

UNIVERSITY OF BELGRADE
FACULTY OF MECHANICAL ENGINEERING

Luka N. Miličić

**NEW ALGORITHM FOR OPTIMAL IMPACT
ANGLE GUIDANCE**

Doctoral dissertation

Belgrade, 2026.

УНИВЕРЗИТЕТ У БЕОГРАДУ
МАШИНСКИ ФАКУЛТЕТ

Лука Н. Миличић

**НОВИ АЛГОРИТАМ ОПТИМАЛНОГ
ВОЂЕЊА СА ОГРАНИЧЕЊЕМ УПАДНОГ
УГЛА**

Докторска дисертација

Београд, 2026.

Doctoral Dissertation Mentor:

Dr. Miloš Marković, Associate Professor,
University of Belgrade, Faculty of Mechanical Engineering

Doctoral Dissertation Defense Committee Members:

Dr. Predrag Elek, Full Professor,
University of Belgrade, Faculty of Mechanical Engineering

Dr. Miloš Marković, Associate Professor,
University of Belgrade, Faculty of Mechanical Engineering

Dr. Aleksandar Obradović, Full Professor,
University of Belgrade, Faculty of Mechanical Engineering

Dr. Milan Ristanović, Full Professor,
University of Belgrade, Faculty of Mechanical Engineering

Dr. Oleg Cherkasov, Associate Professor,
Lomonosov Moscow State University,
Faculty of Mechanics and Mathematics

Defense Date: _____.

Ментор докторске дисертације:

др Милош Марковић, ванредни професор,
Универзитет у Београду, Машински факултет

Чланови комисије за одбрану докторске дисертације:

др Предраг Елек, редовни професор,
Универзитет у Београду, Машински факултет

др Милош Марковић, ванредни професор,
Универзитет у Београду, Машински факултет

др Александар Обрадовић, редовни професор,
Универзитет у Београду, Машински факултет

др Милан Ристановић, редовни професор,
Универзитет у Београду, Машински факултет

др Олег Черкасов, доцент,
Московски државни универзитет „Ломоносов“,
Факултет за механику и математику

Датум одбране:_____.

Acknowledgements

The completion of this dissertation would not have been possible without the support, patience, and knowledge of people who accompanied me throughout my education and professional development.

First and foremost, I would like to express my deepest gratitude to my family for their unconditional support throughout my entire education. Some of the people who contributed greatly to the person I have become are, unfortunately, no longer with us. I sincerely wish they could have known that this dissertation has been completed, and I dedicate a part of this achievement to their memory.

I am grateful to my closest friends for their patience and understanding throughout all these years, especially during the periods when my work on this dissertation left little time for anything else.

I would also like to thank my colleagues from the Weapon Systems Department of the Faculty of Mechanical Engineering, as well as the members of the Doctoral Dissertation Defense Committee for their careful evaluation of the dissertation and for their valuable comments and suggestions.

I would like to express my gratitude to Dr. Aleksandar Pejčev for his time, our joint work on several mathematical topics related to this dissertation, and, perhaps most importantly, for his genuine interest in how the dissertation was progressing.

I am grateful to EDePro and to all colleagues involved in the projects related to this research for providing the opportunity to apply and verify the ideas in practical engineering environments.

Two people deserve particular recognition for their contribution to the professional development of this work.

I am deeply grateful to Dr. Ivana Todić, with whom the idea that ultimately led to this dissertation began. I am especially grateful for her many useful guidelines and discussions during our everyday work, as well as for the practical foundations she established through her work, upon which many of the ideas developed in this dissertation could subsequently be practically verified.

My deepest professional gratitude goes to Dr. Aleksandar Obradović, who introduced me to optimal control theory and whose mentorship was fundamental to this research. I am grateful for his effort in deriving and verifying several of the results presented in this dissertation, for his innovative ideas on the problems we encountered, and for his guidance throughout my doctoral studies.

Finally, and most importantly, I would like to thank my wife, Milica, and our three daughters, Ana, Mitra, and Nikolija, for their love, patience, and understanding throughout these years. This dissertation is, in many ways, also a reflection of the patience and sacrifices of the people closest to me, and I am deeply grateful to them for that.

NEW ALGORITHM FOR OPTIMAL IMPACT ANGLE GUIDANCE

Abstract

This dissertation presents a computationally efficient nonlinear optimal guidance method for flight vehicles required to achieve a prescribed flight-path angle at the terminal point. Although the methodology is primarily motivated by tactical missile applications, the proposed formulation is general and applicable to a broad class of flight vehicles regardless of their mission profile or control architecture. A comprehensive review of existing guidance methods reveals that most optimal impact-angle guidance laws rely on linearized engagement models and require additional navigation quantities, such as time-to-go or line-of-sight rate estimates, which reduce their applicability outside the vicinity of the nominal collision geometry. While exact nonlinear formulations have been reported, their computational complexity has largely prevented practical real-time implementation on embedded flight computers. The primary objective of this dissertation is therefore to bridge the gap between nonlinear optimal control theory and practical onboard implementation by developing an exact nonlinear guidance method with computational requirements compatible with real operational systems. The proposed guidance law is derived directly from the nonlinear engagement model without employing linearization or small-angle approximations. Assuming constant vehicle speed and planar motion, an optimal control formulation minimizing control effort is established. The resulting optimal solution is transformed into a computationally efficient algorithm requiring the solution of only a single nonlinear equation. Furthermore, the guidance law is obtained in closed-feedback form using only standard navigation data, namely vehicle position and velocity, eliminating the need for additional measurements. A closed-form analytical solution is additionally derived for the practically important climb-to-cruise-altitude problem. Special attention is devoted to the numerical aspects of the proposed method. Efficient algorithms are developed for evaluating the nonlinear equation and the associated elliptic integrals, accompanied by a detailed error analysis and execution-time optimization. The complete algorithm is implemented on representative embedded hardware, where deterministic computational burden and substantial real-time timing margins are demonstrated. The proposed guidance method is validated through extensive numerical simulations, hardware-in-the-loop, and field testing. Its performance is examined for cruise and ballistic missile applications under realistic disturbances, navigation errors, propulsion uncertainties, and implementation constraints. The results demonstrate that the proposed guidance law generates unique energy-optimal trajectories, accurately achieves prescribed impact angles, and remains suitable for real-time execution on resource-constrained flight computers.

Key words: guidance laws, optimal control, flight-path angle, impact angle, numerical simulation, real-time algorithm

Scientific discipline: Mechanical engineering

Scientific subdiscipline: Military engineering - Weapon systems, Missile systems

UDK number:

НОВИ АЛГОРИТАМ ОПТИМАЛНОГ ВОЂЕЊА СА ОГРАНИЧЕЊЕМ УПАДНОГ УГЛА

Резиме

У дисертацији је представљена рачунарски ефикасна метода нелинеарног оптималног вођења за летелице код којих се захтева постизање жељеног угла вектора брзине у тачки циља. Иако се примењена методологија првенствено односи на тактичке ракете, предложена метода је општег карактера и применљива на широку класу летелица, без обзира на њихову мисију или начин управљања. Детаљним прегледом постојећих метода вођења установљено је да се већина оптималних метода за остваривање жељених углова путање заснива на линеаризованим моделима сусрета ракете и циља, и да такве методе захтевају естимацију додатних величина током лета, као што су преостало време до сусрета са циљем или угаона брзина линије визирања циља. Због уведених апроксимација опада и применљивост ових метода ван околине почетног троугла претицања. Иако постоје методе засноване на тачним нелинеарним моделима сусрета, њихова рачунска комплексност углавном ограничава практичну употребу у реалном времену на летним рачунарима. Због тога је примарни задатак ове дисертације развој методе оптималног вођења засноване на нелинеарном моделу сусрета, која је у погледу рачунске комплексности компатибилна са постојећим системима у оперативној употреби. Предложени закон вођења изведен је без апроксимација у смислу линеаризације или претпоставки о малим угловима. Усвајајући равански модел краткања летелице константне брзине, установљен је оптимизациони проблем одређивања управљачке величине - команде вођења, у циљу минималног утrophка енергије. Процедура одређивања оптималне команде сведена је на решавање једне нелинеарне једначине по једној непознатој. Штавише, закон вођења је дат у затвореној форми по стањима која су део стандардних навигационих података: позиције и брзине. Додатно, за специјалан случај пењања крстареће ракете предложен је закон вођења у затвореном облику без итеративног поступка. Посебна пажња посвећена је нумеричким методама које су саставни део предложеног поступка. Предложени су ефикасни алгоритми израчунавања свих величина неопходних за решавање нелинеарне једначине, уз анализу њихових грешака и оптимизацију у циљу примене у реалном времену. Комплетан алгоритам имплементиран је на реалном хардверу и постигнути су фиксно време извршавања и значајне резерве у погледу рачунарског времена. Предложени поступак валидиран је кроз нумеричке симулације, лабораторијска и летна испитивања. Верификација је обављена за два система од интереса: крстарећу и балистичку ракету, и то у присуству реалних поремећаја, навигационих грешака, несавршености у изради, и ограничених маневарских способности. Резултатима је показано да се предложеним поступком остварују јединствене трајекторије које су оптималне са становишта управљачке енергије, да се постижу жељени упадни углови, и да је метода погодна за примену у реалном времену на летним рачунарима ограничених рачунарских могућности.

Кључне речи: закони вођења, оптимално управљање, угао путање, упадни угао, нумеричка симулација, алгоритми у реалном времену

Научна област: Машинство

Ужа научна област: Војно машинство - Системи наоружања, Ракетно наоружање

УДК број:

Contents

1	Introduction	1
1.1	General Considerations	1
1.2	Literature Review and Historical Background	6
1.2.1	Markov–Dubins Model	6
1.2.2	Zermelo Navigation Problem	8
1.2.3	Proportional Navigation	9
1.2.4	Explicit Guidance	14
1.2.5	Linear Optimal Guidance Laws with Impact Angle Constraint	19
1.2.6	Nonlinear Optimal Guidance Laws with Impact Angle Constraint	27
1.2.7	Non-Optimal Guidance Laws with Impact Angle Constraint	33
1.2.8	Three-Dimensional Guidance Laws with Impact Angle Constraint	38
1.3	Subject and Objectives of the Dissertation	40
1.3.1	Starting Hypotheses	42
1.3.2	Scientific Research Methods	43
1.3.3	Expected Scientific Contribution	44
1.4	Structure of the Dissertation	45
2	Optimal Guidance	46
2.1	Introduction	46
2.2	Optimal Guidance Algorithms	47
2.2.1	Kinematic Model of Missile Motion	47
2.2.2	Optimization Problem	48
2.2.3	Shooting Method	54
2.2.4	General Case of Prescribed Terminal Flight-Path Angle	55
2.2.5	Special Case: Cruise Missile Climb Phase	82
2.3	Comparison with Existing Methods	90
2.3.1	Numerical Simulation Setup	91
2.3.2	Comparison with Linear Optimal Guidance Laws	92
2.3.3	Comparison with Nonlinear Optimal Guidance Laws	101
2.3.4	Comparison with Non-Optimal Guidance Laws	106
2.3.5	Overview of Existing Methods Results	116
3	Application of Optimal Guidance Algorithms	119

3.1	Introduction	119
3.2	Real-Time Application Analysis	120
3.2.1	Numerical Simulation Setup	120
3.2.2	Solution to the Nonlinear Equation	122
3.2.3	Numerical Integration	132
3.2.4	Influence of Variable Velocity	151
3.2.5	Influence of Delay in Updating Optimal Guidance Parameters	153
3.3	Hardware Implementation	156
3.4	Numerical Simulation of Missile Flight with Six Degrees of Freedom	168
3.4.1	Cruise Missile	169
3.4.2	Ballistic Missile	204
3.5	Hardware-in-the-Loop Validation and Results	210
3.6	Flight Test	216
4	Conclusion	223
	Bibliography	228
	Appendices	236
A	Minimum Principle	237
B	Numerical Methods	241
B.1	Solving the Nonlinear Equation	241
B.1.1	Bisection Method	241
B.1.2	Derivatives Required for Root-Finding via Newton's Method	242
B.2	Numerical Integration	244
B.2.1	Gauss–Legendre Quadrature Formula	244
B.3	Minimax Polynomial Approximation	246
C	Six-Degree-of-Freedom Simulation Model	247
C.1	Coordinate Frames	247
C.2	Mathematical Model	250
	Biography	257

List of Figures

1.1	The V-2 rocket and its contribution to the early development of space exploration. . .	2
1.2	Schematic representation of a closed-loop Guidance–Navigation–Control system. . .	2
1.3	Representative applications of guidance with orientation constraint.	5
1.4	Planar kinematic model of the Markov–Dubins problem.	8
1.5	Zermelo Navigation Problem.	9
1.6	The <i>Lark</i> surface-to-air missile was the first missile on which proportional navigation was implemented as a guidance method.	10
1.7	Missile–target engagement geometry.	11
1.8	Block diagram of the linearized missile–target engagement model employing proportional navigation (PN).	13
1.9	Specifying the velocity-vector components at the interception point is equivalent to specifying the impact angle.	15
1.10	Kinematic model for determining relative deviation for the formulation of the optimal guidance problem.	19
1.11	Missile–stationary-target engagement geometry.	23
1.12	A possible trajectory between two points with specified directions consists of a circular arc and two line segments.	31
1.13	Three-dimensional guidance in two phases: steering into the terminal velocity vector plane (trajectory segment <i>MP</i>) and planar guidance (segment <i>PT</i>)	39
2.1	Planar kinematic models of missile motion.	48
2.2	Optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -30^\circ$, and velocity $V = 160$ m/s.	57
2.3	Flight parameters along the optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -30^\circ$, and velocity $V = 160$ m/s.	58
2.4	Singularity analysis and solution of Eq. (2.46) for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -30^\circ$, and velocity $V = 160$ m/s.	60
2.5	Singularity analysis and solution of Eq. (2.46) for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 35^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.	61
2.6	Optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 35^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.	61

2.7	Solutions of Eq. (2.52) for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $x_1 = 1000$ m, $y_1 = 0$ m, and velocity $V = 160$ m/s.	62
2.8	Phase trajectory for Case A: $\gamma' < 0$ throughout the entire trajectory.	62
2.9	Optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.	63
2.10	Flight parameters along the optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.	63
2.11	Optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 500$ m, $\gamma_0 = -60^\circ$, $x_1 = 500$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.	64
2.12	Flight parameters along the optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 500$ m, $\gamma_0 = -60^\circ$, $x_1 = 500$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.	64
2.13	Phase trajectory for Case B: $\gamma' > 0$ along the first segment of the trajectory and $\gamma' < 0$ along the second segment.	65
2.14	Solutions of Eq. (2.52) for different examples of Case B. $x_0 = 0$ m, $y_1 = 0$ m, $V = 160$ m/s.	66
2.15	Phase trajectory for Case C: $\gamma' < 0$ on the first segment of the trajectory, $\gamma' > 0$ on the second segment of the trajectory.	67
2.16	Optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -10^\circ$, and velocity $V = 160$ m/s.	68
2.17	Flight parameters along the optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -10^\circ$, and velocity $V = 160$ m/s.	68
2.18	Optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 500$ m, $\gamma_0 = -60^\circ$, $x_1 = 200$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.	69
2.19	Flight parameters along the optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 500$ m, $\gamma_0 = -60^\circ$, $x_1 = 200$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.	69
2.20	Solutions of Eq. (2.52) for different examples of Case C. $x_0 = 0$ m, $y_1 = 0$ m, $V = 160$ m/s.	70
2.21	Phase trajectory for Case D: $\gamma' > 0$ on the entire trajectory.	70
2.22	Optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = -45^\circ$, $x_1 = 500$ m, $y_1 = 0$ m, $\gamma_1 = -10^\circ$, and velocity $V = 160$ m/s.	71
2.23	Flight parameters along the optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = -45^\circ$, $x_1 = 500$ m, $y_1 = 0$ m, $\gamma_1 = -10^\circ$, and velocity $V = 160$ m/s.	72
2.24	Solution of Eq. (2.52) for the example analyzed in Case D.	72
2.25	Example in which the proposed procedure cannot determine the optimal trajectory – unrealistic case: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 85^\circ$, $x_1 = 200$ m, $y_1 = 0$ m, $\gamma_1 = -85^\circ$, $V = 160$ m/s.	75
2.26	Absence of a zero of the function $\mathcal{N}(\alpha)$ for the example shown in Fig. 2.25.	75

2.27	Example in which the proposed procedure cannot determine the optimal trajectory – realistic case: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = -45^\circ$, $x_1 = 200$ m, $y_1 = 0$ m, $\gamma_1 = -45^\circ$, $V = 160$ m/s.	76
2.28	Absence of a zero of the function $\mathcal{N}(\alpha)$ for the example shown in Fig. 2.27.	77
2.29	Possible initial positions of the missile relative to the line defined by the desired velocity vector at the target.	78
2.30	Geometric classification of engagement scenarios based on relative separation and velocity vector orientation.	78
2.31	Optimal trajectories for the boundary conditions: $x_0 = 0$ m, $\gamma_0 = 0^\circ$, $x_1 = 300$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and $V = 160$ m/s.	79
2.32	Flowchart representation of the Algorithm 2.	81
2.33	Kinematic model of the cruise missile climb phase.	82
2.34	Optimal trajectory for boundary conditions: $x_0 = 200$ m, $y_0 = 100$ m, $\gamma_0 = 25^\circ$, $y_1 = 200$ m, $\gamma_1 = 0^\circ$, and velocity $V = 160$ m/s. x_1 is free.	86
2.35	Flight parameters on the optimal trajectory for boundary conditions: $x_0 = 200$ m, $y_0 = 100$ m, $\gamma_0 = 25^\circ$, $y_1 = 200$ m, $\gamma_1 = 0^\circ$, and velocity $V = 160$ m/s. x_1 is free.	86
2.36	Comparison of trajectories obtained using (2.89) — fixed x_1 , and (2.85) — free x_1	87
2.37	Comparison of trajectories obtained using (2.89) — fixed x_1 , and (2.85) — free x_1	88
2.38	Comparison of trajectories obtained using Algorithm 2 — fixed x_1 , and using ex- pression (2.85) — free x_1	88
2.39	Comparison of trajectories obtained using Algorithm 2 — fixed x_1 , and using ex- pression (2.85) — free x_1	89
2.40	Dependence of the performance index, obtained by solving optimization problem (2.6) using Algorithm 2, on the prescribed value of x_1	90
2.41	Comparison of results obtained using Algorithm 2 (OPT) and TSG guidance.	93
2.42	Comparison of results obtained using expression (2.103) for $n = 0$ (OGL/IAC-0, $n = 0$), and different methods for determining t_{go}	95
2.43	Comparison of results obtained using Algorithm 2 (OPT) and expression (2.103) (OGL/IAC-0) for $n = 1$, with $t_{go} = R/V_c$	97
2.44	Comparison of results obtained using Algorithm 2 (OPT) and expression (2.109) (OGL-CTIA), with $t_{go} = R/V_c$	98
2.45	Comparison of results obtained using Algorithm 2 (OPT) and proportional naviga- tion (PN) given by expressions (2.110) and (2.111).	100
2.46	Comparison of results obtained using the Algorithm 2 (OPT) and SDRE guidance.	102
2.47	Results obtained using the guidance law NOIACG (2.116) for different values of parameters N and M , for the case $\gamma_1 = -30^\circ$	103
2.48	Results obtained using the guidance law NOIACG (2.116) for different values of parameters N and M , for the case $\gamma_1 = -60^\circ$	104
2.49	Results obtained using the guidance law NOIACG (2.116) for different values of parameters N and M , for the case $\gamma_1 = -90^\circ$	105

2.50	Comparison of results obtained using the Algorithm 2 (OPT) and NOIACG guidance law (2.116) with $N = 3$ and $M = 1$	106
2.51	Comparison of results obtained using Algorithm 2 (OPT) and the guidance law Kim1998.	108
2.52	Comparison of results obtained using Algorithm 2 (OPT) and the guidance law Ratnoo2008, for the case of terminal guidance after cruise phase.	109
2.53	Comparison of results obtained using Algorithm 2 (OPT) and and the guidance law Ratnoo2008, for the case of direct engagement.	110
2.54	Comparison of results obtained using Algorithm 2 (OPT) and the guidance law Erer2012, for the case: $t_{\text{beg}} = 0$ s, $t_{\text{end}} = 5$ s, $k = 3$	112
2.55	Comparison of results obtained using Algorithm 2 (OPT) and the guidance law Erer2012, for the case: $t_{\text{beg}} = 0$ s, and t_{end} equal to the flight time of the optimal trajectory.	113
2.56	Comparison of results obtained using Algorithm 2 (OPT) and the guidance law IACG.	115
3.1	Results obtained by applying Algorithm 2 in real time. Numerical solution of $\mathcal{N}(\alpha) = 0$ using the bisection method, numerical integration using the trapezoidal rule.	121
3.2	Control input for different values of the absolute error of the solution of the equation $\mathcal{N}(\alpha) = 0$, for the scenario described in Table 3.1, $\gamma_1 = -60^\circ$	122
3.3	Upper bound of the absolute error ϵ_α for the adopted control input error ϵ_u , for the scenario described in Table 3.1. The figure on the right is a magnified view of the left figure.	125
3.4	Comparison of the number of iterations required to achieve an absolute accuracy of $\epsilon = 10^{-6}$ using different methods for solving the equation $\mathcal{N}(\alpha) = 0$	127
3.5	Correct subinterval selection and potential errors in its determination.	128
3.6	Variation of the solution of the equation $\mathcal{N}(\alpha) = 0$ during the numerical simulation for the case $\gamma_1 = -60^\circ$	129
3.7	Comparison of the number of iterations required by different methods using an adaptive interval.	130
3.8	Shape of the function $\mathcal{N}(\alpha)$ in the vicinity of the inflection point ($300 \text{ m} < x < 400 \text{ m}$) for the case $\gamma_1 = -60^\circ$	130
3.9	Solution of the equation $\mathcal{N}(\alpha) = 0$ expressed as a percentage of the interval (2.50).	131
3.10	The allowable error of the control variable determines the allowable error of numerical procedures.	136
3.11	Allowable error $\epsilon_{\mathcal{I}_1}$ for the adopted value of the control variable error ϵ_u , for the scenario described in Table 3.1. The right figure is a zoomed-in version of the left one.	137
3.12	Absolute error in the evaluation of the integral I_1 using the trapezoidal rule for different numbers of subintervals (n).	137
3.13	Integrand of the integral I_1 , given by (2.42), over the interval $\xi \in [-\pi/2, \pi/2]$	138

3.14	Absolute error in the evaluation of the integral I_1 using the Gauss-Legendre quadrature formula with 10 nodes.	139
3.15	Absolute error in the evaluation of the integral I_1 using the Gauss-Legendre quadrature formula with 10 nodes.	140
3.16	Absolute error in the evaluation of the integral I_1 using the Gauss-Legendre quadrature formula with 10 nodes, computed in two different ways. The figure on the right is a zoomed-in version of the figure on the left.	141
3.17	Spatial partitioning of integration limits: possible positions of upper limit b relative to threshold c^* and the resulting domains (colored area) for lower limit a	144
3.18	Absolute error in the evaluation of the integral I_1 using Algorithm 3 and a quadrature formula with 10 nodes.	144
3.19	Absolute error in the evaluation of the integral I_1 using Algorithm 3 and a quadrature formula with 10 nodes, for the upper integration limit value at which the maximum error may occur.	145
3.20	Allowable approximation error of integral (3.51) for different values of the control error ϵ_u	148
3.21	Absolute error in the evaluation of the integral (3.51) using the Gauss-Legendre quadrature formula with 10 nodes, computed in two different ways. The figure on the right is a zoomed-in version of the figure on the left.	149
3.22	Integrand of the integral I_S , given by expression (3.51), over the interval $\xi \in [0, \pi/2]$	149
3.23	Absolute error in the evaluation of the integral I_S using a Gauss-Legendre quadrature formula with 10 nodes and the combined method.	151
3.24	Different actual velocity profiles for the scenario described in Table 3.1.	152
3.25	Control-input profiles for different choices of the velocity used in the optimal guidance law, under variable actual velocity conditions.	152
3.26	Control-input profile for different delay levels d in the optimal guidance law (3.75).	154
3.27	Variation of optimal guidance parameters along the trajectory for the scenario described in Table 3.1.	155
3.28	Variation of optimal guidance parameters for different values of the time constant τ in the control-system transfer function.	155
3.29	Control-input profile for different values of the time constant τ in the control-system transfer function.	156
3.30	MCB1700 development board with an LPC1768 microcontroller.	157
3.31	Absolute error in determining the integral I_1 using Algorithm 3, a 10-node quadrature formula, and single-precision arithmetic.	157
3.32	Absolute error in determining the integral I_1 using Algorithm 3, a 10-node quadrature formula, single-precision arithmetic, and minimax polynomial approximations of different orders n	160
3.33	Schematic representation of the device network used during <i>SIL</i> testing.	163
3.34	Execution time of the main flight computer algorithm when proportional navigation is used as the terminal guidance method of a cruise missile.	163

3.35 Schematic flow of the main algorithm with update of guidance parameters using three iterations per basic cycle.	165
3.36 Comparison of the execution time of the main flight computer algorithm when optimal guidance is implemented with different numbers of iterations per main algorithm cycle.	165
3.37 Schematic flow of the main algorithm with update of guidance parameters using two iterations per basic cycle.	166
3.38 Execution time of the main flight computer algorithm of a ballistic missile when different guidance methods are employed.	167
3.39 Kinematic quantities relevant to guidance in the vertical plane.	168
3.40 Typical trajectory of a cruise missile engaging a target at a range of 6.5 km.	170
3.41 Kinematic model of straight-line tracking used for implementation of the guidance law (3.82).	170
3.42 Comparison of results obtained using the trajectory-tracking guidance law (3.83a) (TT) and the optimal guidance law (3.85) (OPT) for a desired cruise altitude of $h_{\text{des}} = 200$ m.	173
3.43 Comparison of results obtained using the trajectory-tracking guidance law (3.83a) (TT) and the optimal guidance law (3.85) (OPT) for a desired cruise altitude of $h_{\text{des}} = 300$ m.	174
3.44 Comparison of results obtained during optimal climbs to different cruise altitudes.	175
3.45 Comparison of results obtained during optimal climbs to different cruise altitudes h_{des} and different values of the altitude deviation Δh_{TT} used for switching to trajectory tracking.	176
3.46 Comparison of results obtained during optimal climb in the presence of flight-path-angle estimation errors.	178
3.47 Comparison of the results obtained using the guidance law (3.85) in the presence of a positive flight-path-angle estimation error ($\hat{\gamma} > \gamma$), with (safety ON) and without (safety OFF) a time-based criterion for switching to trajectory tracking.	179
3.48 Comparison of optimal climb results obtained using the exact flight-path angle and its approximation by the pitch angle.	180
3.49 Comparison of the trajectory obtained using proportional navigation (3.92a) and optimal guidance (3.95) for the desired impact angle $\gamma_1 = -45^\circ$	182
3.50 Influence of the cruise altitude h_{cruise} at which terminal guidance begins, $\gamma_1 = -45^\circ$	184
3.51 Influence of the cruise altitude h_{cruise} at which terminal guidance begins, $\gamma_1 = -65^\circ$	185
3.52 Influence of the cruise altitude h_{cruise} at which terminal guidance begins, $\gamma_1 = -85^\circ$	186
3.53 Influence of the distance Δx_{start} at which terminal guidance begins, $\gamma_1 = -45^\circ$	188
3.54 Influence of the distance Δx_{start} at which terminal guidance begins, $\gamma_1 = -65^\circ$	189
3.55 Influence of the distance Δx_{start} at which terminal guidance begins, $\gamma_1 = -85^\circ$	190
3.56 Effect of mission parameters on final results, $\gamma_1 = -45^\circ$	192
3.57 Effect of mission parameters on final results, $\gamma_1 = -65^\circ$	193
3.58 Effect of mission parameters on final results, $\gamma_1 = -85^\circ$	194

3.59	Shape of the cost function (3.98) for different values of weights w_i . $\gamma_1 = -65^\circ$	196
3.60	Comparison of trajectories for optimal mission parameter selection, $\gamma_1 = -65^\circ$	197
3.61	Effect of errors in the estimated flight-path angle, $\gamma_1 = -65^\circ$	199
3.62	Effect of errors in the estimated pitch angle, $\gamma_1 = -65^\circ$	201
3.63	Effect of erroneous estimated pitch angle, $\gamma_1 = -65^\circ$	202
3.64	Effect of errors in estimated angles on the final results, $\gamma_1 = -65^\circ$	203
3.65	Nominal ballistic trajectory of an unguided artillery rocket at a range of 180 km. . .	205
3.66	Comparison of results for an unguided rocket with total impulse deviations.	206
3.67	Comparison of results obtained using optimal guidance (Algorithm 2) with total impulse deviations, $\gamma_1 = -49.25^\circ$	207
3.68	Comparison of results obtained using optimal guidance (Algorithm 2) with total impulse deviations, $\gamma_1 = -45^\circ$	209
3.69	Schematic of the HIL architecture and communication interfaces.	210
3.70	HIL results for the nominal case, $\gamma_1 = -25^\circ$	212
3.71	Results for the nominal case, highlighting the trajectory-convexity cases.	214
3.72	Results for the six HIL test cases summarized in Table 3.12.	215
3.73	Fifth-generation cruise missiles for land attacks.	216
3.74	Missile trajectory in the horizontal plane.	217
3.75	Flight test results (whole trajectory), $\gamma_1 = -35^\circ$	218
3.76	Flight test results (terminal phase), $\gamma_1 = -35^\circ$	220
3.77	Difference between the estimated altitude profiles during the final phase of flight. . .	221
3.78	Flight test results, highlighting the trajectory-convexity cases.	221
3.79	Fixed forward-looking nose camera used for real-time flight video transmission. . . .	222
3.80	Selected video frames captured during the final phase of flight. The number in the upper-left corner indicates the remaining downrange distance to the target.	222
C.1	Coordinate frames for terrestrial navigation.	248
C.2	Azimuth coordinate frame.	249
C.3	Body coordinate frame.	249
C.4	Aeroballistic coordinate frame.	250
C.5	Roll autopilot block diagram.	256
C.6	Pitch autopilot block diagram.	256

List of Tables

2.1	Simulation parameters for comparison between optimal and linear guidance laws. . .	92
2.2	Control effort obtained using Algorithm 2 (OPT) and TSG guidance.	94
2.3	Control effort obtained using expression (2.103) for $n = 0$ (OGL/IAC-0, $n = 0$), and different methods for determining t_{go}	96
2.4	Control effort obtained using Algorithm 2 (OPT) and expression (2.103) (OGL/IAC-0) for $n = 1$, with $t_{go} = R/V_c$	97
2.5	Control effort obtained using Algorithm 2 (OPT), expression (2.109) (OGL-CTIA), and TSG guidance.	99
2.6	Simulation parameters for comparison of optimal guidance and proportional navigation.	99
2.7	Control effort obtained using Algorithm 2 (OPT) and proportional navigation (PN) given by expressions (2.110) and (2.111).	100
2.8	Control effort obtained using the Algorithm 2 (OPT) and SDRE guidance.	102
2.9	Control effort obtained using the Algorithm 2 (OPT) and the guidance law NOIACG (2.116) for different values of parameters N and M	105
2.10	Control effort obtained using Algorithm 2 (OPT) and the guidance law Kim1998. . .	108
2.11	Control effort obtained using Algorithm 2 (OPT) and the guidance law Ratnoo2008, for the case of terminal guidance after cruise phase.	109
2.12	Simulation parameters for comparison between optimal guidance and the guidance law Ratnoo2008.	110
2.13	Control effort obtained using Algorithm 2 (OPT) and the guidance law Ratnoo2008, for the case of direct engagement.	111
2.14	Control effort obtained using Algorithm 2 (OPT) and the guidance law Erer2012, for the case: $t_{beg} = 0$ s, $t_{end} = 5$ s, $k = 3$	112
2.15	Control effort and impact angle error $\Delta\gamma_1$ obtained using Algorithm 2 (OPT) and the guidance law Erer2012, for $t_{end} = 5$ s, and for t_{end} equal to the flight time of the optimal trajectory. In both cases $t_{beg} = 0$ s was assumed.	113
2.16	Control effort obtained using Algorithm 2 (OPT) and the guidance law IACG. . . .	115
2.17	Results obtained using different methods for the scenario of missile transition from cruise phase to terminal guidance phase toward the target (Table 2.1). $\Delta x = x_1 -$ $x(t_1)$, $\Delta y = y_1 - y(t_1)$, $\Delta\gamma_1 = \gamma_1 - \gamma(t_1)$	117
2.18	Overview of methods used for comparison with optimal guidance.	118
3.1	Simulation parameters for real-time analysis.	120

3.2	Comparison of results for different values of the absolute error ε of the solution of the equation $\mathcal{N}(\alpha) = 0$	123
3.3	Execution times of the elementary functions used in the evaluation of the integral I_1 . $\mathbf{x} = 1.23456\mathbf{f}$, $\mathbf{y} = 0.32993874\mathbf{f}$	159
3.4	Execution time of the function used to evaluate the integral I_1 by Algorithm 3. Integration limits: $\mathbf{a} = -1.5$, $\mathbf{b} = 1.5$	160
3.5	Execution time of the function $\mathcal{N}(\alpha)$, defined by Algorithm 5, for $\mathbf{alpha} = 1.2217$	161
3.6	Values of the gains in the guidance law (3.82) used in the numerical simulations.	171
3.7	Control effort (3.81) for different cruise altitudes, $\Delta x_{\text{start}} = 2$ km.	183
3.8	Control effort (3.81) for different distances from the target at which terminal guidance begins, $h_{\text{cruise}} = 300$ m.	187
3.9	Final results for an unguided rocket with total impulse deviations.	206
3.10	Final results obtained using optimal guidance (Algorithm 2) with total impulse deviations, $\gamma_1 = -49.25^\circ$	207
3.11	Final results obtained using optimal guidance (Algorithm 2) with total impulse deviations, $\gamma_1 = -45^\circ$	209
3.12	Subset of standard HIL test cases and corresponding results.	214
B.1	Positive nodes and corresponding weights for the 10-point Gauss–Legendre quadrature rule.	245
B.2	Coefficients of the minimax polynomial approximations of the cosine function.	246

Nomenclature

Greek symbols

Symbol	Description	Units
α	Auxiliary variable / Angle of attack	[rad]
α^*	Solution to the characteristic equation $\mathcal{N}(\alpha) = 0$	[rad]
β	Auxiliary variable / Sideslip angle	[rad]
$\Delta\gamma_1$	Final flight-path angle error	[rad]
$\dot{\gamma}_c$	Commanded flight-path angle rate	[rad/s]
ϵ_α	Error of the approximation $\hat{\alpha}$	[rad]
ϵ_u	Control variable error	[m/s ²]
$\epsilon_{\mathcal{I}_1}, \epsilon_{I_1}$	Integration error bounds	[-]
η	Velocity lead angle / Pitch control surface deflection	[rad]
η_1, η_2	Lagrange multipliers	[m/s ³]
η_3	Lagrange multiplier	[m ² /s ³]
γ	Flight-path angle	[rad]
γ_{co}	Flight-path angle on the collision triangle	[rad]
γ_e	Heading error	[rad]
$\hat{\alpha}$	Approximation of the optimal solution α^*	[rad]
$\hat{\gamma}$	Estimated flight-path angle	[rad]

$\hat{\theta}$	Estimated pitch angle	[rad]
λ	Line-of-sight angle	[rad]
λ_γ	System costate variable	[m ² /s ³]
λ_x, λ_y	System costate variables	[m/s ³]
λ_z, λ_0	System costate variables	[-]
$\mathcal{C}$	Cost function for multi-parameter optimization	[-]
$\mathcal{D}$	Admissible set of mission flight parameters	[-]
$\mathcal{E}$	Error function	[-]
$\mathcal{F}$	Alternative characteristic root-finding function	[-]
$\mathcal{G}$	Gauss-Legendre quadrature operator	[-]
$\mathcal{I}_1$	First generalized composite integral component	[-]
$\mathcal{I}_2$	Second generalized composite integral component	[-]
$\mathcal{N}$	Primary characteristic function evaluated for root-finding	[m]
σ	Impact angle	[rad]
τ	Autopilot lag time constant	[s]
θ	Pitch angle	[rad]
ξ	Auxiliary variable / Roll control surface deflection	[rad]
ζ	Yaw control surface deflection	[rad]

Latin symbols

Symbol	Description	Units
$\mathbf{r}$	Position vector	[m]
$\mathbf{v}$	Velocity vector	[m/s]
Δx_1	Terminal downrange miss distance	[m]
Δx	Missile-target downrange separation distance	[m]

Δx_{start}	Downrange separation at terminal guidance initiation	[m]
Δy	Missile-target crossrange/vertical separation distance	[m]
a	Lower boundary integration limit	[-]
a^*, c^*	Switching points for fundamental integral evaluation	[-]
b	Upper boundary integration limit / bias	[-], [rad/s]
$f, \mathbf{f}$	Specific force magnitude / specific force vector	[m/s ²]
$g, \mathbf{g}$	Gravitational acceleration magnitude / gravitational vector	[m/s ²]
H	Hamiltonian	[m ² /s ⁴]
h	Altitude	[m]
h_{cruise}	Cruise altitude	[m]
I_1	First fundamental integral	[-]
I_2	Second fundamental integral	[-]
I_D	Auxiliary denominator integral	[-]
I_N	Auxiliary numerator integral	[-]
I_S	Elliptic integral evaluated for climb phase	[-]
J	Performance index / Cost function	[m ² /s ³]
k	Guidance law gain	[Varies]
l_b, l'_b	Lower mathematical bounds	[-]
N'	Effective navigation ratio for proportional navigation	[-]
R	Relative distance to target	[m]
r	Auxiliary transformation variable	[m/s ³]
t	Time	[s]
t_{go}	Time until intercept (time-to-go)	[s]
$u, \mathbf{u}$	Control variable magnitude / control vector	[m/s ²]
u_b, u'_b	Upper mathematical bounds	[-]
u_c	Commanded normal acceleration	[m/s ²]

V	Total velocity magnitude	[m/s]
V_c	Closing velocity magnitude	[m/s]
v_N, v_E, v_D	Velocity vector components resolved in NED frame	[m/s]
$x, \boldsymbol{x}$	Downrange position coordinate / State vector / Axis label	[m], [-]
y	Crossrange or height coordinate	[m]
z	Cross-axis coordinate / Linearized deviation parameter	[m]

Subscripts

Symbol Meaning

$\tilde{x}, \tilde{y}, \tilde{z}$ Projections relative to the aeroballistic reference axes

d, des Desired value

f Final boundary value condition

M Missile

T Target

x, y, z Projections relative to the body reference axes

0 Initial or entry state value

1 Final or terminal intercept value

Az Azimuth coordinate frame

cmd Commanded value

hor Horizontal trajectory component projection

nom Nominal value

ref Reference value

rhp Right-half plane

ver Vertical trajectory component projection

Abbreviations

Acronym Meaning

6DOF Six-degrees-of-freedom

ALTI Altimeter

CAN Controller area network communication protocol

GNC Guidance, navigation, and control

GNSS Global navigation satellite system

GPS Global positioning system

HIL Hardware-in-the-loop

IMU Inertial measurement unit

INS Inertial navigation system

IPP Impact point prediction

LOS Line-of-sight

NED North-East-Down local coordinate frame

OBC Onboard computer

OPT Optimal

PN Proportional navigation

RHP Right-half plane

SIL Software-in-the-loop

TPBVP Two-point boundary-value problem

TT Trajectory-tracking

UDP User Datagram Protocol communication standard

Chapter 1

Introduction

1.1 General Considerations

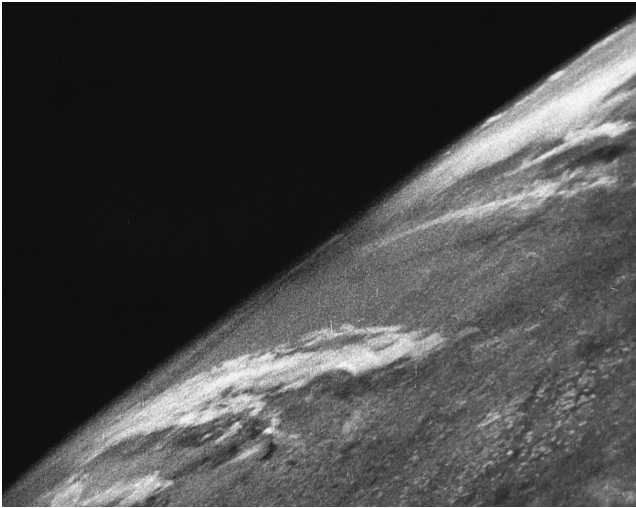
“At first we inevitably have an idea, fantasy, fairy tale, and then come scientific calculations; finally execution crowns the thought.”

— Konstantin E. Tsiolkovsky, *The Exploration of Cosmic Space by Means of Reaction Devices* (1903; expanded 1911–1912)

Although the fundamental ideas on the use of rockets for space exploration were already presented at the beginning of the 20th century in the works of K. Tsiolkovsky [1], their practical application firmly established the foundations of astronautics only several decades later. The first vehicle to exceed all previous limits and provide humanity with its first view from space (Fig. 1.1a) was also the first tactical ballistic missile deployed in operational use—the V-2 rocket (Fig. 1.1b). Immediately after the end of the war, this weapon of mass destruction was repurposed as a platform for the early development of space exploration.

In addition to the many boundaries this rocket crossed in terms of altitude, range, flight duration, and other performance metrics, it also established a fundamental constraint in system design: it became clear that a guidance system is an essential subsystem of long-range tactical missiles. The first cruise (V-1) and ballistic (V-2) missiles did not possess a guidance system in the sense in which it is understood today; instead, they generated control commands based on onboard sensor signals primarily for flight stabilization.

Apart from a small number of manually guided missiles used during World War II (simple radio-command guidance directed by an operator), the first missiles equipped with true autonomous guidance capability were early pursuit-type homing missiles developed in the late 1940s and 1950s, which implemented continuous target-based steering. These were followed, from the 1950s onward, by systems based on the now well-known proportional navigation principle.



(a) The first photograph of Earth from above the Kármán line, 1946 [2].



(b) V-2 rocket during Allied tests in Germany, 1945 [3].

Figure 1.1: The V-2 rocket and its contribution to the early development of space exploration.

The second half of the twentieth century saw a rapid expansion in the application of guided rocket technology, in both military and civilian domains.

The primary task of a guidance system is to generate a command based on the information available about the guided object (projectile, rocket, unmanned aerial vehicle, etc.) and the point to which the object is guided (most commonly referred to as the target). A command is understood as a signal representing one of the kinematic quantities along the trajectory, whose subsequent tracking ensures interception of the target by the guided object.

On the other hand, the task of the control system is, based on the known dynamic characteristics of the guided object and the quantities measured during flight, to generate a control signal for its actuator that enables accurate tracking of the commanded reference.

The navigation system (or another tracking system for the missile and target) provides the information required by the guidance system, thereby forming a closed-loop system composed of these three disciplines: Guidance, Navigation, and Control (GNC). This process is schematically illustrated in Fig. 1.2 using a missile application as an example.

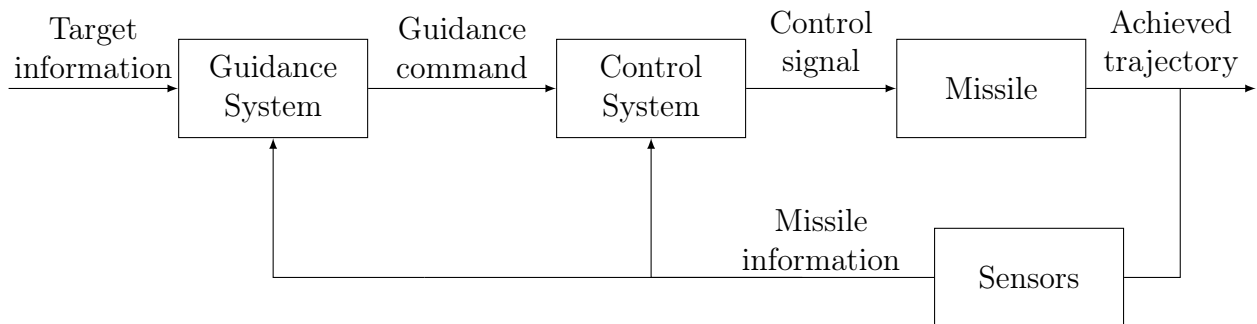


Figure 1.2: Schematic representation of a closed-loop Guidance–Navigation–Control system.

For successful mission completion, merely achieving an encounter between the flight vehicle and the target is often insufficient. Additional guidance requirements depend on the type of vehicle, the nature of the target, and the mission objectives. For example, in orbital rendezvous missions (Fig. 1.3a), it is not enough for the vehicles to reach the same position in space; their relative velocity must also be reduced to zero. Similar requirements arise in planetary and Earth landing missions, where the vehicle must additionally enter the atmosphere at a prescribed flight-path angle (Fig. 1.3b).

The effectiveness of a missile warhead may also depend strongly on the vehicle's terminal orientation relative to the target. This is particularly evident for directional warheads, whose effectiveness is highly sensitive to the direction of approach. Since armored targets are generally most vulnerable in their upper hemisphere, top-attack trajectories are often preferred, as illustrated in Fig. 1.3c.

Guidance of cruise missiles along predefined paths requires passage through waypoints with appropriately prescribed velocity vector orientations, such that the resulting trajectory smoothly connects consecutive path segments while minimizing unnecessary maneuvering. A similar requirement arises during the launch phase of cruise missiles, as illustrated in Fig. 1.3d, where the vehicle must transition from its initial ascent to a nearly horizontal flight condition before entering the cruise phase.

Although the specific requirements vary among applications, all of the previously mentioned examples share two common objectives.

1. Achieving an encounter between the guided object and the target;
2. Achieving the desired orientation at the target point.

The required orientation at the terminal point may refer either to the vehicle attitude, typically specified through pitch and/or yaw angles, or to the orientation of the velocity vector, described by two angles in mutually perpendicular planes (typically the horizontal and vertical planes). The angle between the velocity vector and the local horizontal plane is commonly referred to as the flight-path angle. While certain applications require a specific vehicle attitude to accomplish their mission objectives, almost all scenarios require appropriate trajectory shaping through control of the velocity vector orientation.

Moreover, suitable guidance can often establish the desired direction of the velocity vector well before reaching the target, allowing the vehicle to approach from the required direction. In such cases, the difference between the vehicle attitude and the velocity vector orientation remains sufficiently small. For this reason, the second requirement is generally formulated in terms of the desired terminal velocity vector orientation.

This category also includes missions requiring control of the relative velocity at encounter, such as orbital rendezvous and landing operations, since prescribing the velocity components at the terminal point is closely related to prescribing the velocity vector orientation.

Although the term *target* may refer to an intermediate waypoint, as in path-following guidance, the terminology commonly used in missile guidance is generally based on the final point of the

trajectory. Consequently, the desired terminal velocity angle in the relevant plane is often referred to as the *impact angle*.

This terminology does not imply that the resulting guidance methods are restricted to the terminal phase of flight. Rather, any waypoint can be treated as a target point, allowing the same methods to be applied sequentially along a trajectory. In this way, the developed guidance laws are applicable not only to missiles but also to a wide range of flight vehicles and guidance scenarios.

Besides the requirements imposed on the terminal state of the vehicle (terminal position and velocity vector orientation), additional constraints may be placed on the trajectory itself. Infinitely many trajectories satisfy the two basic requirements, since countless curves can connect two points while preserving a prescribed terminal direction. From a practical perspective, however, not all such trajectories are feasible, and from a mission perspective, not all are acceptable.

Additional requirements may concern the trajectory length, flight time, or bounds on kinematic quantities such as velocity, acceleration, flight-path angle, or deviation from a nominal path. Because real flight vehicles possess limited maneuvering capability, the most common constraint is a bound on the maximum available normal acceleration. Such constraints typically appear as inequalities in the guidance formulation, still leaving infinitely many admissible trajectories.

An alternative to imposing explicit bounds on normal acceleration is to consider the total control effort, commonly represented by an integral measure of the commanded normal acceleration over the flight duration. Minimization of this quantity leads to minimum-energy trajectories, i.e., trajectories that are optimal with respect to control effort. The guidance problem can therefore be augmented with a third requirement:

3. Minimization of control effort.

Guidance methods designed to satisfy these requirements belong to the field of optimal guidance, where optimality is defined with respect to a prescribed performance index. The corresponding guidance laws are generally derived using optimal control theory. Since optimal impact-angle guidance constitutes the central topic of this dissertation, the following sections review the relevant literature and formulate the research problem in the context of the current state of the art.



(a) Orbital rendezvous and docking [4].



(b) Atmospheric entry and landing [5].



(c) Explosively formed projectile top attack [6].



(d) Cruise missile transition to horizontal flight [7].

Figure 1.3: Representative applications of guidance with orientation constraint.

1.2 Literature Review and Historical Background

Due to its broad range of applications, the problem of achieving a desired flight-path angle has attracted considerable attention over the past five decades. Consequently, a large number of guidance methods have been developed to address this problem. Classifying these methods according to a single criterion is therefore difficult. Nevertheless, two dominant classifications can be identified:

1. According to the methodology used in the derivation of the guidance law:
 - methods based on optimal control theory;
 - methods based on modifications of proportional navigation.
2. According to the kinematic model of the missile–target engagement:
 - methods based on linearized engagement models, commonly referred to as linear guidance laws;
 - nonlinear guidance laws.

The following sections present an attempt to systematically classify these methods, as well as those that cannot be readily assigned to either of the above categories, while preserving their chronological development. Before reviewing the literature on guidance laws designed to achieve prescribed flight-path angles, a brief overview of several historically significant works related to guidance problems is provided.

First, the Markov–Dubins model is introduced, as it may be regarded as a fundamental model for determining the shortest trajectory between two points with prescribed velocity-vector directions. Next, Zermelo’s navigation problem is presented as the problem of minimizing travel time in the presence of known disturbances. In addition to these problems, which focus primarily on the determination of time-optimal trajectories, proportional navigation is reviewed as the classical optimal missile-guidance problem in which the control effort is minimized.

