptimizeUnc solver managed to converge in all cases. Using FD gradients proved to be faster, even though the number of function calls is much higher than that for AD. However, the solutions are always better using the AD, both the final cost value and the gradient norm are much smaller. OptimizeUnc solver was faster than IPOT and managed to converge to a smaller value of the cost function. However, the gradient magnitude was one few orders of magnitude smaller for the IPOPT cases. It has to be pointed out that IPOPT uses a rather complicated algorithms best suited for a large-scale constrained problems, which was not the case in those calculations. OptimizeUnc used an algorithm tailored for the problem and simpler, as all of the cases were unconstrained problems.

Figures 5 - 7 present the optimal trajectory of the missile and the resulting control parameter for the OptimizeUnc AD cases.

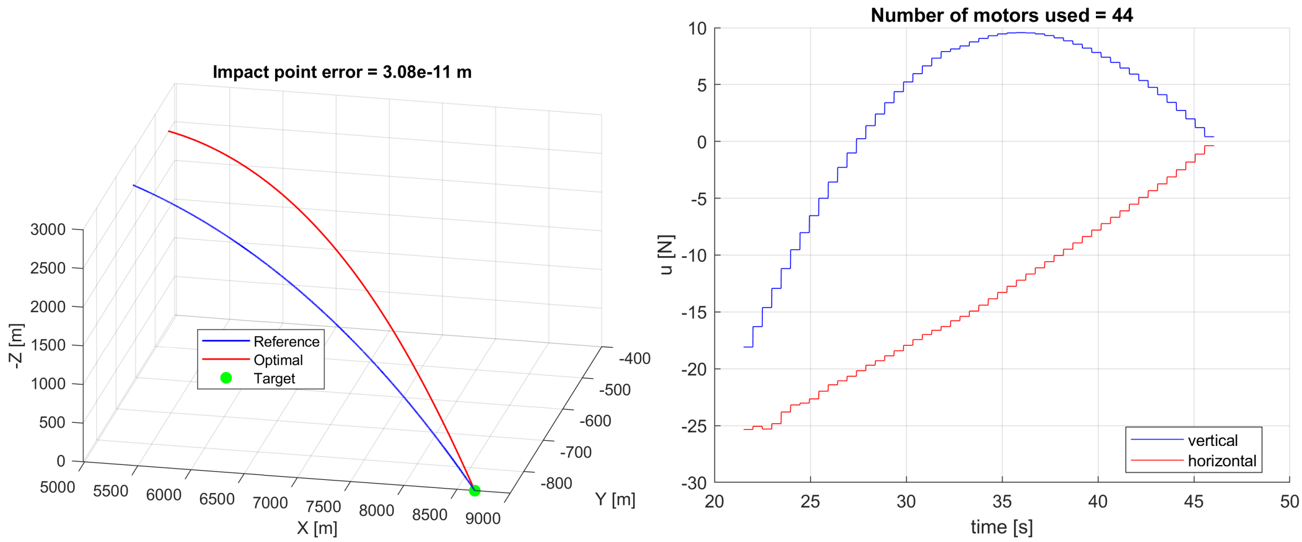

Figure 5 – Missile's trajectory after optimization in North-East-Down frame (left) and resulting optimal values of the control parameter (right) for case 1