1.2.1 Markov–Dubins Model

Due to the simple geometric procedure used to determine its optimal solution, the Markov–Dubins model is one of the most commonly employed kinematic models of planar constant-speed motion in guidance problems.

The original form of the problem was introduced by *A. A. Markov* in 1889 in his work [8]. The objective is to determine the shortest curve connecting two points, subject to a lower bound on the radius of curvature and a prescribed tangent direction at the initial point. The problem was originally formulated in the context of railway track design. Nearly seventy years later, *L. E. Dubins* demonstrated in [9] the solution of an extended problem in which the tangent direction at the terminal point is also prescribed. He showed that the optimal trajectory consists of at most three segments, each of which is either a straight line or an arc of a circle having the minimum admissible radius of curvature.

The Markov–Dubins model has frequently been studied in problems involving vehicles with limited maneuverability. One notable example is the work [10], which established the foundations of differential game theory. The problem was formulated as the motion of a constant-speed vehicle attempting to minimize the time required to approach a pedestrian who actively seeks to avoid the encounter. The initial direction of the vehicle velocity vector is prescribed. In the special case of a stationary pedestrian, it is evident that the vehicle seeks a trajectory of minimum length, which corresponds to the original Markov problem.

The formulation of the Markov–Dubins model as an optimal control problem has been the subject of research over the past decades using Pontryagin’s minimum principle (see Appendix A). One formulation of such a problem is given as follows:

$$\begin{aligned}
 &\text{Minimize} && J = \int_{t_0}^{t_1} dt \\
 &\text{subject to} && \dot{x} = V \cos \gamma \\
 &&& \dot{y} = V \sin \gamma \\
 &&& \dot{\gamma} = \frac{u}{V} \\
 &\text{and constraint} && |u| \leq u_{\text{MAX}} \\
 &\text{and boundary conditions} && x(t_0) = x_0 \\
 &&& y(t_0) = y_0 \\
 &&& \gamma(t_0) = \gamma_0 \\
 &&& x(t_1) = x_1 \\
 &&& y(t_1) = y_1 \\
 &&& \gamma(t_1) = \gamma_1
 \end{aligned} \tag{1.1}$$

In these equations, the position of the material point M is defined by the planar Cartesian coordinates x and y , while γ denotes the flight-path angle. The boundary conditions specify the initial state at time t_0 and the terminal state at time t_1 .

The parameter V represents the speed of the material point M , which, in the general case, need not be constant. However, the kinematic model most commonly employed in guidance problems neglects variations in speed magnitude, since accounting for such variations would require consideration of the vehicle dynamics. This simplification is justified in most guidance applications of practical interest [11].

For a clearer physical interpretation, the control variable u is chosen as the normal acceleration. The sign convention for u in this dissertation is chosen to be consistent with the definition of the flight-path angle γ , for which a positive change requires a positive normal acceleration u as depicted in Fig. 1.4. Imposing a bound on the normal acceleration through the maximum allowable value u_{MAX} is equivalent, for a given speed V , to imposing a lower bound on the trajectory radius of curvature.

On the other hand, minimizing the performance index J yields the shortest-time transfer to the terminal point M_1 . Since the speed is constant, this is equivalent to minimizing the trajectory length. Consequently, the planar constant-speed motion problem defined by (1.1) is equivalent to the Markov–Dubins model.

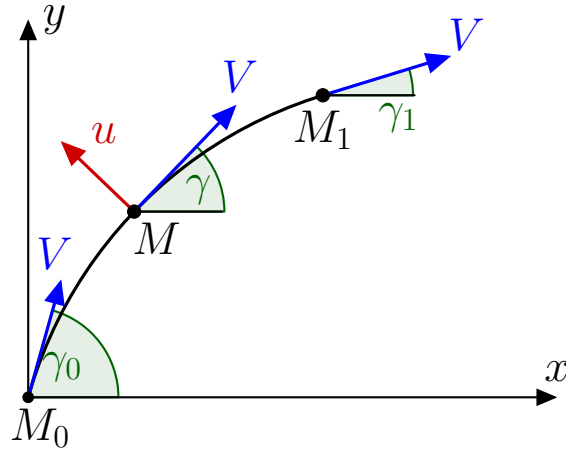


Figure 1.4: Planar kinematic model of the Markov–Dubins problem.

1.2.2 Zermelo Navigation Problem

The Zermelo navigation problem, named after the German mathematician *Ernst Zermelo*, who first formulated it in 1931 [12], is a classical problem in optimal control theory.

It considers the motion of a vessel (modeled as a material point M in Fig. 1.5) moving with constant speed V relative to a flowing medium. During its motion, the vessel is subject to a disturbance in the form of a current, described by a position-dependent vector field $w_x(x, y)$, $w_y(x, y)$. The vessel dynamics are therefore governed by the following equations:

$$\begin{aligned}\dot{x} &= V \cos \gamma + w_x(x, y) \\ \dot{y} &= V \sin \gamma + w_y(x, y)\end{aligned}\tag{1.2}$$

The objective is to reach a desired point M_1 , or an island whose boundary is described by the function $G(x, y)$, in minimum time, by controlling the velocity-vector angle γ .

Although originally formulated in a nautical context, the Zermelo navigation problem also describes a wide range of kinematic control problems, including aircraft motion in the presence of wind disturbances, as in [13]. Due to its generality, it is frequently used in aerospace engineering as a basis for deriving more complex motion models. Examples related to vehicle guidance can be found in [14, 15], where the concept of the so-called simple boat guidance problem (SBGP) was first introduced in the context of differential games [16] and target interception problems. More complex orbital mechanics problems can also be reduced to the Zermelo problem through an appropriate change of coordinates, as shown in [17].

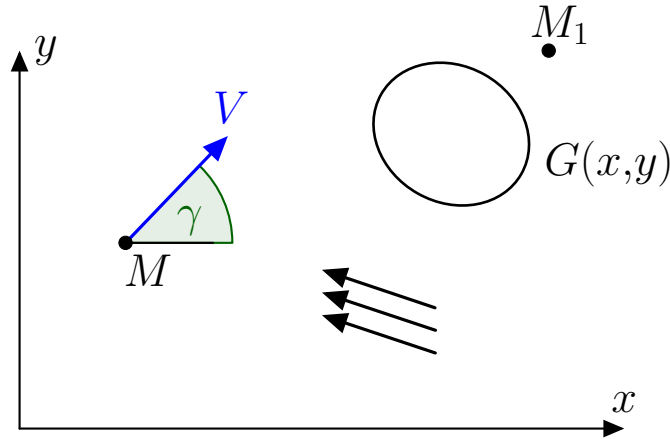


Figure 1.5: Zermelo Navigation Problem.

Unlike the Markov–Dubins problem, this kinematic model explicitly accounts for external disturbances; however, the initial and terminal velocity-vector angles are not prescribed.

Cases involving both disturbances and prescribed terminal conditions are treated using a combination of the Zermelo and Markov–Dubins models (ZMD – Zermelo–Markov–Dubins), as presented in [18] for the case of a constant wind field. Stochastic wind variations are considered in [19] for a small unmanned aerial vehicle, but without constraints on the terminal flight-path angle.

In practice, wind disturbances are not known a priori with sufficient accuracy to enable exact optimal feedback control. One common approach is therefore to neglect disturbances during trajectory generation (i.e., when computing nominal normal-acceleration profiles) [20], and to compensate for their effect using feedback control theory, with the objective of tracking the nominal trajectory. In this way, suboptimal solutions of problem (1.1) are obtained, which are nevertheless practically implementable on real systems.

Missile–target engagement problems can also be reduced to the Zermelo problem in the case of highly maneuverable missiles, where changes in the flight-path angle can be considered instantaneous. This has been demonstrated for air-to-air missiles in [21, 22]. For highly maneuverable missiles employing a collision triangle strategy and non-maneuvering targets, the engagement kinematics can be interpreted as a Zermelo problem with constant drift, resulting in parallel navigation. Parallel navigation represents a guidance strategy in which the line-of-sight angular rate is driven to zero, and it is closely related to guidance laws based on proportional navigation.

1.2.3 Proportional Navigation

Proportional navigation is the most widely known missile guidance law, which has been in use since approximately 1950, when the *Lark* surface-to-air missile (Fig. 1.6) achieved its first successful flight tests. Since then, proportional navigation has been implemented on nearly all tactical missiles employing radar, infrared, or television-based guidance [23].

The popularity of this guidance law stems from its simplicity and effectiveness, as its implementation requires only two easily measurable quantities: the closing velocity and the angular rate of the line-of-sight (LOS) to the target¹. [22]



(a) Museum model. [24]



(b) Missile launch, circa 1949. [25]

Figure 1.6: The *Lark* surface-to-air missile was the first missile on which proportional navigation was implemented as a guidance method.

The proportional navigation law is most intuitively understood when expressed in terms of the commanded rate of change of the flight-path angle. The guidance law can then be written as:

$$\dot{\gamma}_c = k\dot{\lambda} \quad (1.3)$$

which is highly intuitive: the rate of change of the flight-path angle should be k times the rate of change of the line-of-sight angle.

Since the normal acceleration can be expressed in terms of the flight-path angle rate as $u = V\dot{\gamma}$, the commanded acceleration can be written as:

$$u_c = kV\dot{\lambda} \quad (1.4)$$

where:

- u_c — commanded normal acceleration (perpendicular to the velocity vector),
- k — proportional navigation gain,
- V — missile speed,
- $\dot{\lambda}$ — line-of-sight angular rate.

For $k = 1$, the law reduces to pure pursuit, while for $k > 1$ the missile initially generates a corrective maneuver in order to cancel future LOS rotation and establish a suitable lead angle (the angle between the missile velocity vector and the line of sight) [26].

¹The line-of-sight (LOS) is the line connecting the missile and the target.

Figure 1.7 shows the planar kinematic engagement model between a missile and a target. The initial positions are denoted by M_0 and T_0 , respectively. Both velocities are assumed constant, and for simplicity the target is assumed to be non-maneuvering. Without loss of generality, the initial LOS is aligned with the x -axis.

Assuming that the missile is launched with an initial flight-path angle γ_{co} , interception at point C is achieved if the following condition holds:

$$V \sin \gamma_{co} = V_T \sin \gamma_T \quad (1.5)$$

The angle γ_{co} represents the lead angle, and the triangle M_0T_0C defines the ideal interception geometry, i.e., the collision triangle, which is why the subscript “co” is used. The initial separation distance is R_0 , and the nominal closing velocity is given by:

$$V_{c,nom} = V \cos \gamma_{co} - V_T \cos \gamma_T \quad (1.6)$$

By setting $t_0 = 0$, the nominal flight time is:

$$t_{1,nom} = \frac{R_0}{V_{c,nom}}. \quad (1.7)$$

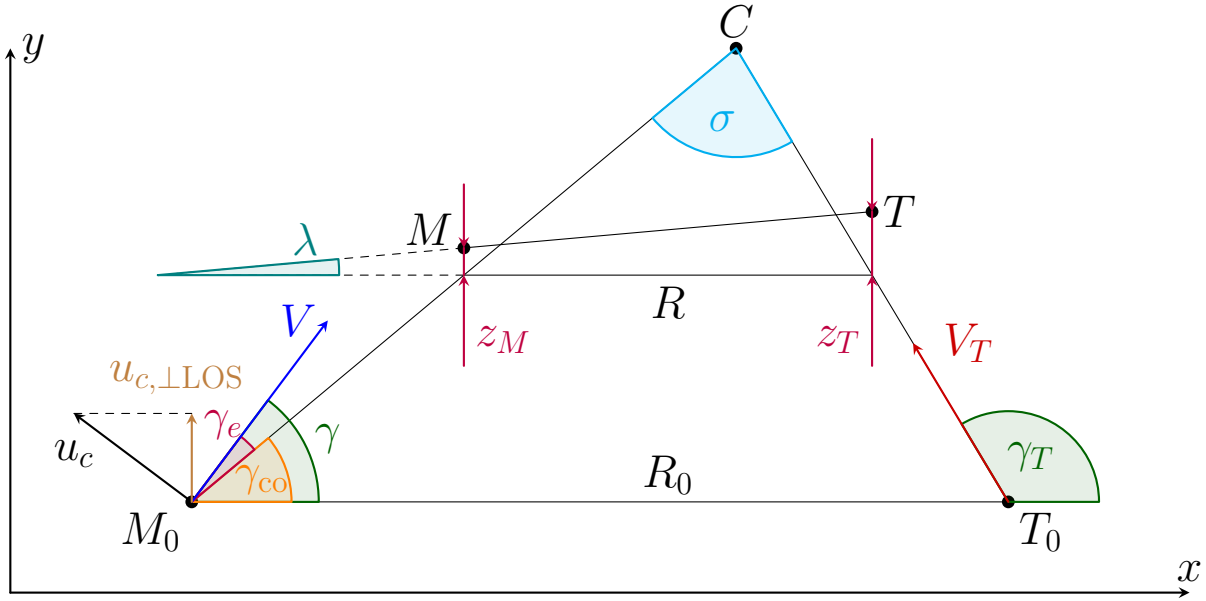


Figure 1.7: Missile–target engagement geometry.

The actual flight-path angle γ may differ from the lead angle γ_{co} due to an error in establishing the lead angle, called the heading error γ_e . Consequently, the closing velocity is given by:

$$V_c = V \cos \gamma - V_T \cos \gamma_T \quad (1.8)$$

If the heading error is small, the following approximation holds:

$$\cos \gamma = \cos(\gamma_{co} \pm \gamma_e) = \cos \gamma_{co} \cos \gamma_e \mp \sin \gamma_{co} \sin \gamma_e \approx \cos \gamma_{co} \mp \gamma_e \sin \gamma_{co} \quad (1.9)$$

Neglecting the effect of heading error on the closing velocity, it can be approximated by its nominal value:

$$V_c \triangleq V \cos \gamma_{co} - V_T \cos \gamma_T \quad (1.10)$$

Under the same assumption, the effect of the heading error on the total flight time can also be neglected, so that the nominal flight time is given by:

$$t_1 \triangleq \frac{R_0}{V_c} \quad (1.11)$$

At an arbitrary time instant t , the missile and target positions may deviate from their nominal trajectories, as shown in Fig. 1.7 by the points M and T . The nominal separation distance at that instant is:

$$R = V_c(t_1 - t) = V_c t_{go} \quad (1.12)$$

where t_{go} denotes the nominal remaining time to interception (time-to-go).

The deviations of the missile and target in the direction perpendicular to the initial line-of-sight are denoted by z_M and z_T , respectively. The instantaneous line-of-sight angle can then be expressed as:

$$\lambda = \tan^{-1} \left(\frac{z_T - z_M}{R} \right) \quad (1.13)$$

It should be noted that in Fig. 1.7 the initial line-of-sight direction is assumed to be horizontal for simplicity. As will be shown, the key kinematic quantities in proportional navigation are defined relative to the initial line-of-sight direction. Therefore, at the beginning of the engagement, it is always possible to rotate the coordinate system such that the x -axis is aligned with the initial line-of-sight direction, and the engagement geometry can be analyzed using the model in Fig. 1.7.

For this reason, small LOS angles can be assumed, which is a common situation in both head-on and tail-chase engagements [23]. Hence:

$$\lambda \approx \frac{z_T - z_M}{R} = \frac{z}{R} \quad (1.14)$$

Based on these assumptions, the linearized missile–target engagement model using proportional navigation can be represented in block-diagram form in Fig. 1.8. In this model, the seeker dynamics are neglected and approximated as an ideal differentiator, while the missile dynamics are also neglected by assuming an ideal autopilot, such that the achieved normal acceleration u is equal to the commanded acceleration u_c . Furthermore, the effect of the heading error on the projection of the normal acceleration onto the direction perpendicular to the initial line-of-sight is neglected:

$$\ddot{z}_M = u_{\perp LOS} = u_{c, \perp LOS} = u_c \cos \gamma \approx u_c \cos \gamma_{co} \quad (1.15)$$

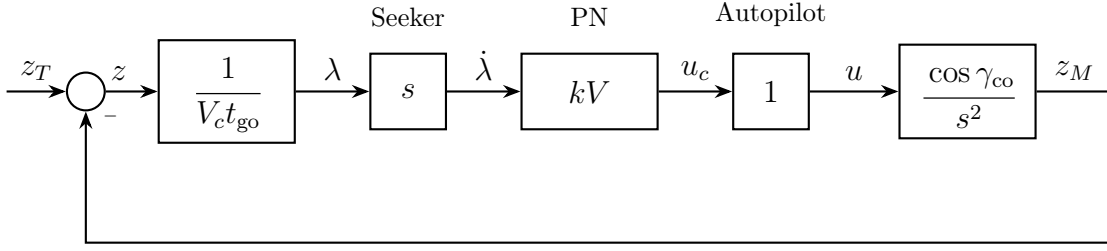


Figure 1.8: Block diagram of the linearized missile–target engagement model employing proportional navigation (PN).

The open-loop gain is equal to N'/t_{go} , where N' is the effective navigation ratio and is given by:

$$N' = \frac{kV \cos \gamma_{co}}{V_c} \quad (1.16)$$

Projecting (1.4) onto the direction perpendicular to the initial line of sight (while neglecting the effect of the heading error) yields:

$$u_{c,\perp LOS} = kV \dot{\lambda} \cos \gamma_{co} \quad (1.17)$$

Combining the previous two equations gives the proportional navigation guidance law in the following form:

$$u_{c,\perp LOS} = N' V_c \dot{\lambda} \quad (1.18)$$

Owing to its clear physical interpretation and the technology available at the time, proportional navigation was widely adopted long before its optimality was established mathematically, nearly two decades later (see Section 1.2.5). The optimality of this guidance law is reflected in the minimization of the following performance index:

$$J = \int_0^{t_1} u_{c,\perp LOS}^2 dt \quad (1.19)$$

For the linearized missile–target engagement model presented above, assuming a non-maneuvering target, it has been shown that proportional navigation in the form (1.18) is optimal when the effective navigation ratio is chosen as $N' = 3$ [27].

If the target performs a maneuver such that the projection of its normal acceleration onto the direction perpendicular to the initial line of sight is $u_{T,\perp LOS}$, the resulting optimal guidance law becomes the augmented proportional navigation (APN) law:

$$u_{c,\perp LOS} = N' V_c \dot{\lambda} + \frac{1}{2} N' u_{T,\perp LOS} \quad (1.20)$$

again with the effective navigation ratio $N' = 3$ [23].

Different formulations of the optimal guidance problem, particularly through alternative choices of the performance index, lead to different optimal values of the effective navigation ratio [28].

1.2.4 Explicit Guidance

The first guidance laws that, in addition to ensuring interception of the target, also provided a desired flight-path angle at the instant of interception emerged from research on spacecraft guidance algorithms. The guidance problems most commonly encountered in space missions, besides orbital insertion, are rendezvous with another spacecraft and landing on a celestial body. In both cases², the corresponding two-point boundary value problem (TPBVP) can be stated as follows. Given the spacecraft position and velocity at the initial time t_0 , expressed in a geocentric inertial Cartesian coordinate system:

$$\mathbf{r}(t_0) = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}, \quad \mathbf{v}(t_0) = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} \quad (1.21)$$

the objective is to determine the time history of the control vector:

$$\mathbf{u}(t) = \begin{bmatrix} u_x(t) \\ u_y(t) \\ u_z(t) \end{bmatrix} \quad (1.22)$$

such that, at the prescribed terminal time t_1 , the desired position and velocity are reached:

$$\mathbf{r}(t_1) = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}, \quad \mathbf{v}(t_1) = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{z}_1 \end{bmatrix} \quad (1.23)$$

Although the flight-path angle does not appear explicitly in this formulation, its value at both the initial and terminal points is fully determined by the components of the velocity vector, as illustrated in Fig. 1.9.

The spacecraft motion is described by the vector differential equation:

$$\frac{d^2 \mathbf{r}}{dt^2} = \mathbf{g} + \mathbf{u} \quad (1.24)$$

where $\mathbf{g}$ is the gravitational-acceleration vector, a nonlinear function of the position vector.

²In rendezvous problems, however, the formulation is most often expressed in relative rather than absolute coordinates.

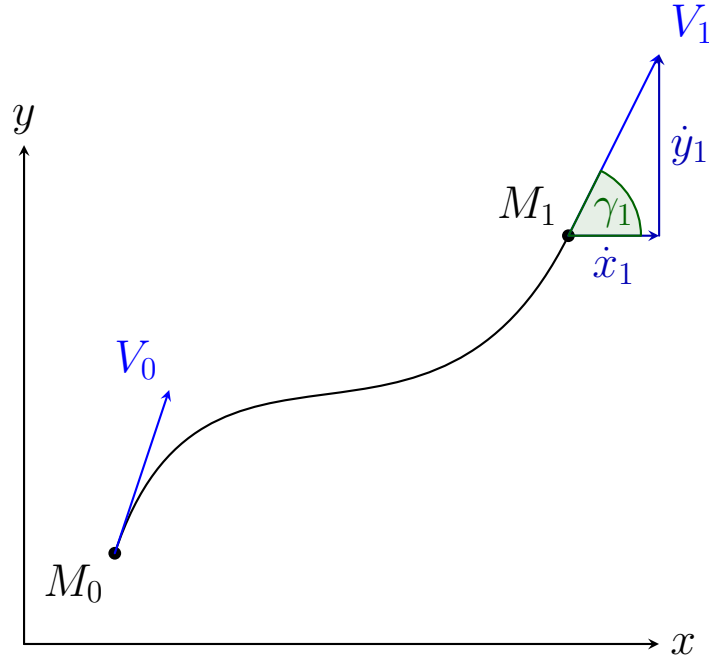


Figure 1.9: Specifying the velocity-vector components at the interception point is equivalent to specifying the impact angle.

In the general case, the vehicle is subjected not only to gravitational forces but also to rocket thrust and, during atmospheric flight, aerodynamic forces. Depending on the application, the aerodynamic force may contribute to the control force together with thrust or may even constitute the sole control force. Furthermore, the terminal time t_1 may be free rather than prescribed.

Since problems of this type were originally studied in the context of space missions, where aerodynamic forces are absent, the components of the control vector $\mathbf{u}(t)$ represented accelerations generated by the propulsion system along the axes of the geocentric inertial frame. The terminal time t_1 was then determined by the total engine burn time.

A solution to this problem, developed for lunar landing applications, was proposed by G. Cherry in 1964 in the form of explicit guidance (*E-guidance*) [29]. Explicit guidance laws are characterized by command expressions written directly as functions of the current and desired position and velocity components and are often obtained as direct solutions of the equations of motion.

By decomposing the vector problem into three scalar problems, Cherry proposed the component of the total acceleration along the y -axis (the same form applies to the remaining two axes) as:

$$\ddot{y}(t) = c_1 p_1(t) + c_2 p_2(t) \quad (1.25)$$

where $p_1(t)$ and $p_2(t)$ are linearly independent predefined functions of time, while the coefficients c_1 and c_2 are computed during flight from the current deviations from the desired terminal conditions:

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} \dot{y}_1 - \dot{y}_0 \\ y_1 - y_0 - \dot{y}_0 t_{\text{go}} \end{bmatrix} \quad (1.26)$$

where t_{go} (*time-to-go*) denotes the remaining time until arrival at the target, i.e., $t_{\text{go}} = t_1 - t$. The elements of the matrix $E[e_{ij}]$ depend on the selected forms of the functions $p_1(t)$ and $p_2(t)$ and are therefore known a priori.

Once the required total-acceleration component is obtained from (1.25), the corresponding control input is computed as

$$u_y(t) = \ddot{y}(t) - g_y(t) = c_1 p_1(t) + c_2 p_2(t) - g_y(t) \quad (1.27)$$

For practical implementation, and owing to the fact that any time function can be approximated by a Taylor-series expansion in the vicinity of the terminal time t_1 , the functions $p_1(t)$ and $p_2(t)$ were chosen as polynomials in $(t_1 - t)$. The following form was therefore proposed:

$$\begin{aligned} p_1(t) &= a_0 + a_1 t_{\text{go}} + a_2 t_{\text{go}}^2 + \dots + a_n t_{\text{go}}^n \\ p_2(t) &= p_1(t) t_{\text{go}} \end{aligned} \quad (1.28)$$

under the condition that at least one coefficient a_i , $i = 0, \dots, n$, is nonzero.

In [29], the optimal selection of the coefficients a_i was not investigated with respect to any performance criterion defined through a cost function, as is customary in optimal control theory. Nevertheless, it was pointed out that the “optimal” values of these coefficients could be determined offline, regardless of the computational effort required, and subsequently stored in the onboard computer for real-time implementation of (1.26).

As the simplest example, the functions $p_1(t)$ and $p_2(t)$ were chosen as follows:

$$\begin{aligned} p_1(t) &= 1 \\ p_2(t) &= t_{\text{go}} \end{aligned} \quad (1.29)$$

for which (1.26) takes the form:

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \frac{4}{t_{\text{go}}} & -\frac{6}{t_{\text{go}}^2} \\ -\frac{6}{t_{\text{go}}^2} & \frac{12}{t_{\text{go}}^3} \end{bmatrix} \begin{bmatrix} \dot{y}_1 - \dot{y}_0 \\ y_1 - y_0 - \dot{y}_0 t_{\text{go}} \end{bmatrix} \quad (1.30)$$

Substituting (1.29) and (1.30) into (1.25) yields the final expression for the total acceleration component:

$$\ddot{y}(t) = \frac{6}{t_{\text{go}}^2} (y_1 - y_0 - \dot{y}_0 t_{\text{go}}) - \frac{2}{t_{\text{go}}} (\dot{y}_1 - \dot{y}_0) \quad (1.31)$$

In practice, however, the actual trajectory differs from the nominal one because of imperfections in the flight-control system. Consequently, the coefficients c_1 and c_2 must be updated whenever new estimates of the position and velocity, y_0 and $\dot{y}_0$, become available. For this reason, the acceleration command given by (1.31) is computed at the control-system update rate, whereas the quantities

y_0 and $\dot{y}_0$ are updated according to the navigation-system update rate, which is generally lower than that of the control system.

Although the problem was not originally formulated within the framework of optimal control theory, the proposed procedure yields, even with the simplest choice of the functions $p_1(t)$ and $p_2(t)$, an optimal solution with respect to the control effort cost function:

$$\int_{t_0}^{t_1} u^2(t) dt \quad (1.32)$$

Even in this intuitive form of explicit guidance, as proposed by G. Cherry [29], the final expressions contain the time-to-go, arguably the most problematic yet indispensable component of virtually all optimal guidance laws.

A generalization of explicit guidance was presented by Ohlmeyer and Phillips in 2006 [30]. They considered minimization of the cost function:

$$\int_0^{t_1} \frac{u^2}{2t_{\text{go}}^n} dt \quad (1.33)$$

where $n \geq 0$ is a design parameter whose selection reduces control effort during the final stage of flight. Larger values of n result in a faster decay of the control command toward zero as the terminal point is approached.

The equations of motion were represented in state-space form:

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}u \quad (1.34)$$

subject to the boundary conditions:

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_1) = \mathbf{0} \quad (1.35)$$

The optimal control law is given by:

$$\mathbf{u} = -(M\mathbf{b})^T Q^{-1} M\mathbf{x} t_{\text{go}}^n \quad (1.36)$$

where

$$Q(t_{\text{go}}) = \int_0^{t_{\text{go}}} (M\mathbf{b})(M\mathbf{b})^T t_{\text{go}}^n dt_{\text{go}} \quad (1.37)$$

and M is the fundamental matrix satisfying:

$$\frac{dM}{dt_{\text{go}}} = MA, \quad M(t_{\text{go}} = 0) = I \quad (1.38)$$

The missile and target position vectors are denoted by:

$$\mathbf{r}_M = \begin{bmatrix} x_M \\ y_M \\ z_M \end{bmatrix}, \quad \mathbf{r}_T = \begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix} \quad (1.39)$$

For the stationary-target engagement problem, the following state variables were selected (considering only the y -axis; an identical formulation applies to the remaining two axes):

$$\mathbf{x} = \begin{bmatrix} y_T - y_M - \dot{y}_M t_{\text{go}} \\ \dot{y}_M(t_1) - \dot{y}_M \end{bmatrix} \quad (1.40)$$

where the first state variable represents the Zero Effort Miss (ZEM), while the second represents the difference between the current and desired velocity component at the target point. The control input is chosen as the missile acceleration component along the considered axis, i.e., $u = \ddot{y}_M$.

It was shown that the optimal control law³ takes the following form:

$$u = \frac{K_1}{t_{\text{go}}^2}(y_T - y_M - \dot{y}_M t_{\text{go}}) + \frac{K_2}{t_{\text{go}}}(\dot{y}_M(t_1) - \dot{y}_M) \quad (1.41)$$

where the coefficients K_1 and K_2 are given by:

$$K_1 = (n + 2)(n + 3) \quad (1.42a)$$

$$K_2 = -(n + 1)(n + 2) \quad (1.42b)$$

From (1.41) it can be seen that the control law consists of two components. The first, similarly to proportional navigation (see Section 1.2.3), attempts to eliminate the lead-angle error, whereas the second is proportional to the difference between the current and desired missile velocity at the target point.

For the special case $n = 0$, the coefficients become $K_1 = 6$ and $K_2 = -2$, which are identical to those obtained in (1.31).

It should be noted that both [29] and [30] also considered the vector form of explicit guidance:

$$\mathbf{u} = \frac{K_1}{t_{\text{go}}^2}[\mathbf{r}_T - \mathbf{r}_M - \mathbf{v}_M t_{\text{go}}] + \frac{K_2}{t_{\text{go}}}[\mathbf{v}_M(t_1) - \mathbf{v}_M] \quad (1.43)$$

which classifies explicit guidance as a three-dimensional guidance law.

Using the same guidance-law structure, C. F. Lin [31] determined the optimal values of the coefficients K_1 and K_2 for maximizing the terminal velocity and established the so-called κ -guidance law [32]. Since the notion of optimality in this case is not associated with minimum control effort but rather with achieving maximum velocity prior to the terminal phase of guidance, this approach is not considered further in the present work.

³Excluding gravitational acceleration, which is the most common assumption in guidance-law derivation.

The principal advantage of explicit guidance is that its derivation does not rely on a linearized engagement model and that the guidance law is formulated directly in vector form. Although a three-dimensional formulation is essential for most spaceflight applications, aerodynamic missiles are generally unable to realize arbitrary acceleration commands independently along all three axes.

Furthermore, a major drawback of explicit guidance is that its implementation requires knowledge of the remaining time before reaching the target point.

1.2.5 Linear Optimal Guidance Laws with Impact Angle Constraint

This section reviews guidance laws derived using a linearized kinematic model of the missile–target engagement. When included in the formulation, the dynamics of the control system are also assumed to be representable as a linear state-space system.

Trajectory Shaping Guidance

The optimality of proportional navigation for the linearized missile–target engagement model was first demonstrated by A. E. Bryson in 1965 [33]. This work established the foundations for the application of linear–quadratic regulator theory to guidance problems, which were later presented in greater detail in his book [27].

Using the terminology introduced in Section 1.2.3, the model describing deviations from the nominal collision trajectory can be represented by the block diagram shown in Fig. 1.10, where $u_{T,\perp\text{LOS}}$ denotes the target normal acceleration projected onto the direction perpendicular to the initial line of sight.

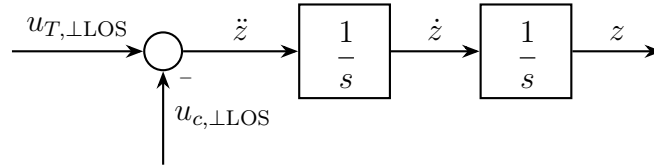


Figure 1.10: Kinematic model for determining relative deviation for the formulation of the optimal guidance problem. [23]

The presented model can be represented in state-space form⁴:

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{u}_{T,\perp\text{LOS}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ u_{T,\perp\text{LOS}} \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} u_{c,\perp\text{LOS}} \quad (1.44)$$

⁴Model (1.44) assumes a constant target maneuver, such that $\dot{u}_{T,\perp\text{LOS}} = 0$.

Assuming, for the sake of simplicity, a non-maneuvering target, Bryson showed in [33] that the control minimizing the performance index:

$$J = \frac{1}{2}[c_z z^2(t_1) + c_{\dot{z}} \dot{z}^2(t_1)] + \frac{1}{2} \int_{t_0}^{t_1} u_{c,\perp\text{LOS}}^2 dt \quad (1.45)$$

is given by:

$$u_{c,\perp\text{LOS}}(t) = \lambda_{\dot{z}0} - \lambda_z(t - t_0) \quad (1.46)$$

where $\lambda_{\dot{z}0}$ and λ_z are constants obtained by solving the corresponding two-point boundary-value problem. If every instant is treated as a new initial time, the optimal control law takes the form:

$$u_{c,\perp\text{LOS}}(t) = \lambda_{\dot{z}0}(z, \dot{z}, t_{\text{go}}) \quad (1.47)$$

The optimality criterion (1.45) seeks to minimize control effort while penalizing the terminal values of the state variables z and $\dot{z}$ through the weighting coefficients c_z and $c_{\dot{z}}$, respectively. Such a mixed performance index, combining the integral of the squared control input with weighted terminal-state penalties rather than imposing terminal constraints explicitly, became the standard formulation for a large class of single-objective guidance optimization problems based on linear-quadratic regulator theory (see, for example, [28]).

Two limiting cases considered in [33] correspond to extreme values of the weighting coefficients c_z and $c_{\dot{z}}$:

1. Target interception without a terminal relative-velocity constraint (*pure intercept*): $c_{\dot{z}} = 0$, $c_z \rightarrow \infty$;
2. Target interception with zero terminal relative velocity (*rendezvous*): $c_{\dot{z}} \rightarrow \infty$, $c_z \rightarrow \infty$.

For the first case, the optimal guidance law is:

$$u_{c,\perp\text{LOS}} = 3 \frac{z + \dot{z}t_{\text{go}}}{t_{\text{go}}^2} \quad (1.48)$$

Differentiating the LOS-angle expression (1.14) with respect to time and using (1.12) yields:

$$\dot{\lambda} = \frac{z + \dot{z}t_{\text{go}}}{V_c t_{\text{go}}^2} \quad (1.49)$$

Comparing (1.48) and (1.49), it is immediately evident that the resulting guidance law is proportional navigation in the form of (1.18) with an effective navigation ratio of $N' = 3$, thereby proving its optimality.

For the second case, the optimal guidance law becomes:

$$u_{c,\perp\text{LOS}} = \frac{6z + 4\dot{z}t_{\text{go}}}{t_{\text{go}}^2} \quad (1.50)$$

If the explicit-guidance expression (1.31) is specialized by requiring zero missile velocity at the target point, $\dot{y}_1 = 0$, while interpreting $y_1 - y_0$ as the relative separation z and $\dot{y}_0$ as the relative velocity $\dot{z}$, expression (1.50) is recovered.

This guidance law was employed during the Apollo program for the lunar landing mission in 1969. [23]

The optimality criterion (1.45) requires not only target interception with minimum control effort but also zero relative velocity at the terminal point. As discussed previously, specifying an impact angle is equivalent to specifying a velocity component at the terminal point. Consequently, when a nonzero terminal relative velocity is required, the performance index can be written as:

$$J = \frac{1}{2}[c_z z^2(t_1) + c_{\dot{z}}(\dot{z}(t_1) - \dot{z}_d)^2] + \frac{1}{2} \int_{t_0}^{t_1} u_{c,\perp\text{LOS}}^2 dt \quad (1.51)$$

where $\dot{z}_d$ denotes the desired terminal relative velocity.

The solution for this problem yields:

$$u_{c,\perp\text{LOS}} = \frac{6z + 4\dot{z}t_{\text{go}} + 2\dot{z}_d t_{\text{go}}}{t_{\text{go}}^2} \quad (1.52)$$

As expected, rendezvous guidance, given by (1.50), is obtained as a special case of (1.52) when the desired terminal relative velocity is zero.

The same guidance law can also be derived using the explicit-guidance framework⁵, Eqs. (1.34)–(1.38), for the parameter value $n = 0$, by selecting the following state variables (shown only for the y -axis, with the same form applying to the remaining axes):

$$\mathbf{x} = \begin{bmatrix} y_T - y_M + (\dot{y}_T - \dot{y}_M)t_{\text{go}} \\ [\dot{y}_T(t_1) - \dot{y}_M(t_1)]_d - [\dot{y}_T(t_1) - \dot{y}_M(t_1)] \end{bmatrix} = \begin{bmatrix} z + \dot{z}t_{\text{go}} \\ \dot{z}_d - \dot{z}(t_1) \end{bmatrix} \quad (1.53)$$

Since explicit guidance is inherently a vector guidance law and, as already noted, aerodynamic missiles cannot generally realize arbitrary acceleration commands in all directions, explicit guidance will not be considered separately from the guidance formulation presented in this section.

Equation (1.52) can also be written in the following form [23]:

$$u_{c,\perp\text{LOS}} = \frac{4(z + \dot{z}t_{\text{go}})}{t_{\text{go}}^2} + \frac{2(z + \dot{z}_d t_{\text{go}})}{t_{\text{go}}^2} = 4V_c \dot{\lambda} + \frac{2(\lambda V_c + \dot{z}_d)}{t_{\text{go}}} \quad (1.54)$$

If the relative velocity is expressed from (1.49) as:

$$\dot{z} = \frac{\dot{\lambda} V_c t_{\text{go}}^2 - z}{t_{\text{go}}} = \frac{\dot{\lambda} V_c t_{\text{go}}^2 - \lambda V_c t_{\text{go}}}{t_{\text{go}}} = \dot{\lambda} V_c t_{\text{go}} - \lambda V_c \quad (1.55)$$

⁵In this case, a moving target is considered, unlike the explicit-guidance formulation presented in [30].

and noting that, in the ideal terminal condition when $t_{\text{go}} = 0$,

$$\dot{z}(t_1) = \dot{z}_d \implies -\lambda(t_1)V_c = -\lambda_d V_c \quad (1.56)$$

the guidance law (1.52) can be rewritten in the simplified form:

$$u_{c,\perp\text{LOS}} = 4V_c\dot{\lambda} + \frac{2V_c(\lambda - \lambda_d)}{t_{\text{go}}} \quad (1.57)$$

where λ_d denotes the desired terminal LOS angle.

Both problems discussed above (target interception with and without a terminal relative-velocity constraint) were also treated in [23] using Schwarz's inequality. The same optimal guidance laws were obtained, together with an additional term accounting for target maneuvers, resulting in the general form of Trajectory Shaping Guidance (TSG):

$$u_{c,\perp\text{LOS}} = 4V_c\dot{\lambda} + \frac{2V_c(\lambda - \lambda_d)}{t_{\text{go}}} + u_{T,\perp\text{LOS}} \quad (1.58)$$

Like most advanced optimal guidance laws, implementation of (1.58) requires knowledge of the time-to-go t_{go} .

In practical implementations of closed-loop optimal guidance laws, the nominal total flight time defined by (1.11) cannot be used directly. Instead, the time-to-go t_{go} must be estimated online. The reason is that the nominal kinematic quantities are introduced only during the derivation of the guidance law, whereas during actual flight each instant is effectively treated as a new initial condition. Consequently, the predicted total flight time, and therefore t_{go} , continuously changes throughout the engagement.

Errors in time-to-go estimation may significantly degrade the performance of optimal guidance laws [34]. Consequently, accurate determination of t_{go} has been extensively studied in the literature, either as an integral component of newly proposed guidance laws (as will be discussed later in this dissertation) or as an independent research topic [35,36]. Ben-Asher and Yaesh proposed an optimal guidance law with reduced sensitivity to time-to-go estimation errors in [37], considering a first-order control system. Their results were compared with proportional navigation, representing a guidance law inherently insensitive to t_{go} estimation errors, and with the Minimum Effort Guidance Law (MEL) for a first-order control system [34, 38]. Determining accurate time-to-go estimates for different guidance laws remains an open research problem, and one potential application of the method proposed in this dissertation is precisely the estimation of the remaining flight time for optimal trajectories.

Engagements against stationary targets constitute a special case for all of the aforementioned optimal guidance laws, since:

$$V_T = 0 \quad (1.59a)$$

$$u_T = 0 \quad (1.59b)$$

Therefore, based on the interception condition (1.5), the lead angle is $\gamma_{co} = 0$, as illustrated in Fig. 1.11. In this case, the actual closing velocity in the presence of a heading error is:

$$V_c = V \cos \gamma_e \quad (1.60)$$

while the projection of the normal acceleration onto the direction perpendicular to the line of sight is:

$$u_{c,\perp\text{LOS}} = u_c \cos \gamma_e \quad (1.61)$$

If the proportional navigation guidance law is expressed in the form of (1.18), the commanded normal acceleration becomes:

$$u_c = \frac{u_{c,\perp\text{LOS}}}{\cos \gamma_e} = \frac{N' V_c \dot{\lambda}}{\cos \gamma_e} \quad (1.62)$$

and, taking into account the expression for the closing velocity (1.60), the final expression for the commanded normal acceleration in the case of stationary targets is:

$$u_c = N' V \dot{\lambda} \quad (1.63)$$

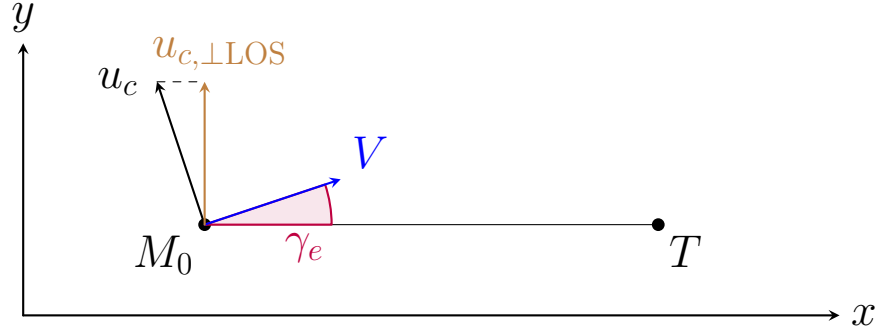


Figure 1.11: Missile–stationary–target engagement geometry.

A similar analysis can be carried out for the trajectory shaping guidance law, given by (1.58), which, in the case of engagements against stationary targets, takes the following form [23]:

$$u_c = 4V \dot{\lambda} + \frac{2V(\lambda - \gamma_1)}{t_{go}} \quad (1.64)$$

where γ_1 is the desired impact angle.

The guidance law (1.64) will be used as one of the representatives of traditional impact-angle-constrained guidance laws for comparison with the optimal guidance algorithm proposed in this dissertation.

Pioneering works

The work most commonly regarded as pioneering in the field of impact-angle-constrained guidance is the 1973 paper [39], in which Kim and Grider considered a vertical impact on a stationary or slowly moving target. The guidance law is given as a linear combination of state variables: relative separation, relative velocity in the horizontal direction, and flight-path angle, where the coefficients depend on the remaining time until interception (t_{go}). Two versions of the control system were considered: a first-order autopilot (*single lag autopilot*) and an ideal autopilot (*lag free autopilot*). For the case with control constraints, feasible regions of initial conditions were defined, i.e., sets of initial states for which the prescribed miss distance and terminal flight-path angle error can be achieved under the given control constraints. The main drawbacks of the proposed approach are high sensitivity to the accuracy of time-to-go estimation, an unjustified approximation used during linearization assuming $\cos \gamma$ to be constant, as well as an unclear procedure for selecting the weighting coefficients of the cost function to be minimized.

This work was directly extended by York and Pastrick in 1977 [40], who investigated the influence of control-system time delay on the performance of the guidance laws proposed in [39]. They also proposed a methodology for formulating the optimal guidance problem when the missile angle of attack is taken into account, although without indicating that a closed-form solution could be obtained in that case. Another extension relative to [39] is the demonstration that, with a slight modification of the state variables, the proposed algorithm can also be used to achieve arbitrary impact angles, not only vertical impact as originally considered. Due to these limitations, the methods presented in [39] and [40] will not be further analyzed in this dissertation.

Optimal guidance for the case of a maneuvering target and variable missile speed was studied by Song et al. in the 1999 paper [41]. The coefficients appearing in the guidance law are expressed as functions of the time-to-go and a constant describing the rate of change of missile speed. The state variables are defined in a coordinate frame rotating with the target velocity vector; therefore, the proposed guidance law is not directly applicable to stationary targets, which constitute the primary benchmark model for all guidance laws considered in this dissertation.

Guidance algorithms including autopilot dynamics

A systematic attempt to unify previously published optimal guidance methods with impact-angle constraints (*OGL/IAC*⁶) was presented by Ryoo, Cho, and Tahk in 2005 [42], based on the observation that all such methods are structurally similar to linear-quadratic guidance laws. The main contribution of their work is that the proposed framework is applicable to arbitrary missile dynamics against stationary targets, with closed-form guidance solutions provided for two representative cases: a first-order autopilot (*OGL/IAC-1*) and an ideal autopilot (*OGL/IAC-0*), for which the guidance law is given in the following form:

$$u = -\frac{V}{t_{go}}(-6\lambda + 4\gamma + 2\gamma_1) \quad (1.65)$$

⁶*Optimal Guidance Law/Impact Angle Control*

Since, as in all linear optimal guidance laws, the time-to-go explicitly appears in the control expression, Ryoo et al. [42] also proposed two methods for estimating this quantity. The first method computes the remaining time-to-go as the ratio between the remaining trajectory length and the missile speed, assumed to be constant, while the second method estimates it as the ratio between the missile–target distance and the projection of the relative velocity onto the line of sight. Both approaches rely on approximations; however, better results in [42] were obtained using the second method, according to which the time-to-go is computed as follows:

$$t_{\text{go}} = \frac{R}{V \left[1 - \frac{\bar{\gamma}^2 + \bar{\gamma}_1^2}{15} + \frac{\bar{\gamma}\bar{\gamma}_1}{30} + \frac{\bar{\gamma}^4 + \bar{\gamma}_1^4}{420} - \frac{\bar{\gamma}\bar{\gamma}_1(\bar{\gamma}^2 + \bar{\gamma}_1^2 - \bar{\gamma}\bar{\gamma}_1)}{840} \right]} \quad (1.66)$$

where:

$$\bar{\gamma} = \gamma + \lambda \quad (1.67a)$$

$$\bar{\gamma}_1 = \gamma_1 + \lambda \quad (1.67b)$$

The same authors, in [43] in 2006, proposed a generalization of the optimal guidance methods with impact angle constraint presented in their previous work [42], by minimizing the cost⁷:

$$\int_0^{t_1} \frac{u^2}{2t_{\text{go}}^n} dt \quad (1.68)$$

For the case of an ideal autopilot (*OGI/IAC-0*), the guidance law is given in the form:

$$u = -\frac{V}{t_{\text{go}}}(-n_\lambda\lambda + n_\gamma\gamma + n_{\gamma_1}\gamma_1) \quad (1.69)$$

where:

$$n_\lambda = (n+2)(n+3) \quad (1.70a)$$

$$n_\gamma = 2(n+2) \quad (1.70b)$$

$$n_{\gamma_1} = (n+1)(n+2) \quad (1.70c)$$

For the case $n = 0$, the guidance law (1.65) is obtained. Non-zero values of the parameter n are desirable because they ensure that the commanded normal acceleration decays to zero faster as the flight approaches the terminal phase, thereby increasing system robustness by leaving more maneuvering margin in the presence of disturbances, actuator constraints, and similar effects.

In this work, a procedure for determining the remaining time-to-go is also proposed, namely:

$$t_{\text{go}} = \frac{R}{V}(1+k) \quad (1.71)$$

⁷The same cost function was also considered in explicit guidance presented in [30], described here in Section 1.2.4

where:

$$k = \begin{cases} \frac{2\bar{\gamma}^2 - \bar{\gamma}_1\bar{\gamma} + 2\bar{\gamma}_1^2}{30}, & \text{for } n = 0 \\ \frac{3\bar{\gamma}^2 - 2\bar{\gamma}_1\bar{\gamma} + 12\bar{\gamma}_1^2}{70}, & \text{for } n = 1 \\ \frac{4\bar{\gamma}^2 - 3\bar{\gamma}_1\bar{\gamma} + 36\bar{\gamma}_1^2}{126}, & \text{for } n = 2 \end{cases} \quad (1.72)$$

and $\bar{\gamma}$ and $\bar{\gamma}_1$ are given by Eq. (1.67).

Although the optimality criterion from [43] differs for $n \neq 0$ from the criterion considered in this dissertation ($n = 0$), the guidance law (1.69) will be used for comparison purposes due to its robustness-related advantages.

Optimal guidance laws against maneuvering targets using linear quadratic regulator theory (*OGL-CTIA*⁸) and differential game theory (*LQDG-CTIA*⁹) were proposed by Shaferman and Shima in [44]. Although arbitrary vehicle dynamics were assumed in the problem formulation, closed-form analytical expressions were derived only for the case of an ideal autopilot. The cost function to be minimized in the linear-quadratic regulator formulation is given by:

$$J = \frac{a}{2}Z_1^2(t_1) + \frac{b}{2}[Z_2(t_1) - \sigma_d]^2 + \frac{1}{2} \int_0^{t_1} u_{c,\perp\text{LOS}}^2 dt \quad (1.73)$$

where a and b are weighting coefficients, and σ_d represents the desired impact angle (see Fig. 1.7). The state variable Z_1 represents the zero-effort miss (ZEM), which in the case of engagement with a non-maneuvering target is given by:

$$Z_1 = z + \dot{z}t_{go} \quad (1.74)$$

while the state variable Z_2 , by analogy with Z_1 , represents the angle between the missile and the target at impact under zero-effort continuation, which in the case of a non-maneuvering target is given by:

$$Z_2 = \gamma_T - \gamma \quad (1.75)$$

This quantity, due to its physical interpretation, is referred to as the zero-effort angle (ZEA), and it is clear that minimizing the cost function (1.73) in the limiting case $a \rightarrow \infty$, $b \rightarrow \infty$ results in achieving impact with the desired impact angle.

The optimal guidance law is given in the following form:

$$u_{c,\perp\text{LOS}} = \frac{N_{\text{ZEM}}}{t_{go}^2}(z + \dot{z}t_{go}) + N_{\text{ZEA}} \frac{V \cos \gamma}{t_{go}}((\gamma_T - \gamma) - \sigma_d) \quad (1.76)$$

where the coefficients N_{ZEM} and N_{ZEA} in the limiting case are:

$$N_{\text{ZEM}}(a \rightarrow \infty, b \rightarrow \infty) = 6, \quad N_{\text{ZEA}}(a \rightarrow \infty, b \rightarrow \infty) = 2 \quad (1.77)$$

⁸Optimal Guidance Law - Constrained Terminal Intercept Angle

⁹Linear Quadratic Differential Game - Constrained Terminal Intercept Angle

Based on Eq. (1.49), expression (1.76) takes the following form:

$$u_{c,\perp\text{LOS}} = 6V_c\dot{\lambda} + 2\frac{V\cos\gamma}{t_{\text{go}}}((\gamma_T - \gamma) - \sigma_d) \quad (1.78)$$

If, instead of the direction normal to the line-of-sight, the command direction is taken to be normal to the missile velocity vector, the commanded normal acceleration becomes:

$$u_c = 6\frac{V_c}{\cos\gamma}\dot{\lambda} + 2\frac{V}{t_{\text{go}}}((\gamma_T - \gamma) - \sigma_d) \quad (1.79)$$

In the case of engagement with a stationary target, the zero-effort angle reduces to the flight-path angle γ , and, using expression (1.60), the optimal guidance law for a desired impact angle γ_1 is given as:

$$u_c = 6V\dot{\lambda} + 2\frac{V}{t_{\text{go}}}(\gamma - \gamma_1) \quad (1.80)$$

A similar methodology to that in [44] was applied by Rusnak, Weiss, Eliav, and Shima in [45] in 2014, by taking missile angle of attack into account such that, in addition to target interception, a desired pitch angle must also be achieved, as opposed to the desired flight-path angle considered in most previous works. In the special case where the dynamics of the angle of attack can be neglected, the guidance law (1.80) is recovered.

1.2.6 Nonlinear Optimal Guidance Laws with Impact Angle Constraint

Excluding spacecraft guidance methods, such as explicit guidance (presented in Section 1.2.4), the first work addressing the nonlinear formulation of the optimal guidance problem with a prescribed impact angle was that of V. Y. Glizer in [46], published in 1996. For a planar kinematic model of missile and target motion with constant speeds, using singular perturbation theory (*Singular Perturbation Theory - SPT*), optimal guidance was considered by minimizing the following cost:

$$J = \int_0^{t_1} (1 + ku^2) dt \quad (1.81)$$

where k is a positive constant that, in addition to minimizing the flight time, also penalizes control effort.

Although the paper states that the optimal guidance is given in closed-form, i.e., expressed as a function of state variables and fixed parameters, its values must be determined by solving a system of 3 or 4 nonlinear equations that include elliptic integrals. Although this procedure is computationally less demanding than solving the original two-point boundary value problem, an approximate solution was also proposed in order to improve practical applicability.

Minimization of the same¹⁰ cost function, but for a pre-known maneuvering target trajectory, was studied by Idan, Golan, and Guelman in [47], published in 1995¹¹. The optimal guidance is also given in closed form as a function of state variables and unknown parameters that must be determined numerically by solving a system of 4 nonlinear equations. It was found that two solutions exist, and the optimal one is the one yielding the smaller value of the functional (1.81), which must be determined by comparing both solutions.

Due to the computational complexity of the procedures proposed in [46] and [47], and due to the mixed structure of the cost function, which in addition to the control input also includes total flight time, these approaches will not be further considered in this dissertation.

Although the optimality of proportional navigation for the linearized missile-target engagement model was shown as early as 1965 [33], its relationship to optimal guidance for the nonlinear kinematic model remained largely unexplored until 2006, when P. Lu and F. R. Chavez presented their work on nonlinear optimal guidance [48]. The paper considers a planar kinematic model of a missile with arbitrary speed¹² against a stationary target, for both the unconstrained terminal flight-path angle case and the constrained case. The approach is also valid for a non-maneuvering moving target, provided that missile speed is replaced by the relative missile-target speed.

It was already shown in previous works [50] and [51] that proportional navigation (1.3), with a constant k satisfying the following inequality:

$$k = \frac{\gamma_1 - \gamma_0}{\gamma_1 - \lambda_0} > 2 \quad (1.82)$$

will lead to interception with the desired impact angle γ_1 , given the initial flight-path angle γ_0 and the initial line-of-sight angle λ_0 . Even when this inequality is not satisfied, proportional navigation can still achieve the desired impact angle by using a time-varying gain k [51].

The proportional navigation law (1.3) with constant k defined by (1.82) was compared with the nonlinear optimal guidance law, and the following conclusions were drawn [48]:

- When the terminal angle is unconstrained, proportional navigation with $k = 3$ and the nonlinear optimal guidance law are nearly identical;
- When the terminal angle is constrained, the two guidance laws produce significantly different trajectories.

Although the approach presented in this work is of significant importance for understanding nonlinear optimal guidance laws, it will not be further analyzed for the following two reasons:

¹⁰The cost function considered in [47] is: $J = kt_1 + \frac{1}{2} \int_0^{t_1} u^2 dt$

¹¹Although published later, [46] was known to the authors of [47] prior to its publication

¹²Analysis of proportional navigation for time-varying missile speed was carried out in [49]

1. The derivation procedure of the optimal guidance law is similar to the one presented in this dissertation¹³;
2. The procedure for determining unknown parameters of the optimal guidance law is not presented in [48], which is one of the main contributions of this dissertation.

It is known that optimal gains for full state feedback control are obtained by formulating the optimal control problem as a linear quadratic regulator and solving the Riccati equation in which the state matrix and input matrix appear, which for linear systems are constant [27]. One way to extend a similar methodology to nonlinear systems is to express these matrices as functions of the state variables so that the system can be described in the following form:

$$\dot{\mathbf{x}} = A(\mathbf{x})\mathbf{x} + \mathbf{b}u \quad (1.83)$$

and then solve the Riccati equation for such state-dependent matrices (State-Dependent Riccati-Equation - SDRE). This approach was applied by Ratnoo and Ghose in 2009 in [52], proposing the optimal guidance command in the following form:

$$u = - \left(\frac{N}{t_{\text{go}}} \right)^2 R \sin(\gamma_1 - \lambda) + \frac{\sqrt{2}NV}{t_{\text{go}}} \sqrt{\frac{\sin(\gamma - \gamma_1)}{\gamma - \gamma_1}} (\gamma - \gamma_1) \quad (1.84)$$

The time-to-go is determined as follows:

$$t_{\text{go}} = \frac{R}{V_{\text{ref}}} \quad (1.85)$$

where:

$$V_{\text{ref}} = \begin{cases} V_c, & \text{for } V_c > V/2 \\ V/2, & \text{for } V_c \leq V/2 \end{cases} \quad (1.86)$$

Numerical flight simulations for a single interception scenario and different values of the parameter N in Eq. (1.84) showed that an optimal value exists with respect to the control effort criterion. Although the procedure for determining the optimal value of this parameter for a specific scenario was not provided in the paper, the guidance law (1.84) will be used in the comparative analysis of optimal guidance methods in this dissertation.

In the decade that followed, a paradigm shift occurred in the fields of guidance and control of aerial vehicles. Namely, the traditional approach based on analytical closed-form expressions for guidance laws was gradually replaced by computational guidance and control (CG&C), whose main characteristics are [53]:

1. Instead of fixed-structure controllers, numerical algorithms are used to determine the control input;
2. These algorithms are executed in real time onboard the vehicle;

¹³ [48] was not known to the author at the time the optimal guidance algorithm presented in Chapter 2 was developed

3. The control computation is based on a model (model-based) or data (data-based), and does not require prior determination of a nominal trajectory, planning, or gain tuning dependent on a specific mission or vehicle.

The greatest impact of computational guidance and control methods has been observed in space-flight applications, particularly in atmospheric entry missions such as Earth re-entry or Mars landing missions (see e.g. [54]).

This trend has also introduced a large number of new methods that make it possible to solve the otherwise complex problem of nonlinear optimal missile guidance, now including constraints that are inherent to real systems (limits on maneuverability, field-of-view constraints, etc.), as well as system dynamics involving mechanical properties and aerodynamic and propulsion forces.

Combining ideas from model predictive control (MPC) and approximate dynamic programming, Padhi and Kothari proposed in 2009 in [55] a computationally efficient method for a class of optimal control problems, called Model Predictive Static Programming (MPSP). In the original paper, the method was validated on a ballistic missile guidance example, while later works [56, 57] considered guidance algorithms for tactical missiles with impact angle constraints.

Using proportional navigation in the form (1.63) as a guidance law for stationary target interception with a desired impact angle, Liu, Shen, and Lu proposed in 2016 in [58] an algorithm for computing the optimal navigation gain in closed form, while considering constraints on maximum normal acceleration and field of view. The optimization problem is formulated such that the control variable is the navigation gain, and the cost function to be minimized is the control effort given by (1.32). Since the optimal gain must be computed at each guidance cycle, fast convergence and reliability of the numerical algorithm are required. Therefore, the original optimization problem was approximated by a formulation suitable for convex optimization methods (specifically Second Order Cone Programming - SOCP), for which advanced algorithms with convergence-time guarantees exist.

Although the efficiency of model predictive control methods and numerical convex optimization algorithms has significantly improved due to recent advances driven by strong research interest, their use is justified when traditional approaches cannot be applied in real time. Namely, solving two-point boundary value problems arising from optimization formulations using indirect methods [59] is often not feasible at the speeds required for guidance purposes¹⁴. For this reason, authors have often resorted to approximate methods, abandoning the original concept of optimality.

These approximate methods are justified in cases where system dynamics are highly complex and constraints are present, making an analytical solution to the optimization problem practically impossible. However, when there is a possibility that the original optimization problem, solved via indirect methods, can be reduced to a form that is computationally feasible for numerical implementation, such reduction is meaningful and desirable due to its high solution accuracy.

For these reasons, traditional methods remain the subject of active research. Chen and Shima considered nonlinear optimal guidance to a stationary target in 2019 in [60]. It was shown that

¹⁴For example, guidance system update rates may be on the order of a few milliseconds

the classical formulation of the nonlinear optimal guidance problem, in which the control effort is minimized (as in [48]), is not well posed because a global solution does not exist.

Namely, starting from point M_0 in the direction defined by the velocity angle γ_0 (Fig. 1.12), it is possible to reach point M_1 under an impact angle γ_1 ¹⁵ along a trajectory consisting of two straight-line segments and one circular arc. If the normal acceleration on the circular segment is denoted by u_ρ , the control effort along the entire trajectory for a unit velocity is given by:

$$J = \int_0^{t_1} u^2(t) dt = \int_{\gamma_0}^{\gamma_1} u_\rho d\gamma = u_\rho(\gamma_1 - \gamma_0) = \frac{\gamma_1 - \gamma_0}{\rho} \leq \frac{2\pi}{\rho} \quad (1.87)$$

where ρ denotes the radius of curvature of the circular segment of the trajectory.

It is clear that for any $\epsilon > 0$, there exists $\rho > 0$ such that:

$$J = \int_0^{t_1} u^2(t) dt < \epsilon \quad (1.88)$$

i.e., it is always possible to choose a circle with a sufficiently large radius that yields an arbitrarily small value of the cost function.

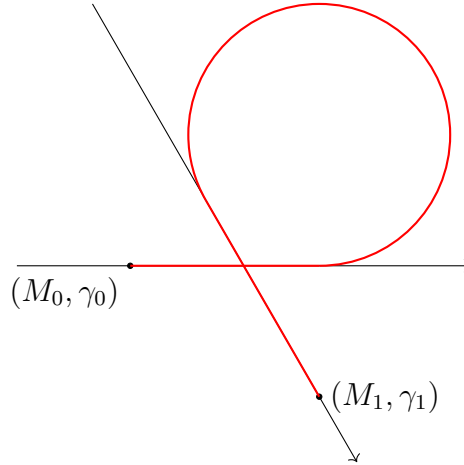


Figure 1.12: A possible trajectory between two points with specified directions consists of a circular arc and two line segments. [60]

In this way, the authors in [60] concluded that the problem of finding an energy-optimal trajectory with free final time, which does not appear in the cost function, is not a well-posed problem, and that optimal trajectories should be determined by minimizing the following performance index:

$$J = \int_0^{t_1} \left[k + \frac{1}{2}(1 - k)u^2(t) \right] dt, \quad k \in (0, 1) \quad (1.89)$$

where t_1 is still free. It was shown that the problem formulated in this way has a global solution, and a procedure for its determination was presented, which reduces to the numerical solution of a

¹⁵ γ_0 and γ_1 belong to the interval $[0, 2\pi]$

single nonlinear equation¹⁶ using the bisection method. A similar execution time of the procedure on a personal computer was reported as in [58], with the remark that the solutions obtained by the proposed method are global optima of the optimal guidance problem. However, the value of the parameter k in the functional (1.89) must be carefully selected in order to achieve a balance between flight time and energy consumption.

Although the analysis in [60], which shows that a global solution to the energy-optimal trajectory does not exist, is valid for a general planar motion model such as the one shown in Fig. 1.12, the question arises whether such an analysis is justified for a realistic missile–stationary target engagement scenario. Namely, it is sufficient to assume strict monotonicity of one of the coordinates, and it becomes clear that the stated theorem on the non-existence of a global optimum no longer holds. Since in most realistic scenarios the traveled distance along one coordinate (most often the x coordinate used to measure range) is a monotonically increasing quantity, this dissertation will not further use the approach presented in [60], nor the mixed form of the cost function (1.89).

Adopting a relative coordinate system with the origin at the moving maneuvering target, as originally introduced in [61], Li, Wang, He, and Lee proposed in 2021 in [62] an optimal guidance law based on a nonlinear kinematic model (NOIACG¹⁷). It was shown that some existing linear guidance laws (such as TSG [23]) are special cases of this law under the small-angle approximation. Since the derivation does not rely on a linearized missile–target engagement model, the proposed nonlinear guidance law can also be used in cases of large impact-angle errors. The guidance command is composed of two parts:

$$u = u_{PN} + u_B \quad (1.90)$$

where u_{PN} is the command component responsible for target interception (elimination of the zero-effort miss - ZEM), and u_B is the component responsible for eliminating the error between the desired impact angle and its prediction in the case where only the first component is active. For the case of stationary target interception, the nonlinear optimal guidance law is given in the following form:

$$u = NV\dot{\lambda} + NMV \cos \eta \frac{-V\eta}{R} - \frac{M(N-1)V}{R/|\dot{R}|}(\gamma_1 - \gamma) \quad (1.91)$$

where:

$$\eta = \gamma - \lambda, \quad |\dot{R}| = V \cos \eta \quad (1.92)$$

The parameters N and M in (1.91) are selected according to the desired convergence rate of $\eta \rightarrow 0$ and $\gamma \rightarrow \gamma_1$, at the cost of higher required acceleration. Since one of the main objectives of this dissertation is to develop an optimal guidance algorithm for the nonlinear missile–target engagement model, the guidance law (1.91) will be used in comparative analysis in later chapters.

One of the more recent examples addressing three-dimensional optimal guidance for a nonlinear kinematic model of stationary target interception without approximations leading to linearized models is the work [63] from 2025. The main advantage of the proposed method lies in its computational efficiency for solving the optimization problem, which is based on parameterization of the state and costate variables of the two-point boundary value problem. Using numerical flight simulations for different values of the desired impact angle, average execution times of the pro-

¹⁶This equation, as well as those that will be solved using the procedure proposed in this dissertation, contains elliptic integrals

¹⁷*Nonlinear Optimal Impact-Angle-Constrained Guidance*

posed algorithm were reported, reaching 15 ms in the best case. However, since the first step of the proposed method involves parameterization of state and costate variables using a linear combination of polynomial functions (splines), it belongs to the class of methods that approximate the original two-point boundary value problem; therefore, the method proposed in [63] will not be further analyzed.

1.2.7 Non-Optimal Guidance Laws with Impact Angle Constraint

The largest group of guidance laws for achieving a desired impact angle that are not derived using optimal control theory consists of guidance laws based on an extension of proportional navigation with an additional term commonly referred to in the literature as a *bias*, hence the name of this class of methods: Biased Proportional Navigation (BPN). The basic form of such a guidance law is given by:

$$\dot{\gamma}(t) = k\dot{\lambda}(t) + b(t) \quad (1.93)$$

where b is bias.

In addition to introducing a bias term, a modified form of proportional navigation frequently used to achieve desired impact angles is proportional navigation with time-varying gain:

$$\dot{\gamma}(t) = k(t)\dot{\lambda}(t) \quad (1.94)$$

Although these guidance laws are not optimal in the sense that they are not derived from a cost function to be minimized during their use¹⁸, they are often highly efficient and easy to implement in real systems. Therefore, the following sections present some of the most representative works from this class of methods.

The pioneering work in this area was published by Kim, Lee, and Han in 1998 [64]. The paper proposes a mixed-structure guidance law¹⁹ of the form:

$$u = N(t)V(\dot{\lambda}(t) - b(t)) \quad (1.95)$$

where the bias term is time-varying and computed as:

$$b(t) = \frac{\eta V(\gamma_1 - \lambda(t))}{N(t)R(t) \cos(\gamma(t) - \lambda(t))} \quad (1.96)$$

To mitigate excessively large commanded normal acceleration in the terminal phase while ensuring good target tracking in the initial phase, the gain N is also time-varying and defined as:

$$N(t) = N' + \frac{\rho + 0.5\alpha R(t)/V}{\cos(\gamma(t) - \lambda(t))} \quad (1.97)$$

¹⁸This does not imply that such a function does not exist, but rather that its minimization was not the primary design objective during derivation

¹⁹mixed structure in the sense of expressions (1.93) and (1.94)

The guidance parameters in expressions (1.96) and (1.97) for the stationary-target case are:

$$\eta = 1.3, \quad \alpha = 0.28, \quad N' = 2.9, \quad \rho = 0$$

The main advantage of the proposed guidance law is that it does not require knowledge of time-to-go, which is a characteristic of most methods in this class, and it does not rely on any approximations that would lead to linearized models. This allows analytical conditions for achieving the desired impact angle to be derived. However, it has been shown that when the missile and target do not deviate significantly from the collision triangle, the method approximates optimal guidance laws.

Although it was shown in earlier works [50] and [51] that proportional navigation (1.3), with a constant gain satisfying condition (1.82), ensures impact at a desired angle, Ratnoo and Ghose in 2008 [65] proposed a time-varying gain k that enables desired impact angles even when this condition is not satisfied. By integrating the proportional navigation law (1.3), the following relation is obtained:

$$\frac{\gamma(t_1) - \gamma_0}{\lambda(t_1) - \lambda_0} = k \quad (1.98)$$

Assuming $\lambda(t_1) = \gamma(t_1)$, which holds in the case of successful interception, the expression becomes:

$$\frac{\gamma(t_1) - \gamma_0}{\gamma(t_1) - \lambda_0} = k \quad (1.99)$$

This implies that the impact angle is a function of the navigation gain:

$$\gamma(t_1) = \frac{\lambda_0}{1 - 1/k} - \frac{\gamma_0}{k - 1} \quad (1.100)$$

The boundary values of achievable impact angles are:

$$\gamma(t_1) = \begin{cases} 2\lambda_0 - \gamma_0, & \text{for } k = 2 \\ \lambda_0, & \text{for } k \rightarrow \infty \end{cases} \quad (1.101)$$

thus:

$$\gamma(t_1) \in [2\lambda_0 - \gamma_0, \lambda_0], \quad k \geq 2 \quad (1.102)$$

Values of $k < 2$ are not feasible, since in such cases the commanded normal acceleration tends to infinity near the end of flight [23, 26].

To achieve the full range of impact angles $\gamma(t_1) \in [-\pi, 0]$, considering a ground-to-ground scenario with $\lambda_0 = 0$, the authors in [65] proposed a two-stage guidance scheme. In the first stage, if the desired impact angle does not belong to interval (1.102), the gain is set as:

$$k = \frac{0 - \gamma_0}{-\pi/2 - 0} = \frac{2}{\pi} \gamma_0 \quad (1.103)$$

Comparing this expression with (1.98), it follows that the main idea of the first phase is to bring the missile to a position directly above the target with zero flight-path angle: $\lambda = -\pi/2$, $\gamma = 0$, after which the desired impact angle $\gamma_1 = -\pi$ can be achieved in the second phase with constant gain $k = 2$. Since achieving impact angle of $\gamma_1 = -\pi$ is not always necessary, the first phase is terminated when expression (1.99) reaches the value 2.

If the switching instant is denoted by t^* , the condition is:

$$\frac{\gamma_1 - \gamma(t^*)}{\gamma_1 - \lambda(t^*)} = 2 \quad (1.104)$$

The guidance law can then be written as:

$$u = k(t)V\dot{\lambda} \quad (1.105)$$

- if $\frac{\gamma_1 - \gamma_0}{\gamma_1 - \lambda_0} \geq 2$, then $k(t) = k$ computed using (1.99):

$$k(t) = k = \frac{\gamma_1 - \gamma_0}{\gamma_1 - \lambda_0} \quad (1.106)$$

- if $\frac{\gamma_1 - \gamma_0}{\gamma_1 - \lambda_0} < 2$, then:

$$k(t) = \begin{cases} \frac{2}{\pi}\gamma_0, & \forall t < t^* \\ 2, & \forall t \geq t^* \end{cases} \quad (1.107)$$

This procedure is valid only under ideal command tracking, which is not realistic in practice. Therefore, a real-time version is also proposed in [65], in which the gain is updated at each guidance cycle:

$$k(t) = \begin{cases} \frac{\gamma_1 - \gamma(t)}{\gamma_1 - \lambda(t)}, & \frac{\gamma_1 - \gamma(t)}{\gamma_1 - \lambda(t)} \geq 2 \\ \frac{0 - \gamma(t)}{-\pi/2 - \lambda(t)} = \frac{\gamma(t)}{\pi/2 + \lambda(t)}, & \frac{\gamma_1 - \gamma(t)}{\gamma_1 - \lambda(t)} < 2 \end{cases} \quad (1.108)$$

The same authors extended the presented approach in 2010 [66] by considering non-maneuvering moving targets instead of stationary ones. Due to the two-stage guidance structure, as in [65], the main drawback of these approaches lies in abrupt changes in the commanded normal acceleration at the switching instants between the first and second guidance phases.

Considering a guidance law of the form (1.93), Erer and Merttopçuoğlu in 2012 [67] proposed a method for computing the bias term b during flight.

By integrating expression (1.93), the following is obtained:

$$\gamma(t) = \gamma_0 + k(\lambda(t) - \lambda_0) + \int_{t_0}^t b(t) dt \quad (1.109)$$

Thus, the final impact angle, taking into account that $\lambda(t_1) = \gamma(t_1)$ in the case of successful target interception, can be computed as follows:

$$\gamma(t_1) = \frac{k\lambda_0 - \gamma_0 - \int_{t_0}^{t_1} b(t) dt}{k - 1} \quad (1.110)$$

From the previous expression, it is clear that the impact angle can be controlled by shaping the integral of the bias term. The profile of the bias term b can be completely arbitrary, including intervals where b is zero, and the proposed guidance scheme assumes the following profile:

$$b(t) = \begin{cases} 0, & \forall t < t_{\text{beg}} \\ \frac{k\lambda_0 - \gamma_0 - (k-1)\gamma_1}{t_{\text{end}} - t_{\text{beg}}}, & \forall t_{\text{beg}} \leq t < t_{\text{end}} \\ 0, & \forall t \geq t_{\text{end}} \end{cases} \quad (1.111)$$

where t_{beg} and t_{end} denote the initial and final time instants in which the bias term is active, respectively. These instants are arbitrary, and since trajectory shaping is preferably performed in the early phase of flight, while $\dot{\lambda}$ is still small, the bias term is most often active from the beginning of flight, i.e., $t_{\text{beg}} = 0$.

Although the presented guidance procedure is very practical—since its implementation requires only knowledge of the line-of-sight rate—it can be interpreted as a form of open-loop control when it comes to shaping the desired impact angle. Namely, after the bias term ceases to act, guidance continues with classical proportional navigation with constant gain k , which in the presence of disturbances still ensures target interception, but the resulting trajectory will differ from the nominal one for which (1.110) holds. However, since such disturbances are usually small, and small deviations in the achieved impact angle are often acceptable compared to miss distance errors, the use of expression (1.93) together with (1.111) remains justified for impact-angle-constrained guidance.

A similar guidance formulation as in [62] was considered by Lee, Kim, and Tahk in 2013 [68], for a linearized model of missile interception of a stationary target. Unlike the previous two works, [65] and [67], which assume a two-stage guidance structure, this work proposes a full-flight guidance law based on the form (1.90):

$$u(t) = u_{PN}(t) + u_B(t) \quad (1.112)$$

where u_{PN} is the command component responsible for target interception (proportional navigation law (1.63)), and u_B is the component responsible for compensating the error between the desired

impact angle and its predicted value when only the first control component is active:

$$u_B(t) = \frac{V[k\lambda(t) - \gamma(t) - (k-1)\gamma_1]}{t_{go}} \quad (1.113)$$

Comparing the previous expression with the bias term proposed in [67], i.e., expression (1.111), it is clear that in this case the bias term is active over the entire trajectory and is updated at each guidance cycle using the instantaneous values of the flight-path angle and the line-of-sight angle. Moreover, it has been shown that for $k = 3$, it is identical to the optimal guidance law OGL/IAC-0 from [42], presented in the section on optimal linear guidance laws. This implies that the guidance law (1.93) together with (1.111) is optimal when the bias term is applied over the entire trajectory with continuous updating, which also requires knowledge of the time-to-go.

From the expression for the bias term (1.111), it is clear that its magnitude depends on the time interval during which it is active. For shorter activation intervals, larger values are required in order to satisfy (1.110), i.e., to ensure that the integral of the bias term achieves the required value along the trajectory. Shorter intervals have the advantage that the open-loop shaping phase ends earlier, leaving more time for classical proportional navigation, which is more robust to disturbances since it operates in closed-loop form. However, larger bias magnitudes in the early phase of flight, especially for large impact angles, may lead to excessive seeker line-of-sight angles, which can result in loss of target lock.

Taking into account the limitation of the maximum allowable line-of-sight angle, Kim, Park, and Tahk proposed in 2013 [69] (as well as in [70] from 2017) a guidance law of the form (1.93), where the bias term varies exponentially:

$$b(t) = \frac{B_{\text{ref}}}{\tau} e^{-t/\tau} \quad (1.114)$$

where B_{ref} is the value of the integral of the bias term over the entire trajectory, which, based on Eq. (1.110), for the case $\lambda_0 = 0$, is given by:

$$B_{\text{ref}} = \int_{t_0}^{t_1} b(t) dt = (1-k)\gamma_1 - \gamma_0 \quad (1.115)$$

where τ is a time constant chosen such that $\tau \leq R_0/7V$.

If the maximum allowable line-of-sight angle is reached during flight, the bias term becomes:

$$b(t) = (1-k)\dot{\lambda} \quad (1.116)$$

and the guidance continues with proportional navigation with gain $k = 1$, in order to maintain the maximum line-of-sight angle, until the value of the bias term given by (1.114) becomes smaller than that given by (1.116).

Although the implementation of the method presented in [69] requires only knowledge of the line-of-sight angular rate, the insufficiently defined selection procedure for the time constant τ raises questions regarding the effectiveness of the method in achieving desired impact angles.

A similar guidance formulation for stationary targets as in [65] was investigated by Tekin and Erer in 2015 [71]. The gains in the guidance law are determined numerically, and the proposed approach, based on a nonlinear kinematic model, also accounts for constraints on maximum normal acceleration and line-of-sight angle.

Although two-stage guidance methods remain an active research topic [72], the unavoidable issue of abrupt jumps in commanded normal acceleration remains their major drawback.

Apart from the previously listed guidance methods for achieving the desired impact angle, among those that are not based on optimal control theory and do not belong to the class of modified proportional navigation methods, the most numerous are those belonging to the class of nonlinear control systems, in particular Sliding Mode Control (SMC) methods [73–77].

Other methods assume a geometric approach, such as Circular Navigation Guidance [78, 79], Differential Geometric Guidance [80], as well as approaches in which the guidance command is given in terms of the flight path angle [81] (instead of its rate of change $\dot{\gamma}$), for which implementation requires knowledge of the line-of-sight angle (instead of its rate of change $\dot{\lambda}$).

1.2.8 Three-Dimensional Guidance Laws with Impact Angle Constraint

Although missile guidance is inherently a spatial problem that must be described in three dimensions using nonlinear vector equations, the majority of guidance methods have been derived by analyzing a planar model of the missile–target engagement. Such a model is justified in cases where the system of equations describing the motion can be separated into two independent identical sets of equations describing guidance in two mutually orthogonal planes [82], i.e., when the missile has two identical and independent guidance channels and when the missile trajectory can be linearized in the vicinity of the initial collision triangle [11].

Even when the missile has two independent guidance channels²⁰, linearized models that assume small deviations from the collision triangle are rarely justified, especially in the case of targets with large impact angles, where trajectory shaping is required before steering toward the target itself.

For this reason, three-dimensional guidance laws, in addition to planar ones, have been the subject of significant research interest. Some of the previously mentioned guidance laws are inherently spatial, such as explicit guidance [29, 30], which in vector form is given by (1.43). Guidance algorithms belonging to the class of computational guidance and control, due to their ability to solve large systems of equations with multiple constraints, are most often presented through three-dimensional guidance examples, such as [54, 56, 63].

In [83] from 2024, the convergence and optimality of the proposed three-dimensional guidance law (*NOIVCG* - *Near-Optimal Impact-Vector-Control Guidance*), based on a predictor–corrector concept, were theoretically analyzed, thereby improving the reliability compared to earlier three-dimensional guidance methods, such as [84, 85]. It was also shown that the previously mentioned

²⁰most commonly in the form of two planes in which the command is generated

optimal guidance laws *TSG* [23] and *OGL/IAC-0* [42,43] are special cases of the proposed method in planar guidance, and therefore its implementation also requires knowledge of the time-to-go t_{go} .

Sliding mode control (*SMC*) is another example of nonlinear spatial guidance methods [73, 86].

In contrast to three-dimensional guidance methods expressed in classical form as a single guidance law that can be directly applied to compute the command [30,87], there also exist methods in which the desired orientation of the missile velocity vector at the impact point is achieved in multiple phases during flight. One such example involves first steering the missile into a plane defined by the desired terminal velocity vector, followed by planar guidance within that plane in order to achieve the desired impact angle, as shown in Fig. 1.13. Different guidance laws may be used in these two phases. For example, in [88] from 2022, Erer, Tekin, and Özgören proposed a mixed guidance approach in which Apollo guidance (1.50) is used to approach the desired plane, while simultaneously the missile is guided within that plane using the *TSG* method (see Section 1.2.5).

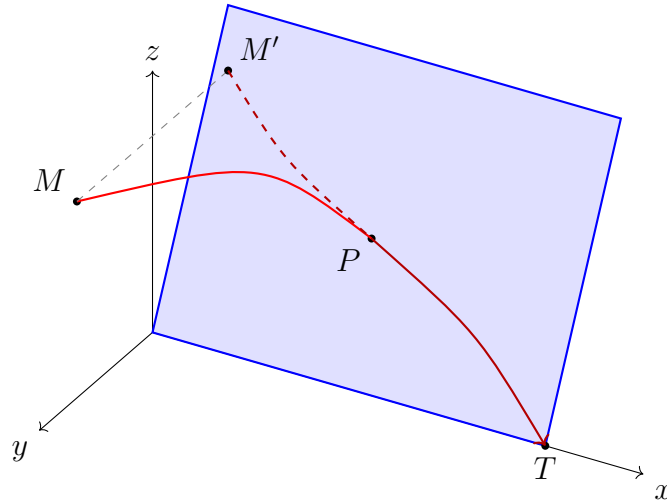


Figure 1.13: Three-dimensional guidance in two phases: steering into the terminal velocity vector plane (trajectory segment MP) and planar guidance (segment PT)

Considering that, in the vast majority of cases, a missile possesses two independent guidance channels, three-dimensional guidance is most commonly reduced to two planar guidance problems. Since the trajectory-shaping requirements in the horizontal plane are generally less demanding than those in the vertical plane, classical proportional navigation is often employed for guidance in the horizontal plane. Guidance in the vertical plane is then performed using one of the guidance laws discussed previously. In particular, there are also cases in which proportional navigation is employed in both planes to achieve the desired orientation of the velocity vector at the target, as in [50,51], with the proportional navigation gain being adaptively determined during each guidance cycle.

Based on the foregoing discussion, it is evident that the planar formulation of motion for achieving a desired impact angle remains justified, whether considering two-channel guidance in the horizontal and vertical planes or a two-phase guidance scheme consisting of steering toward the plane containing the desired terminal velocity vector followed by motion within that plane. Consequently, the planar missile–target engagement model remains a justified and widely applicable framework for the analysis and development of guidance laws with impact-angle constraints [89].

1.3 Subject and Objectives of the Dissertation

The literature review presented in the previous sections was not intended to provide an exhaustive survey of all guidance methods capable of achieving a desired terminal flight-path angle. Rather, it focused on the most influential contributions that are relevant for understanding both the underlying concepts and the historical development of the field. Particular emphasis was placed on optimal guidance laws, although several representative non-optimal approaches were also discussed.

In the context of missile guidance, the term *optimal guidance law* refers to a guidance strategy obtained by minimizing a prescribed performance index. The most commonly considered formulations are time-optimal and energy-optimal problems, as well as various combinations of the two objectives.

The majority of published works on optimal guidance belong to the class of classical guidance laws expressed in closed form, where the commanded acceleration is computed directly from the state variables through an explicit analytical expression. Such solutions are commonly referred to as *closed-form guidance laws*. Alongside these classical approaches, an increasing number of modern guidance methods have emerged that require computational procedures substantially more complex than the evaluation of a single analytical expression.

The principal advantage of closed-form guidance laws lies in their extremely low computational burden. Since the command can be evaluated almost instantaneously, such methods are particularly attractive for real-time applications. However, obtaining explicit analytical solutions generally requires the introduction of simplifying assumptions. Most commonly, these assumptions take the form of linearization of the engagement geometry around the nominal collision triangle. As a consequence, the resulting solutions are valid only within a relatively limited region of the state space where the linearization remains accurate [90]. Therefore, although these methods are often referred to as optimal guidance laws, they do not represent purely optimal solutions of the original nonlinear problem. Another significant limitation is the requirement for accurate knowledge of the time-to-go, a quantity that cannot be measured directly and must instead be estimated.

In contrast, computationally intensive guidance methods typically address the original optimal control problem without resorting to linearization or similar simplifying approximations. Such methods generally fall into two categories: indirect and direct approaches.

Indirect methods employ the minimum principle and related necessary conditions of optimality to transform the original optimal control problem into a two-point boundary-value problem. The resulting solution satisfies the optimality conditions directly and is therefore characterized by a high level of accuracy. For this reason, indirect methods are often regarded as the most rigorous approach for obtaining truly optimal trajectories.

Direct methods, on the other hand, parameterize the continuous optimal control problem as a finite-dimensional optimization problem. The resulting constrained optimization problem can then be solved numerically using well-established nonlinear programming algorithms. Although direct methods are generally more robust and less susceptible to convergence difficulties than indirect methods, the obtained solutions satisfy the optimality conditions only approximately and are therefore typically less accurate.

From the perspective of guidance-system design, the superior accuracy of indirect methods makes them particularly attractive whenever a truly optimal solution of the original nonlinear problem is desired. Their principal disadvantage, however, lies in the computational effort required to solve the associated boundary-value problem. In many practical applications, especially those involving real-time onboard implementation, this computational burden becomes the dominant obstacle.

Although several examples presented in the literature review pursued the optimal-control formulation through indirect methods, relatively little attention has been devoted to their implementation on actual flight hardware with limited computational resources. In this regard, it is important to distinguish between the numerical method itself and its implementation. As emphasized in [59], a numerical method is described by mathematical equations and algorithmic logic, whereas implementation involves translating that method into software and, ultimately, into onboard hardware capable of executing it in real time. Consequently, the development of an optimal guidance method and the development of an efficient implementation algorithm should be regarded as related but distinct challenges.

The objectives of this dissertation are therefore twofold:

- the development of an optimal guidance method capable of achieving a prescribed impact angle;
- the development of a computationally efficient algorithm that enables the proposed method to be implemented in real time on hardware with limited computational resources.

The guidance problem considered in this dissertation is formulated as an energy-optimal interception problem against a stationary target with a prescribed terminal flight-path angle. The underlying engagement model is fully nonlinear and does not rely on trajectory linearization or small-angle approximations. As noted in [46], the problem is inherently complex and, consequently, most previously published works have either adopted simplified models or focused on the development of efficient numerical procedures.

The objective pursued herein is to derive a guidance law in closed-feedback form while preserving the nonlinear nature of the original problem. In accordance with the philosophy advocated in [46], the optimal command should be expressed explicitly as a function of the state variables and a limited set of unknown parameters, whose values can be obtained through a computational procedure substantially simpler than solving the complete nonlinear two-point boundary-value problem.

Several studies reviewed previously address problems closely related to the one considered here. Nevertheless, the proposed formulation differs from existing approaches in several important respects. First, unlike the majority of guidance laws that address impact-angle-constrained interception within a linearized framework, the proposed method is derived directly from the nonlinear engagement kinematics and therefore remains valid without relying on small-heading-error approximations. Second, unlike [60], which considers optimal interception without a prescribed terminal flight-path angle, the present formulation explicitly incorporates the impact-angle constraint. Third, compared with [46, 47], where the optimal solution is obtained through the solution of multiple nonlinear equations involving elliptic integrals, the proposed formulation reduces the

computational procedure to the solution of a single nonlinear equation. Finally, although [48] derives a closed-form feedback structure capable of achieving both interception and a desired impact angle, it does not provide a practical procedure for determining the associated unknown parameters in real time.

A significant portion of this dissertation is devoted to the practical implementation of the proposed guidance method. In addition to the development of the guidance methodology itself, an important objective of the present research is to demonstrate that the proposed method can be executed on resource-constrained onboard hardware representative of actual flight vehicles, thereby bridging the gap between theoretical optimal guidance and practical real-time implementation.

1.3.1 Starting Hypotheses

Based on the considerations presented in the preceding sections, the following hypotheses are adopted throughout this dissertation:

- The guided vehicle is considered to be a missile, although the resulting formulation is sufficiently general to be applied to other flight vehicles equipped with either aerodynamic control surfaces or thrust-vector-control systems.
- The engagement scenario considered in this research involves a stationary target. The proposed guidance method is primarily intended for cruise and ballistic missiles against fixed ground targets. Extensions to moving targets are possible provided that the target motion is known or can be accurately estimated by the guidance system.
- During the derivation of the guidance law, the missile speed is assumed to be constant. This assumption is widely accepted in missile guidance theory because longitudinal accelerations are generally much smaller than the lateral accelerations used for maneuvering. Furthermore, in practical implementation the guidance algorithm is executed repeatedly in real time, such that the optimization problem is reformulated using updated flight parameters, including the current missile speed.
- The missile is assumed to possess two independent and decoupled guidance and control channels corresponding to two mutually perpendicular planes. Consequently, the three-dimensional engagement can be decomposed into two planar guidance problems, and the guidance-law derivation is performed using a planar engagement model.
- During the derivation of the guidance law, the missile is represented by a planar kinematic model. The inner-loop flight-control system is assumed to be sufficiently fast and well tuned so that autopilot dynamics can be neglected and the commanded normal acceleration can be considered to be achieved instantaneously.
- The optimal guidance problem is formulated within the framework of optimal control theory, with the missile normal acceleration adopted as the control input. Since the normal acceleration represents the quantity generated by the guidance law and tracked by the flight-control system, it naturally serves as the optimization variable.

- No explicit hard constraint is imposed on the control input during the derivation of the optimal guidance law. Instead, the control effort is minimized through the selected performance index, resulting in a soft limitation of the commanded normal acceleration. The influence of maneuverability limits is subsequently evaluated through higher-fidelity simulations and experimental validation.

1.3.2 Scientific Research Methods

The research presented in this dissertation employs the following scientific methods:

- Analysis of relevant literature sources and systematization of existing knowledge in the field of missile guidance, with particular emphasis on guidance laws incorporating impact-angle constraints;
- Analytical modeling of the missile–target engagement geometry, and definition of the corresponding optimal control problem;
- Application of optimal control theory, primarily Pontryagin’s Minimum Principle, for deriving the necessary conditions of optimality and formulating the associated two-point boundary-value problem;
- Analytical transformation and simplification of the resulting system of differential equations in order to obtain a computationally efficient formulation suitable for real-time implementation;
- Numerical solution of the reduced-order problem for determining the optimal guidance command and evaluating the performance of the proposed guidance method;
- Numerical verification through six-degree-of-freedom (6DOF) missile flight simulations under realistic operating conditions;
- Laboratory experimental evaluation of the proposed algorithm on available resource-constrained onboard hardware representative of real missile systems;
- Field experimental validation through real-world flight test to verify the robustness, accuracy, and overall effectiveness of the guidance system in operational environment.

1.3.3 Expected Scientific Contribution

Taking into account the research problem, the stated objectives, the adopted hypotheses, the applied scientific methods, and the analysis of the available literature, the expected scientific contribution of this dissertation is:

- The development of a novel guidance method for energy-optimal interception of stationary targets with a prescribed terminal flight-path angle, based on a nonlinear engagement model and formulated in a computationally efficient closed-feedback form suitable for real-time implementation;
- The development of an efficient numerical algorithm for real-time computation of the optimal guidance command on resource-constrained onboard computing platforms.

In addition to the expected scientific contribution, the anticipated professional contribution includes:

- The development of software optimized for implementation on commercially available micro-controllers intended for onboard flight-computer applications;
- Experimental validation of the proposed guidance method on representative embedded hardware platforms.

Potential practical applications of the proposed method include:

- Guidance during the launch and transition-to-cruise phases of cruise missile flight;
- Waypoint guidance with prescribed flight-path angles at intermediate waypoints, enabling smooth trajectory transitions;
- Midcourse guidance applications in which the terminal conditions are shaped to facilitate a reliable transition to a subsequent terminal guidance phase;
- Terminal guidance requiring the achievement of a prescribed impact angle at the target;
- All-flight guidance of ballistic missiles based on a predefined flight-path-angle profile;
- Ballistic missile applications where near-vertical target impact may reduce miss distance caused by navigation errors [42].

1.4 Structure of the Dissertation

The research presented in this dissertation is organized into four main parts.

The first part provides a comprehensive review of guidance methods with impact-angle constraints. Particular emphasis is placed on optimal guidance laws, although representative non-optimal approaches are also discussed for completeness. The review covers both guidance methods derived from linearized missile–target engagement models and those based on nonlinear formulations, with the aim of identifying their principal advantages, limitations, and areas of applicability.

The second part is devoted to the formulation and analytical solution of the optimal guidance problem considered in this research. Following the development of the mathematical model and the corresponding optimal control formulation, the original problem is transformed into a computationally efficient form suitable for practical implementation. An algorithm for determining the unknown parameters defining the optimal trajectory is then developed and analyzed. The obtained results are compared with existing guidance methods addressing similar objectives. In addition, a special case of the general problem, corresponding to the optimal climb phase of a cruise missile, is investigated in greater detail.

The third part focuses on the real-time implementation of the proposed guidance method. Particular attention is devoted to computational efficiency, numerical robustness, and implementation on resource-constrained onboard computing platforms. The proposed algorithm is first evaluated through numerical six-degree-of-freedom missile flight simulations involving representative cruise-missile and ballistic-missile scenarios. The implementation is then further assessed through experimental testing on actual hardware representative of onboard flight-computer systems.

The final part presents a critical analysis of the obtained results and discusses the extent to which the research objectives have been achieved. The dissertation concludes with a summary of the main findings and an outline of possible directions for future research.

Chapter 2

Optimal Guidance

2.1 Introduction

In accordance with the first research objective stated in Section 1.3, this chapter is devoted to the development of an optimal guidance method for achieving a prescribed flight-path angle at a specified point in space. The term *specified point* is intentionally used instead of *target*, since the proposed formulation is equally applicable to intermediate waypoints and other mission-defined locations in addition to the final target point.

Optimality is defined in terms of minimizing control effort, resulting in energy-optimal trajectories. This choice is motivated by two considerations. First, the proposed method is primarily intended for cruise and ballistic missile applications involving stationary ground targets, for which the terminal time is generally not a design objective. Consequently, the performance index does not contain a term associated with flight time, thereby yielding a purely energy-optimal solution. Second, minimizing a quadratic measure of control effort results in a soft limitation of the control input. In contrast to formulations involving explicit inequality constraints, often referred to as *hard constraints*, the control authority is indirectly restricted through the optimization criterion itself. This significantly simplifies the mathematical formulation while preserving practical relevance.

Although real flight vehicles are inherently subject to maneuverability limits, the absence of explicit control constraints during the guidance-law derivation is justified by the intended application domain. The objective is to obtain optimal trajectories representative of realistic missile missions rather than mathematically admissible but operationally impractical solutions. The influence of physical limitations, including maneuverability constraints and control-system dynamics, is addressed separately during the verification and validation stages.

The underlying engagement model employed throughout the guidance-law derivation is based on the kinematics of a lag-free missile. This assumption is adopted solely for the purpose of analytical development. The applicability of the resulting guidance method under realistic operating conditions is subsequently investigated in Chapter 3, where the proposed algorithm is evaluated

using six-degree-of-freedom missile models, realistic control-system dynamics, and maneuverability limitations.

The principal contribution of this chapter is presented in Section 2.2.4, where the complete algorithm is derived for the most general case of achieving a prescribed flight-path angle at an arbitrary point in space. Particular attention is also devoted to the special case of the cruise-missile climb phase, for which an analytical solution in closed-feedback form is obtained.

The chapter concludes with a comparative analysis of the proposed guidance method and several representative guidance laws addressing the same objective. The comparison serves to demonstrate the uniqueness of the obtained solution and to verify its optimality with respect to the adopted performance criterion.

2.2 Optimal Guidance Algorithms

2.2.1 Kinematic Model of Missile Motion

Consistent with the assumptions and starting hypotheses adopted in this dissertation, a planar constant-speed kinematic model of missile motion is considered. The derivation begins with motion in the vertical plane, since this represents the more general case due to the presence of gravitational acceleration. The corresponding geometry is shown in Fig. 2.1a.

The position of the missile, represented by the material point M , is defined by the Cartesian coordinates x and h , denoting the horizontal range and altitude, respectively. The angle γ represents the flight-path angle. The boundary conditions specify the initial state at time t_0 and the terminal state at time t_1 . Furthermore, V denotes the missile velocity, g is the gravitational acceleration, and U is the component of the resultant specific force acting perpendicular to the velocity vector (it includes all contributions except gravity).

Under these assumptions, the planar kinematic model is described by the following equations:

$$\dot{x} = V \cos \gamma \quad (2.1a)$$

$$\dot{h} = V \sin \gamma \quad (2.1b)$$

$$\dot{\gamma} = \frac{U - g \cos \gamma}{V} \quad (2.1c)$$

If the specific force U is assumed to include a gravity-compensation term, it can be expressed as:

$$U = u + g \cos \gamma \quad (2.2)$$

Substituting (2.2) into (2.1c) yields:

$$\dot{\gamma} = \frac{u}{V} \quad (2.3)$$

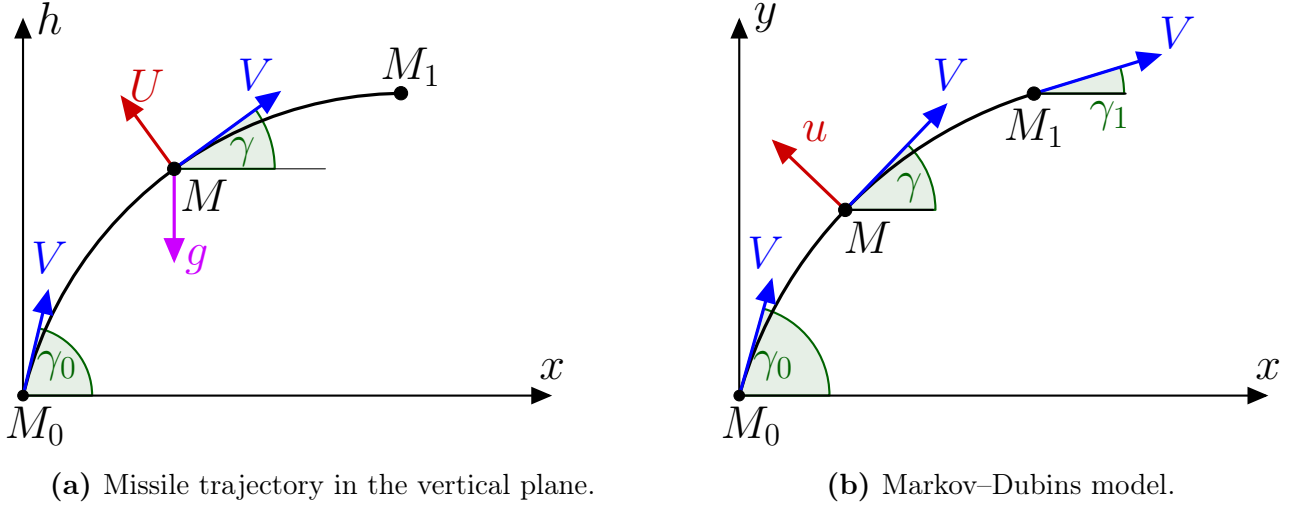


Figure 2.1: Planar kinematic models of missile motion.

In this manner, the gravitational acceleration is eliminated from the equations describing missile motion, resulting in a model that is independent of the particular plane in which the motion is considered. Such an approach is justified because gravity compensation can be performed independently of the selected guidance law, as indicated by (2.2). Consequently, the control variable adopted throughout the remainder of this work is the component of the specific force unrelated to gravity compensation, denoted by u .

It should also be noted that the required nongravitational specific force can be generated through different control mechanisms. From the perspective of the guidance law, the quantity u represents the commanded specific force required for trajectory shaping, regardless of whether the vehicle is controlled through aerodynamic control surfaces, thrust-vector control, or another actuation mechanism. This further contributes to the generality of the adopted kinematic model.