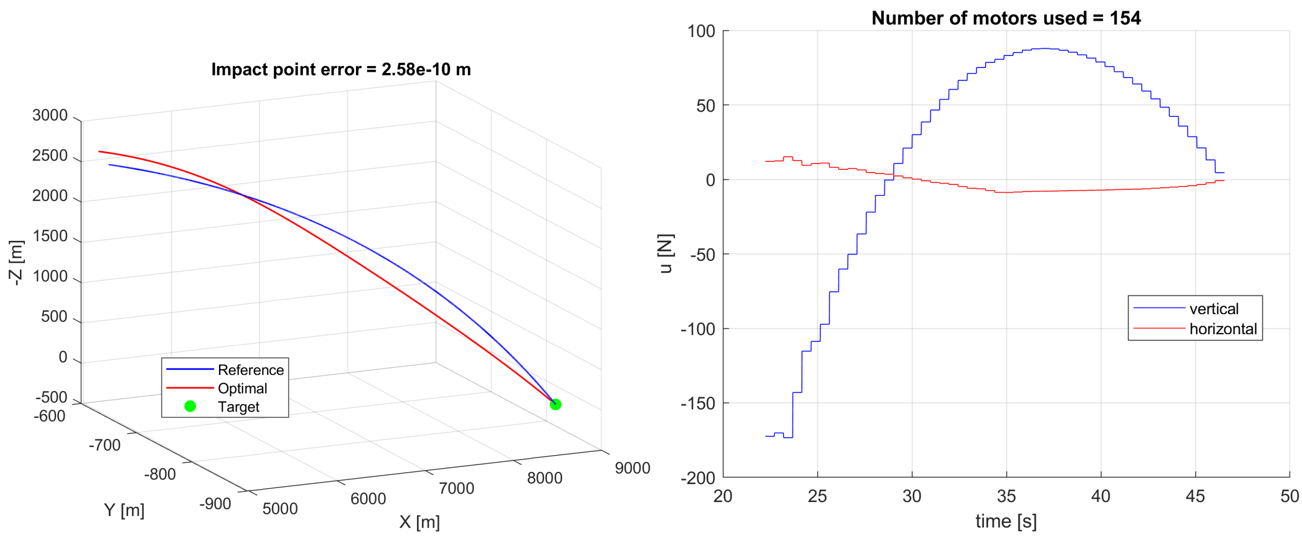

Figure 6 – Missile's trajectory after optimization in North-East-Down frame (left) and resulting optimal values of the control parameter (right) for case 2

The missile trajectory is presented on the left side of the Figures. The reference trajectory is shown in blue, and the optimal one in red. Additionally, the target impact point is shown as a green dot.

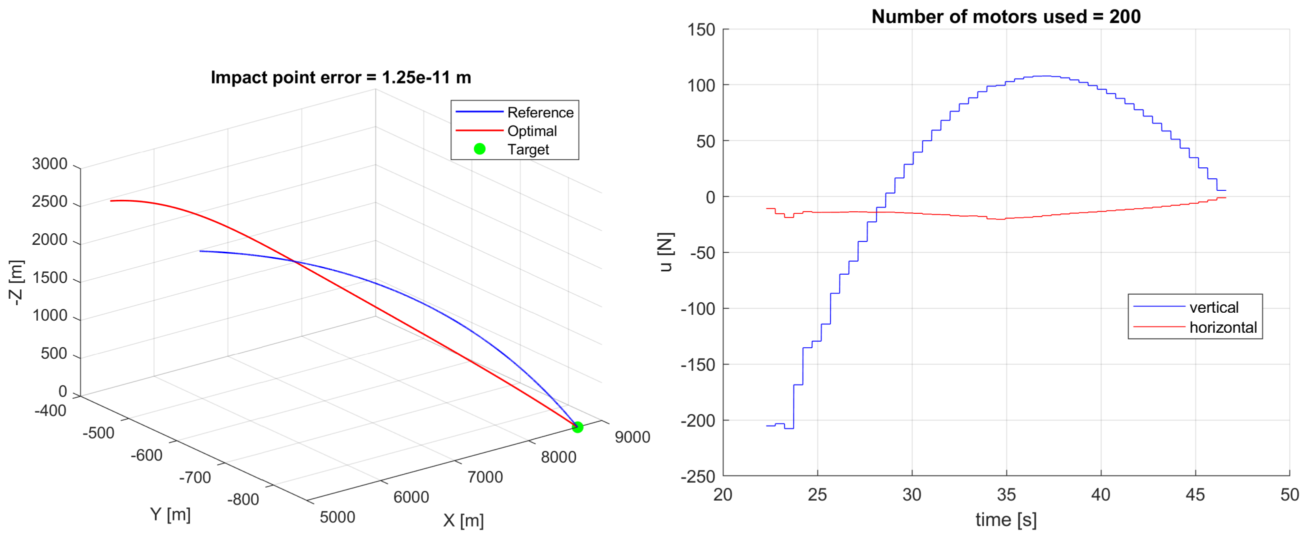

Figure 7 – Missile's trajectory after optimization in North-East-Down frame (left) and resulting optimal values of the control parameter (right) for case 3

Both trajectories start at the origin of the North-East-Down frame (point 0,0,0), but are shown only in the descending phase of the flight. Missile initial attitude angles are 0 degrees in azimuth and 45 degrees in elevation for the reference trajectory, to which the disturbances are added depending on the case. The impact point error was calculated as the square root of the final cost value. In every case, it was a perfect hit. The right-hand side of the Figures show the resulting control parameter values. They are the values of the lateral thrust of the simplified control system. It was assumed that one real thruster can give about 680 N of thrust during 30 ms of burning time. Knowing also the total impulse of a single motor, which can be calculated as:

$$I_c = \int_0^T F(t)dt \tag{13}$$

the total number of required motors can be calculated using the equation:

$$NUM = ceil\left(\frac{1}{I_c} \cdot \int_0^T (|u_1(t)| + |u_2(t)|)dt\right) \tag{14}$$

where $ceil$ is a function that finds the nearest greater integer value. It can be seen that for the simulated cases the missile should be equipped with many thrusters to ensure the direct hit.

6. Conclusions

The article presented the use of direct optimal control methods to steer a missile towards a predefined ground-based target. Two solvers where used, open-source IPOPT, for large-scale non-linear constrained optimization, and OptimizeUnc developed by the author, for the unconstrained non-linear optimization problems. Also, the accuracy of the results was compared depending on the procedure for calculating the cost function gradient required by the solvers, forward finite difference approximation was compared with automatic differentiation using ADOL-C package. The use of automatic differentiation was shown to give much more accurate results. The final cost value was lower for every case, and the gradient norm at the solution, which is one of the optimality conditions, was also much lower. Despite the need for much more cost function evaluations for the finite difference cases, the calculations were much faster. However, the IPOPT solver did not manage to converge to a solution using the FD approach with the default solver settings. It could be fixed by lowering the convergence criteria. The missile hit the target with great accuracy in all of the cases. The used cost function however does not ensure that the missile will hit the target using minimum energy. The control cost, presented as the number of motors used, was rather high especially in cases 2 and 3. In the future work different types of cost functions could be compared to try to lower the energy consumption by the control system. The approaches with penalty parameters or constraints for the control parameters could be used. The indirect methods for the optimal control problems are under development by the author, where the use of Pontryagin's Maximum Principle solved with the reduced gradients or shooting methods will be performed.

7. Contact Author Email Address

mailto: dariusz.miedzinski2.dokt@pw.edu.pl

8. Copyright Statement

References

[1] Gupta, S. K. Saxena, S. Singhal, A. Ghosh, A. K. Trajectory Correction Flight Control System using Pulsejet on an Artillery Rocket, *Defence Science Journal*, Vol. 58, No. 1, pp. 15-33, 2008

[2] Ożóg, R. Jacewicz, M. Głębocki, R. Modified Trajectory Tracking Guidance for Artillery Rocket, *Journal of Theoretical and Applied Mechanics*, Vol. 58, No. 3, pp. 611-622, 2020

[3] Głębocki, R. Jacewicz, M. Parametric Study of Guidance of a 160-mm Projectile Steered with Lateral Thrusters, *aerospace*, Vol. 7, No. 61, 2020

[4] Pavic, M. Pavkovic, B. Mandic, S. Zivkovic, S. Cuk, D. Pulse-frequency modulated guidance laws for a mortar missile with a pulse jet control mechanism, *The Aeronautical Journal*, Vol. 119, No. 1213, pp. 389-405, 2015

[5] Jitpraphai, T. Burchett, B. Costello, M. A Comparison of Different Guidance Schemes for a Direct Fire Rocket With a Pulse Jet Control Mechanism, *Army Research Laboratory*, ARL-CR-493, 2002