After removing the gravitational term from (2.1), the resulting planar model becomes identical to the Markov–Dubins model introduced in Section 1.2.1. Therefore, the kinematic model shown in Fig. 2.1b will be used throughout the remainder of this dissertation, irrespective of the plane in which the motion is analyzed.

2.2.2 Optimization Problem

As shown in Section 2.2.1, missile motion can be represented by a general planar model identical to the Markov–Dubins model. Therefore, the optimization problem considered in this dissertation is obtained by modifying the optimization problem (1.1) as follows:

- An optimal trajectory is defined as the trajectory connecting points M_0 and M_1 for which the control effort, represented by the performance index

$$J = \int_{t_0}^{t_1} u^2(t) dt \quad (2.4)$$

attains its minimum value.

The quadratic form of the performance index is adopted both for its favorable mathematical properties in optimal control problems and for its physical interpretation as a measure of control effort. For aerodynamically controlled missiles, the control variable is directly related to the angle of attack, while the control-induced aerodynamic losses are approximately proportional to its square. A similar interpretation applies to thrust-vector-controlled systems, where control deflection results in losses in the effective thrust. Therefore, minimization of the quadratic control effort provides a convenient measure of the energy expenditure associated with trajectory control.

- The initial time is fixed as $t_0 = 0$. The total flight time from point M_0 to point M_1 is not of interest; therefore, the terminal time t_1 is considered free.¹
- No explicit constraint is imposed on the control input:

$$|u| \leq \infty \quad (2.5)$$

In practice, such an assumption is unrealistic because the maximum achievable normal acceleration is limited by the maneuvering capability of the flight vehicle. Nevertheless, removing the control constraint considerably simplifies the optimization problem and, in certain cases, enables an analytical solution to be obtained. The assumption is further justified by the nature of the problem itself, since the objective is the minimization of control effort. As a result, the optimal trajectories obtained tend to require moderate control inputs, which, as will be demonstrated in subsequent examples, may remain within realistic maneuverability limits throughout the entire flight.

Based on the above modifications of problem (1.1), the optimization problem considered in this dissertation can be stated as follows:

$$\begin{aligned}
 &\text{Minimize} \quad J = \int_0^{t_1} u^2(t) dt \\
 &\text{subject to} \quad \dot{x} = V \cos \gamma \\
 &\quad \quad \quad \dot{y} = V \sin \gamma \\
 &\quad \quad \quad \dot{\gamma} = \frac{u}{V} \\
 &\text{and boundary conditions} \quad x(0) = x_0 \\
 &\quad \quad \quad y(0) = y_0 \\
 &\quad \quad \quad \gamma(0) = \gamma_0 \\
 &\quad \quad \quad x(t_1) = x_1 \\
 &\quad \quad \quad y(t_1) = y_1 \\
 &\quad \quad \quad \gamma(t_1) = \gamma_1
 \end{aligned} \quad (2.6)$$

¹Using the terminology of optimal control theory, a variable is said to be *free* when its value is not prescribed by the boundary conditions.

By introducing an additional state variable z , whose value evolves from the initial condition $z(0) = 0$ according to:

$$\dot{z} = u^2 \quad (2.7)$$

the optimization problem in Lagrange form is transformed into an equivalent Mayer problem, in which the objective is to minimize the terminal value:

$$P = z(t_1) \quad (2.8)$$

The resulting optimal control problem will be solved using Pontryagin's Minimum Principle, following the procedure outlined in Appendix A.

By forming the Pontryagin function (Hamiltonian):

$$H = \lambda_x V \cos \gamma + \lambda_y V \sin \gamma + \lambda_\gamma \frac{u}{V} + \lambda_z u^2 \quad (2.9)$$

the corresponding costate equations can be obtained as:

$$\dot{\lambda}_x = -\frac{\partial H}{\partial x} = 0 \quad (2.10a)$$

$$\dot{\lambda}_y = -\frac{\partial H}{\partial y} = 0 \quad (2.10b)$$

$$\dot{\lambda}_\gamma = -\frac{\partial H}{\partial \gamma} = V(\lambda_x \sin \gamma - \lambda_y \cos \gamma) \quad (2.10c)$$

$$\dot{\lambda}_z = -\frac{\partial H}{\partial z} = 0 \quad (2.10d)$$

For problems with terminal constraints of the form $\Psi(\mathbf{x}_f, t_f) = 0$, the transversality conditions are given by [22]:

$$\begin{aligned} \boldsymbol{\lambda}(t_f) &= (\lambda_0 P_{\mathbf{x}_f} + \mathbf{L}^T \boldsymbol{\Psi}_{\mathbf{x}_f}) \\ H(t_f) &= -(\lambda_0 P_{t_f} + \mathbf{L}^T \boldsymbol{\Psi}_{t_f}) \end{aligned} \quad (2.11)$$

In the problem under consideration, the terminal position and terminal flight-path angle are prescribed. Consequently, the terminal constraint manifold is defined as:

$$\boldsymbol{\Psi} = \begin{bmatrix} x(t_1) - x_1 \\ y(t_1) - y_1 \\ \gamma(t_1) - \gamma_1 \end{bmatrix} \quad (2.12)$$

and the associated vector of Lagrange multipliers is $\mathbf{L}^T = [\eta_1, \eta_2, \eta_3]$, since the terminal constraint consists of three components.

The transversality conditions (2.11) therefore become:

$$\lambda_x(t_1) = \eta_1 \quad (2.13a)$$

$$\lambda_y(t_1) = \eta_2 \quad (2.13b)$$

$$\lambda_\gamma(t_1) = \eta_3 \quad (2.13c)$$

$$\lambda_z(t_1) = \lambda_0 \quad (2.13d)$$

From (2.10) and (2.13), it follows that the costates λ_x , λ_y , and λ_z remain constant throughout the trajectory, i.e.,

$$\lambda_x(t) = \eta_1 \quad (2.14a)$$

$$\lambda_y(t) = \eta_2 \quad (2.14b)$$

$$\lambda_z(t) = \lambda_0 \quad (2.14c)$$

In addition, the terminal Hamiltonian condition reduces to:

$$H(t_1) = 0 \quad (2.15)$$

Since time does not appear explicitly in the expression (2.9), the Pontryagin function remains constant along the optimal trajectory. Therefore, using the terminal condition (2.15), it follows that:

$$H(t) = 0 \quad (2.16)$$

By evaluating the Hamiltonian (2.9) at time t_1 and imposing the terminal condition (2.15), and taking into account the relations (2.13), the following relationship is obtained:

$$|\lambda_\gamma(t_1)| = |\eta_3| = \sqrt{4V^3(\eta_1 \cos \gamma_1 + \eta_2 \sin \gamma_1)} \quad (2.17)$$

From Eq. (A.20), the optimal control is obtained as:

$$\frac{\partial H}{\partial u} = 0 \implies u = -\frac{\lambda_\gamma}{2V\lambda_z} = -\frac{\lambda_\gamma}{2V\lambda_0} \quad (2.18)$$

Assuming $\lambda_0 = 1$ [22], this reduces to:

$$u = -\frac{\lambda_\gamma}{2V} \quad (2.19)$$

This implies that determining the optimal control requires the evaluation of $\lambda_\gamma(t)$, which is obtained by solving a two-point boundary-value problem (TPBVP). The TPBVP consists of seven boundary conditions (six originating from the optimal control problem formulation (2.6) and one given by (2.17)) and four differential equations:

$$\begin{aligned}
\dot{x} &= V \cos \gamma \\
\dot{y} &= V \sin \gamma \\
\dot{\gamma} &= -\frac{\lambda_\gamma}{2V^2} \\
\dot{\lambda}_\gamma &= V(\eta_1 \sin \gamma - \eta_2 \cos \gamma) \\
x(0) &= x_0 \\
y(0) &= y_0 \\
\gamma(0) &= \gamma_0 \\
x(t_1) &= x_1 \\
y(t_1) &= y_1 \\
\gamma(t_1) &= \gamma_1 \\
|\lambda_\gamma(t_1)| &= \sqrt{4V^3(\eta_1 \cos \gamma_1 + \eta_2 \sin \gamma_1)}
\end{aligned} \tag{2.20}$$

where η_1 and η_2 are unknown constants.

Since, in missile guidance problems, the coordinate x is a monotonically increasing function, the variables y , γ , and λ_γ can be expressed as functions of downrange distance. Consequently, the last three differential equations of system (2.20) can be divided by the first equation, yielding the following equivalent form of the two-point boundary-value problem:

$$\begin{aligned}
\frac{dy}{dx} &= \tan \gamma \\
\frac{d\gamma}{dx} &= -\frac{\lambda_\gamma}{2V^3 \cos \gamma} \\
\frac{d\lambda_\gamma}{dx} &= \eta_1 \tan \gamma - \eta_2 \\
y(x_0) &= y_0 \\
\gamma(x_0) &= \gamma_0 \\
y(x_1) &= y_1 \\
\gamma(x_1) &= \gamma_1 \\
|\lambda_\gamma(x_1)| &= \sqrt{4V^3(\eta_1 \cos \gamma_1 + \eta_2 \sin \gamma_1)}
\end{aligned} \tag{2.21}$$

By combining the first and third equations of system (2.21) and integrating from the initial to the terminal point, the following relation is obtained:

$$\frac{d\lambda_\gamma}{dx} = \eta_1 \frac{dy}{dx} - \eta_2 \implies \int_{\lambda_\gamma(t_0)}^{\lambda_\gamma(t_1)} d\lambda_\gamma = \eta_1 \int_{y_0}^{y_1} dy - \eta_2 \int_{x_0}^{x_1} dx \tag{2.22}$$

or, equivalently:

$$\lambda_\gamma(t_0) = \lambda_\gamma(t_1) - \eta_1(y_1 - y_0) + \eta_2(x_1 - x_0) \tag{2.23}$$

Taking into account the relationship (2.13c), the previous expression becomes:

$$\lambda_\gamma(t_0) = \eta_3 - \eta_1(y_1 - y_0) + \eta_2(x_1 - x_0) \quad (2.24)$$

Using Eqs. (2.19) and (2.24), the optimal control at the initial instant can be determined as:

$$u(t_0) = [\eta_1(y_1 - y_0) - \eta_2(x_1 - x_0) - \eta_3] / (2V) \quad (2.25)$$

By omitting the subscript 0 from the preceding equation, an expression for the required control as a function of the current position is obtained:

$$u(x, y) = [\eta_1(y_1 - y) - \eta_2(x_1 - x) - \eta_3] / (2V) \quad (2.26)$$

where η_3 is given by (2.17). From Eq. (2.19), it is evident that the sign of λ_γ determines the sign of the control input. Therefore, based on the expected direction of the control action, or equivalently the anticipated convexity of the trajectory at time t_1 , the sign of $\lambda_\gamma(t_1)$, and consequently the sign of η_3 , can be determined.

Equation (2.26) can be used to compute the required control in a feedback form. However, the parameters η_1 , η_2 , and η_3 must first be determined by solving the two-point boundary-value problem (2.21). The resulting parameter values are valid only for the trajectory that is optimal with respect to the initial conditions under which they were obtained. In practical applications, if the realized trajectory begins to deviate from the nominal optimal trajectory—for example, due to imperfect command tracking—the parameters η_1 , η_2 , and η_3 must be recomputed in order to continue using Eq. (2.26). Since missile motion is inevitably affected by disturbances that may cause deviations from the desired trajectory, practical implementation requires these parameters to be recalculated at each guidance update. In other words, every guidance cycle is treated as a new initial-value problem, and the required optimal control is computed using Eq. (2.25).

The remainder of this chapter presents several approaches for solving the optimal control problem (2.6), or equivalently the two-point boundary-value problem (2.21) that follows from it, depending on the particular missile mission under consideration.

First, a solution algorithm based on the shooting method, the most commonly used numerical approach for solving two-point boundary-value problems, is presented. Since such numerical techniques are computationally demanding and one of the primary objectives of this dissertation is the real-time implementation of the resulting guidance laws, alternative solution procedures will subsequently be developed. These procedures determine the solution of the optimal control problem (2.6) without relying on the numerical solution of the underlying two-point boundary-value problem (2.21).

2.2.3 Shooting Method

Among the numerical methods commonly used for solving two-point boundary-value problems [22, 59, 91, 92], the shooting method is by far the most widely employed. The method is based on assuming initial values of the unknown parameters and solving the corresponding system of differential equations starting from the boundary conditions specified at one end of the trajectory (for example, the terminal point). By integrating the system, estimates of the state variables at the opposite end of the trajectory are obtained. In general, these estimates do not satisfy the prescribed boundary conditions exactly.

For the problem under consideration, defined by (2.21), an *error function* can be introduced to quantify the deviation between the values obtained through integration of the differential equations, $(\hat{y}_0, \hat{\gamma}_0)$, and the values prescribed by the boundary conditions, (y_0, γ_0) :

$$\mathcal{E}(\hat{y}_0, \hat{\gamma}_0) = \|\hat{y}_0 - y_0\|^2 + \|\hat{\gamma}_0 - \gamma_0\|^2 \quad (2.27)$$

The central idea of the shooting method is to iteratively update the unknown parameters (in the present case η_1 and η_2) until the value of the error function (2.27) is minimized, ideally approaching zero.

Because the error function may contain quantities of different physical nature and scale, such as position and angle in the present example, it can be advantageous to introduce appropriate scaling factors for the individual error components [93]. Such scaling is often employed to improve convergence toward the optimal solution.

In practice, achieving an exactly zero value of the error function is rarely possible. Therefore, the requirement is usually relaxed by introducing a prescribed tolerance ϵ , and the solution is accepted when the error function falls below this threshold.

The general algorithm for solving the two-point boundary-value problem (2.21) using the shooting method is summarized below. The unknown parameters η_1 and η_2 are considered determined once the error function becomes smaller than the prescribed tolerance ϵ , at which point the corresponding trajectory is regarded as optimal.

Algorithm 1 Solution of the two-point boundary-value problem (2.21) using the shooting method.

- 1: Initialize the variables $x_0, y_0, \gamma_0, V, x_1, y_1, \gamma_1$.
 - 2: Select the integration step size dx .
 - 3: Choose initial guesses for the unknown parameters η_1 and η_2 .
 - 4: **Repeat**
 - 5: Integrate system (2.21) from the terminal position x_1 to the initial position x_0 .
 - 6: Compute the error function (2.27).
 - 7: Update the parameter values η_1 and η_2 .
 - 8: **Until** $\mathcal{E}(\hat{y}_0, \hat{\gamma}_0) < \epsilon$.
-

Since the values $\hat{y}_0$ and $\hat{\gamma}_0$ obtained after integration depend on the selected values of η_1 and η_2 , the error function can be regarded as a function of these parameters. Their update may therefore be formulated as the solution of the equation:

$$\mathcal{E}(\hat{y}_0(\eta_1, \eta_2), \hat{\gamma}_0(\eta_1, \eta_2)) = \mathcal{E}(\eta_1, \eta_2) = \|\hat{y}_0(\eta_1, \eta_2) - y_0\|^2 + \|\hat{\gamma}_0(\eta_1, \eta_2) - \gamma_0\|^2 = 0. \quad (2.28)$$

Since analytical expressions for the derivatives of the error function are generally unavailable, gradient-based optimization methods would require derivative approximations, typically obtained using finite-difference techniques. An attractive alternative is the *Nelder–Mead simplex* algorithm [94], which does not require derivative information and can therefore be implemented more efficiently.

The principal limitation of numerical methods for solving two-point boundary-value problems, particularly shooting methods, is their computational burden. Furthermore, since the costate variables lack a direct physical interpretation, selecting suitable initial guesses for the unknown parameters often becomes a challenging task and may significantly affect convergence to the correct solution.

For these reasons, modified formulations of the original boundary-value problem are frequently sought in order to obtain guidance laws that can be implemented efficiently in real time. The remainder of this chapter presents an alternative solution procedure that avoids the use of the shooting method and does not require the numerical solution of the original two-point boundary-value problem. As will be shown, the resulting algorithm retains the optimality properties of the original formulation while being considerably more suitable for practical real-time implementation.

2.2.4 General Case of Prescribed Terminal Flight-Path Angle

If the unknown parameters η_1 and η_2 are represented as:

$$\begin{aligned} \eta_1 &= r \cos \alpha \\ \eta_2 &= r \sin \alpha \end{aligned} \quad (2.29)$$

where:

$$\begin{aligned} r &\geq 0 \\ -\pi &\leq \alpha \leq \pi \end{aligned} \quad (2.30)$$

the quantities r and α become the new unknown parameters to be determined. Using expressions (2.14) and (2.19), the Pontryagin function (2.9) can be written in the form:

$$H = rV(\cos \gamma \cos \alpha + \sin \gamma \sin \alpha) - \frac{\lambda_\gamma^2}{4V^2} \quad (2.31)$$

Taking into account condition (2.16), the following relationship is obtained:

$$\lambda_\gamma^2 = 4rV^3 \cos(\gamma - \alpha) \quad (2.32)$$

or equivalently:

$$\lambda_\gamma = \pm \sqrt{4rV^3} \sqrt{\cos(\gamma - \alpha)} \quad (2.33)$$

Since the left-hand side of (2.32) is nonnegative, it follows that:

$$-\frac{\pi}{2} \leq \gamma - \alpha \leq \frac{\pi}{2} \quad (2.34)$$

Using the flight-path-angle dynamics (2.3) together with (2.19) and (2.32), the following relation is obtained:

$$\dot{\gamma}^2 = \frac{r}{V} \cos(\gamma - \alpha) \quad (2.35)$$

which will be useful in the subsequent analysis of the phase trajectories.

Substituting (2.33) into the second equation of system (2.21) yields:

$$\gamma' = \frac{d\gamma}{dx} = \mp \sqrt{\frac{r}{V^3} \frac{\sqrt{\cos(\gamma - \alpha)}}{\cos \gamma}} \quad (2.36)$$

The sign on the right-hand side depends on the sign of γ' , that is, on the convexity of the trajectory.

Since, according to (2.34), the quantity $\gamma - \alpha$ can span an interval of at most π , it follows that γ' can have at most one zero crossing. Consequently, the trajectory can change its convexity at most once. Therefore, the following cases are possible:

- A) $\gamma' < 0$ along the entire trajectory;
- B) $\gamma' > 0$ over the first part of the trajectory and $\gamma' < 0$ over the second part;
- C) $\gamma' < 0$ over the first part of the trajectory and $\gamma' > 0$ over the second part;
- D) $\gamma' > 0$ along the entire trajectory.

Combining (2.36) with the first equation of system (2.21) gives the following system of two equations:

$$dx = \mp \sqrt{\frac{V^3}{r} \frac{\cos \gamma}{\sqrt{\cos(\gamma - \alpha)}}} d\gamma \quad (2.37a)$$

$$dy = \mp \sqrt{\frac{V^3}{r} \frac{\sin \gamma}{\sqrt{\cos(\gamma - \alpha)}}} d\gamma \quad (2.37b)$$

The above system must be solved in order to determine the two unknown quantities, r and α , after which the parameters η_1 and η_2 can be obtained directly from (2.29).

In the following sections, each of the previously identified cases will be analyzed separately, and procedures for solving system (2.37) will be developed.

The proposed procedures are used to determine the *nominal trajectory* at each guidance update during flight. The term *nominal trajectory* is used because, owing to imperfect command tracking and the presence of disturbances, the actual trajectory generally differs from the trajectory that was computed as optimal at the previous guidance cycle. Consequently, different trajectory cases may arise during a single flight. In particular, the case applicable at the beginning of the terminal guidance phase (most commonly Case A or Case B) does not necessarily remain valid throughout the entire engagement.

However, since the objective of this chapter is not the analysis of the guidance and control system from the perspective of command-tracking performance, but rather the development of computationally efficient methods for determining the required optimal control, the examples presented in the following sections assume ideal command tracking, i.e., exact realization of the computed optimal trajectory for the specific case under consideration.

A) $\gamma' < 0$ throughout the entire trajectory

This case corresponds to a continuously decreasing flight-path angle, i.e., a trajectory characterized by continuous descent. An example of such a scenario is shown in Fig. 2.2. For clarity of presentation, the following graphical conventions are used in all subsequent examples:

- The missile position and velocity vector at the initial instant are shown in blue;
- The target position and the desired terminal velocity-vector direction are shown in red;
- The missile–target line (LOS) at the initial instant is shown as a black dashed line;
- The optimal trajectory corresponding to the specified boundary conditions is shown in green.

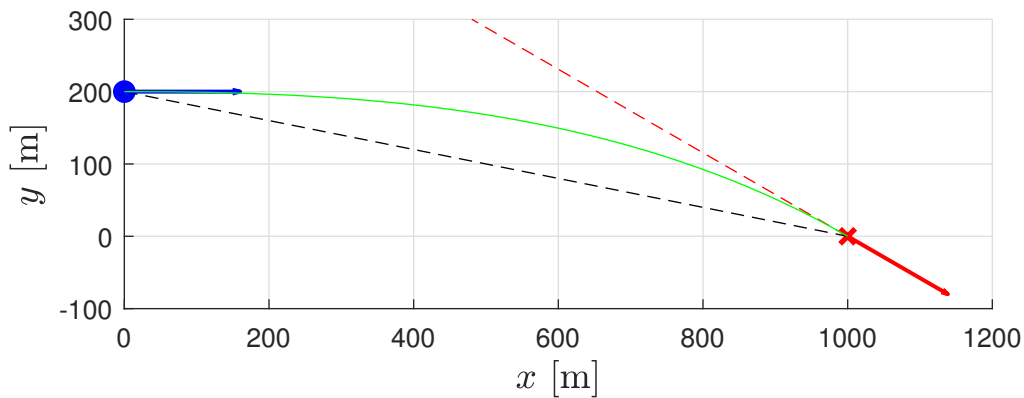


Figure 2.2: Optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -30^\circ$, and velocity $V = 160$ m/s.

The trajectory shown in Fig. 2.2 was obtained by numerical simulation of the planar kinematic model (1.1), with the control input u computed from (2.26) and (2.29). The unknown parameters r and α were determined using the procedure presented later in this subsection.

Figure 2.3 shows the control input (normal acceleration) and the flight-path angle as functions of downrange distance along the optimal trajectory.

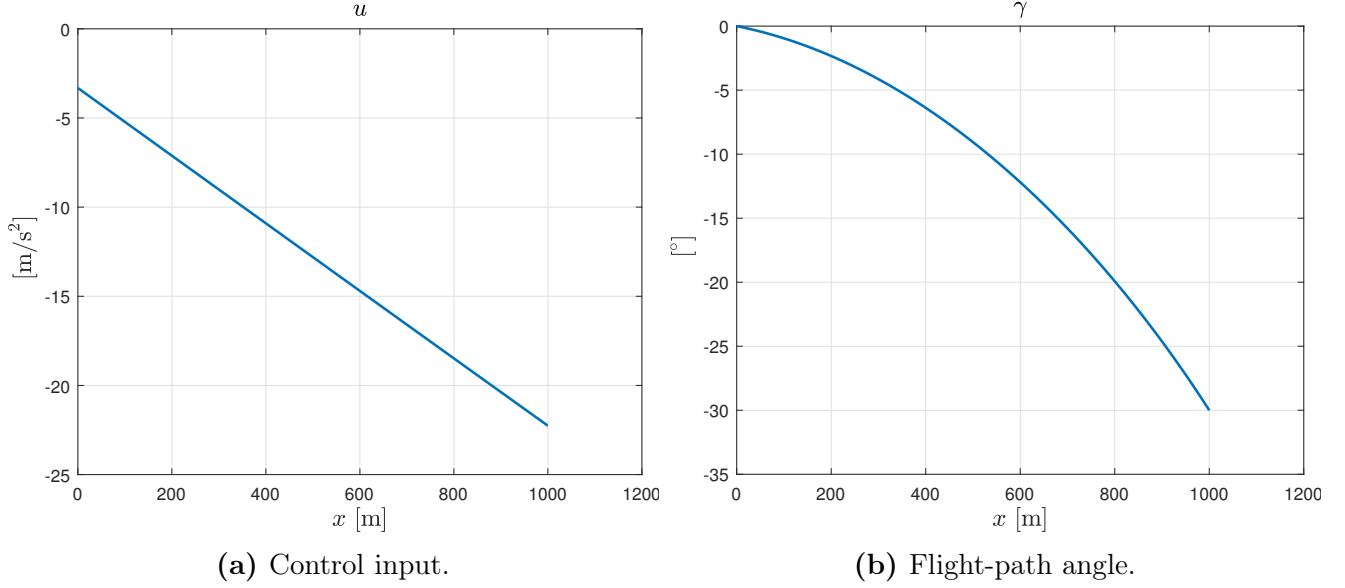


Figure 2.3: Flight parameters along the optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -30^\circ$, and velocity $V = 160$ m/s.

Starting from system (2.37), selecting the negative sign in both equations and integrating from the initial to the terminal point yields the following relations:

$$\Delta x = x_1 - x_0 = -\sqrt{\frac{V^3}{r}} \int_{\gamma_0}^{\gamma_1} \frac{\cos \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma \quad (2.38a)$$

$$\Delta y = y_1 - y_0 = -\sqrt{\frac{V^3}{r}} \int_{\gamma_0}^{\gamma_1} \frac{\sin \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma \quad (2.38b)$$

If the integrals appearing in the previous expressions are denoted by $I_N(\alpha)$ and $I_D(\alpha)$, respectively, and the substitution:

$$\beta = \gamma - \alpha, \quad d\beta = d\gamma \quad (2.39)$$

is introduced, the following expressions are obtained:

$$I_N(\alpha) = \int_{\gamma_0 - \alpha}^{\gamma_1 - \alpha} \frac{\cos \beta \cos \alpha - \sin \beta \sin \alpha}{\sqrt{\cos \beta}} d\beta \quad (2.40a)$$

$$I_D(\alpha) = \int_{\gamma_0 - \alpha}^{\gamma_1 - \alpha} \frac{\sin \beta \cos \alpha + \cos \beta \sin \alpha}{\sqrt{\cos \beta}} d\beta \quad (2.40b)$$

or equivalently:

$$I_N(\alpha) = \cos \alpha \int_{\gamma_0 - \alpha}^{\gamma_1 - \alpha} \sqrt{\cos \beta} d\beta - \sin \alpha \int_{\gamma_0 - \alpha}^{\gamma_1 - \alpha} \frac{\sin \beta}{\sqrt{\cos \beta}} d\beta \quad (2.41a)$$

$$I_D(\alpha) = \cos \alpha \int_{\gamma_0 - \alpha}^{\gamma_1 - \alpha} \frac{\sin \beta}{\sqrt{\cos \beta}} d\beta + \sin \alpha \int_{\gamma_0 - \alpha}^{\gamma_1 - \alpha} \sqrt{\cos \beta} d\beta \quad (2.41b)$$

Two types of integrals appear in the preceding expressions:

•

$$I_1(a, b) = \int_a^b \sqrt{\cos \xi} d\xi = 2 \left[E \left(\sqrt{2}, b/2 \right) - E \left(\sqrt{2}, a/2 \right) \right] \quad (2.42)$$

where:

$$E(k, \Phi) = \int_0^\Phi \sqrt{1 - k^2 \sin^2 \theta} d\theta \quad (2.43)$$

is the incomplete elliptic integral of the second kind;

•

$$I_2(a, b) = \int_a^b \frac{\sin \xi}{\sqrt{\cos \xi}} d\xi = 2 \left(\sqrt{\cos a} - \sqrt{\cos b} \right) \quad (2.44)$$

Therefore, the final form of (2.41) becomes:

$$I_N(\alpha) = \cos \alpha I_1(\gamma_0 - \alpha, \gamma_1 - \alpha) - \sin \alpha I_2(\gamma_0 - \alpha, \gamma_1 - \alpha) \quad (2.45a)$$

$$I_D(\alpha) = \cos \alpha I_2(\gamma_0 - \alpha, \gamma_1 - \alpha) + \sin \alpha I_1(\gamma_0 - \alpha, \gamma_1 - \alpha) \quad (2.45b)$$

Dividing the first equation of system (2.38) by the second yields a single equation with a single unknown:

$$\mathcal{F}(\alpha) = 0 \quad (2.46)$$

where:

$$\mathcal{F}(\alpha) = \frac{\Delta x}{\Delta y} - \frac{I_N(\alpha)}{I_D(\alpha)} \quad (2.47)$$

Once Eq. (2.46) is solved for α , the parameter r can be obtained from either the first or the second equation of system (2.38):

$$r = V^3 \left(\frac{I_N(\alpha)}{\Delta x} \right)^2 \quad (2.48)$$

Equation (2.46) can be solved numerically. However, before attempting its solution, it is necessary to verify whether singularities occur within the considered interval, i.e., whether the denominator integral becomes zero for some value of α .

From inequality (2.34), the solution for α must satisfy:

$$\gamma - \frac{\pi}{2} \leq \alpha \leq \gamma + \frac{\pi}{2} \quad (2.49)$$

where the lower bound is obtained using the maximum value of the flight-path angle, while the upper bound is obtained using its minimum value:

$$\max(\gamma_0, \gamma_1) - \frac{\pi}{2} \leq \alpha \leq \min(\gamma_0, \gamma_1) + \frac{\pi}{2} \quad (2.50)$$

Figure 2.4a shows the dependence of the integral I_D , given by (2.45b), on α over the interval of interest for the boundary conditions $\gamma_0 = 0^\circ$ and $\gamma_1 = -30^\circ$. Since I_D does not vanish within the interval, the root of $\mathcal{F}(\alpha)$ can be searched over the entire interval. The corresponding function is shown in Fig. 2.4b, where its intersection with the horizontal axis is highlighted in red.

Figure 2.5 presents the same functions for identical initial and terminal positions but for the boundary angles $\gamma_0 = 35^\circ$ and $\gamma_1 = -60^\circ$. In this case, the integral I_D possesses a zero within the considered interval, indicated in Fig. 2.5a. Consequently, the root of $\mathcal{F}(\alpha)$ must be sought within two subintervals separated by the singular point. The corresponding optimal trajectory is shown in Fig. 2.6.

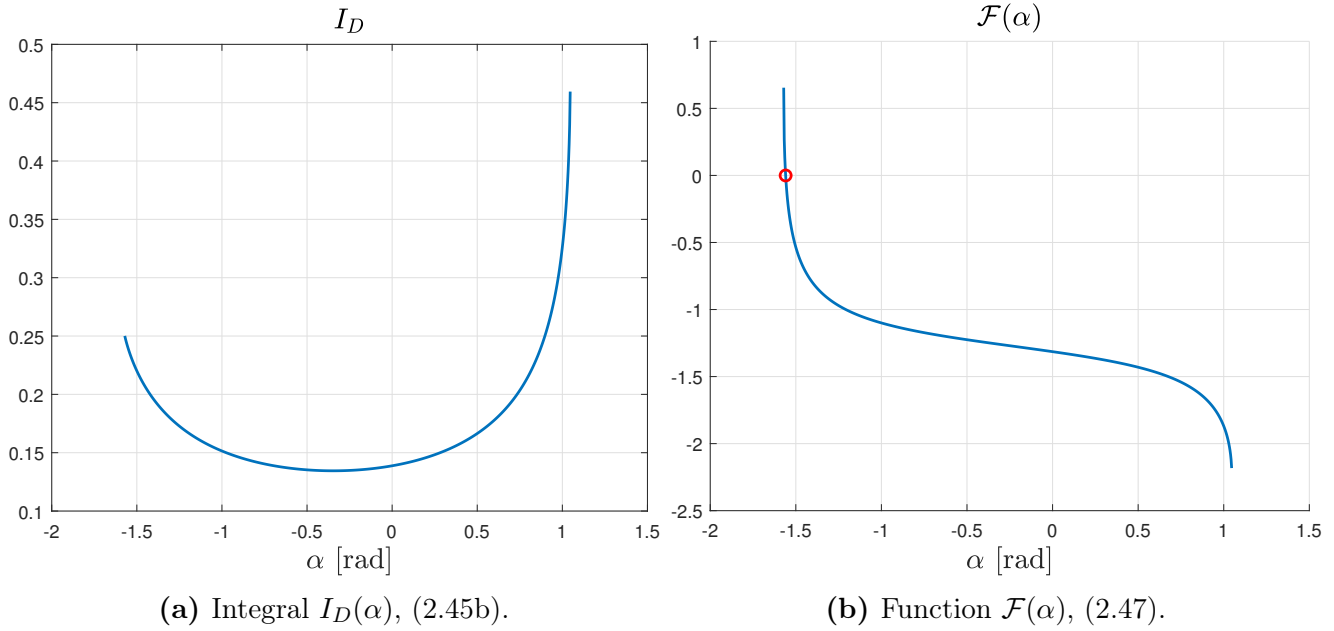


Figure 2.4: Singularity analysis and solution of Eq. (2.46) for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -30^\circ$, and velocity $V = 160$ m/s.

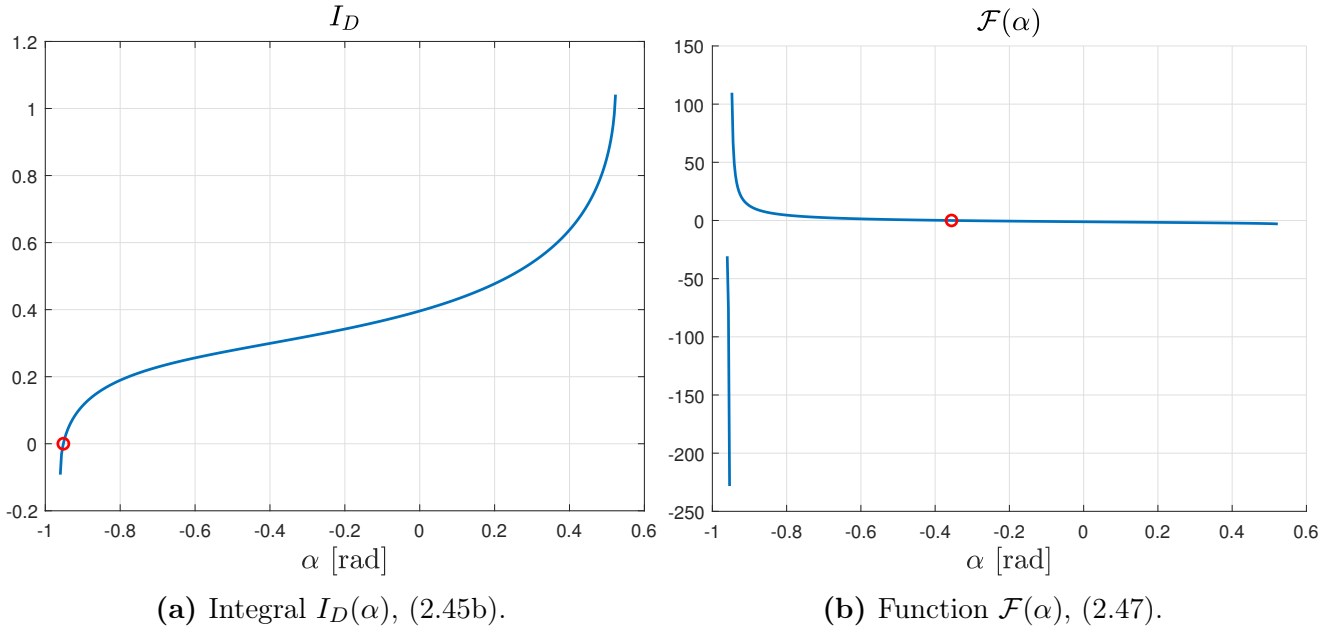


Figure 2.5: Singularity analysis and solution of Eq. (2.46) for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 35^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.

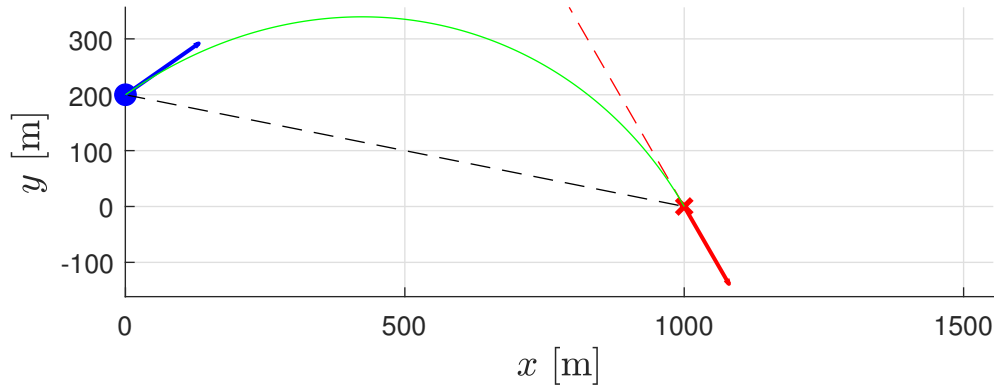


Figure 2.6: Optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 35^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.

To eliminate the singularity issue, the function $\mathcal{F}(\alpha)$ may be rewritten as:

$$\mathcal{F}(\alpha) = \frac{\Delta x I_D(\alpha) - \Delta y I_N(\alpha)}{\Delta y I_D(\alpha)} \quad (2.51)$$

and the solution for α obtained from:

$$\mathcal{N}(\alpha) = 0 \quad (2.52)$$

where:

$$\mathcal{N}(\alpha) = \Delta x I_D(\alpha) - \Delta y I_N(\alpha) \quad (2.53)$$

Figure 2.7 shows the functions $\mathcal{N}(\alpha)$ corresponding to the examples presented in Figs. 2.4 and 2.5. Since this function contains no singularities over the interval of interest, its root may be sought directly over the entire interval, without any need to examine the behavior of $I_D(\alpha)$.

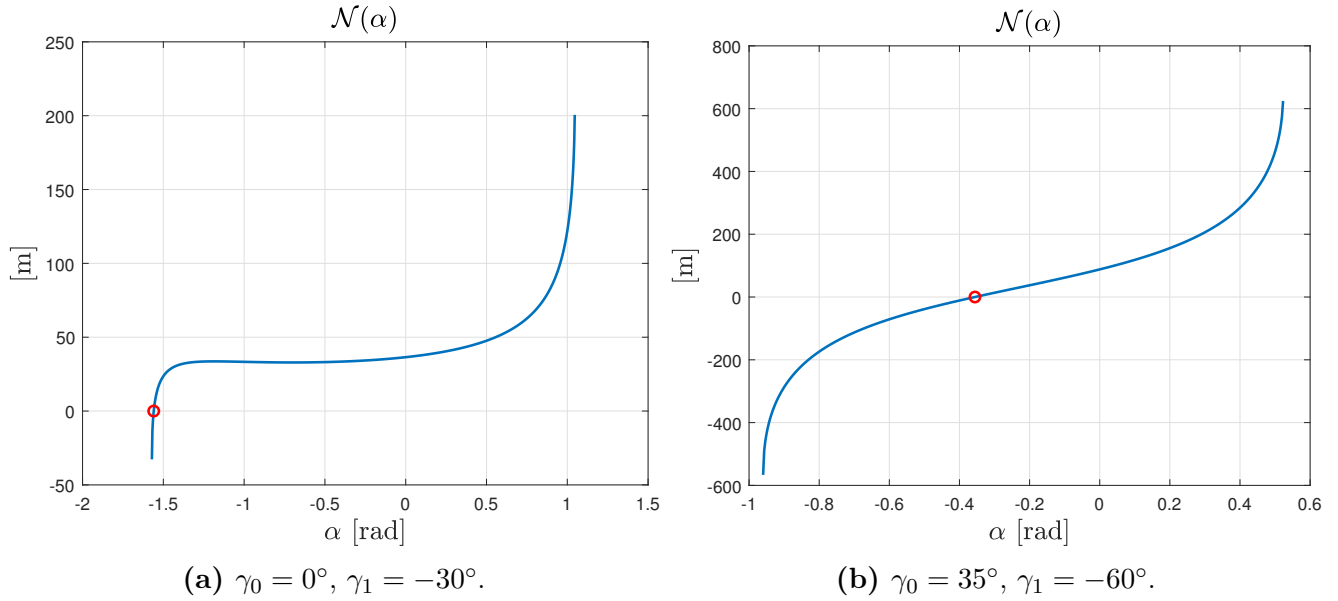


Figure 2.7: Solutions of Eq. (2.52) for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $x_1 = 1000$ m, $y_1 = 0$ m, and velocity $V = 160$ m/s.

Finally, for the considered case of continuous descent, the corresponding phase trajectory, obtained from (2.35), is shown in Fig. 2.8.

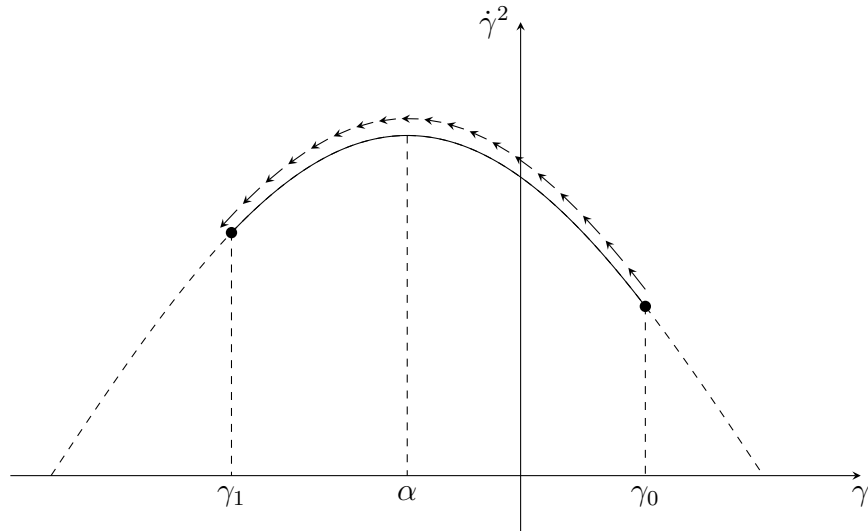


Figure 2.8: Phase trajectory for Case A: $\gamma' < 0$ throughout the entire trajectory.

B) $\gamma' > 0$ along the first segment of the trajectory and $\gamma' < 0$ along the second segment

This case corresponds to an initial increase in the flight-path angle up to the inflection point, after which the trajectory follows the behavior described in Case A. Optimal trajectories for two representative examples of this case are shown in Figs. 2.9 and 2.11. The first example corresponds to a transition from the cruise phase, where an initial climb is required to achieve a steeper terminal impact angle, while the second example corresponds to trajectory correction during the terminal phase of guidance.

The corresponding control input (normal acceleration) and flight-path angle as functions of down-range distance are shown in Figs. 2.10 and 2.12.

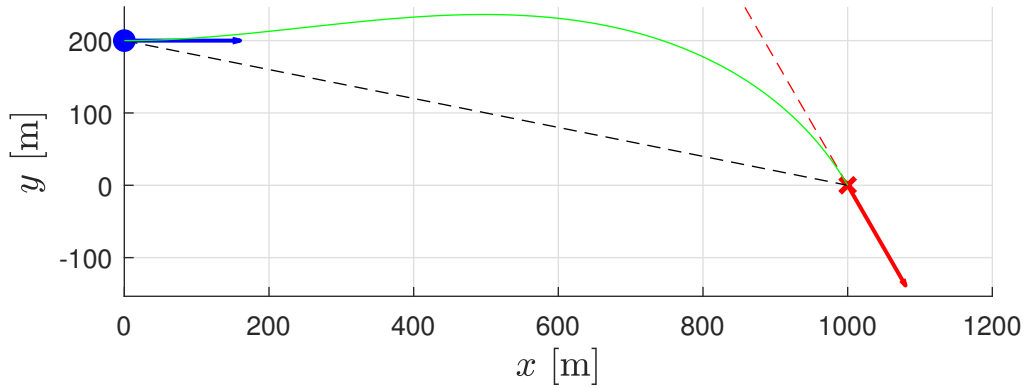


Figure 2.9: Optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.

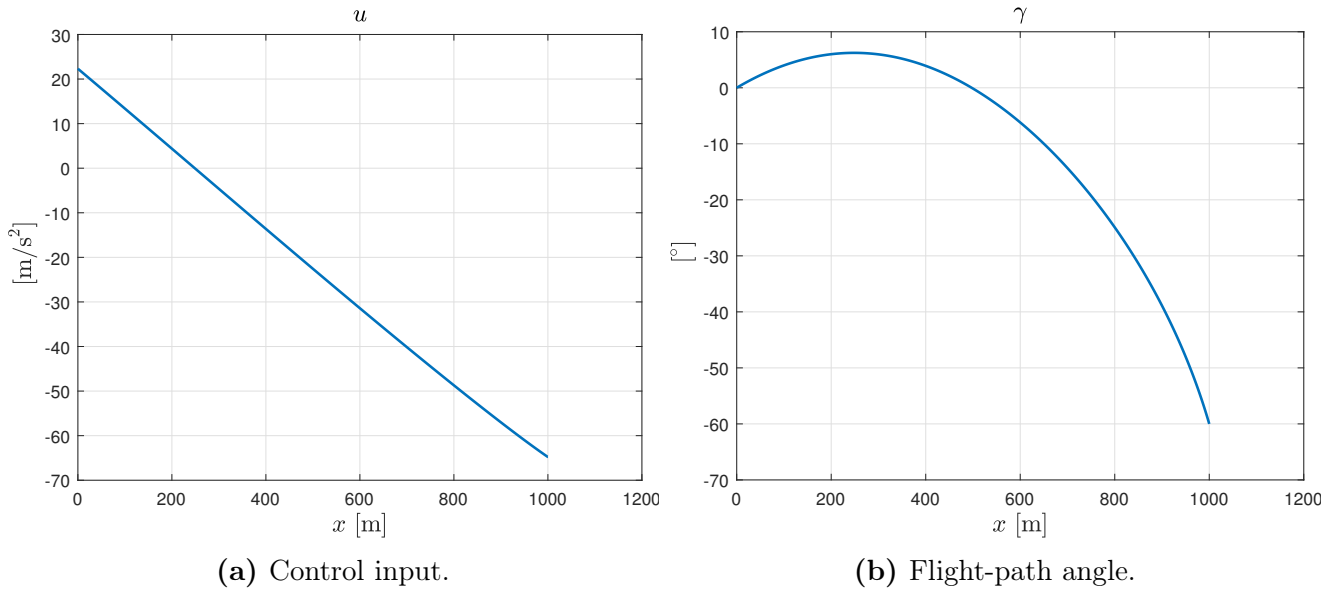


Figure 2.10: Flight parameters along the optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.

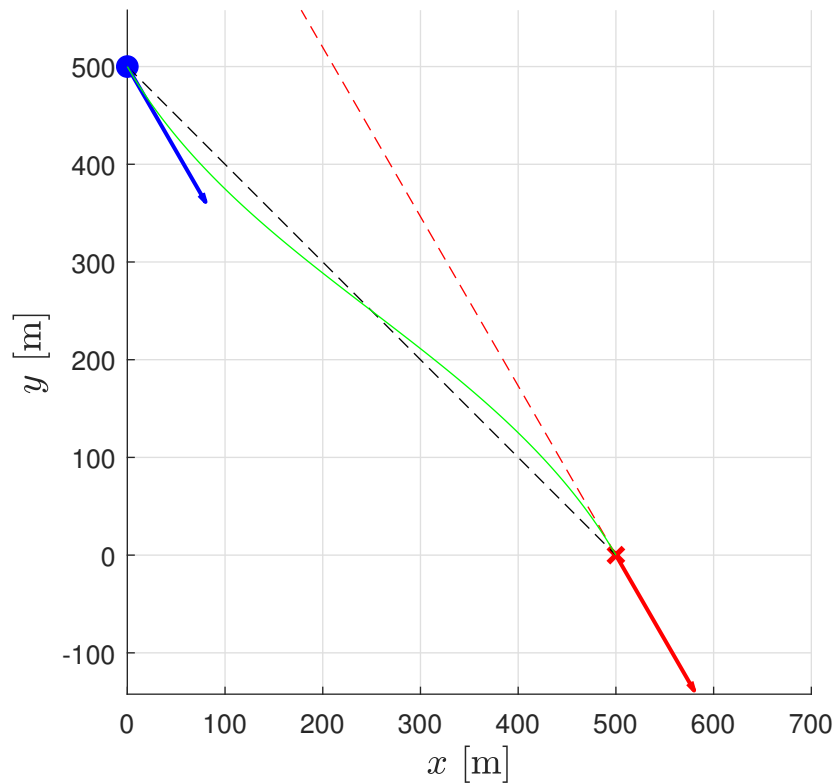


Figure 2.11: Optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 500$ m, $\gamma_0 = -60^\circ$, $x_1 = 500$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.

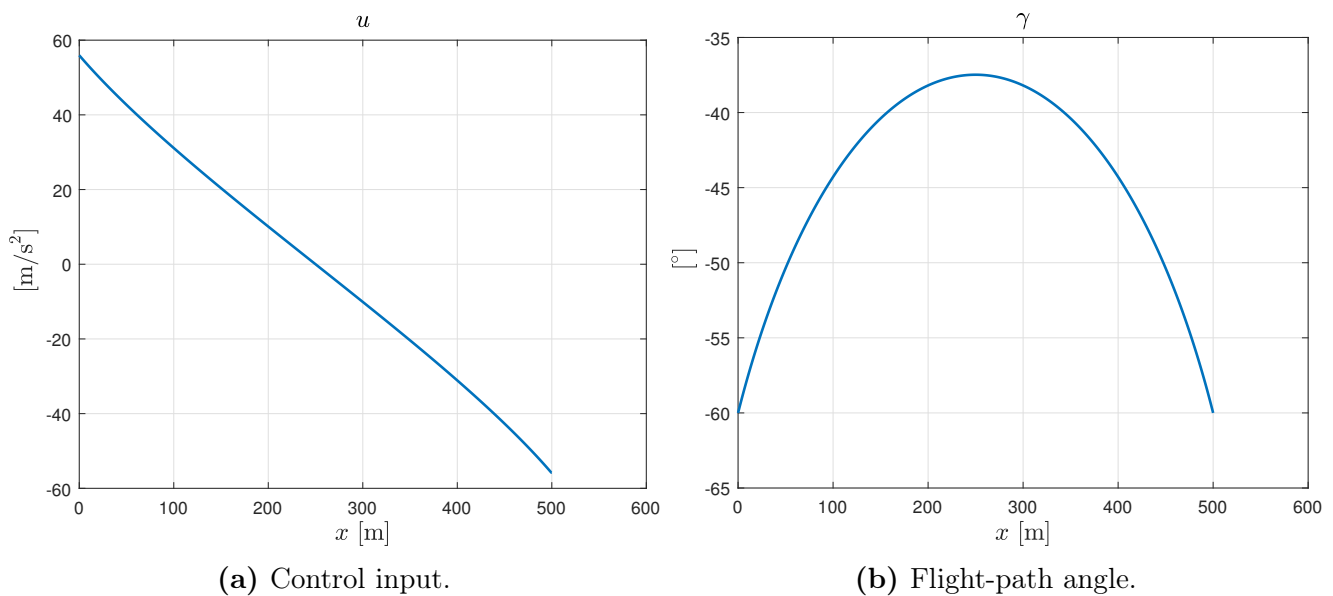


Figure 2.12: Flight parameters along the optimal trajectory for the boundary conditions: $x_0 = 0$ m, $y_0 = 500$ m, $\gamma_0 = -60^\circ$, $x_1 = 500$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.

The phase trajectory corresponding to this case, obtained from (2.35) and taking into account the existence of an inflection point at which $\dot{\gamma} = 0$, is shown in Fig. 2.13.

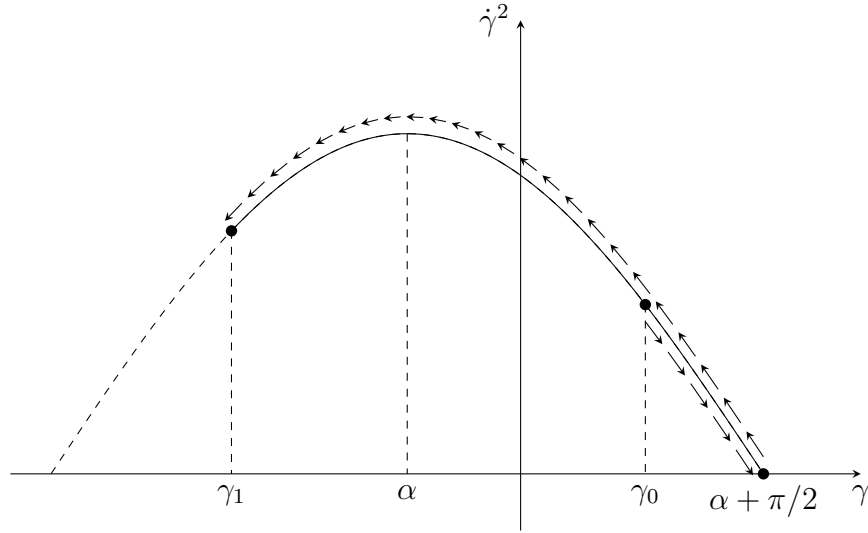


Figure 2.13: Phase trajectory for Case B: $\gamma' > 0$ along the first segment of the trajectory and $\gamma' < 0$ along the second segment.

Because γ' changes sign along the trajectory, the point at which $\gamma = \alpha + \pi/2$ corresponds to a change in trajectory convexity. Consequently, the integral form of system (2.37) becomes:

$$\Delta x = \sqrt{\frac{V^3}{r}} \left[\int_{\gamma_0}^{\alpha+\pi/2} \frac{\cos \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma - \int_{\alpha+\pi/2}^{\gamma_1} \frac{\cos \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma \right] \quad (2.54a)$$

$$\Delta y = \sqrt{\frac{V^3}{r}} \left[\int_{\gamma_0}^{\alpha+\pi/2} \frac{\sin \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma - \int_{\alpha+\pi/2}^{\gamma_1} \frac{\sin \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma \right] \quad (2.54b)$$

Following the same procedure as in Case A and introducing the substitution (2.39), the same functional form of Eq. (2.47) is obtained, where the integrals I_N and I_D are now given by:

$$I_N(\alpha) = \cos \alpha I_1(\gamma_0 - \alpha, \pi/2) - \sin \alpha I_2(\gamma_0 - \alpha, \pi/2) \\ + \cos \alpha I_1(\gamma_1 - \alpha, \pi/2) - \sin \alpha I_2(\gamma_1 - \alpha, \pi/2) \quad (2.55a)$$

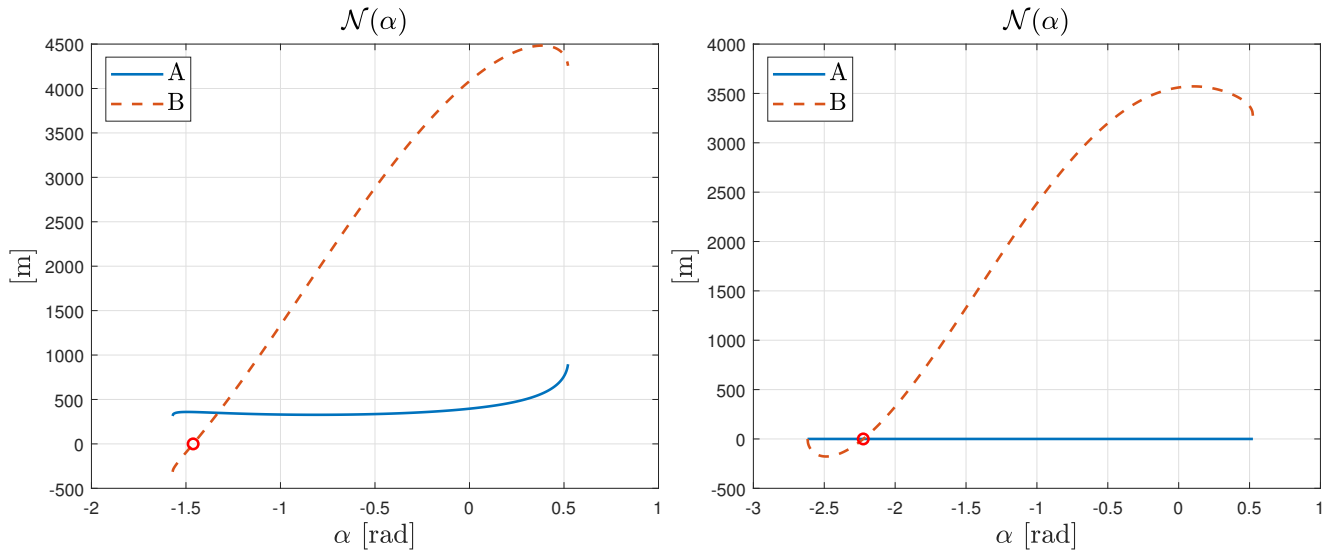
$$I_D(\alpha) = \cos \alpha I_2(\gamma_0 - \alpha, \pi/2) + \sin \alpha I_1(\gamma_0 - \alpha, \pi/2) \\ + \cos \alpha I_2(\gamma_1 - \alpha, \pi/2) + \sin \alpha I_1(\gamma_1 - \alpha, \pi/2) \quad (2.55b)$$

Therefore, the unknown parameter α can again be determined by solving Eq. (2.52), using the expressions given by (2.55).

Figure 2.14 shows the functions $\mathcal{N}(\alpha)$ corresponding to the examples presented in Figs. 2.9 and 2.11. The curves shown in red were obtained using expressions (2.55), corresponding to Case B, whereas the curves shown in blue were obtained using expressions (2.45), corresponding to Case A.

The first example is particularly interesting because, at the instant when the optimal trajectory is being determined, it is not possible to know in advance whether Case A or Case B should be applied. Figure 2.7a shows that when the expressions derived for Case A are used, the function $\mathcal{N}(\alpha)$ has no root within the considered interval. The existence of a root may therefore be used as a criterion for selecting the appropriate algorithm. In practical implementation, Case A should always be assumed first; if no root exists within the admissible interval, the procedure should then switch to Case B.

In the second example, the flight-path angle has the same value at the initial and terminal points. It is evident that, in such situations, achieving the desired terminal impact angle is possible only if the trajectory changes convexity², as illustrated in Fig. 2.11. Consequently, Case A can be immediately excluded, since the function $\mathcal{F}(\alpha)$ is not defined when the initial and terminal flight-path angles are identical.



(a) $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $\gamma_1 = -60^\circ$. (b) $y_0 = 500$ m, $\gamma_0 = -60^\circ$, $x_1 = 500$ m, $\gamma_1 = -60^\circ$.

Figure 2.14: Solutions of Eq. (2.52) for different examples of Case B. $x_0 = 0$ m, $y_1 = 0$ m, $V = 160$ m/s.

C) $\gamma' < 0$ on the first segment of the trajectory, $\gamma' > 0$ on the second segment of the trajectory

As in Case B, this scenario also involves a change in trajectory convexity, i.e., a change in the sign of the derivative γ' , but now occurring at the point $\gamma = \alpha - \pi/2$. The phase trajectory, reflecting a decrease in the flight-path angle up to the inflection point and its subsequent increase until the desired impact angle is reached, is shown in Fig. 2.15.

²Except for the special case in which the initial velocity vector lies on the line of sight.

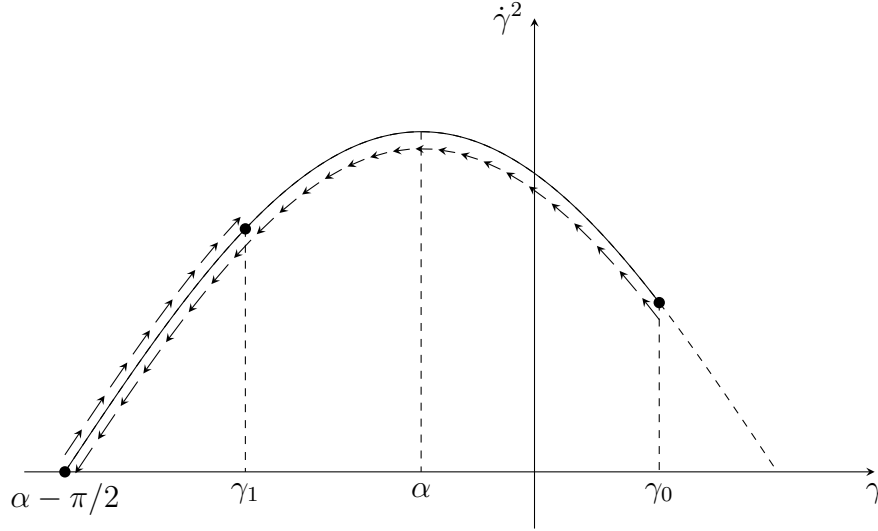


Figure 2.15: Phase trajectory for Case C: $\gamma' < 0$ on the first segment of the trajectory, $\gamma' > 0$ on the second segment of the trajectory.

The system of equations (2.37) in integral form for this case takes the following form:

$$\Delta x = \sqrt{\frac{V^3}{r}} \left[- \int_{\gamma_0}^{\alpha - \pi/2} \frac{\cos \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma + \int_{\alpha - \pi/2}^{\gamma_1} \frac{\cos \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma \right] \quad (2.56a)$$

$$\Delta y = \sqrt{\frac{V^3}{r}} \left[- \int_{\gamma_0}^{\alpha - \pi/2} \frac{\sin \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma + \int_{\alpha - \pi/2}^{\gamma_1} \frac{\sin \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma \right] \quad (2.56b)$$

Following the same procedure as in Cases A and B, and introducing the substitution (2.39), the same structure as Eq. (2.47) is obtained, where the integrals I_N and I_D take the following form:

$$I_N(\alpha) = \cos \alpha I_1(\gamma_0 - \alpha, -\pi/2) - \sin \alpha I_2(\gamma_0 - \alpha, -\pi/2) \\ + \cos \alpha I_1(\gamma_1 - \alpha, -\pi/2) - \sin \alpha I_2(\gamma_1 - \alpha, -\pi/2) \quad (2.57a)$$

$$I_D(\alpha) = \cos \alpha I_2(\gamma_0 - \alpha, -\pi/2) + \sin \alpha I_1(\gamma_0 - \alpha, -\pi/2) \\ + \cos \alpha I_2(\gamma_1 - \alpha, -\pi/2) + \sin \alpha I_1(\gamma_1 - \alpha, -\pi/2) \quad (2.57b)$$

The trajectories for two representative examples of this case are shown in Figs. 2.16 and 2.18. The first example corresponds to a transition from cruise flight with an initial descent to achieve small impact angles, while the second example corresponds to trajectory correction in the terminal guidance phase toward the target.

The control variable (normal acceleration) and the flight-path angle as functions of range for these two examples are shown in Figs. 2.17 and 2.19.

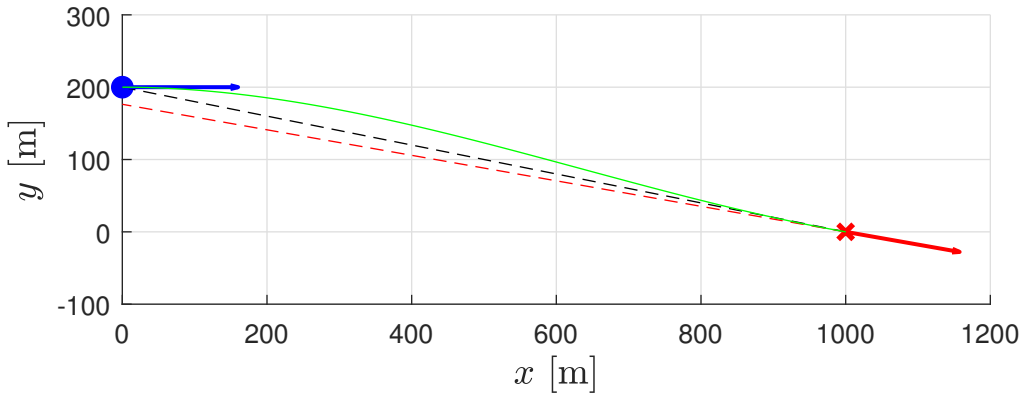


Figure 2.16: Optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -10^\circ$, and velocity $V = 160$ m/s.

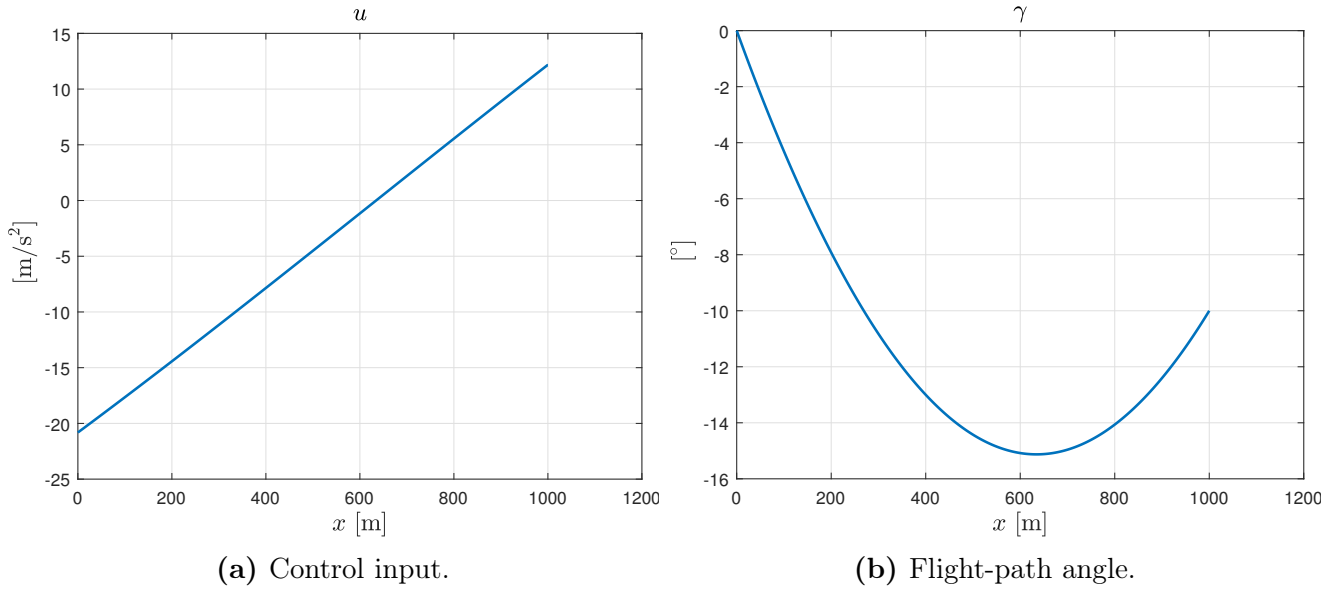


Figure 2.17: Flight parameters along the optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $y_1 = 0$ m, $\gamma_1 = -10^\circ$, and velocity $V = 160$ m/s.

The functions $\mathcal{N}(\alpha)$ for the examples presented in Figs. 2.16 and 2.18 are plotted in Fig. 2.20. The yellow curves correspond to expressions (2.57), which define Case C, while the red and blue curves correspond to expressions (2.55) and (2.45), associated with Cases B and A, respectively.

As already noted in the discussion of Case B, the absence of a root of $\mathcal{N}(\alpha)$ on the considered interval, or the undefined nature of $\mathcal{F}(\alpha)$, indicates whether the selected formulation is appropriate for a given scenario or whether the formulation from the subsequent case should be applied. From Fig. 2.20, it is evident that in the considered examples, a solution can only be obtained using the procedure corresponding to Case C.

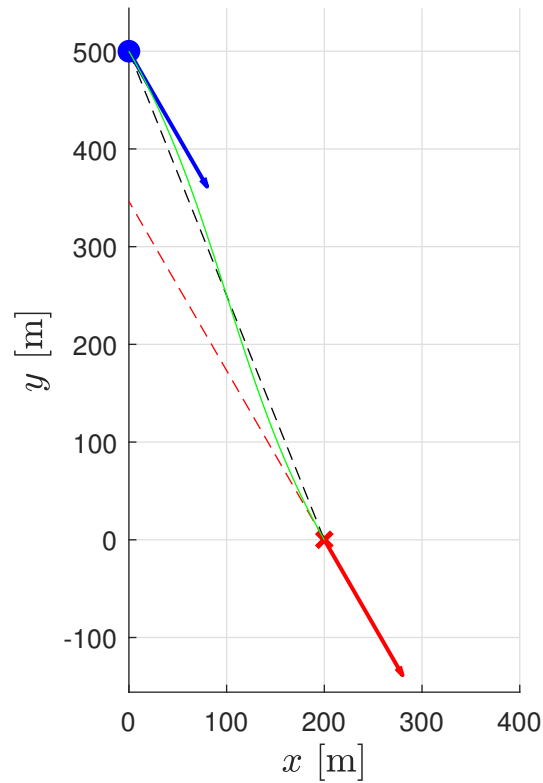


Figure 2.18: Optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 500$ m, $\gamma_0 = -60^\circ$, $x_1 = 200$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.

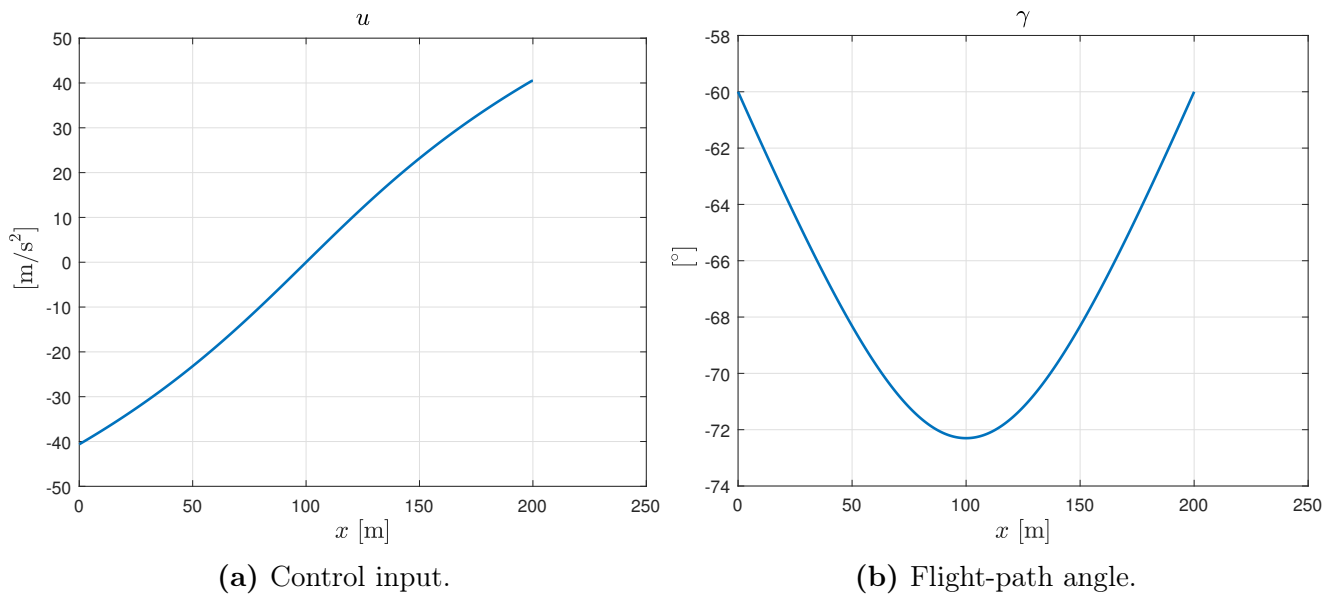
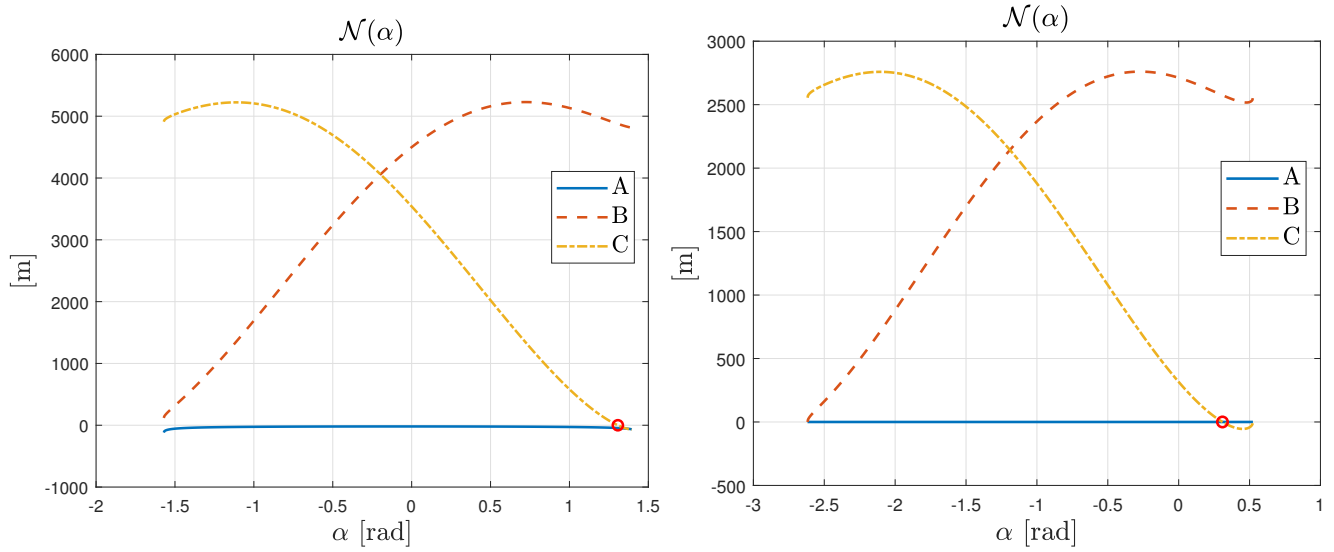


Figure 2.19: Flight parameters along the optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 500$ m, $\gamma_0 = -60^\circ$, $x_1 = 200$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and velocity $V = 160$ m/s.



(a) $y_0 = 200$ m, $\gamma_0 = 0^\circ$, $x_1 = 1000$ m, $\gamma_1 = -10^\circ$. (b) $y_0 = 500$ m, $\gamma_0 = -60^\circ$, $x_1 = 200$ m, $\gamma_1 = -60^\circ$.

Figure 2.20: Solutions of Eq. (2.52) for different examples of Case C. $x_0 = 0$ m, $y_1 = 0$ m, $V = 160$ m/s.

D) $\gamma' > 0$ on the entire trajectory

This case corresponds to a continuously increasing flight-path angle, i.e., a trajectory with no change in convexity. The phase trajectory is shown in Fig. 2.21.

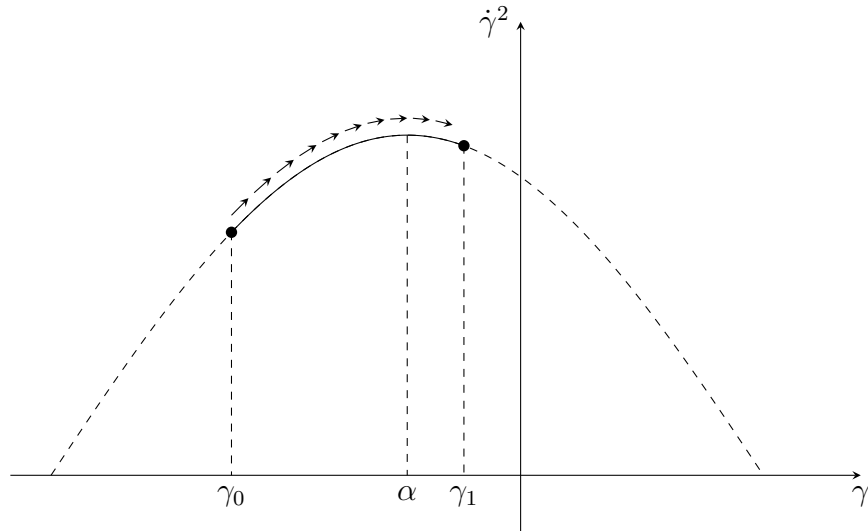


Figure 2.21: Phase trajectory for Case D: $\gamma' > 0$ on the entire trajectory.

The system of equations (2.37), after integration from the initial to the final time, takes the form:

$$\Delta x = \sqrt{\frac{V^3}{r}} \int_{\gamma_0}^{\gamma_1} \frac{\cos \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma \quad (2.58a)$$

$$\Delta y = \sqrt{\frac{V^3}{r}} \int_{\gamma_0}^{\gamma_1} \frac{\sin \gamma}{\sqrt{\cos(\gamma - \alpha)}} d\gamma \quad (2.58b)$$

Following the substitution (2.39) and division of the two equations in system (2.58), the same form of function $\mathcal{F}(\alpha)$ as in Case A is obtained. Moreover, due to the absence of an inflection point, the integrals I_N and I_D also retain the same form as in Case A, and thus the procedure for determining the optimal trajectory in this case is identical to that of Case A.

Figure 2.22 shows a representative trajectory for this case. Since this work focuses on optimal trajectories of missiles engaging ground targets, the most common example of this case corresponds to trajectory correction in the terminal guidance phase.

The control variable (normal acceleration) and the flight-path angle as functions of range are shown in Fig. 2.23, while the function $\mathcal{N}(\alpha)$ is shown in Fig. 2.24.

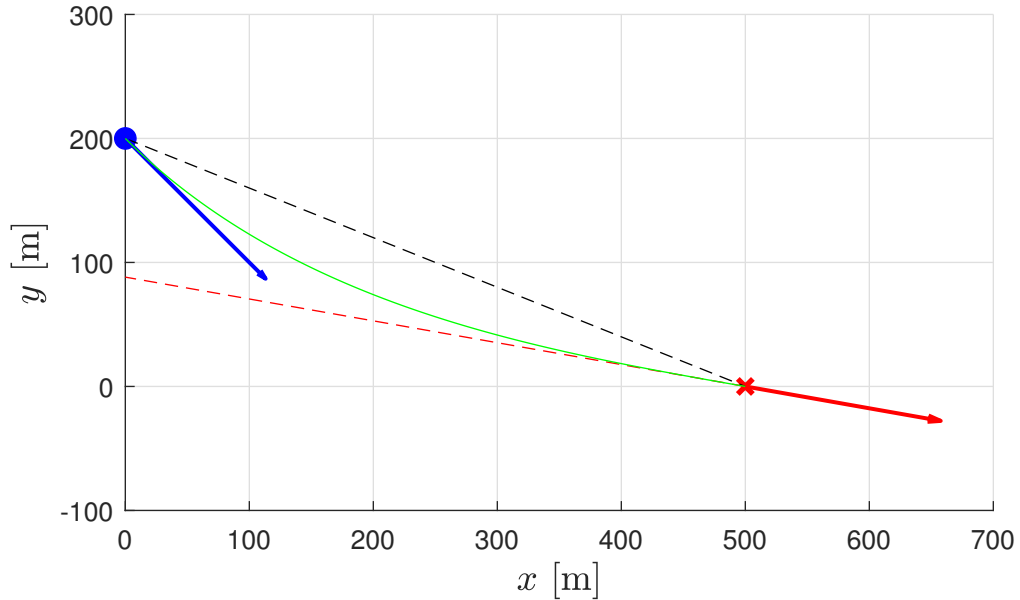


Figure 2.22: Optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = -45^\circ$, $x_1 = 500$ m, $y_1 = 0$ m, $\gamma_1 = -10^\circ$, and velocity $V = 160$ m/s.

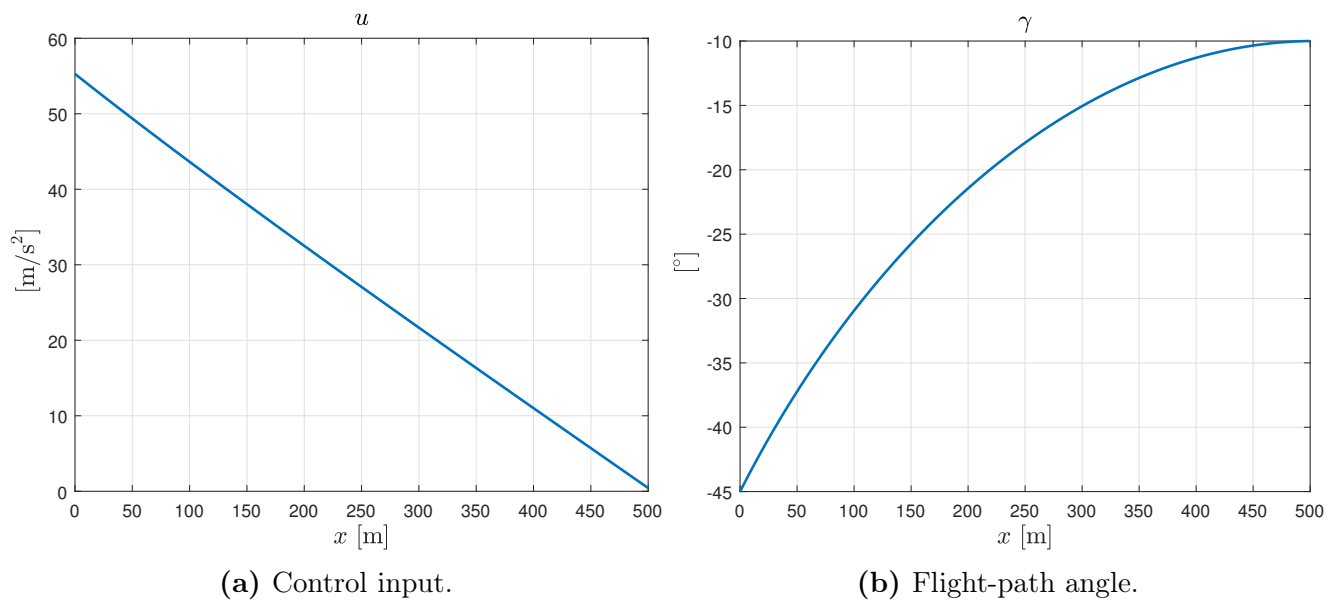


Figure 2.23: Flight parameters along the optimal trajectory for boundary conditions: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = -45^\circ$, $x_1 = 500$ m, $y_1 = 0$ m, $\gamma_1 = -10^\circ$, and velocity $V = 160$ m/s.

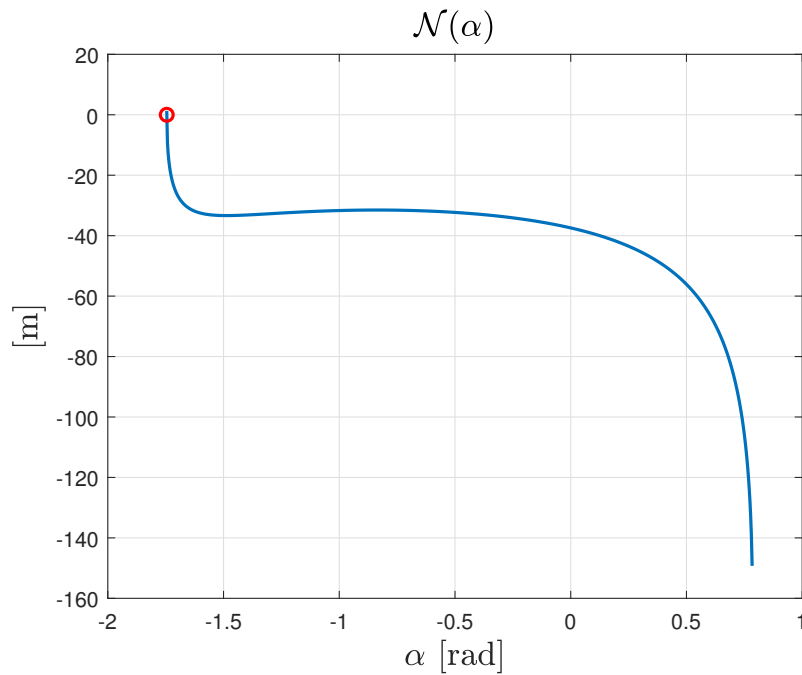


Figure 2.24: Solution of Eq. (2.52) for the example analyzed in Case D.

Algorithm

The procedures presented in the previous part of the paper for different trajectory convexity cases can be summarized as follows:

- At the very beginning, it is necessary to examine the existence of a root of the function $\mathcal{N}(\alpha)$ for Case A. If an intersection of the curve $\mathcal{N}(\alpha)$ with the abscissa exists on the interval (2.50), this case is considered *valid*. Otherwise, the existence of a root of this function must be examined for Case B, and if no root exists on the considered interval, Case C is adopted as *valid*.
- The existence of a root can be checked by evaluating the function $\mathcal{N}(\alpha)$ at the endpoints of the interval. If the function has the same sign at the beginning and at the end of the interval, the corresponding case should be rejected and the procedure continued with the next case. If the function has opposite signs at the interval endpoints, the considered case is accepted as valid and the root of the function is then computed.

Since the function $\mathcal{N}(\alpha)$ is continuous, as it is composed of continuous functions, the proposed root-existence test is valid under the assumption that at most one root exists on the considered interval. In this work, it is not analytically proven that only one root exists; however, numerical examples show that when a root exists, it is unique.

- What distinguishes Cases A, B, and C³ is the form of the integrals $I_N(\alpha)$ and $I_D(\alpha)$ given by Eqs. (2.45), (2.55), and (2.57), which can be written in the general form:

$$I_N(\alpha) = \cos \alpha \mathcal{I}_1(\alpha) - \sin \alpha \mathcal{I}_2(\alpha) \quad (2.59a)$$

$$I_D(\alpha) = \cos \alpha \mathcal{I}_2(\alpha) + \sin \alpha \mathcal{I}_1(\alpha) \quad (2.59b)$$

where:

$$\mathcal{I}_1(\alpha) = \begin{cases} I_1(\gamma_0 - \alpha, \gamma_1 - \alpha), & \text{for case A} \\ I_1(\gamma_0 - \alpha, \pi/2) + I_1(\gamma_1 - \alpha, \pi/2), & \text{for case B} \\ I_1(\gamma_0 - \alpha, -\pi/2) + I_1(\gamma_1 - \alpha, -\pi/2), & \text{for case C} \end{cases} \quad (2.60a)$$

$$\mathcal{I}_2(\alpha) = \begin{cases} I_2(\gamma_0 - \alpha, \gamma_1 - \alpha), & \text{for case A} \\ I_2(\gamma_0 - \alpha, \pi/2) + I_2(\gamma_1 - \alpha, \pi/2), & \text{for case B} \\ I_2(\gamma_0 - \alpha, -\pi/2) + I_2(\gamma_1 - \alpha, -\pi/2), & \text{for case C} \end{cases} \quad (2.60b)$$

When testing for the existence of a root or solving for it (once a case is considered *valid*), the expressions corresponding to that specific case must be used.

³Case D is identical to Case A and is therefore not treated separately.

- What is common to all three cases is that solving the Eq. (2.53):

$$\mathcal{N}(\alpha) = \Delta x I_D(\alpha) - \Delta y I_N(\alpha) = 0$$

yields the unknown parameter α . Based on this solution, the second unknown parameter is obtained using Eq. (2.48):

$$r = V^3 \left(\frac{I_N(\alpha)}{\Delta x} \right)^2$$

- After determining r and α , the unknown parameters η_1 and η_2 can be computed using (2.29):

$$\eta_1 = r \cos \alpha$$

$$\eta_2 = r \sin \alpha$$

- Finally, the required control at the initial time is determined using (2.25):

$$u(x_0, y_0) = [\eta_1(y_1 - y_0) - \eta_2(x_1 - x_0) - \eta_3] / (2V)$$

where, according to (2.17):

$$\eta_3 = \pm \sqrt{4V^3(\eta_1 \cos \gamma_1 + \eta_2 \sin \gamma_1)}$$

The sign of η_3 depends on the trajectory convexity at the final time, as already discussed in relation to Eq. (2.19). This implies that in Case B η_3 is positive, in Case C it is negative, while in Case A an additional analysis is required to determine the trajectory convexity.

- For each subsequent time instant at which the control input is computed, it is necessary to re-evaluate the valid case based on the updated initial conditions, determine new values of the unknown parameters r , α (i.e., η_1 , η_2), and then compute η_3 and u accordingly.

Based on the above conclusions, it is clear that, for given initial and terminal conditions, the valid case is the one for which the function $\mathcal{N}(\alpha)$ possesses a zero within the considered interval. Consequently, this verification must be performed sequentially, starting with the expressions corresponding to Case A. This raises the question of whether there exists a combination of initial and terminal conditions for which the function $\mathcal{N}(\alpha)$ has no zero in any of the considered cases.

An example of such a scenario is shown in Fig. 2.25, while the corresponding functions $\mathcal{N}(\alpha)$ are presented in Fig. 2.26. It is evident that, according to the proposed procedure, none of the cases is valid. This example clearly illustrates that the proposed method cannot be applied universally for determining optimal trajectories. One possible reason is the adopted assumption of strict monotonicity of the x -coordinate, which may be violated along the optimal trajectory in this example. However, although such a scenario is theoretically admissible within the considered kinematic model, it is not physically realistic. At short distances from the target, the flight-path angle cannot attain large positive values, which are expected only during the initial phase of terminal guidance when large impact angles must be achieved, and at that stage the target is still at a considerable distance.

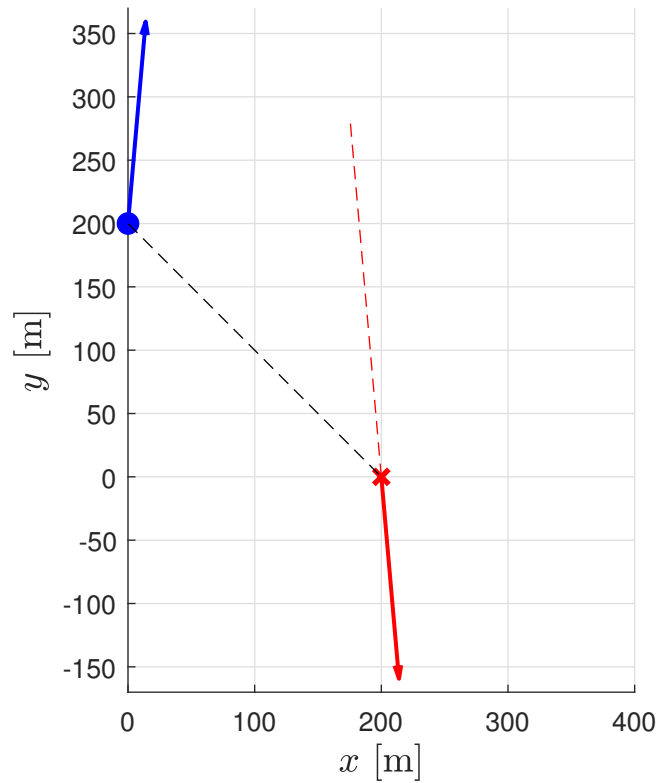


Figure 2.25: Example in which the proposed procedure cannot determine the optimal trajectory – unrealistic case: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = 85^\circ$, $x_1 = 200$ m, $y_1 = 0$ m, $\gamma_1 = -85^\circ$, $V = 160$ m/s.

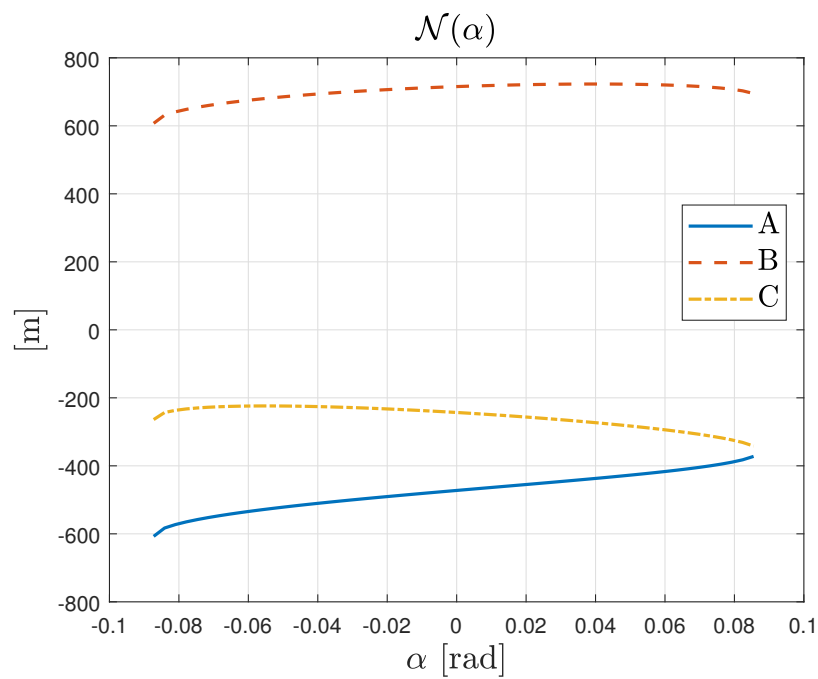


Figure 2.26: Absence of a zero of the function $\mathcal{N}(\alpha)$ for the example shown in Fig. 2.25.

Figures 2.27 and 2.28 present another example in which the proposed procedure fails to determine the optimal trajectory. Unlike the previous example, however, this combination of initial and terminal conditions is physically realistic. The reason why the proposed procedure does not yield a solution is that the optimal trajectory in this case is a straight line. Consequently, none of the analyzed trajectory-convexity cases is valid, since $\gamma' = 0$.

Based on the two presented examples for which the proposed procedure cannot determine the optimal trajectory, the previously stated conclusions may be extended in order to formulate a universal algorithm applicable to all scenarios expected in practice:

- If the verification of the existence of a zero of the function $\mathcal{N}(\alpha)$ shows that none of Cases A, B, or C is valid, then the optimal trajectory is a straight-line trajectory:

$$\eta_1 = 0, \quad \eta_2 = 0 \implies u = 0 \quad (2.61)$$

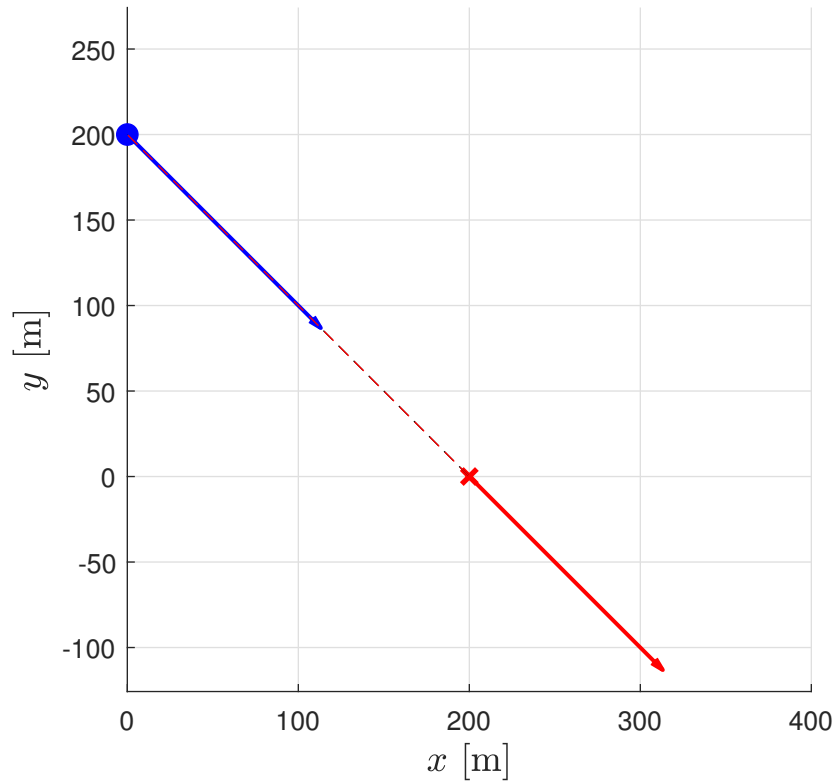


Figure 2.27: Example in which the proposed procedure cannot determine the optimal trajectory – realistic case: $x_0 = 0$ m, $y_0 = 200$ m, $\gamma_0 = -45^\circ$, $x_1 = 200$ m, $y_1 = 0$ m, $\gamma_1 = -45^\circ$, $V = 160$ m/s.

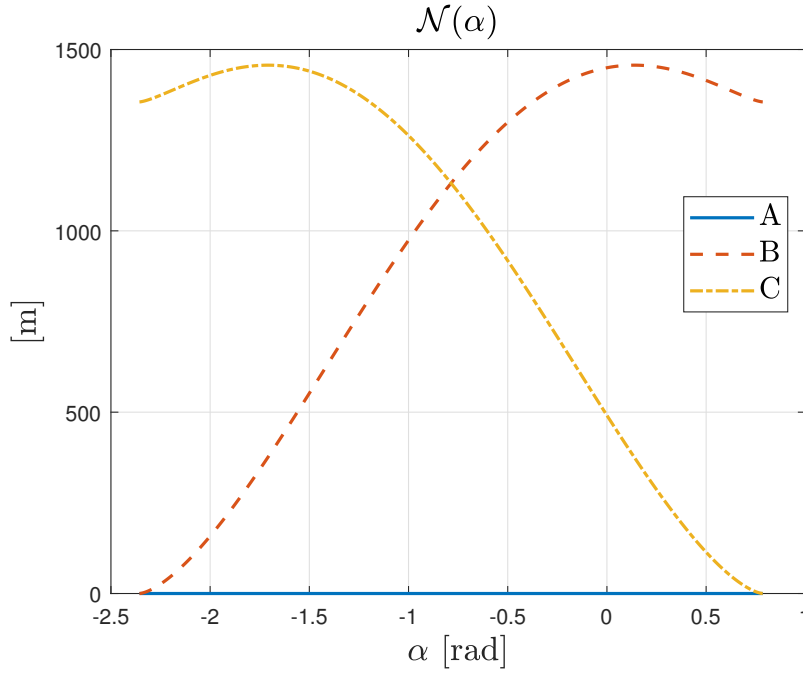


Figure 2.28: Absence of a zero of the function $\mathcal{N}(\alpha)$ for the example shown in Fig. 2.27.

It remains necessary to analyze the sign of η_3 in Case A⁴. As already discussed, in Cases B and C this sign is uniquely determined because the trajectory convexity at the final instant is known, whereas in Case A the trajectory convexity must be inferred.

For this purpose, it is necessary to identify two characteristic regions in which the missile may be located during target engagement, yielding three characteristic cases (illustrated in Fig. 2.29):

- 1° The missile is located to the left of the line defined by the desired velocity vector at the target:

$$x_0 < x_1 - \frac{\Delta y}{\tan \gamma_1} \quad (2.62)$$

- 2° The missile is located on the line defined by the desired velocity vector at the target:

$$x_0 = x_1 - \frac{\Delta y}{\tan \gamma_1} \quad (2.63)$$

- 3° The missile is located to the right of the line defined by the desired velocity vector at the target:

$$x_0 > x_1 - \frac{\Delta y}{\tan \gamma_1} \quad (2.64)$$

⁴and Case D, although these two cases are treated as equivalent.

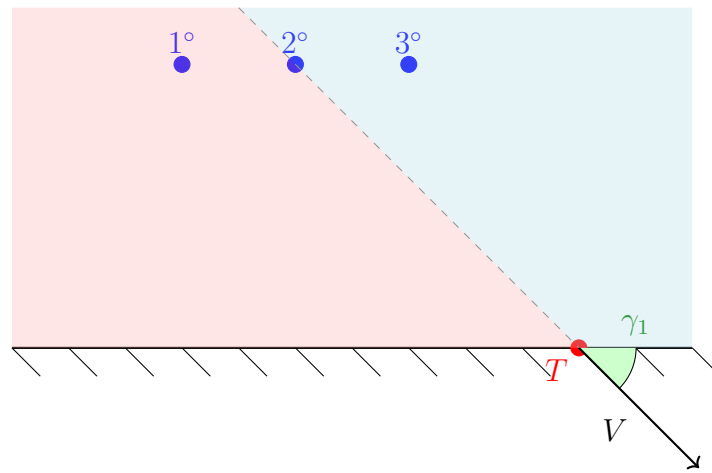


Figure 2.29: Possible initial positions of the missile relative to the line defined by the desired velocity vector at the target.

For each of the above cases, three subcases are possible depending on the orientation of the initial velocity vector relative to the missile–target line-of-sight. These subcases are shown in Fig. 2.30.