[6] Burchett, B. Costello, M. Model Predictive Lateral Pulse Jet Control of an Atmospheric Rocket, *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 5, 2002

[7] Caillau, J. B. Ferretti, R. Trelat, E. Zidani, H. Numerics for finite-dimensional optimal control problems, 2022, hal-03707475

[8] Rao, A. V. A survey of numerical methods for optimal control, *Advances in the Astronautical Sciences*, Vol. 135, 2010

[9] Williams P. Jacobi Pseudospectral Method for Solving Optimal Control Problems. *Journal of Guidance*, Vol. 27, No. 2, pp 293-297, 2003

[10] Vlassenbroeck, J. Van Dooren, R. A Chebyshev Technique for Solving Nonlinear Optimal Control Problems, *IEEE Transactions on Automatic Control*, Vol. 33, No. 4, pp. 333-340, 1988

[11] Vlassenbroeck, J. A Chebyshev Polynomial Method for Optimal Control with State Constraints, *Automatica*, Vol. 24, No. 4, pp. 499-506, 1988

[12] Ross, M. Fahroo, F. Legendre Pseudospectral Approximations of Optimal Control Problems, *New Trends in Nonlinear Dynamics and Control*, LNCIS 295, pp. 327-342, 2003

[13] Williams, P. A Gauss-Lobatto quadrature method for solving optimal control problems. *Proceedings of the 7th Biennial Engineering Mathematics and Applications Conference*, EMAC-2005, Vol. 47, pp 101-115, 2006

[14] von Stryk, O. Numerical Solution of Optimal Control Problems by Direct Collocation, *Optimal Control - Calculus of Variations, Optimal Control Theory and Numerical Methods, International Series of Numerical Mathematics*, Vol. 111, pp. 129-143, 1993

[15] Reddien, G. W. Collocation at Gauss Points as a Discretization in Optimal Control, *SIAM Journal of Control and Optimization*, Vol. 17, No. 2, pp. 298-306, 1979

[16] Adamu, S. Aduroja, O. O. Bitrus, K. Numerical solution to optimal control problems using collocation method via Pontryagin's principle, *FUDMA Journal of Sciences*, Vol. 7, No. 5, pp. 228-233

[17] Trent, A. Venkataraman, R. Doman, D. Trajectory Generation Using A Modified Simple Shooting Method. *IEEE Aerospace Conference Proceedings*, Big Sky, MT, USA, Vol. 4, pp. 2723-2729, 2004

[18] Bonnans, J. F. The shooting approach to optimal control problems, *IFAC International Workshop on Adaptation and Learning in Control and Signal Processing*, Giri, Fouad, Caen, France, pp. 281-292, 2013

[19] Fraser-Andrews, G. Shooting Method for the Numerical Solution of Optimal Control Problems with Bounded State Variables, *Journal of Optimization Theory and Applications*, Vol. 89, No. 2, pp. 351-372, 1996

[20] Jacewicz, M. Lichota, P. Miedziński, D. Głębocki, R. Study of Model Uncertainties Influence on the Impact Point Dispersion for a Gasodynamicaly Controlled Projectile, *Sensors* Vol. 22, No. 3257, 2022

[21] Wachter, A. Short Tutorial: Getting Started With Ipopt in 90 minutes. In Combinatorial Scientific Computing. Dagstuhl Seminar Proceedings, Vol. 9061, pp. 1-17, Schloss Dagstuhl - Leibniz-Zentrum fur Informatik, 2009

[22] Wachter, A. Biegler, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming* Vol. 106, pp 25-57, 2006

[23] Nocedal, J. Wright, S. J. Numerical Optimization, *Springer*, New York, NY, USA, 2006

[24] Walther, A. Griewank, A. ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++, 2016

[25] Walther, A. Getting Started with ADOL-C, In Combinatorial Scientific Computing. Dagstuhl Seminar Proceedings, Vol. 9061, pp. 1-10, Schloss Dagstuhl - Leibniz-Zentrum fur Informatik, 2009