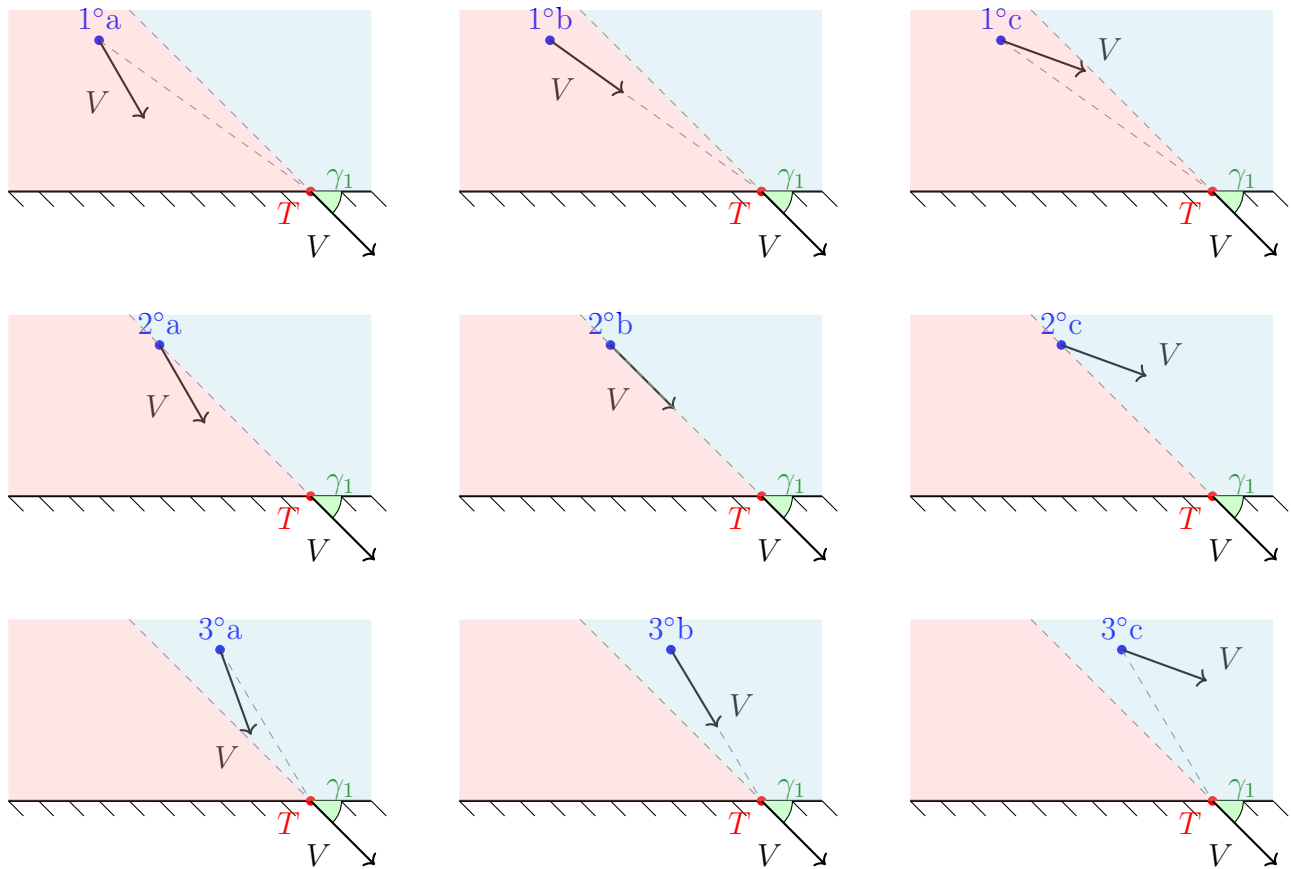


Figure 2.30: Geometric classification of engagement scenarios based on relative separation and velocity vector orientation.

It is evident that, if the missile is located on the line defined by the desired velocity vector at the target (Case 2°), interception with the desired impact angle can be achieved either by changing the trajectory convexity (subcases 2°a and 2°c) or by following a straight-line trajectory (subcase 2°b). Therefore, the procedure corresponding to Case A is not applicable, since it assumes constant trajectory convexity.

If the missile is located to the left of the line defined by the desired velocity vector at the target (Case 1°), the procedure corresponding to Case A can only be valid for subcase 1°c. Since $\gamma' < 0$ throughout the trajectory in this situation, the sign of η_3 is positive.

Similarly, if the missile is located to the right of the line defined by the desired velocity vector at the target (Case 3°), the procedure corresponding to Case A can only be valid for subcase 3°a. Since $\gamma' > 0$ throughout the trajectory in this case, the sign of η_3 is negative.

It should be noted that for subcases 1°c and 3°a, Case A is not always valid, and it is always necessary to verify which of the cases A, B, or C is applicable. Examples illustrating this are shown in Fig. 2.31. Although both examples correspond to subcase 1°c, the valid procedure in the first example is Case A, whereas in the second example the valid procedure is Case C.

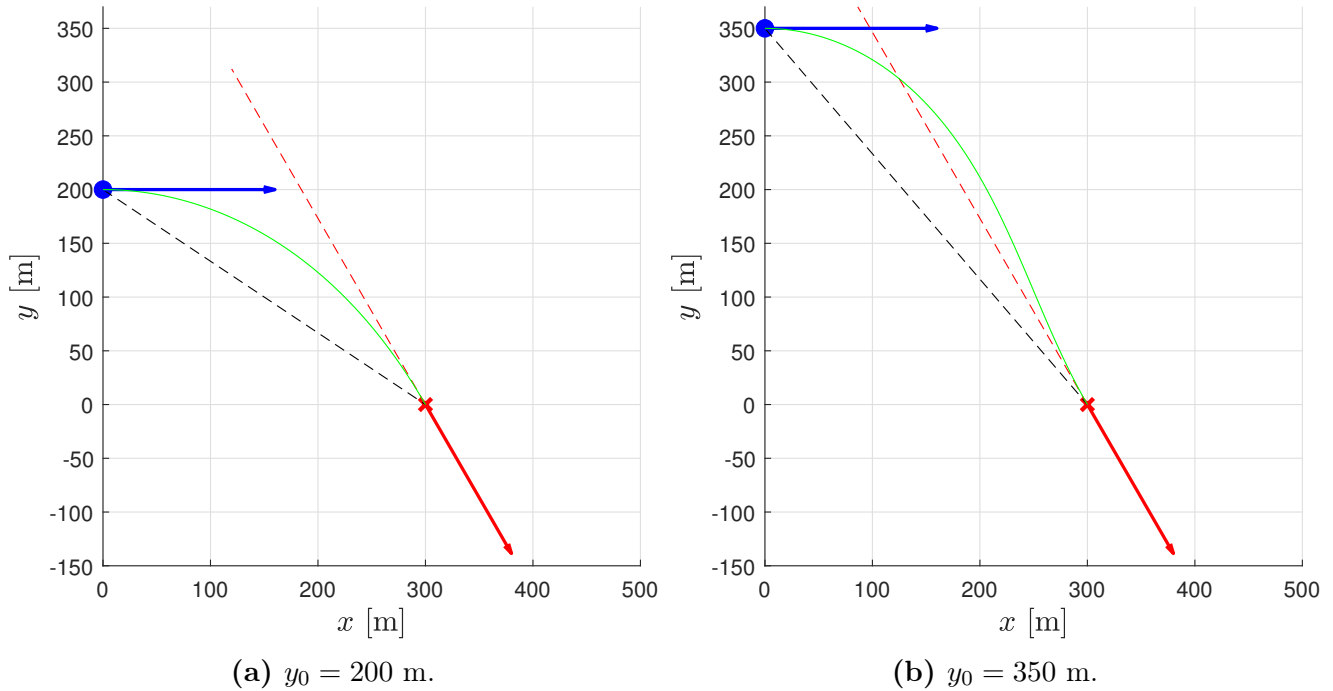


Figure 2.31: Optimal trajectories for the boundary conditions: $x_0 = 0$ m, $\gamma_0 = 0^\circ$, $x_1 = 300$ m, $y_1 = 0$ m, $\gamma_1 = -60^\circ$, and $V = 160$ m/s.

Finally, the general algorithm for solving the two-point boundary value problem (2.21) is presented below. Since the function $\mathcal{N}(\alpha)$, i.e., the integrals $I_N(\alpha)$ and $I_D(\alpha)$ from which it is constructed, take different forms depending on whether Case A, B, or C is considered, an additional argument is introduced to indicate which form of $I_N(\alpha)$ and $I_D(\alpha)$ should be used. For example:

$$\mathcal{N}(\alpha, A) = \Delta x I_D(\alpha, A) - \Delta y I_N(\alpha, A) \quad (2.65)$$

indicates that expressions (2.45) corresponding to Case A should be used; $\mathcal{N}(\alpha, B)$ indicates that expressions (2.55) corresponding to Case B should be used; and $\mathcal{N}(\alpha, C)$ indicates that expressions (2.57) corresponding to Case C should be used.

Numerical methods for evaluating the integrals $I_N(\alpha)$ and $I_D(\alpha)$, as well as for solving the nonlinear scalar equation $\mathcal{N}(\alpha, s) = 0$, $s \in A, B, C$, are provided in Appendix B.

Algorithm 2 Computation of the optimal control

```

1: Input:  $x_0, y_0, \gamma_0, x_1, y_1, \gamma_1, V$ 
2:  $l_b = \max(\gamma_0, \gamma_1) - \pi/2$ 
3:  $u_b = \min(\gamma_0, \gamma_1) + \pi/2$ 
4: if  $\mathcal{N}(l_b, A)\mathcal{N}(u_b, A) < 0$  then
5:   Solve  $\mathcal{N}(\alpha, A) = 0$  for  $\alpha$ 
6:    $r = V^3 [I_N(\alpha, A)/(x_1 - x_0)]^2$ 
7:   if  $x_0 < x_1 - \frac{\Delta y}{\tan \gamma_1}$  then
8:      $\text{sgn}(\eta_3) = 1$ 
9:   else
10:     $\text{sgn}(\eta_3) = -1$ 
11:   end if
12: else if  $\mathcal{N}(l_b, B)\mathcal{N}(u_b, B) < 0$  then
13:   Solve  $\mathcal{N}(\alpha, B) = 0$  for  $\alpha$ 
14:    $r = V^3 [I_N(\alpha, B)/(x_1 - x_0)]^2$ 
15:    $\text{sgn}(\eta_3) = 1$ 
16: else if  $\mathcal{N}(l_b, C)\mathcal{N}(u_b, C) < 0$  then
17:   Solve  $\mathcal{N}(\alpha, C) = 0$  for  $\alpha$ 
18:    $r = V^3 [I_N(\alpha, C)/(x_1 - x_0)]^2$ 
19:    $\text{sgn}(\eta_3) = -1$ 
20: else
21:    $u = 0$ 
22:   return  $u$ 
23: end if
24:  $\eta_1 = r \cos \alpha$ 
25:  $\eta_2 = r \sin \alpha$ 
26:  $\eta_3 = \text{sgn}(\eta_3) \sqrt{4V^3(\eta_1 \cos \gamma_1 + \eta_2 \sin \gamma_1)}$ 
27:  $u = [\eta_1(y_1 - y_0) - \eta_2(x_1 - x_0) - \eta_3] / (2V)$ 
28: return  $u$ 

```

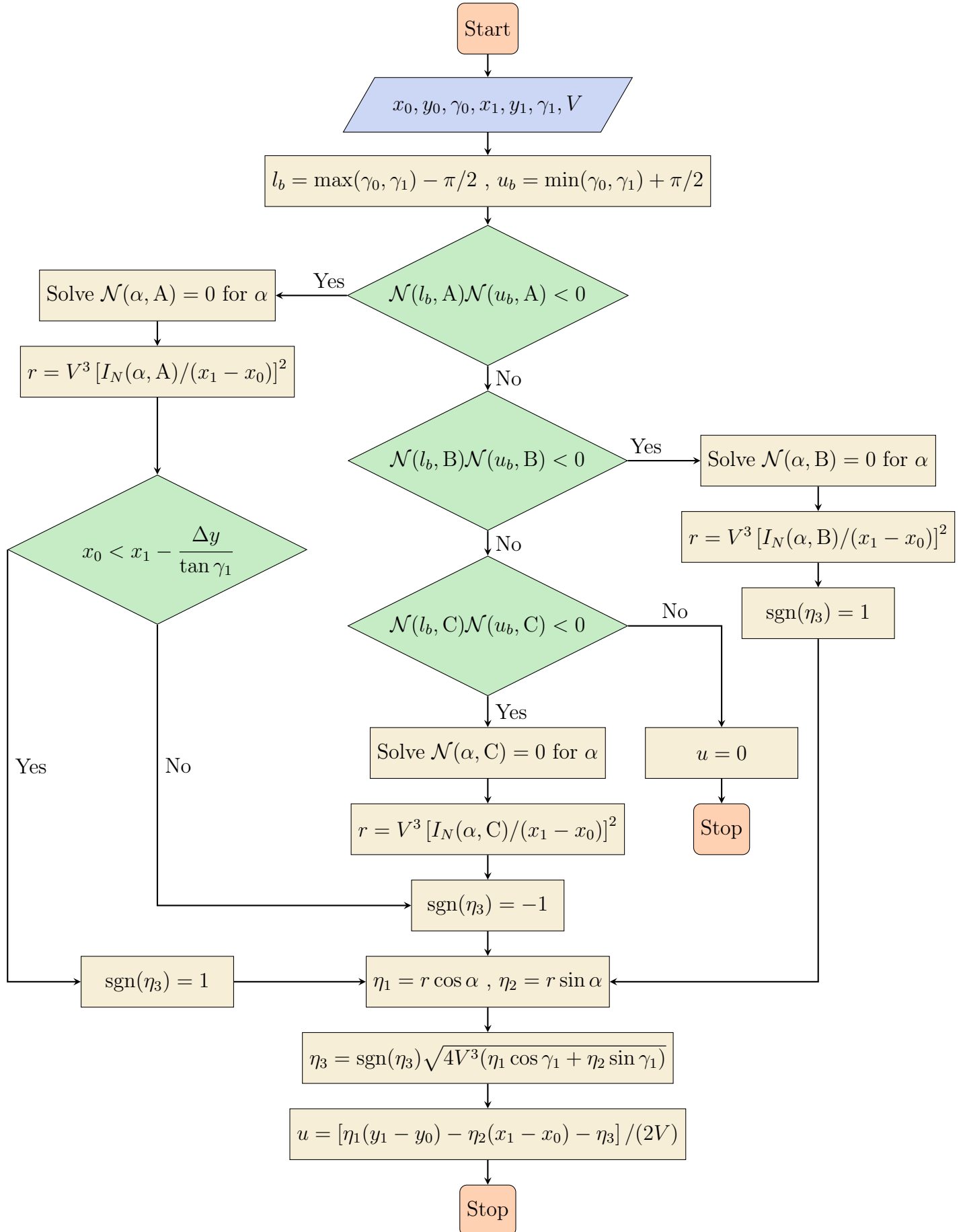


Figure 2.32: Flowchart representation of the Algorithm 2.

2.2.5 Special Case: Cruise Missile Climb Phase

In the previous section, the application of the proposed algorithm for determining the optimal control was demonstrated using examples related to terminal missile guidance. The reason for this is that the problem of achieving a desired flight-path angle is most commonly encountered in situations where an impact-angle constraint is imposed due to missile engagement of a target. However, this problem may also arise during other phases of flight, for example when passing through control points along the trajectory (*waypoint guidance*). Another example in which it is necessary to achieve a desired flight-path angle occurs during the transition of a missile from the climb phase to the cruise phase. This example is characteristic of the initial flight phase of cruise missiles, and its specificity lies in the fact that, during the climb phase, there is usually no range constraint associated with completing the transition to level flight.

Figure 2.33 shows the planar kinematic model of the cruise missile climb phase. The optimization problem can be formulated as follows:

$$\begin{aligned}
 &\text{Minimize} \quad J = \int_0^{t_1} u^2(t) dt \\
 &\text{subject to} \quad \dot{x} = V \cos \gamma \\
 &\quad \quad \quad \dot{y} = V \sin \gamma \\
 &\quad \quad \quad \dot{\gamma} = \frac{u}{V}
 \end{aligned} \tag{2.66}$$

$$\begin{aligned}
 &\text{and boundary conditions} \quad x(0) = x_0 \\
 &\quad \quad \quad y(0) = y_0 \\
 &\quad \quad \quad \gamma(0) = \gamma_0 \\
 &\quad \quad \quad y(t_1) = y_1 \\
 &\quad \quad \quad \gamma(t_1) = 0
 \end{aligned}$$

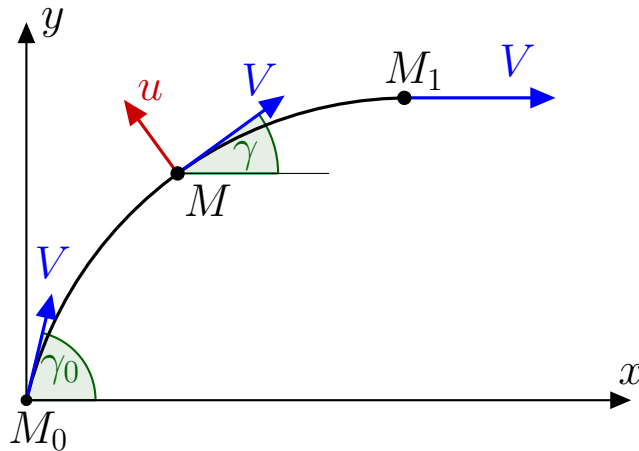


Figure 2.33: Kinematic model of the cruise missile climb phase.

The modifications with respect to the original optimization problem (2.6) are as follows:

1. There is no range constraint. This means that x_1 is free, and therefore $\eta_1 = 0$.
2. The missile transitions from the climb phase to level flight. This means that $\gamma_1 = 0^\circ$, and since $\eta_1 = 0$, it follows from (2.17) that η_3 is also zero.

Based on these modifications and their consequences, Eq. (2.23) takes the following form:

$$\lambda_\gamma(t_0) = \eta_2(x_1 - x_0) = \lambda_\gamma(x_0) \quad (2.67)$$

which leads to a modified expression for (2.26):

$$u(x) = -\frac{1}{2V} \eta_2(x_1 - x) \quad (2.68)$$

In order to use the previous expression for determining the required control input, it is necessary to determine not only the unknown parameter η_2 but also the value of x_1 , which is now unknown.

Substituting (2.67) into the second equation of system (2.21) yields:

$$\frac{d\gamma}{dx} = -\frac{\eta_2(x_1 - x)}{2V^3 \cos \gamma} \quad (2.69)$$

The variables in the above equation can be separated and integrated from an arbitrary instant to the terminal instant:

$$2V^3 \int_{\gamma}^{\gamma_1=0} \cos \gamma \, d\gamma = -\eta_2 \int_x^{x_1} (x_1 - x) \, dx \quad (2.70)$$

which results in:

$$2V^3 \sin \gamma = \frac{\eta_2}{2} (x_1 - x)^2 \quad (2.71)$$

To determine the two unknown quantities, η_2 and x_1 , two equations containing them as the only unknowns are required. The first relation can be obtained from (2.71) by selecting the initial instant as the current one:

$$2V^3 \sin \gamma_0 = \frac{\eta_2}{2} (x_1 - x_0)^2 \quad (2.72)$$

If $\sin \gamma$ is expressed from (2.71):

$$\sin \gamma = \frac{\eta_2}{4V^3} (x_1 - x)^2 \quad (2.73)$$

and taking into account that the flight-path angle during the climb phase belongs to the interval

$$0 \leq \gamma < 90^\circ \quad (2.74)$$

the first equation of system (2.21) becomes

$$\frac{dy}{dx} = \frac{\sin \gamma}{\sqrt{1 - \sin^2 \gamma}} = \frac{\eta_2(x_1 - x)^2}{\sqrt{16V^6 - \eta_2^2(x_1 - x)^4}} \quad (2.75)$$

If the optimal trajectory were one involving an altitude overshoot, then, due to the necessary change in trajectory convexity during the climb phase, the flight-path angle interval would be $-90^\circ < \gamma < 90^\circ$. However, as will be shown later, the only practically feasible solution for this optimization problem corresponds to $\gamma' < 0$ throughout the climb phase, which means that the minimum flight-path angle is reached precisely at the end of the maneuver, where $\gamma = 0$.

The variables in (2.75) can also be separated and integrated from the initial to the terminal instant in order to obtain a second relation between η_2 and x_1 :

$$y_1 - y_0 = \eta_2 \int_{x_0}^{x_1} \frac{(x_1 - x)^2}{\sqrt{16V^6 - \eta_2^2(x_1 - x)^4}} dx \quad (2.76)$$

If η_2 is expressed from (2.72):

$$\eta_2 = \frac{4V^3 \sin \gamma_0}{(x_1 - x_0)^2} \quad (2.77)$$

and the constant $S = \frac{\sin \gamma_0}{(x_1 - x_0)^2}$ is introduced, Eq. (2.76) takes the form:

$$S \int_{x_0}^{x_1} \frac{(x_1 - x)^2}{\sqrt{1 - S^2(x_1 - x)^4}} dx = \Delta y \quad (2.78)$$

Introducing the substitution $S(x_1 - x)^2 = \sin w$, the integration limits in terms of the new variable take a particularly simple form:

$$w(x = x_1) = \arcsin(0) = 0 \quad (2.79a)$$

$$w(x = x_0) = \arcsin(S(x_1 - x_0)^2) = \gamma_0 \quad (2.79b)$$

and Eq. (2.78) becomes:

$$\frac{1}{2\sqrt{S}} \int_0^{\gamma_0} \frac{\sin w \cos w}{\cos w \sqrt{\sin w}} dw = \Delta y \quad (2.80)$$

Because of the interval (2.74) to which the flight-path angle belongs during the climb phase, the following relation can be obtained:

$$\int_0^{\gamma_0} \sqrt{\sin w} dw = 2\sqrt{S}\Delta y = 2\frac{\sqrt{\sin \gamma_0}}{x_1 - x_0} \Delta y \quad (2.81)$$

If the integral in the previous equation is denoted by I_S for simplicity, x_1 can be determined as

$$x_1 = x_0 + 2 \frac{\sqrt{\sin \gamma_0}}{I_S} \Delta y \quad (2.82)$$

Based on (2.77), η_2 can now be determined as

$$\eta_2 = \frac{V^3}{(\Delta y)^2} I_S^2 \quad (2.83)$$

Finally, from (2.68), the control input becomes

$$u(x) = -\frac{V^2}{2(\Delta y)^2} I_S^2 \left[(x_0 - x) + \frac{2\sqrt{\sin \gamma_0}}{I_S} \Delta y \right] \quad (2.84)$$

If the initial instant is selected as the current instant, removing the subscript 0 yields the feedback form of the guidance law:

$$u(y, \gamma) = -\frac{V^2 \sqrt{\sin \gamma}}{y_1 - y} \int_0^\gamma \sqrt{\sin w} dw \quad (2.85)$$

By expressing the control input as a function of the state variables, in this case the current altitude and flight-path angle, the need to repeatedly solve the optimization problem whenever the trajectory deviates from the nominal path due to flight disturbances is eliminated [22]. Moreover, the control input can be determined very efficiently, without the iterative procedures that are unavoidable when solving nonlinear equations numerically, as was the case in Section 2.2.4. Unfortunately, due to the integral appearing in (2.85), the solution of the optimization problem cannot be considered fully analytical, since this integral must still be evaluated numerically using the methods presented in Appendix B.2.

The trajectory computed using expression (2.85) for the control input is shown in Fig. 2.34 for a desired cruise altitude of 200 m. Since this corresponds to the initial phase of the missile flight, the starting point is not chosen at the launch location but rather at $(x_0, y_0) = (200, 100)$, because guidance is activated only after the velocity becomes sufficiently high to ensure adequate maneuvering capability.

Figure 2.35 shows the control input and the flight-path angle for the considered example. It can be observed that the control variation with respect to range is linear, which is consistent with expression (2.68). Furthermore, at the end of this flight phase, at $x = x_1$, the control input reaches zero, ensuring that no further change in the flight-path angle occurs. In this way, a smooth transition to level flight and the subsequent cruise phase is achieved.

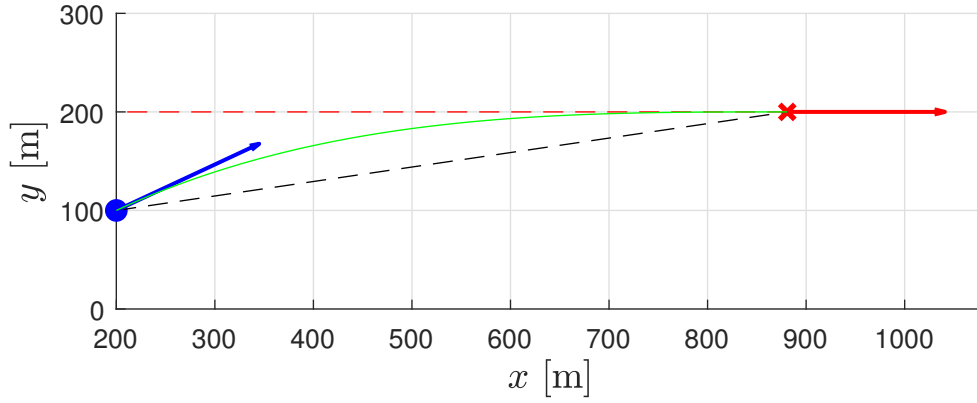


Figure 2.34: Optimal trajectory for boundary conditions: $x_0 = 200$ m, $y_0 = 100$ m, $\gamma_0 = 25^\circ$, $y_1 = 200$ m, $\gamma_1 = 0^\circ$, and velocity $V = 160$ m/s. x_1 is free.

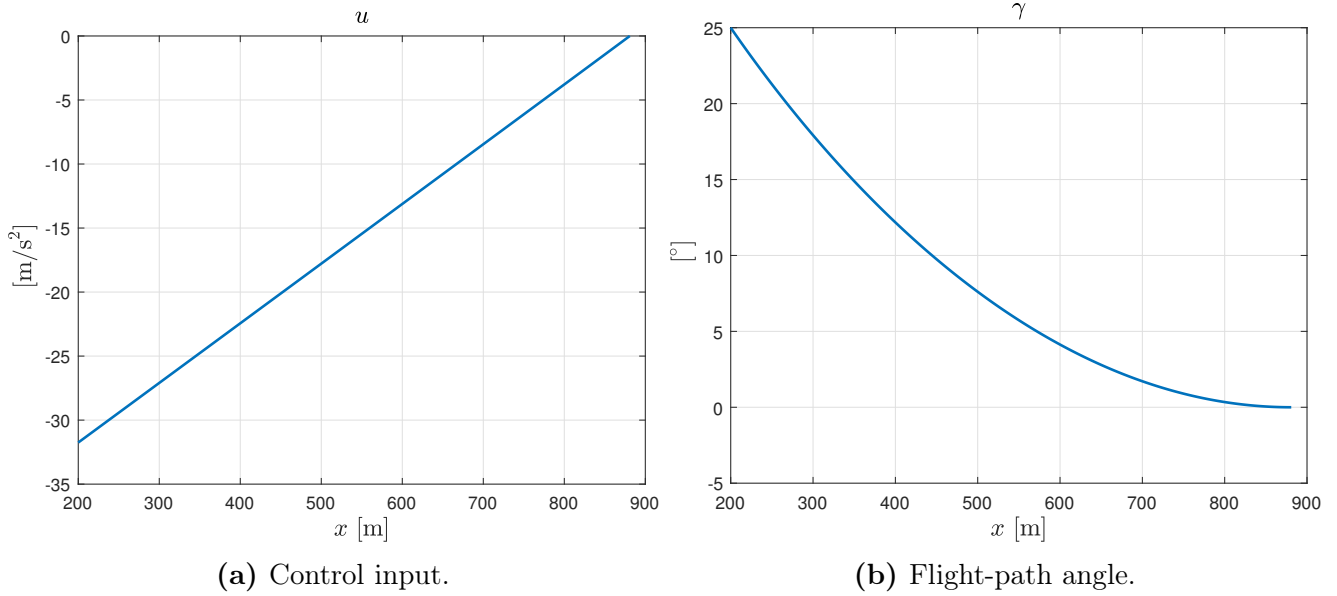


Figure 2.35: Flight parameters on the optimal trajectory for boundary conditions: $x_0 = 200$ m, $y_0 = 100$ m, $\gamma_0 = 25^\circ$, $y_1 = 200$ m, $\gamma_1 = 0^\circ$, and velocity $V = 160$ m/s. x_1 is free.

To verify the optimality of the proposed approach, the climb problem can also be analyzed using the general algorithm for optimal trajectory determination presented in Section 2.2.4. Since the initial flight-path angle is always positive, it follows from interval 2.74 (and also from the example in Fig. 2.35b) that the climb problem should be treated as Case A: $\gamma' < 0$ over the entire trajectory.

As previously stated, in this case $\eta_1 = 0$ because x_1 is not specified. Therefore, using expressions (2.29) and (2.48), it follows:

$$\begin{aligned} \eta_1 = 0 &\implies \alpha = \frac{\pi}{2} \\ \eta_2 = r &= V^3 \left(\frac{I_N(\pi/2)}{\Delta x} \right)^2 \end{aligned} \quad (2.86)$$

For Case A, the integral $I_N(\alpha)$ is given by (2.45), and in the considered case it becomes:

$$I_N(\pi/2) = -I_2(\gamma_0 - \pi/2, 0 - \pi/2) = -2\sqrt{\sin \gamma_0} \quad (2.87)$$

Based on the previous expressions, η_2 is:

$$\eta_2 = 4V^3 \frac{\sin \gamma_0}{(\Delta x)^2} \quad (2.88)$$

which confirms expression (2.77).

Since $\eta_1 = 0$ and $\gamma_1 = 0$, the guidance law from (2.26) becomes:

$$u = -2V^2 \frac{\sin \gamma_0}{\Delta x} \quad (2.89)$$

It should be noted that this expression cannot be used in the general case for arbitrary x_1 . It is derived only to show that, for a known x_1 obtained from (2.82), the procedures for computing the control presented in this section and in Section 2.2.4 are equivalent.

In the previous example (Fig. 2.34), the desired altitude was reached at a range of $x_1 = 880$ m. If this value is imposed as a boundary condition and Case A is applied, i.e., expression (2.89) is used for computing the control, the same trajectory is obtained as when using (2.85). This comparison is shown in Figs. 2.36 and 2.37.

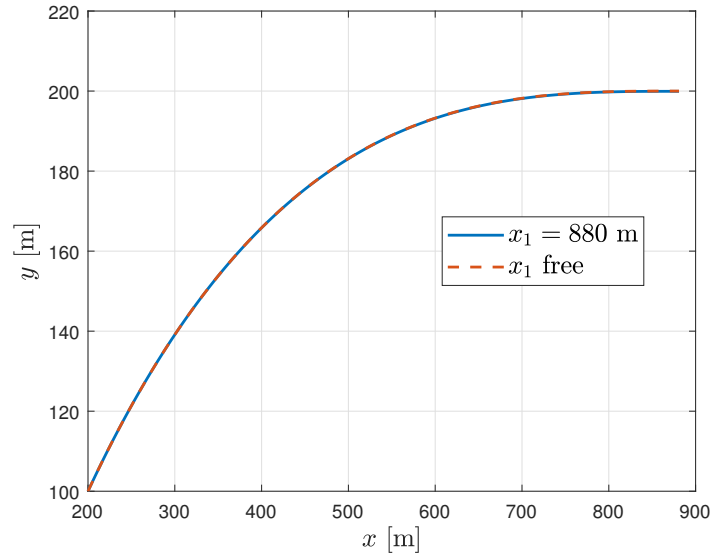


Figure 2.36: Comparison of trajectories obtained using (2.89) — fixed x_1 , and (2.85) — free x_1 .

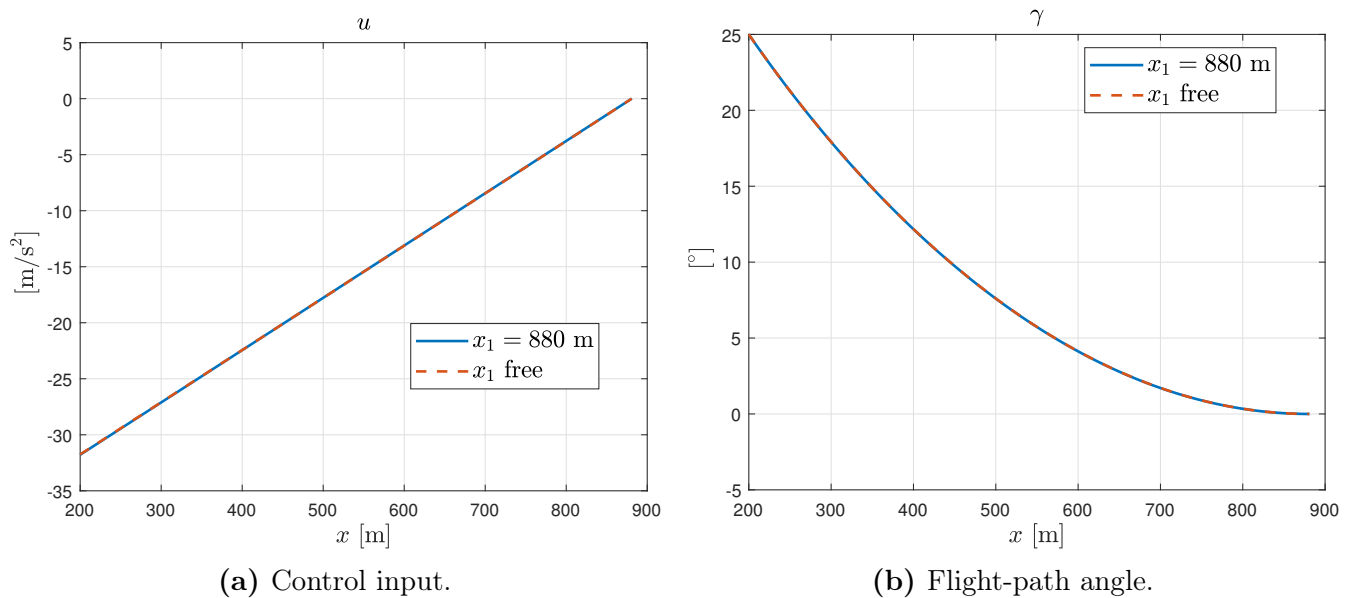


Figure 2.37: Comparison of trajectories obtained using (2.89) — fixed x_1 , and (2.85) — free x_1 .

If instead of the value of x_1 obtained from (2.82), an arbitrary value is chosen, the optimization problem can still be solved using the general Algorithm 2. Figures 2.38 and 2.39 show a comparison of trajectories obtained for $x_1 = 600$ m and $x_1 = 1200$ m, compared to the trajectory from the previous example (Fig. 2.34).

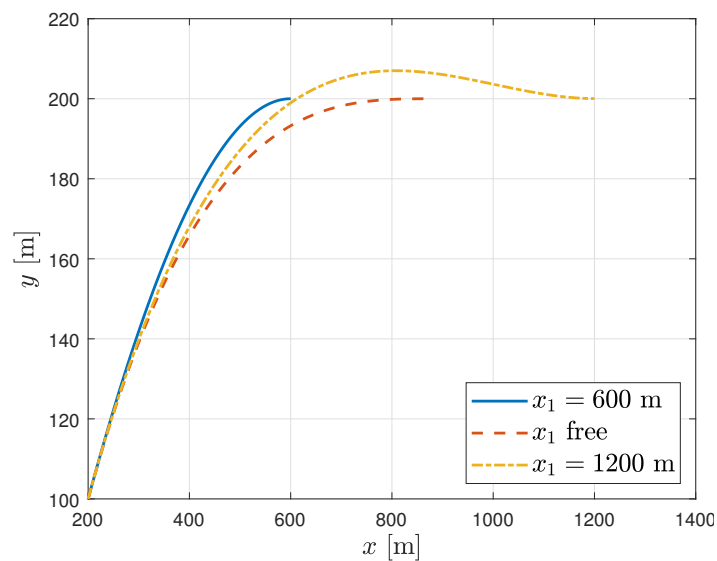


Figure 2.38: Comparison of trajectories obtained using Algorithm 2 — fixed x_1 , and using expression (2.85) — free x_1 .

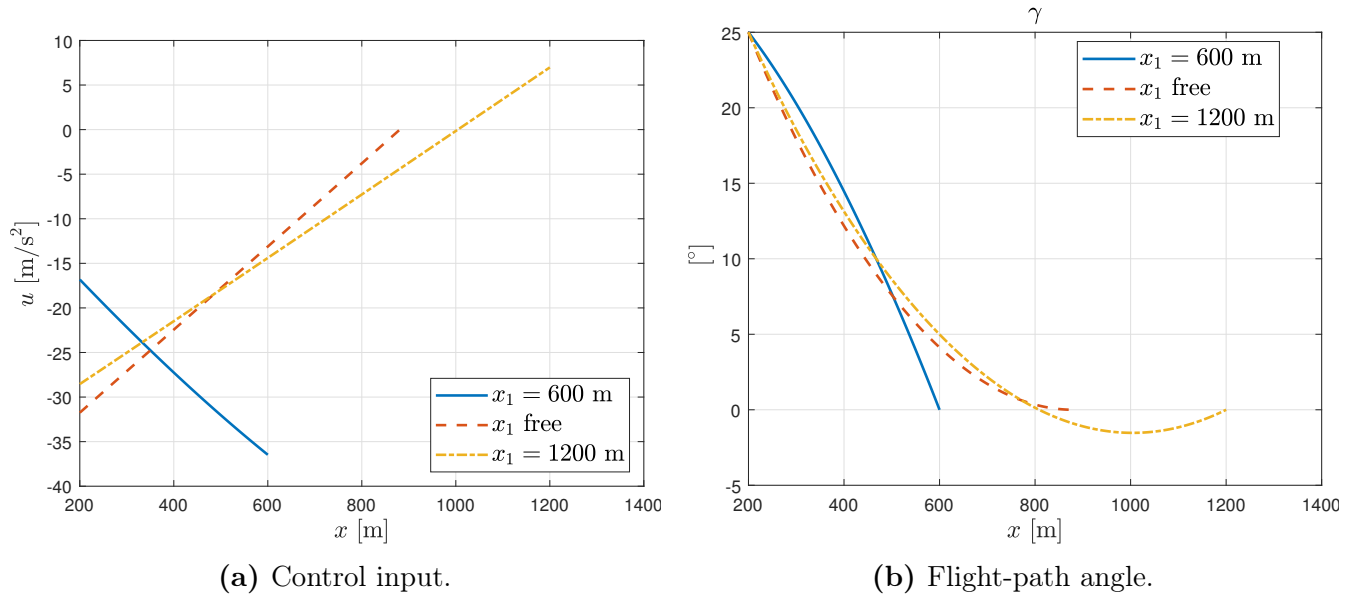


Figure 2.39: Comparison of trajectories obtained using Algorithm 2 — fixed x_1 , and using expression (2.85) — free x_1 .

Although at first glance the trajectories in Fig. 2.38 may not seem directly comparable because in one case x_1 is fixed and in the other it is free, this example is significant because it illustrates that imposing a fixed x_1 can yield a more optimal trajectory than the free- x_1 case. Specifically, for the trajectory obtained using (2.85), the performance index takes the value:

$$J|_{x_1 \text{ free}} = 1492 \text{ m}^2/\text{s}^3$$

while in the case of fixed x_1 the values are:

$$J|_{x_1=600} = 1960 \text{ m}^2/\text{s}^3, \quad J|_{x_1=1200} = 1444 \text{ m}^2/\text{s}^3$$

Figure 2.40 shows the dependence of the performance index J on x_1 over the interval from 600 m to 2000 m. The value of the performance index is obtained by solving the optimization problem with a prescribed final range using Algorithm 2. The red point indicates the solution of the optimization problem in which x_1 is free.

It is evident that increasing x_1 leads to trajectories with a lower performance index, which is expected since at short distances a large maneuver is required to achieve the desired altitude and flight-path angle. The decreasing trend of the performance index continues indefinitely, indicating that the optimal climb trajectory corresponds to $x_1 \rightarrow \infty$. Since such a trajectory is not practically feasible, the question arises as to what type of trajectories are obtained using expression (2.85).

By applying the procedure presented in this section for computing the control in feedback form, trajectories are obtained that satisfy the minimum principle. Using terminology from optimal control theory, such trajectories cannot be called fully optimal because, as shown in Fig. 2.40, by selecting sufficiently large x_1 and solving the optimization problem with fixed final range, a trajectory that yields a lower value of the performance index than in the free- x_1 case can be

obtained. However, using expression (2.85) for computing the control yields trajectories that satisfy the extremal conditions. These trajectories have very convenient properties for practical implementation. Namely, only in the case where x_1 is not specified⁵ is the desired altitude reached at the instant when the control input becomes zero, meaning that no further change in the flight-path angle occurs. In this way, a smooth transition from climb to level flight is achieved, without abrupt changes in normal acceleration and without overshoot relative to the cruise altitude.

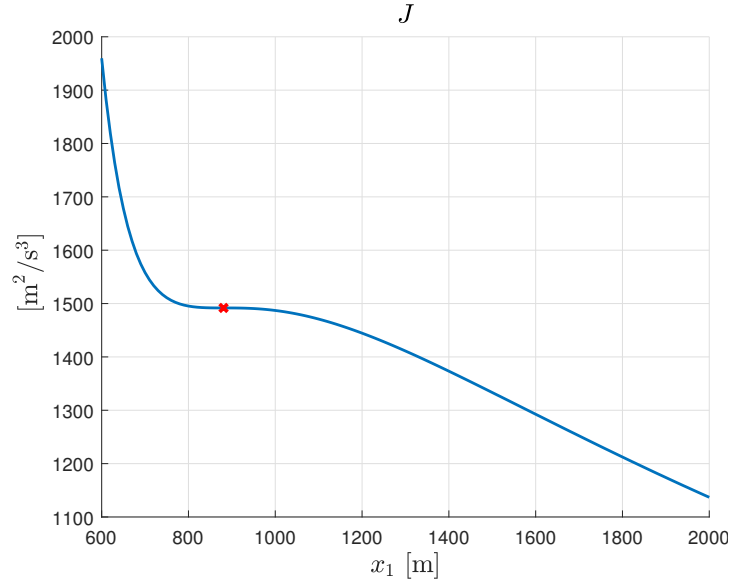


Figure 2.40: Dependence of the performance index, obtained by solving optimization problem (2.6) using Algorithm 2, on the prescribed value of x_1 .

2.3 Comparison with Existing Methods

In this part of the work, the results of numerical simulation of a planar kinematic model of missile–stationary target engagement are presented.

The main purpose of this section is to compare the optimal trajectory obtained using Algorithm 2 with trajectories obtained using existing methods for achieving desired impact angles, presented in Section 1.2. For this purpose, Algorithm 2 is executed only once, at the beginning of the simulation, and based on the input data the unknown parameters η_1 and η_2 are determined, so that the required optimal control, based on Eq. (2.26), is computed during the simulation as follows:

$$u(x, y) = [\eta_1 \Delta y - \eta_2 \Delta x - \eta_3] / (2V) \quad (2.90)$$

where $\Delta x = x_1 - x$ and $\Delta y = y_1 - y$.

The analysis of the use of Algorithm 2 in real time, both in numerical simulation of missile flight with six degrees of freedom and on real hardware, is presented in Chapter 3.

⁵or exactly matches the value given by (2.82)

2.3.1 Numerical Simulation Setup

The kinematic model of the engagement between a missile moving with constant speed V and a stationary target with coordinates (x_1, y_1) is described by the following equations:

$$\begin{aligned}\dot{x} &= V \cos \gamma \\ \dot{y} &= V \sin \gamma \\ \dot{\gamma} &= \frac{u}{V}\end{aligned}\tag{2.91}$$

with initial conditions:

$$\begin{aligned}x(0) &= x_0 \\ y(0) &= y_0 \\ \gamma(0) &= \gamma_0\end{aligned}\tag{2.92}$$

The goal is to hit the target with the desired impact angle γ_1 , and the control input u is computed using the guidance methods presented so far in the thesis.

For the implementation of most existing guidance laws for this purpose, knowledge of the line-of-sight rate is required, which during numerical simulation is computed as follows [23]:

$$\dot{\lambda} = \frac{\Delta x(\dot{y}_1 - \dot{y}) - \Delta y(\dot{x}_1 - \dot{x})}{(\Delta x)^2 + (\Delta y)^2}\tag{2.93}$$

Methods for estimating time-to-go often require the relative closing velocity:

$$V_c = -\frac{\Delta x(\dot{x}_1 - \dot{x}) - \Delta y(\dot{y}_1 - \dot{y})}{\sqrt{(\Delta x)^2 + (\Delta y)^2}}\tag{2.94}$$

and the distance to the target:

$$R = \sqrt{(\Delta x)^2 + (\Delta y)^2}\tag{2.95}$$

Since stationary targets are considered in this chapter, the target velocity components are $\dot{x}_1 = 0$, $\dot{y}_1 = 0$.

The line-of-sight angle is computed as:

$$\lambda = \text{atan}\left(\frac{\Delta y}{\Delta x}\right)\tag{2.96}$$

To avoid the issue present in some guidance laws where the normal acceleration tends to infinity in the final moments of flight, the control input is bounded:

$$-100 \leq u \leq 100\tag{2.97}$$

and in all simulations, guidance is terminated when the distance to the target, given by Eq. (2.95), reaches 1 m, after which the control input is held constant until the end of the simulation.

Since ground targets are considered, the termination condition of the simulation is:

$$x \geq x_1 + 10 \quad || \quad y < y_1 \quad (2.98)$$

The results presented below were obtained using the *Simulink*® package within *Matlab*® version 2024b. For all simulations, a fixed time step $T_s = 0.0001$ s and an ode4 (Runge–Kutta) solver were used.

After the simulation, the control effort, given by:

$$J = \int_0^{t_1} u^2(t) dt \quad (2.99)$$

is computed numerically using the *trapz* function.

2.3.2 Comparison with Linear Optimal Guidance Laws

The most well-known guidance law within the class of linear optimal guidance methods is the Trajectory Shaping Guidance (TSG) law, in which the normal acceleration, based on Eq. (1.64), is defined as:

$$u = 4V\dot{\lambda} + \frac{2V(\lambda - \gamma_1)}{t_{go}} \quad (2.100)$$

where time-to-go is determined using the simplest approach:

$$t_{go} = \frac{R}{V_c} \quad (2.101)$$

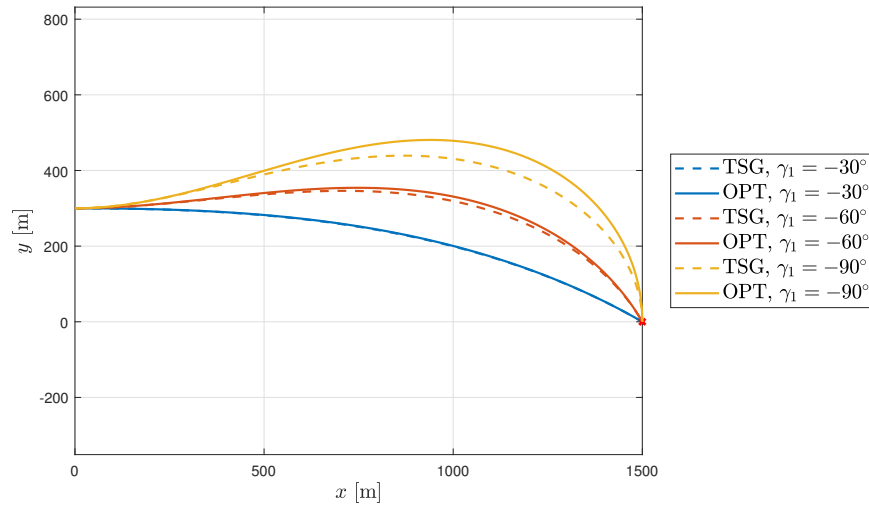
Figure 2.41 shows a comparison between the results obtained using the TSG method and the optimal guidance law (2.90). The presented trajectories correspond to a scenario in which the missile transitions from cruise phase to terminal guidance for different impact angles. The simulation parameters are given in Table 2.1.

Table 2.1: Simulation parameters for comparison between optimal and linear guidance laws.

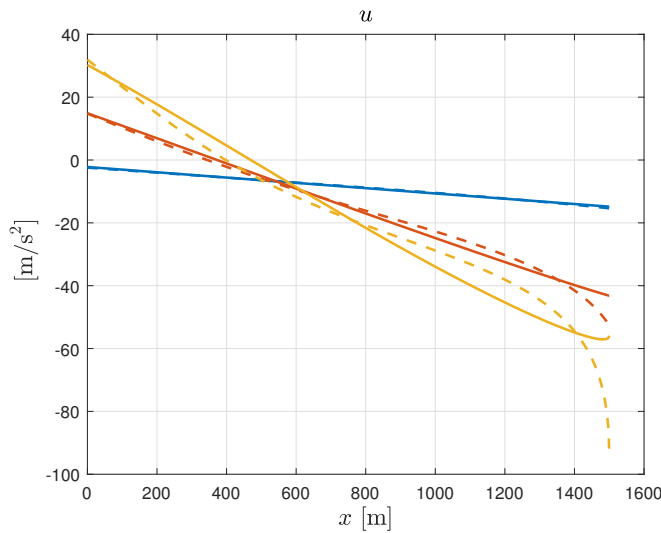
Parameter	Value
x_0	0 m
y_0	300 m
γ_0	0°
x_1	1500 m
y_1	0
γ_1	−30°, −60°, −90°
V	160 m/s

The results show that for smaller impact angles, the trajectory obtained using the TSG method does not significantly differ from the optimal trajectory, and the control effort (2.99), shown in Table 2.2, is identical for the case $\gamma_1 = -30^\circ$. This is expected, since such a scenario (transition from cruise phase and engagement at small impact angles) can be linearized in the vicinity of the collision triangle, and TSG guidance is optimal for the linearized engagement geometry.

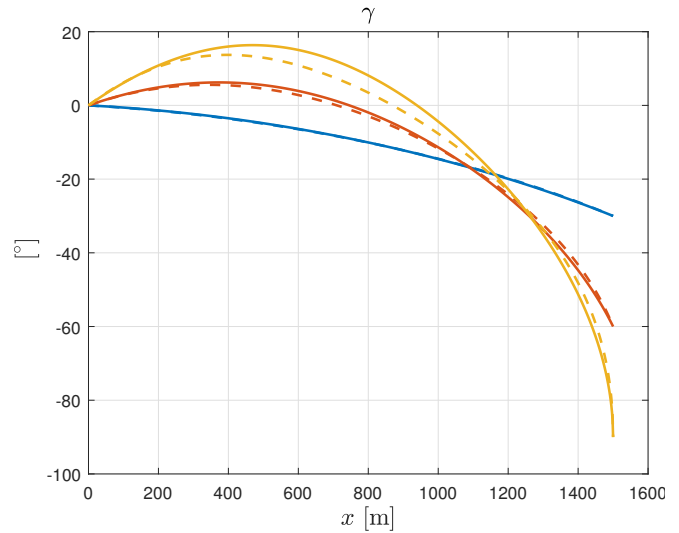
As the desired impact angle increases, the trajectory obtained using the TSG method starts to deviate more significantly from the optimal one, which is clearly visible in Fig. 2.41 for $\gamma_1 = -60^\circ$, and is most pronounced in the limiting case $\gamma_1 = -90^\circ$. Although the desired impact angle is still achieved, the TSG guidance produces large control effort values as the missile approaches the target. Under practical limitations, this may lead to command saturation and possible target miss.



(a) Trajectory.



(b) Control input.



(c) Flight-path angle.

Figure 2.41: Comparison of results obtained using Algorithm 2 (OPT) and TSG guidance.

Table 2.2: Control effort obtained using Algorithm 2 (OPT) and TSG guidance.

γ_1	OPT	TSG
-30°	856	856
-60°	5875	5922
-90°	14452	15148

Minimization of the modified cost function given by expression (1.68):

$$\int_0^{t_1} \frac{u^2}{2t_{go}^n} dt \quad (2.102)$$

was the subject of the work [43], in which the guidance law for the case of an ideal autopilot (OGL/IAC-0) is given by the Eq. (1.69):

$$u = -\frac{V}{t_{go}}(-n_\lambda \lambda + n_\gamma \gamma + n_{\gamma_1} \gamma_1) \quad (2.103)$$

where:

$$n_\lambda = (n+2)(n+3) \quad (2.104a)$$

$$n_\gamma = 2(n+2) \quad (2.104b)$$

$$n_{\gamma_1} = (n+1)(n+2) \quad (2.104c)$$

In a previous work by the same authors [42], where the same cost function is considered but with $n = 0$, the following equation for determining the time-to-go (hereafter Method 2) was proposed:

$$t_{go} = \frac{R}{V \left[1 - \frac{\bar{\gamma}^2 + \bar{\gamma}_1^2}{15} + \frac{\bar{\gamma}\bar{\gamma}_1}{30} + \frac{\bar{\gamma}^4 + \bar{\gamma}_1^4}{420} - \frac{\bar{\gamma}\bar{\gamma}_1(\bar{\gamma}^2 + \bar{\gamma}_1^2 - \bar{\gamma}\bar{\gamma}_1)}{840} \right]} \quad (2.105)$$

where:

$$\bar{\gamma} = \gamma + \lambda \quad (2.106a)$$

$$\bar{\gamma}_1 = \gamma_1 + \lambda \quad (2.106b)$$

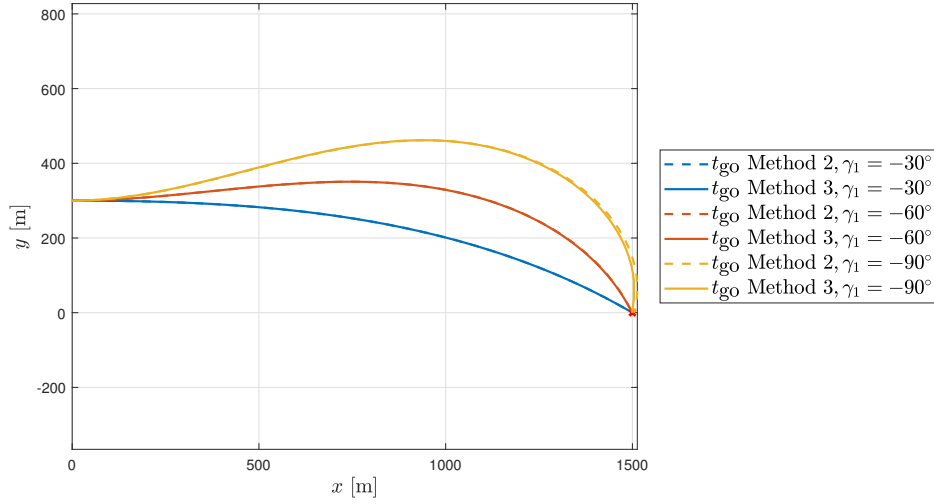
In the work [43], a procedure for determining the time-to-go (hereafter Method 3) is also proposed:

$$t_{go} = \frac{R}{V}(1+k) \quad (2.107)$$

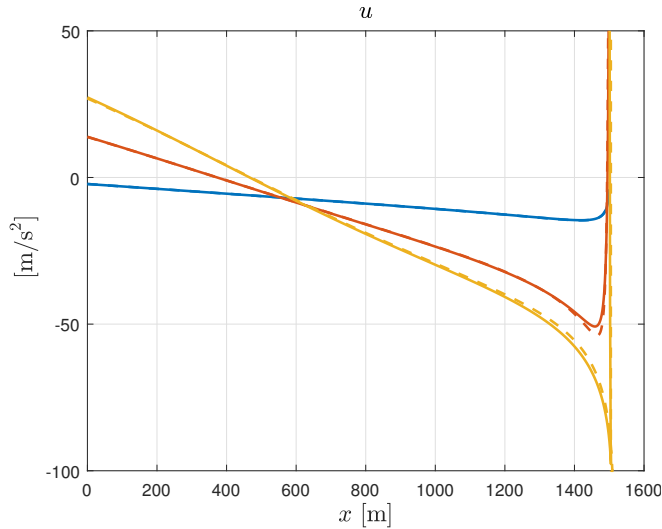
where:

$$k = \begin{cases} \frac{2\bar{\gamma}^2 - \bar{\gamma}_1\bar{\gamma} + 2\bar{\gamma}_1^2}{30}, & \text{for } n = 0 \\ \frac{3\bar{\gamma}^2 - 2\bar{\gamma}_1\bar{\gamma} + 12\bar{\gamma}_1^2}{70}, & \text{for } n = 1 \\ \frac{4\bar{\gamma}^2 - 3\bar{\gamma}_1\bar{\gamma} + 36\bar{\gamma}_1^2}{126}, & \text{for } n = 2 \end{cases} \quad (2.108)$$

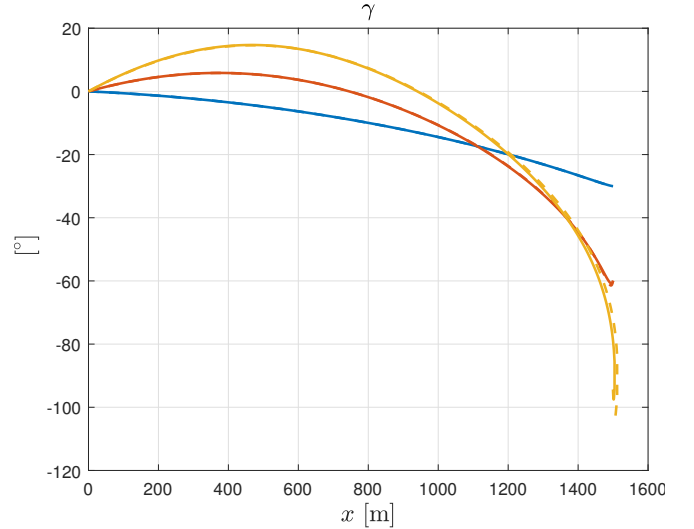
In Fig. 2.42, the results obtained by applying expression (2.103) with $n = 0$ (OGL/IAC-0, $n = 0$) and two methods for determining t_{go} are shown. The values of control effort for different impact angles are summarized in Table 2.3.



(a) Trajectory.



(b) Control input.



(c) Flight-path angle.

Figure 2.42: Comparison of results obtained using expression (2.103) for $n = 0$ (OGL/IAC-0, $n = 0$), and different methods for determining t_{go} proposed in [42] and [43].

Table 2.3: Control effort obtained using expression (2.103) for $n = 0$ (OGL/IAC-0, $n = 0$), and different methods for determining t_{go} proposed in [42] and [43].

γ_1	t_{go} Method 2 [42]	t_{go} Method 3 [43]
-30°	861	860
-60°	6843	6433
-90°	20921	19133

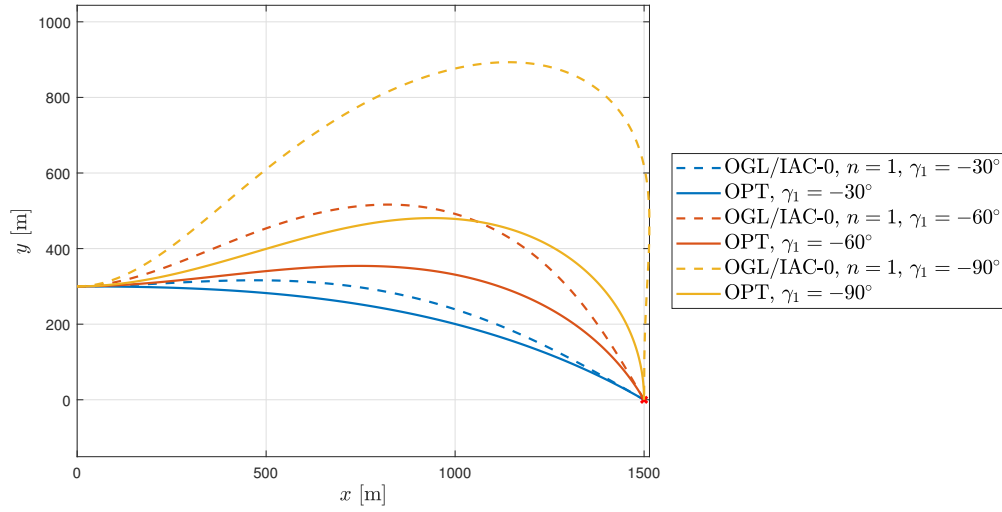
In Fig. 2.42b, it can be seen that both methods lead to a significant increase in control effort in the final moments of flight, which consequently results in high values of the performance index.

The advantage of the guidance methods obtained by minimizing the cost function (2.102) lies in the fact that by selecting the parameter n , which is implicitly present in the guidance law (2.103), the shape of the trajectory is influenced. For values $n \geq 1$, the missile tends to align with the desired line-of-sight direction earlier in the flight, which reduces the requirement for normal acceleration as the missile approaches the target. In this way, the robustness of the system is increased, since more freedom is left for later maneuvers, which is beneficial in the presence of disturbances, actuator limitations, etc.

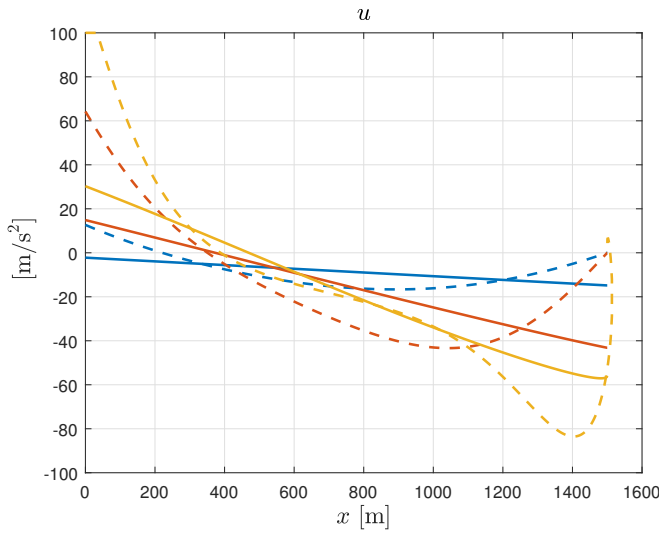
However, this increase in robustness comes at the cost of higher total control effort. Figure 2.43 shows a comparison between the results obtained by optimal guidance (Algorithm 2) and the guidance law (2.103) (OGL/IAC-0) for $n = 1$. Due to issues arising when using Method 2 and Method 3 for determining the remaining time to intercept the target, the simplest approach is used here: $t_{go} = R/V_c$.

Based on the control effort values presented in Table 2.4, it is clear that for $n = 1$ the OGL/IAC-0 guidance is far from optimal. However, it should be noted that this guidance law does not minimize a purely “energy” cost function, but due to its advantages it is suitable for practical implementation, which is why the comparison with optimal guidance is presented. From Fig. 2.43a, these advantages are clearly visible: the missile aligns with the desired direction earlier than in the optimal trajectory case, which corresponds to lower normal acceleration in the final phase of flight, as shown in Fig. 2.43b.

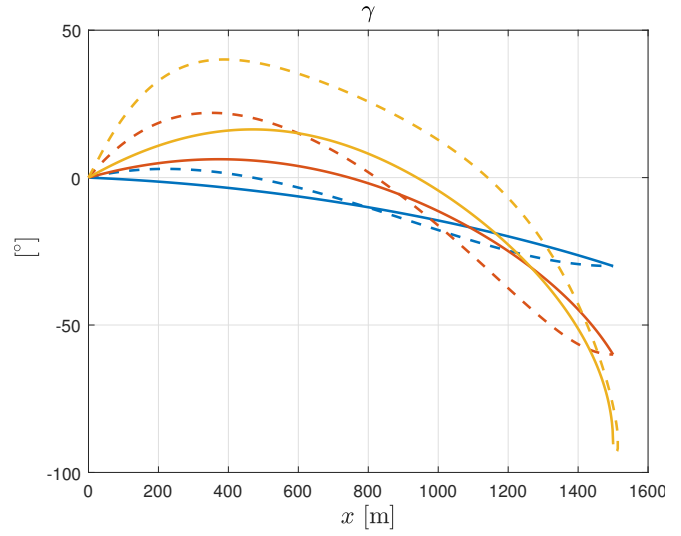
In [42], a formulation of a first-order autopilot guidance law (OGL/IAC-1) is also presented. However, in the expressions for commanded normal acceleration, weighting matrix coefficients appear whose selection is not clearly defined in the paper. Furthermore, the analysis of optimal guidance applied to a realistic missile with a non-ideal control system is presented in Chapter 3. For these reasons, further comparison of optimal guidance with the methods presented in [42] will not be considered in this dissertation.



(a) Trajectory.



(b) Control input.



(c) Flight-path angle.

Figure 2.43: Comparison of results obtained using Algorithm 2 (OPT) and expression (2.103) (OGL/IAC-0) for $n = 1$, with $t_{go} = R/V_c$.

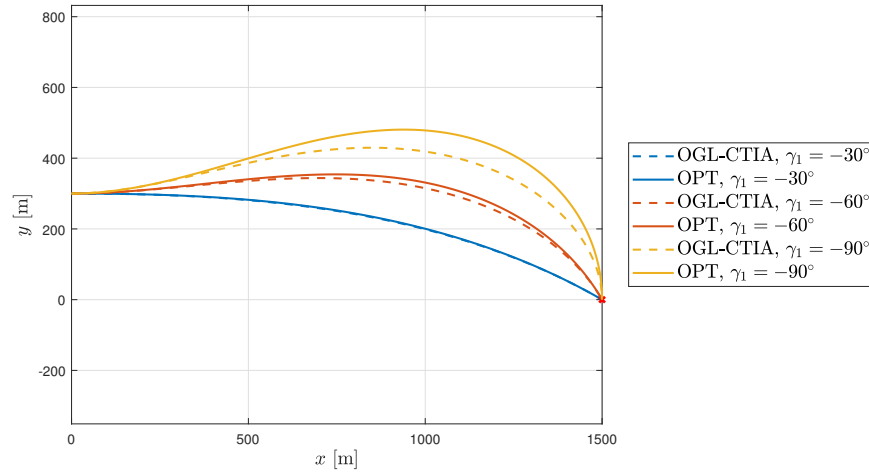
Table 2.4: Control effort obtained using Algorithm 2 (OPT) and expression (2.103) (OGL/IAC-0) for $n = 1$, with $t_{go} = R/V_c$.

γ_1	OPT	OGL/IAC-0, $n = 1$
-30°	856	1286
-60°	5875	10000
-90°	14452	27444

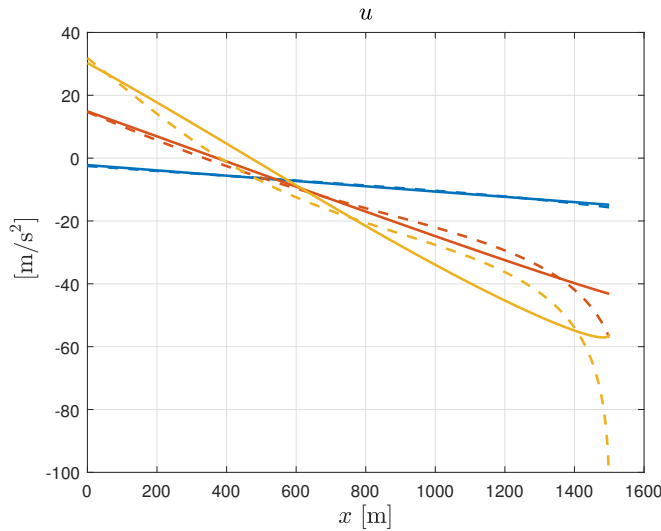
The last guidance law from the group of linear optimal guidance methods with which the optimal guidance (2.90) will be compared is the guidance law (1.80), presented in the work [44], which was derived using linear quadratic regulator theory (OGL-CTIA):

$$u = 6V\dot{\lambda} + 2\frac{V}{t_{go}}(\gamma - \gamma_1) \quad (2.109)$$

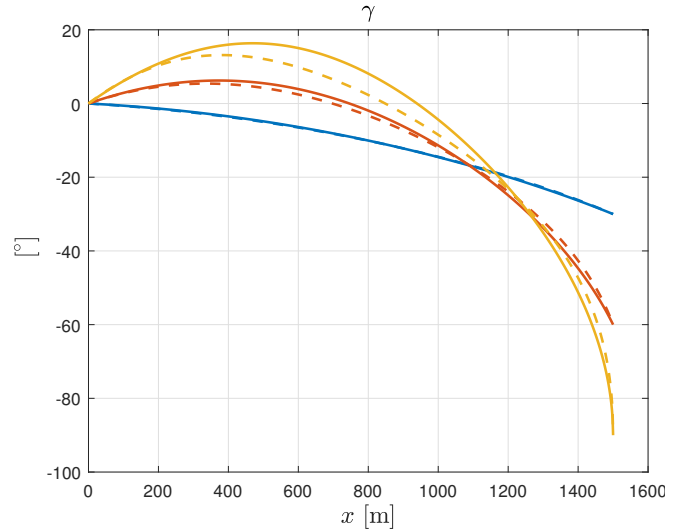
The remaining time to intercept the target was determined using the expression $t_{go} = R/V_c$. Figure 2.44 shows a comparison of the results with optimal guidance. The guidance law (2.109) is similar in form to the TSG guidance law, therefore Table 2.5 presents the values of the cost function obtained using three guidance laws: optimal (OPT), OGL-CTIA, and TSG.



(a) Trajectory.



(b) Control input.



(c) Flight-path angle.

Figure 2.44: Comparison of results obtained using Algorithm 2 (OPT) and expression (2.109) (OGL-CTIA), with $t_{go} = R/V_c$.

Table 2.5: Control effort obtained using Algorithm 2 (OPT), expression (2.109) (OGL-CTIA), and TSG guidance.

γ_1	OPT	OGL-CTIA	TSG
-30°	856	856	856
-60°	5875	5966	5922
-90°	14452	15467	15148

It has already been shown in [50, 51] that the proportional navigation guidance law:

$$u = kV\dot{\lambda} \quad (2.110)$$

with a constant k of the form:

$$k = \frac{\gamma_1 - \gamma_0}{\gamma_1 - \lambda_0} \quad (2.111)$$

can lead to target interception with a desired impact angle provided that $k > 2$. Although this guidance law is not optimal, due to the wide application of proportional navigation in real systems, its comparison with optimal guidance is presented below. The simulation parameters are given in Table 2.6.

Table 2.6: Simulation parameters for comparison of optimal guidance and proportional navigation.

Parameter	Value
x_0	0 m
y_0	0 m
γ_0	60°
x_1	5000 m
y_1	0
γ_1	$-30^\circ, -60^\circ, -90^\circ$
V	160 m/s

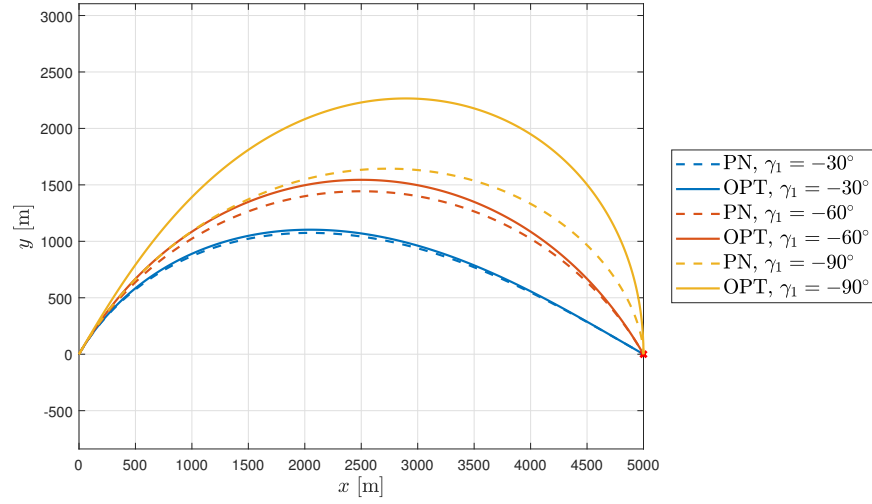
Figure 2.45 shows a comparison of the results obtained using Algorithm 2 (OPT) and proportional navigation (PN) given by expressions (2.110) and (2.111).

The values of the proportional navigation constant given by (2.111) for the considered cases are:

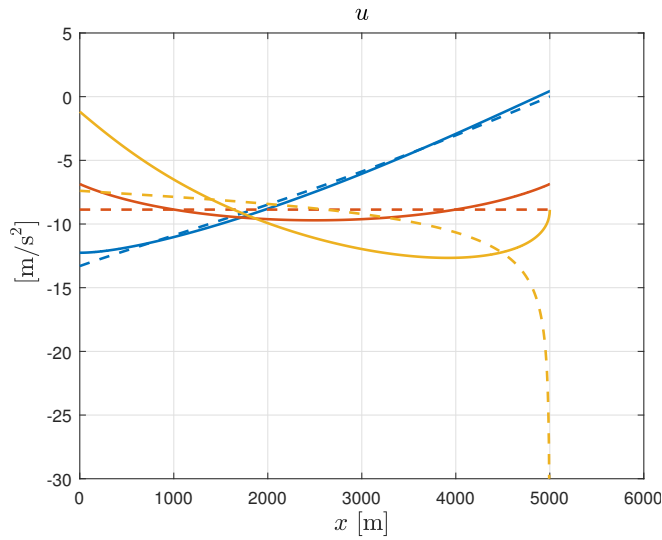
$$k = \begin{cases} 3, & \text{for } \gamma_1 = -30^\circ \\ 2, & \text{for } \gamma_1 = -60^\circ \\ 1.6667, & \text{for } \gamma_1 = -90^\circ \end{cases} \quad (2.112)$$

Although in the case $\gamma_1 = -90^\circ$ the proportional navigation constant does not satisfy $k > 2$, target interception with the desired impact angle is still achieved. Naturally, as with all linear guidance laws, approximate agreement with optimal results is achieved only for smaller impact angles. In

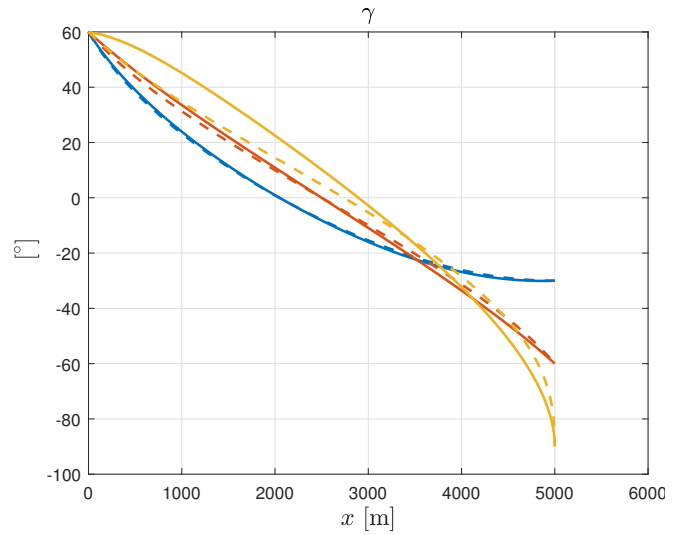
the case $\gamma_1 = -90^\circ$, proportional navigation leads to an increase in the control effort in the final moments of flight, resulting in large values of the cost function shown in Table 2.7.



(a) Trajectory.



(b) Control input.



(c) Flight-path angle.

Figure 2.45: Comparison of results obtained using Algorithm 2 (OPT) and proportional navigation (PN) given by expressions (2.110) and (2.111).

Table 2.7: Control effort obtained using Algorithm 2 (OPT) and proportional navigation (PN) given by expressions (2.110) and (2.111).

γ_1	OPT	PN
-30°	2352	2355
-60°	2945	2972
-90°	4358	5486

2.3.3 Comparison with Nonlinear Optimal Guidance Laws

The first guidance law from the group of nonlinear optimal guidance methods with which the optimal guidance proposed in this dissertation will be compared is the SDRE guidance presented in the work [52], given in the following form:

$$u = - \left(\frac{N}{t_{\text{go}}} \right)^2 R \sin(\gamma_1 - \lambda) + \frac{\sqrt{2}NV}{t_{\text{go}}} \sqrt{\frac{\sin(\gamma - \gamma_1)}{\gamma - \gamma_1}} (\gamma - \gamma_1) \quad (2.113)$$

where the time to intercept is determined as follows:

$$t_{\text{go}} = \frac{R}{V_{\text{ref}}} \quad (2.114)$$

$$V_{\text{ref}} = \begin{cases} V_c, & \text{for } V_c > V/2 \\ V/2, & \text{for } V_c \leq V/2 \end{cases} \quad (2.115)$$

Although in the work [52] no procedure is provided for selecting the optimal parameter N in the guidance law (2.113), the comparison with the optimal guidance, whose results are shown in Fig. 2.46, was performed for the value $N = 3$, as also done in the original paper [52]. The simulation parameters for the considered scenario are given in Table 2.1.

Unlike optimal guidance laws derived for a linearized model of the missile–stationary target engagement, the application of SDRE guidance yields a trajectory that deviates significantly from the optimal one even for small impact angles.

The achieved control effort values are presented in Table 2.8.

In the work [62], a nonlinear optimal guidance law (NOIACG) is proposed in the following form:

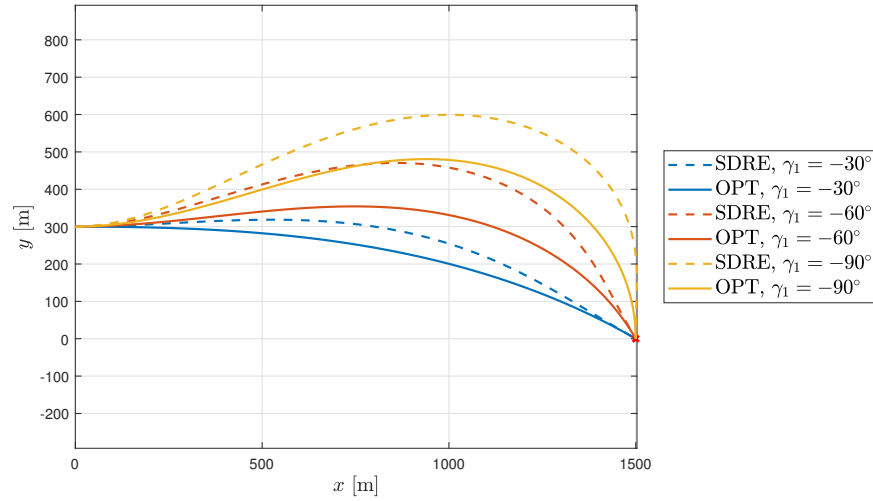
$$u = NV\dot{\lambda} + NMV \cos \eta \frac{-V\eta}{R} - \frac{M(N-1)V}{R/|\dot{R}|} (\gamma_1 - \gamma) \quad (2.116)$$

where:

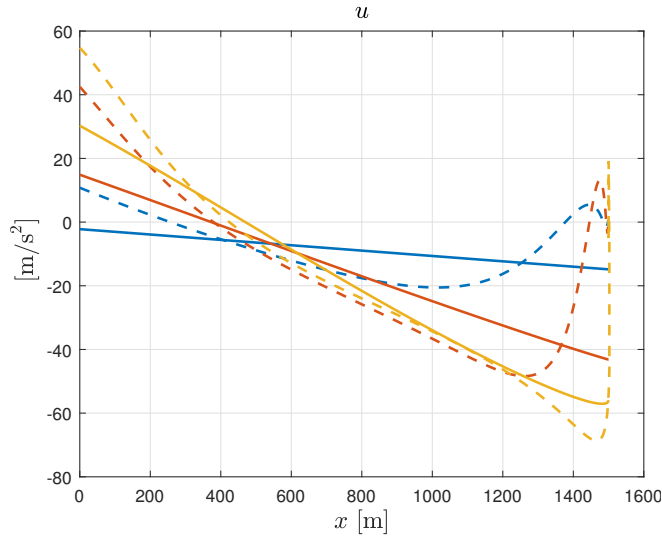
$$\eta = \gamma - \lambda, \quad |\dot{R}| = V \cos \eta \quad (2.117)$$

The parameters N and M in expression (2.116) are selected according to the desired convergence rates $\eta \rightarrow 0$ and $\gamma \rightarrow \gamma_1$.

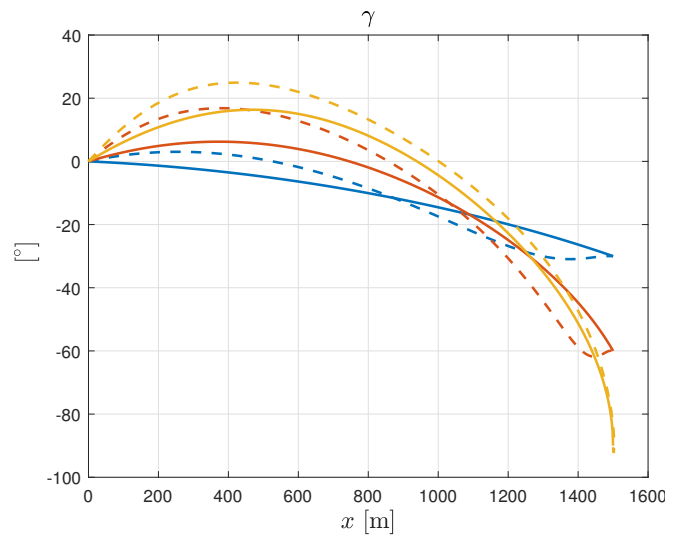
Before comparing the results with optimal guidance (Algorithm 2), an analysis was performed for results obtained using expression (2.116) for different values of parameters N and M . Figures 2.47, 2.48, and 2.49 show results for three impact-angle cases.



(a) Trajectory.



(b) Control input.

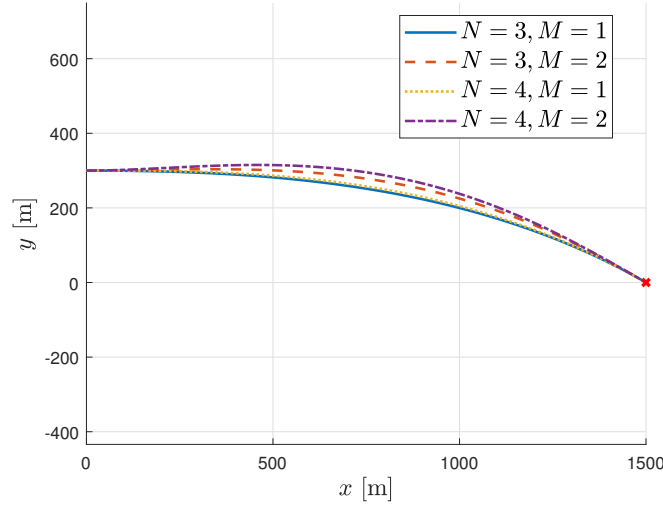


(c) Flight-path angle.

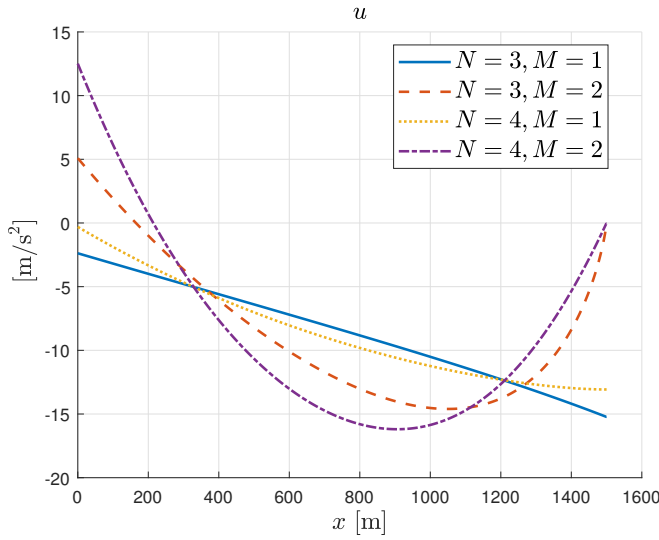
Figure 2.46: Comparison of results obtained using the Algorithm 2 (OPT) and SDRE guidance.**Table 2.8:** Control effort obtained using the Algorithm 2 (OPT) and SDRE guidance.

γ_1	OPT	SDRE
-30°	856	1592
-60°	5875	9256
-90°	14452	18337

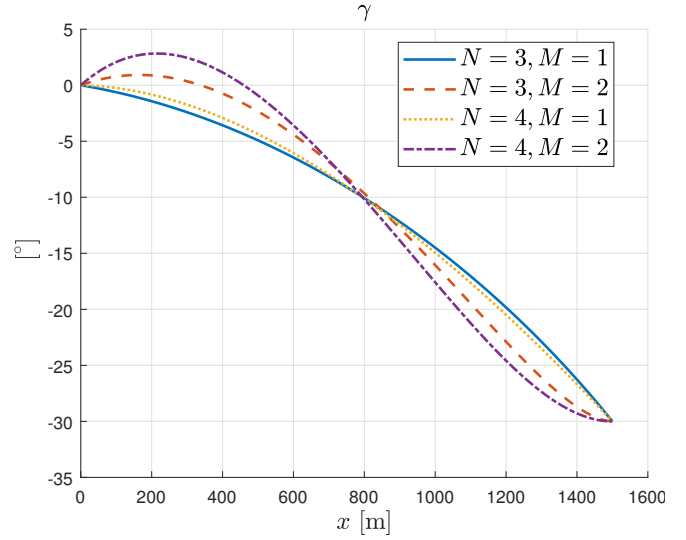
By increasing the parameter M , the missile aligns with the desired line-of-sight direction earlier, achieving lower normal acceleration in the final phase of flight. However, earlier trajectory shaping requires higher normal acceleration in the initial phase, as well as higher total control effort, as shown in Table 2.9.



(a) Trajectory.



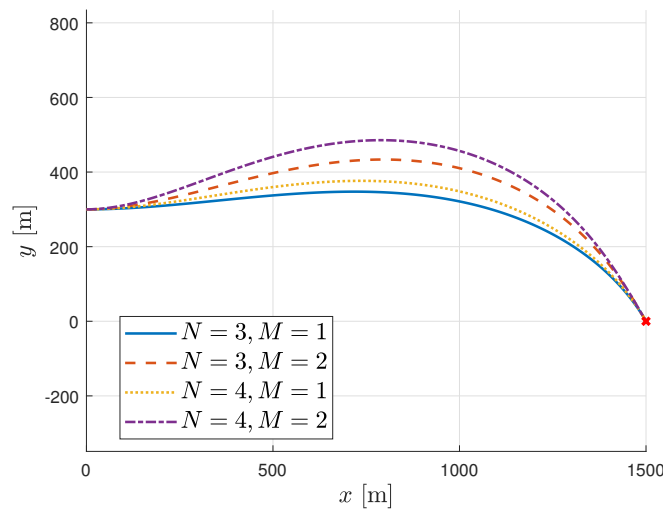
(b) Control input.



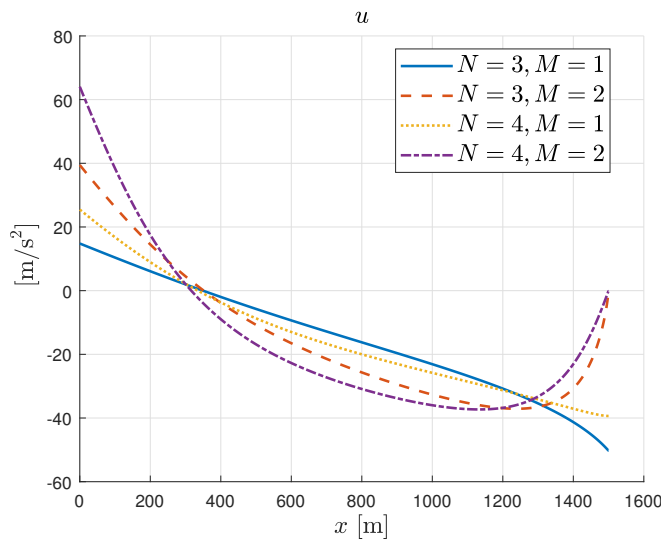
(c) Flight-path angle.

Figure 2.47: Results obtained using the guidance law NOIACG (2.116) for different values of parameters N and M , for the case $\gamma_1 = -30^\circ$.

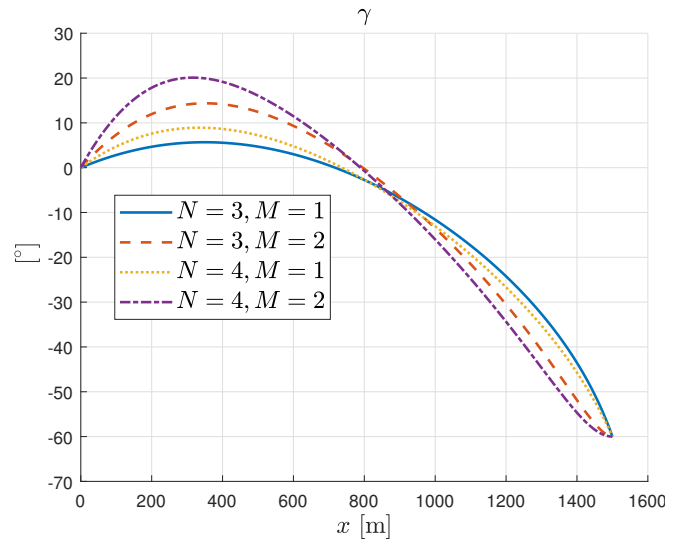
Although the trajectories obtained for parameter value $M = 2$ are favorable from the perspective of control distribution during flight, the most optimal results, as seen in Table 2.9, are achieved with the combination $N = 3$ and $M = 1$. Therefore, the comparison between optimal guidance (using Algorithm 2) and the guidance law given by expression (2.116) is performed exactly for these values and is shown in Fig. 2.50.



(a) Trajectory.

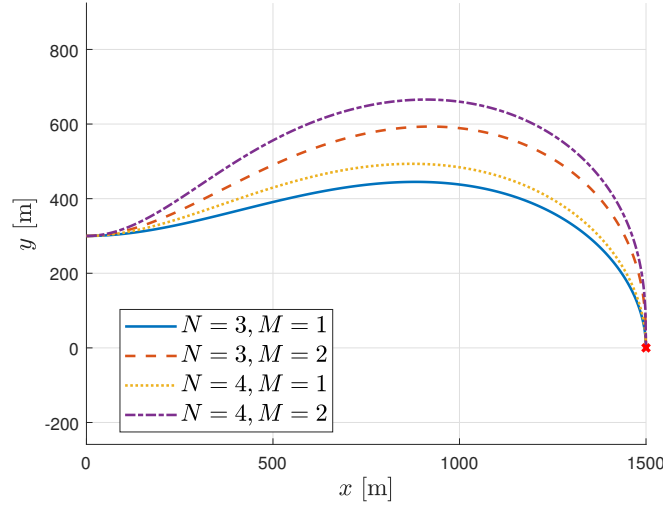


(b) Control input.

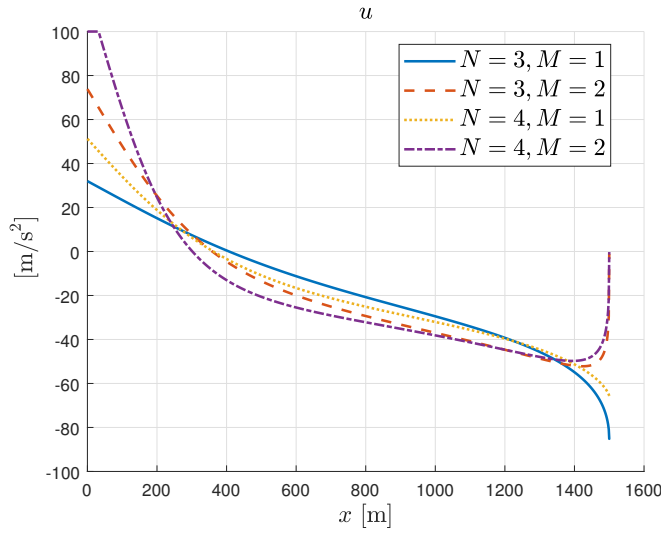


(c) Flight-path angle.

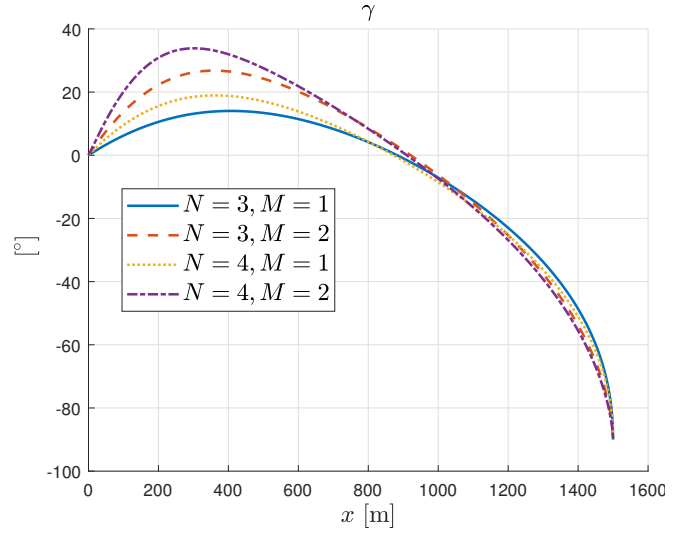
Figure 2.48: Results obtained using the guidance law NOIACG (2.116) for different values of parameters N and M , for the case $\gamma_1 = -60^\circ$.



(a) Trajectory.



(b) Control input.

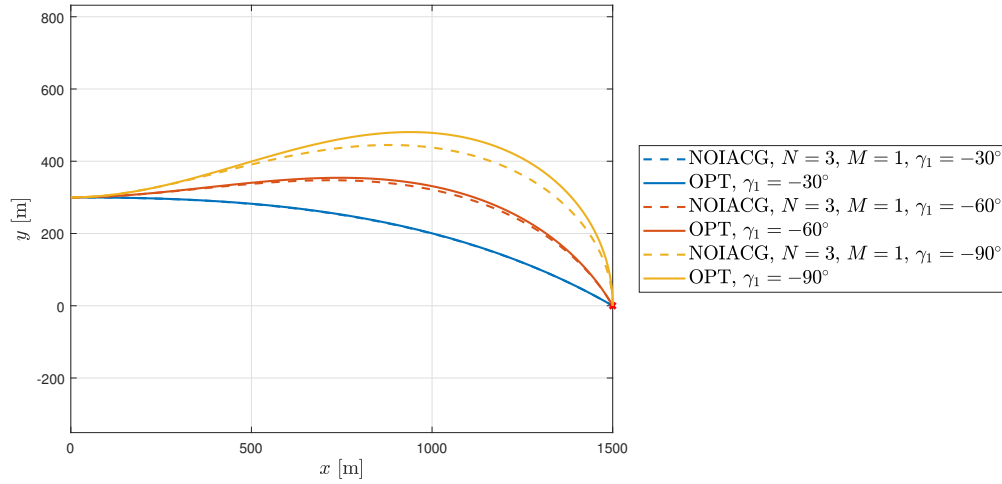


(c) Flight-path angle.

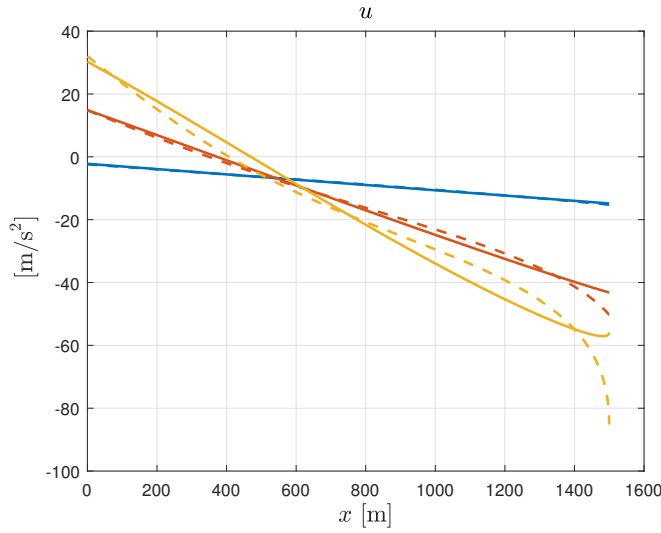
Figure 2.49: Results obtained using the guidance law NOIACG (2.116) for different values of parameters N and M , for the case $\gamma_1 = -90^\circ$.

Table 2.9: Control effort obtained using the Algorithm 2 (OPT) and the guidance law NOIACG (2.116) for different values of parameters N and M .

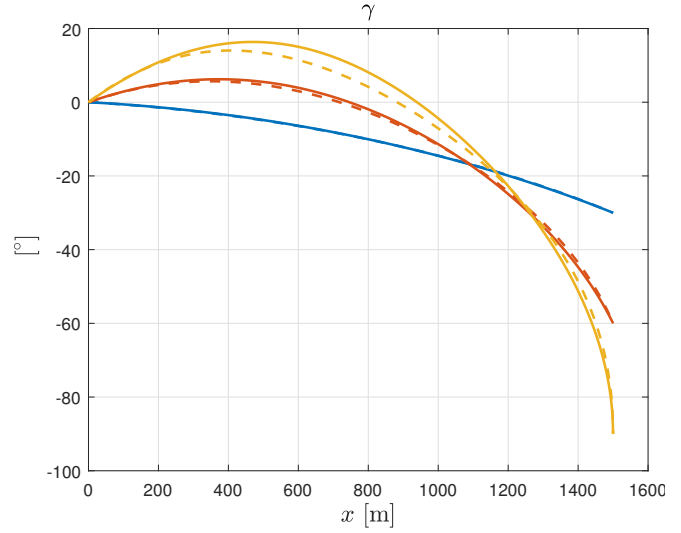
γ_1	OPT	$N = 3, M = 1$	$N = 3, M = 2$	$N = 4, M = 1$	$N = 4, M = 2$
-30°	856	856	1023	862	1255
-60°	5875	5904	7163	5985	8971
-90°	14452	14919	16583	14910	19820



(a) Trajectory.



(b) Control input.



(c) Flight-path angle.

Figure 2.50: Comparison of results obtained using the Algorithm 2 (OPT) and NOIACG guidance law (2.116) with $N = 3$ and $M = 1$.

2.3.4 Comparison with Non-Optimal Guidance Laws

The largest group of guidance laws for achieving a desired impact angle that are not derived using optimal control theory consists of guidance laws based on an extension of proportional navigation with an additional term (*Biased Proportional Navigation - BPN*), described in Section 1.2.7, which in the general case have the following form:

$$\dot{\gamma}(t) = k(t)\dot{\lambda}(t) + b(t) \quad (2.118)$$

Although these guidance laws are not optimal, due to their widespread use and simple implementation in real systems, their comparison with optimal guidance is presented in this section.

In the work [64] from 1998, the following form of the guidance law was proposed:

$$u = N(t)V(\dot{\lambda}(t) - b(t)) \quad (2.119)$$

$$b(t) = \frac{\eta V(\gamma_1 - \lambda(t))}{N(t)R(t)\cos(\gamma(t) - \lambda(t))} \quad (2.120)$$

where the time-varying gain is given by:

$$N(t) = N' + \frac{\rho + 0.5\alpha R(t)/V}{\cos(\gamma(t) - \lambda(t))} \quad (2.121)$$

and the guidance parameters, for the case of stationary target interception, have the following values:

$$\eta = 1.3, \quad \alpha = 0.28, \quad N' = 2.9, \quad \rho = 0 \quad (2.122)$$

Figure 2.51 shows a comparison between the results obtained using the guidance law (2.119) together with (2.120)–(2.122) and optimal guidance (2.90), i.e., Algorithm 2. It can be seen that trajectory shaping occurs later in the flight compared to the optimal trajectory. The later maneuver aimed at achieving the desired impact angle leads to higher control effort in the final moments of flight, which results in higher total control effort, as shown in Table 2.10.

In the work [65], a proportional navigation law with variable gain was proposed:

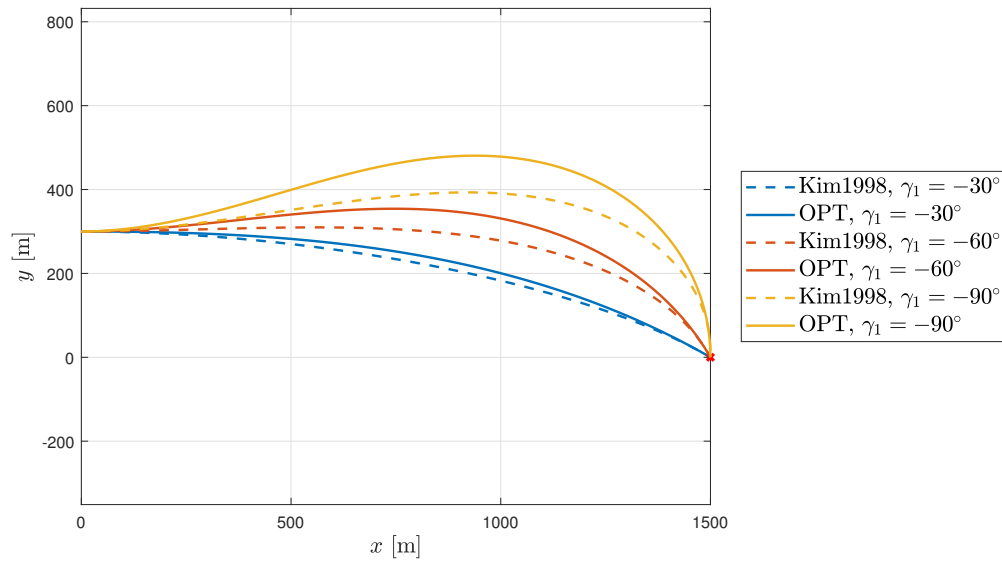
$$u = k(t)V\dot{\lambda} \quad (2.123)$$

where the gain is computed at each guidance cycle as follows:

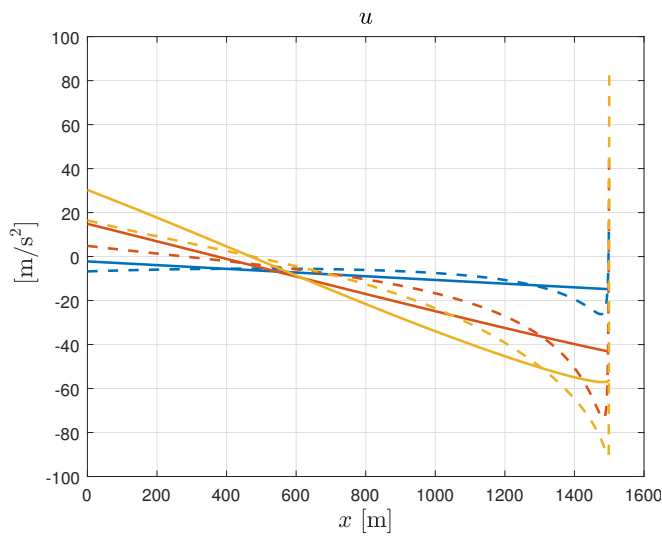
$$k(t) = \begin{cases} \frac{\gamma_1 - \gamma(t)}{\gamma_1 - \lambda(t)}, & \frac{\gamma_1 - \gamma(t)}{\gamma_1 - \lambda(t)} \geq 2 \\ \frac{\gamma(t)}{\pi/2 + \lambda(t)}, & \frac{\gamma_1 - \gamma(t)}{\gamma_1 - \lambda(t)} < 2 \end{cases} \quad (2.124)$$

Figure 2.52 shows a comparison with optimal guidance for a scenario of transition from cruise phase to terminal homing. It can be seen that using the guidance law (2.123)–(2.124), the missile remains in the cruise phase until the condition in (2.124) is violated. At that moment, a sudden change in the control input occurs, which becomes large in order to achieve the desired impact direction by the end of flight. This large control effort is maintained until impact, resulting in higher total control effort, as shown in Table 2.11.

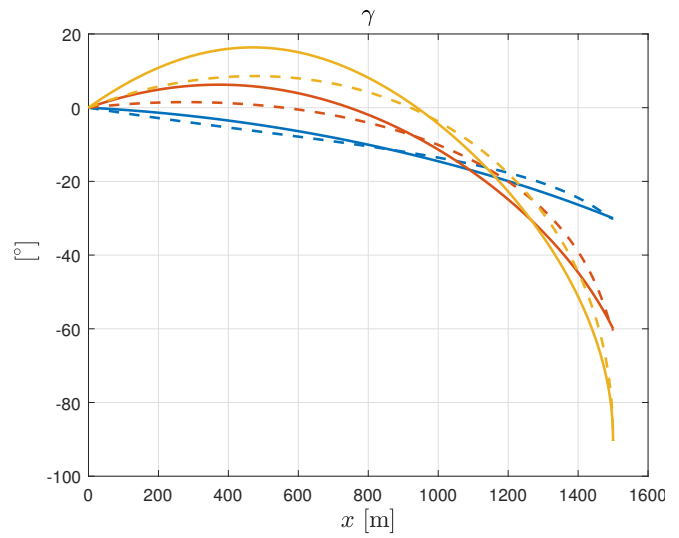
The comparison is also performed for a direct engagement scenario (without cruise phase), for which the simulation parameters are given in Table 2.12. The results are shown in Fig. 2.53. Except for the case $\gamma_1 = -30^\circ$, for which the trajectory obtained by the guidance law (2.123)–(2.124) is close to optimal, sudden changes in the control input occur during flight, leading to later trajectory shaping. As already noted, later trajectory shaping requires higher normal acceleration, which results in large values of the performance index, shown in Table 2.13.



(a) Trajectory.



(b) Control input.

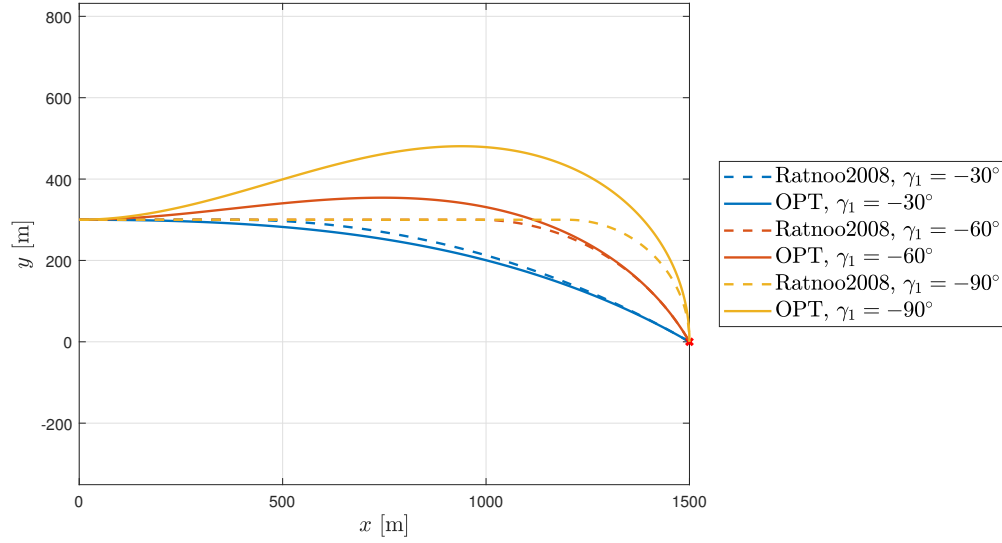


(c) Flight-path angle.

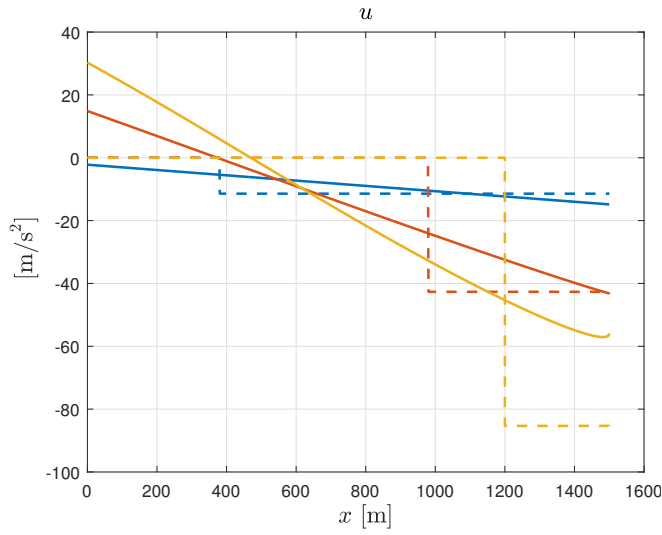
Figure 2.51: Comparison of results obtained using Algorithm 2 (OPT) and the guidance law proposed in [64] (Kim1998).

Table 2.10: Control effort obtained using Algorithm 2 (OPT) and the guidance law proposed in [64] (Kim1998).

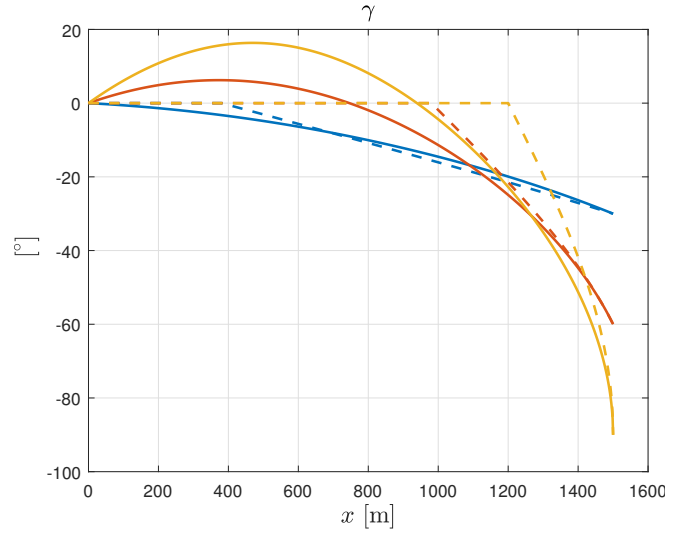
γ_1	OPT	Kim1998
-30°	856	974
-60°	5875	6776
-90°	14452	16215



(a) Trajectory.



(b) Control input.



(c) Flight-path angle.

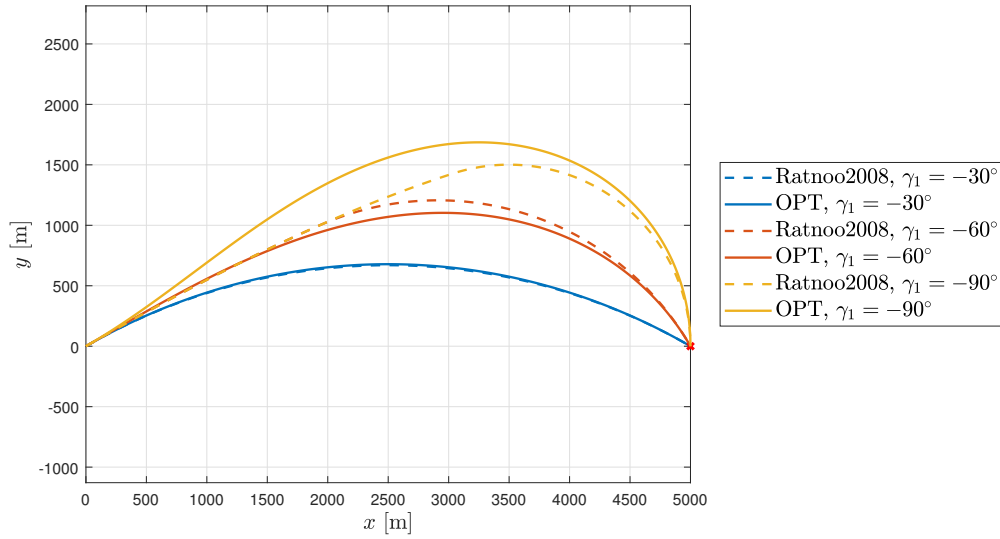
Figure 2.52: Comparison of results obtained using Algorithm 2 (OPT) and the guidance law proposed in [65] (Ratnoo2008), for the case of terminal guidance after cruise phase.

Table 2.11: Control effort obtained using Algorithm 2 (OPT) and the guidance law proposed in [65] (Ratnoo2008), for the case of terminal guidance after cruise phase.

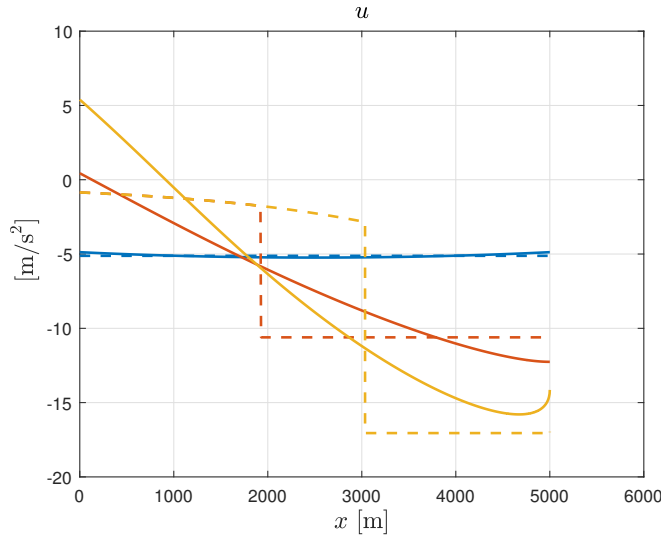
γ_1	OPT	Ratnoo2008
-30°	856	958
-60°	5875	7150
-90°	14452	21449

Table 2.12: Simulation parameters for comparison between optimal guidance and the guidance law proposed in [65].

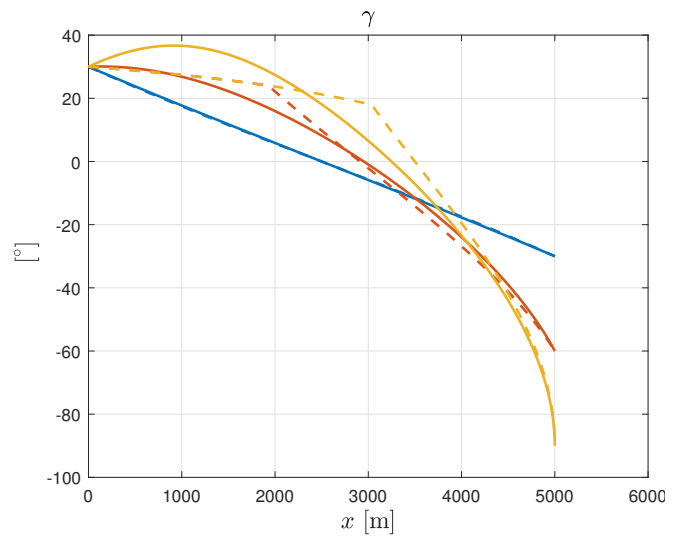
Parameter	Value
x_0	0 m
y_0	0 m
γ_0	30°
x_1	5000 m
y_1	0
γ_1	$-30^\circ, -60^\circ, -90^\circ$
V	160 m/s



(a) Trajectory.



(b) Control input.



(c) Flight-path angle.

Figure 2.53: Comparison of results obtained using Algorithm 2 (OPT) and the guidance law proposed in [65] (Ratnoo2008), for the case of direct engagement.

Table 2.13: Control effort obtained using Algorithm 2 (OPT) and the guidance law proposed in [65] (Ratnoo2008), for the case of direct engagement.

γ_1	OPT	Ratnoo2008
-30°	857	858
-60°	2352	2510
-90°	4657	5203

In the paper [67], for the guidance law of the form:

$$\dot{\gamma}(t) = k(t)\dot{\lambda}(t) + b(t) \quad (2.125)$$

a procedure for computing the bias b is proposed as follows:

$$b(t) = \begin{cases} 0, & \forall t < t_{\text{beg}} \\ \frac{k\lambda_0 - \gamma_0 - (k-1)\gamma_1}{t_{\text{end}} - t_{\text{beg}}}, & \forall t_{\text{beg}} \leq t < t_{\text{end}} \\ 0, & \forall t \geq t_{\text{end}} \end{cases} \quad (2.126)$$

where t_{beg} and t_{end} denote the initial and final instants during which the bias is active, respectively. Although these instants are arbitrary, the time interval during which the bias is active must be less than or equal to the total flight time in order to achieve interception of the target at the desired impact angle.

Figure 2.54 shows results obtained using the guidance law (2.125)-(2.126) with $t_{\text{beg}} = 0$ s, $t_{\text{end}} = 5$ s, and $k = 3$. Although in all cases interception at the desired impact angle is achieved, after the period in which the bias is active, discontinuities in the normal acceleration occur, which also lead to high values of the total control effort (Table 2.14).

Figure 2.55 shows results obtained using the same guidance law, where the initial time is set to $t_{\text{beg}} = 0$ s, while the final time t_{end} is chosen as the total flight time of the optimal trajectory (obtained using Algorithm 2).

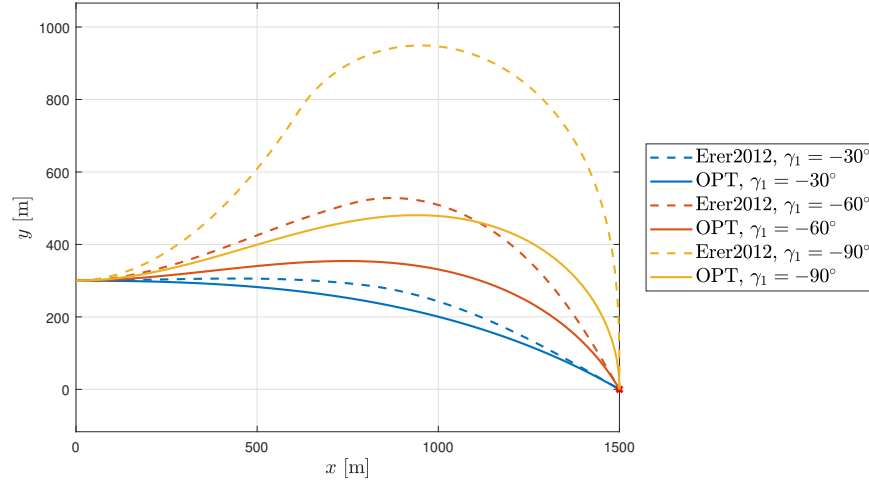
Table 2.15 presents the values of the total control effort, from which it can be seen that this approach yields a lower control effort value than the case with an arbitrarily chosen activation interval of the bias. Moreover, for the desired impact angles $\gamma_1 = -60^\circ$ and $\gamma_1 = -90^\circ$ the obtained cost function value is even lower than that achieved using Algorithm 2. Such a result could lead to the incorrect conclusion that optimal trajectories are achieved in this way. However, Table 2.15 also shows the errors in the achieved impact angle, which are computed as:

$$\Delta\gamma_1 = \gamma_1 - \gamma(t_1) \quad (2.127)$$

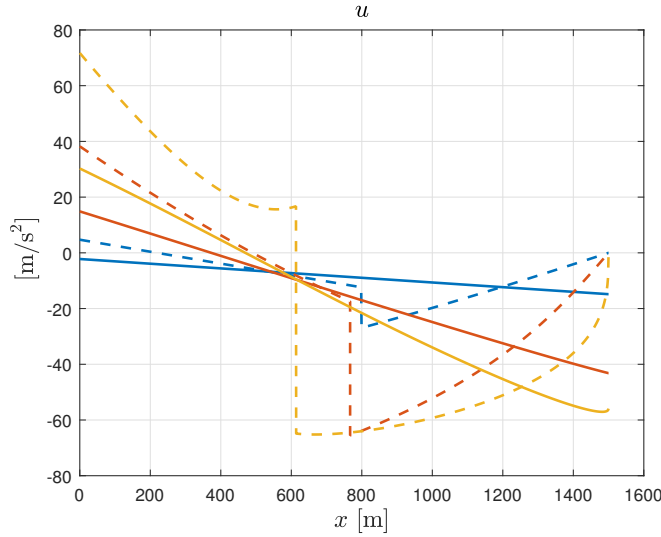
where $\gamma(t_1)$ denotes the value of the flight-path angle at the final simulation time.

Although interception is achieved in all cases, the errors in the achieved angle indicate that this approach does not satisfy all boundary conditions. The reason is that the guidance law (2.125),

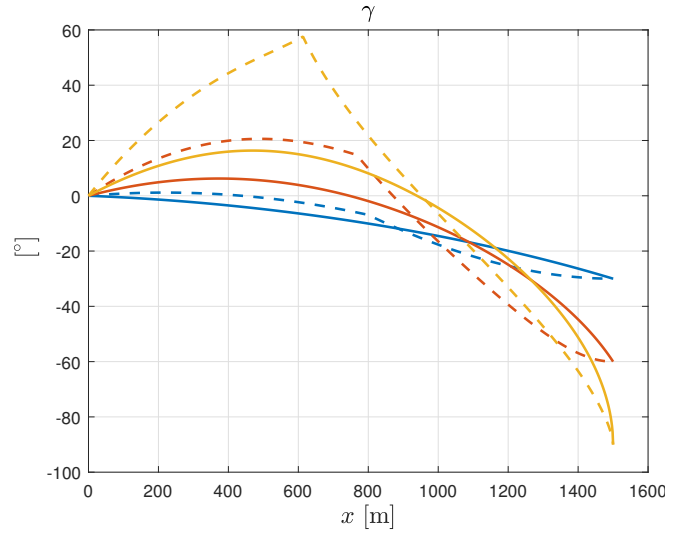
despite the fact that expression (2.126) for the bias is derived by integrating (2.125) under the boundary condition $\gamma(t_1) = \gamma_1$, is not optimal. As a consequence, the flight time on the optimal trajectory cannot be used as the expected flight time for a non-optimal trajectory. Therefore, the guidance law (2.125)–(2.126) should be used with activation intervals of the bias that are certainly shorter than the total flight time, in order to ensure satisfaction of all boundary conditions.



(a) Trajectory.



(b) Control input.

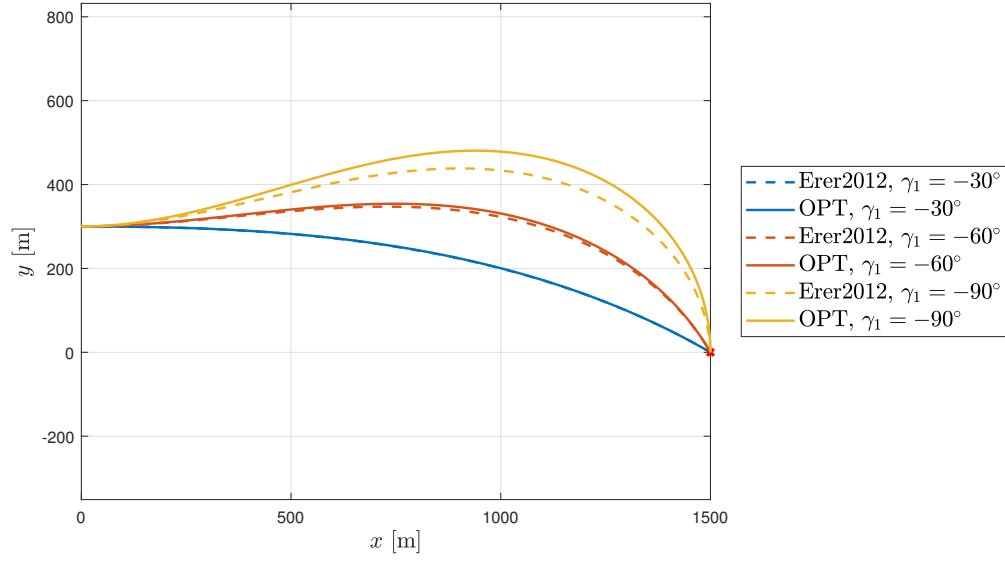


(c) Flight-path angle.

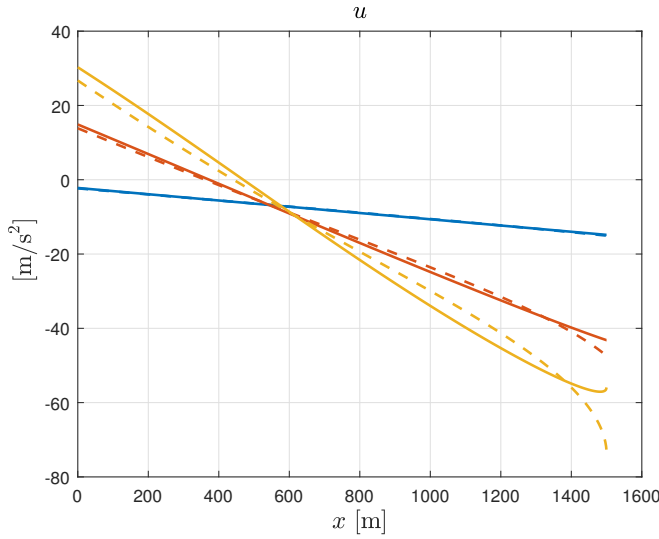
Figure 2.54: Comparison of results obtained using Algorithm 2 (OPT) and the guidance law proposed in [67] (Erer2012), for the case: $t_{\text{beg}} = 0$ s, $t_{\text{end}} = 5$ s, $k = 3$.

Table 2.14: Control effort obtained using Algorithm 2 (OPT) and the guidance law proposed in [67] (Erer2012), for the case: $t_{\text{beg}} = 0$ s, $t_{\text{end}} = 5$ s, $k = 3$.

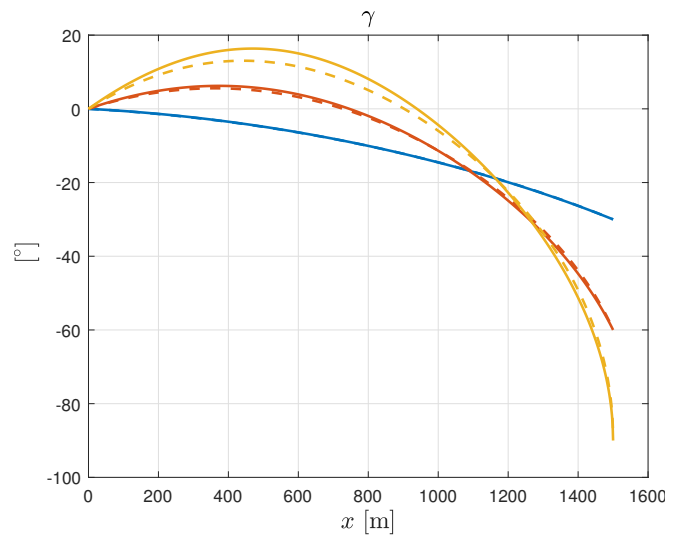
γ_1	OPT	Erer2012
-30°	856	1358
-60°	5875	11089
-90°	14452	27642



(a) Trajectory.



(b) Control input.



(c) Flight-path angle.

Figure 2.55: Comparison of results obtained using Algorithm 2 (OPT) and the guidance law proposed in [67] (Erer2012), for the case: $t_{\text{beg}} = 0$ s, and t_{end} equal to the flight time of the optimal trajectory.

Table 2.15: Control effort and impact angle error $\Delta\gamma_1$ obtained using Algorithm 2 (OPT) and the guidance law proposed in [67] (Erer2012), for $t_{\text{end}} = 5$ s, and for t_{end} equal to the flight time of the optimal trajectory. In both cases $t_{\text{beg}} = 0$ s was assumed.

γ_1	OPT		Erer2012, $t_{\text{end}} = 5$ s		Erer2012, $t_{\text{end}} = t_{1,\text{OPT}}$	
	J	$\Delta\gamma_1$	J	$\Delta\gamma_1$	J	$\Delta\gamma_1$
-30°	856	0.00019°	1358	0°	856	-0.00086°
-60°	5875	0.00065°	11089	-0.00006°	5854	-0.14342°
-90°	14452	0.00060°	27642	-0.00018°	14018	-2.05260°

As already described in Section 1.2.7, the problem with the guidance law given by Eqs. (2.125) and (2.126) is that it, in a certain sense, represents open-loop control. Therefore, in [68], for the same formulation of the guidance law:

$$\dot{\gamma}(t) = k(t)\dot{\lambda}(t) + b(t) \quad (2.128)$$

a time-varying bias active over the entire trajectory is proposed, which is updated at each guidance cycle:

$$b(t) = \frac{k\lambda(t) - \gamma(t) - (k-1)\gamma_1}{t_{\text{go}}} \quad (2.129)$$

where the time-to-go is defined as follows:

$$t_{\text{go}} = t_{\text{go, PN}} + t_{\text{go, B}} \quad (2.130a)$$

$$t_{\text{go, PN}} = \frac{R}{V} \left[1 + \frac{0.5}{2k-1}(\lambda - \gamma)^2 \right] \quad (2.130b)$$

$$t_{\text{go, B}} = \frac{R}{V} \frac{[(\gamma - \gamma_1) + k(\gamma_1 - \lambda)][3(\lambda - \gamma) + (\gamma - \gamma_1) + k(\gamma_1 - \lambda)]}{3(k+1)(2k-1)} \quad (2.130c)$$

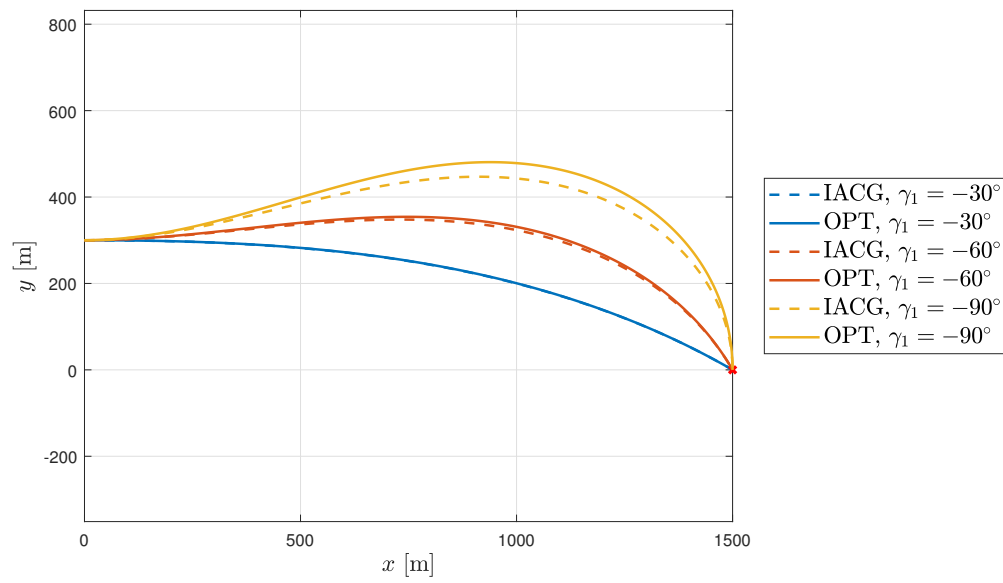
The comparison of results obtained using expressions (2.128), (2.129), and (2.130) with optimal guidance is shown in Fig. 2.56, while Table 2.16 presents the achieved control effort values.

Although the trajectory obtained in this way deviates from the optimal one, which becomes more pronounced for larger desired impact angles, the control effort does not differ significantly from that achieved in the optimal guidance case. Moreover, this guidance law achieves closed-loop control, since the bias is computed at each guidance cycle based on the current (rather than initial) values of the line-of-sight angle and flight-path angle.

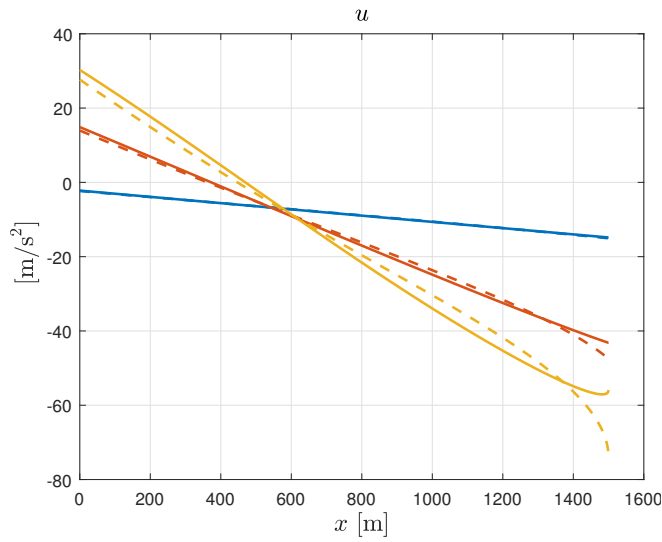
The drawback of this guidance law is that its implementation requires knowledge of the time-to-go. Similar to the analysis performed for the previous guidance law, whose results are shown in Fig. 2.55, the remaining time to intercept can be computed as:

$$t_{\text{go}} = t_{1,\text{OPT}} - t \quad (2.131)$$

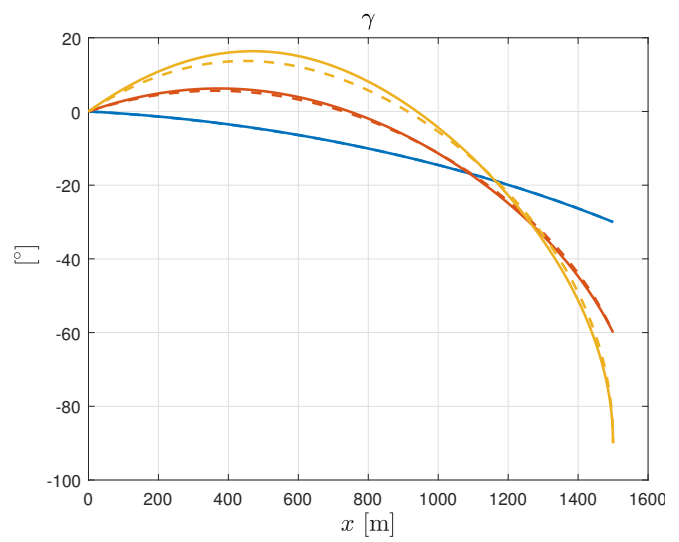
where $t_{1,\text{OPT}}$ is the total flight time of the optimal trajectory (obtained using Algorithm 2). However, for the same reasons discussed for the previous guidance law, this approach will yield lower control effort value but at the cost of increased error in the achieved impact angle.



(a) Trajectory.



(b) Control input.



(c) Flight-path angle.

Figure 2.56: Comparison of results obtained using Algorithm 2 (OPT) and the guidance law proposed in [68] (IACG).

Table 2.16: Control effort obtained using Algorithm 2 (OPT) and the guidance law proposed in [68] (IACG).

γ_1	OPT	IACG
-30°	856	856
-60°	5875	5888
-90°	14452	14673

2.3.5 Overview of Existing Methods Results

Table 2.17 summarizes the results obtained using all guidance laws considered in this chapter for comparison, for the scenario of missile transition from cruise phase to terminal guidance phase, with simulation parameters given in Table 2.1. The exception is the proportional navigation law with a constant k , which must satisfy inequality (1.82) in order for this method to be applicable for achieving the desired impact angles, and the guidance law proposed in [67], due to the arbitrary choice of the time interval during which the bias is active. Moreover, the guidance law proposed in [68] (IACG) represents a closed-loop version of the guidance law from [67], making it more suitable for implementation in real systems.

Guidance methods that require knowledge of the time-to-go, and for which procedures for its computation are provided in the corresponding papers, were used with their respective procedures (methods 3, 4, 7, 11), while for methods for which no such procedure was proposed (methods 2 and 6), the expression $t_{go} = R/V_c$ was used. The exception is method 5 (OGL/IAC-0, $n = 1$), where this expression was also used due to the limitations of the procedures given in [42] and [43].

It is noticeable that for all methods, in cases where the desired impact angles are $\gamma_1 = -30^\circ$ and $\gamma_1 = -60^\circ$, the target is hit with satisfactory accuracy in terms of the achieved impact angle. For the case $\gamma_1 = -90^\circ$, larger errors occur only when using the OGL/IAC-0 $n = 0$ method [42, 43], which is a consequence of the procedure used to compute the remaining time to intercept the target.

A similar observation holds for the achieved control effort values: for smaller impact angles, most guidance laws result in cost function values comparable to those of the optimal trajectory. This is expected, since many guidance laws are derived by linearizing the engagement geometry around the initial collision triangle, which is justified for the considered cruise-to-terminal guidance scenario only when the desired impact angles are small.

Table 2.18 provides an overview of the methods used for comparison with optimal guidance. In addition to the guidance laws, the last column indicates a parameter whose value is, in the general case, arbitrary. In this way, a certain degree of freedom is left in the synthesis of the guidance system, allowing influence over certain trajectory characteristics, although in most studies a fixed value is used in the analysis (e.g. $N = 3$ in proportional navigation-based guidance laws).

Not considering method 10 (Ratnoo2008 [65]) due to sudden changes in normal acceleration at the moment of gain k switching, and method 9 (Kim1998 [64]) due to the large number of parameters that must be selected, all methods (except method 1) require knowledge of the time-to-go (in method 8 (NOIACG [62]), although not explicitly appearing, the expression $R/(V \cos \eta)$ essentially represents t_{go}). In addition, implementation of all methods requires knowledge of the flight-path angle γ and the line-of-sight angle λ (or its rate of change $\dot{\lambda}$). Based on all the above, for optimal guidance (method 1), no additional quantities are required compared to those already determined by a navigation system for existing guidance laws.

Table 2.17: Results obtained using different methods for the scenario of missile transition from cruise phase to terminal guidance phase toward the target (Table 2.1). $\Delta x = x_1 - x(t_1)$, $\Delta y = y_1 - y(t_1)$, $\Delta\gamma_1 = \gamma_1 - \gamma(t_1)$.

Method	Fig.	$\gamma_1 = -30^\circ$				$\gamma_1 = -60^\circ$				$\gamma_1 = -90^\circ$			
		J	Δx	Δy	$\Delta\gamma_1$	J	Δx	Δy	$\Delta\gamma_1$	J	Δx	Δy	$\Delta\gamma_1$
1. OPT	-	856	-0.00523	0.00302	0.00019	5875	-0.00347	0.00615	0.00065	14452	-0.00007	0.00486	0.00060
2. TSG [23]	2.41	856	-0.01104	0.00637	0.00043	5922	-0.00315	0.00545	0.00065	15148	0.00000	0.00964	0.00175
3. OGL/IAC-0 [42], $n=0$, t_{go} method 2	2.42	861	-0.00669	0.00389	0.00404	6843	0.00101	0.00433	-0.05245	20921	-4.60010	0.01331	14.82200
4. OGL/IAC-0 [43], $n=0$, t_{go} method 3	2.42	860	-0.00284	0.00167	0.00397	6433	-0.00373	0.00589	-0.08539	19133	0.38417	0.00668	4.37190
5. OGL/IAC-0, [43], $n=1$, $t_{go} = R/V_c$	2.43	1286	-0.00418	0.00242	0.00005	10000	-0.00426	0.00738	0.00010	27444	0.00000	0.00460	-0.00010
6. OGL-CTIA [44]	2.44	856	-0.00479	0.00277	0.00018	5966	-0.00185	0.00320	0.00031	15467	-0.04612	0.01591	-0.30728
7. SDRE [52]	2.46	1592	-0.00419	0.00242	-0.00037	9256	-0.00080	0.00139	-0.00035	18337	0.00001	0.00937	0.00021
8. NOIACG [62] $N=3$, $M=1$	2.50	856	-0.00593	0.00342	0.00022	5904	-0.00733	0.01269	0.00156	14919	0.00000	0.01492	0.00264
9. Kim1998 [64]	2.51	974	-0.00320	0.00199	0.01863	6776	-0.00471	0.00884	0.04482	16215	0.00025	0.01453	0.01502
10. Ratnoo2008 [65]	2.52	958	-0.00405	0.00234	0.00022	7150	-0.00790	0.01369	0.00237	21449	0.00001	0.00679	0.00356
11. IACG [68]	2.56	856	-0.00585	0.00338	0.00022	5888	-0.00740	0.01283	0.00153	14673	0.00000	0.01022	0.00162

Table 2.18: Overview of methods used for comparison with optimal guidance.

Method	Eq.	Expression	Parameter
1. OPT	(2.26)	$u = (\eta_1 \Delta y - \eta_2 \Delta x - \eta_3) / (2V), \quad \eta_1 = \eta_1(\gamma), \quad \eta_2 = \eta_2(\gamma)$	—
2. TSG [23]	(1.64)	$u = 4V\dot{\lambda} + \frac{2V(\lambda - \gamma_1)}{t_{\text{go}}}$	—
3./4./5. OGL/IAC-0 [43]	(1.69)	$u = -\frac{V}{t_{\text{go}}}(-n_\lambda \lambda + n_\gamma \gamma + n_{\gamma_1} \gamma_1), \quad n_\lambda = n_\lambda(n), \quad n_\gamma = n_\gamma(n), \quad n_{\gamma_1} = n_{\gamma_1}(n)$	n
6. OGL-CTIA [44]	(1.80)	$u = 6V\dot{\lambda} + \frac{2V(\gamma - \gamma_1)}{t_{\text{go}}}$	—
7. SDRE [52]	(1.84)	$u = -\left(\frac{N}{t_{\text{go}}}\right)^2 R \sin(\gamma_1 - \lambda) + \frac{\sqrt{2}NV}{t_{\text{go}}} \sqrt{\frac{\sin(\gamma - \gamma_1)}{\gamma - \gamma_1}} (\gamma - \gamma_1)$	N
8. NOIACG [62]	(1.91)	$u = NV\dot{\lambda} + NMV \cos \eta \frac{-V\eta}{R} - \frac{M(N-1)V}{R/(V \cos \eta)} (\gamma_1 - \gamma), \quad \eta = \gamma - \lambda$	N, M
9. Kim1998 [64]	(1.95)	$u = N(t)V\dot{\lambda} - \frac{\eta V^2(\gamma_1 - \lambda(t))}{R(t) \cos(\gamma(t) - \lambda(t))}, \quad N(t) = N' + \frac{0.5\alpha R(t)/V}{\cos(\gamma(t) - \lambda(t))}$	η, α, N'
10. Ratnoo2008 [65]	(1.105)	$u = k(t)V\dot{\lambda}, \quad k(t) = k(\gamma_1, \gamma(t), \lambda(t))$	—
11. IACG [68]	(1.113)	$\dot{\gamma}(t) = k\dot{\lambda}(t) + b(t), \quad b(t) = \frac{k\lambda(t) - \gamma(t) - (k-1)\gamma_1}{t_{\text{go}}}$	k

Chapter 3

Application of Optimal Guidance Algorithms

3.1 Introduction

In Section 2.2.4, the optimal guidance algorithm (Algorithm 2) was presented, which, based on input data in the form of the current position (x_0, y_0) and the current flight-path angle γ_0 , determines the required control action (normal acceleration) in order to achieve a hit on the target (x_1, y_1) with the desired impact angle γ_1 , while minimizing control effort, as defined by the cost function (2.4). In that section, as well as in Section 2.3 on comparison with existing optimal guidance methods, the focus was on the analysis of the optimal trajectory obtained for a single set of initial and terminal conditions. In reality, due to the presence of disturbances and imperfect tracking of the commanded input, a precomputed optimal trajectory cannot be followed throughout the entire flight. Whenever deviations occur in any state variable relative to the values corresponding to the optimal trajectory, a new optimal trajectory must be recomputed and subsequently followed. Since the considered state variables (position and flight-path angle) are updated at the navigation system sampling rate, it is practical to update the nominal trajectory at the same rate. Moreover, since the full trajectory is typically not required, in each guidance cycle (which follows each navigation update cycle), only the initial value of the normal acceleration required to follow the nominal trajectory is determined.

This implies that, for application of Algorithm 2 in real systems, it is necessary to use the current values of position and flight-path angle as initial conditions ($x \rightarrow x_0, y \rightarrow y_0, \gamma \rightarrow \gamma_0$). In this way, closed-loop state feedback control is achieved, which enables further analysis of the applicability of the proposed algorithm in real time, which is the topic of this part of the work.

3.2 Real-Time Application Analysis

3.2.1 Numerical Simulation Setup

In this section, the application of Algorithm 2 in real time is analyzed using a numerical simulation of the planar kinematic model of the missile-target engagement described in Section 2.3.1. The difference compared to previously presented results is that in the expression (2.90) for the control input, the parameters η_1 and η_2 are computed using Algorithm 2 at each guidance cycle, based on the current state variables:

$$u_c(x, y, \gamma) = [\eta_1(x, y, \gamma)\Delta y - \eta_2(x, y, \gamma)\Delta x - \eta_3(x, y, \gamma)] / (2V) \quad (3.1)$$

The adopted guidance system sampling time is 16 ms, while all other simulation parameters remain unchanged.

To make the model closer to a real system in which imperfect command tracking is present, the control system is described by a first-order transfer function:

$$\frac{u(s)}{u_c(s)} = \frac{1}{\tau s + 1} \quad (3.2)$$

where u_c is the commanded normal acceleration, u is the achieved normal acceleration, and τ is the time constant of the control system.

Table 3.1 presents the simulation parameter values for the scenario analyzed in the following part of the work.

Table 3.1: Simulation parameters for real-time analysis.

Parameter	Value
x_0	0 m
y_0	300 m
γ_0	0°
x_1	1500 m
y_1	0
γ_1	−60°
V	160 m/s
τ	0.1 s

All results presented so far in Chapter 2 were obtained by executing Algorithm 2 once, in order to determine the parameters η_1 and η_2 prior to the flight simulation. For this purpose, the equation $\mathcal{N}(\alpha) = 0$ for the corresponding case (A, B, or C) must be solved numerically, which was done using the bisection method [95], described in Appendix B.1.1. Given that the function $\mathcal{N}(\alpha)$ has a single root in the considered interval, the bisection method is the most robust method for finding its zero, since convergence is guaranteed.

Figure 3.1 shows the results obtained by applying Algorithm 2 in real time, for the simulation parameters given in Table 3.1. The difference compared to previously presented results is that in the control input diagram (3.1c), both the achieved normal acceleration and the command are shown, since a single-lag autopilot is considered. Additionally, Figure 3.1b shows the number of iterations of the bisection method solver required to solve the equation $\mathcal{N}(\alpha) = 0$. The solver stopping criterion is reached when the interval length becomes smaller than 10^{-6} .

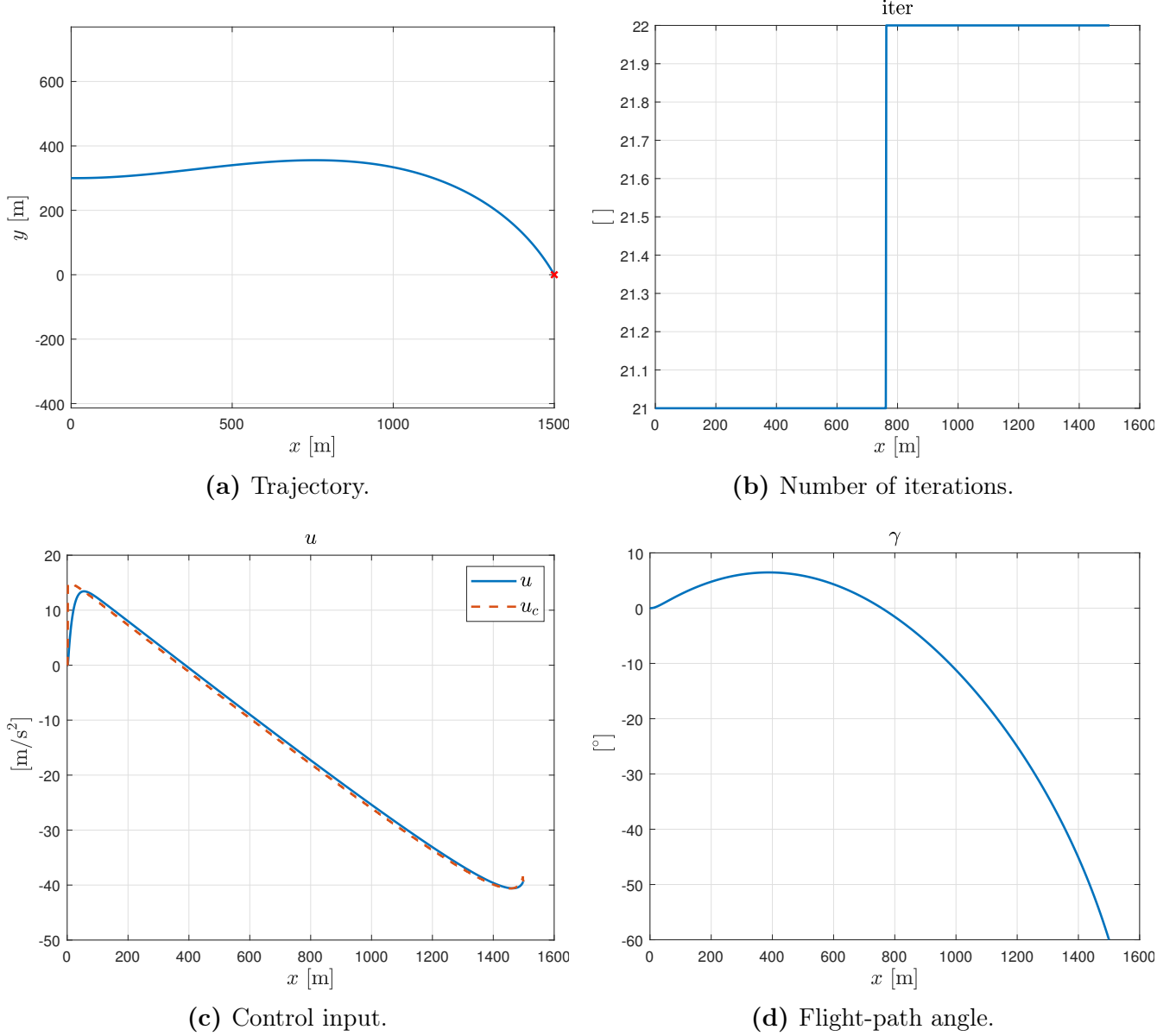


Figure 3.1: Results obtained by applying Algorithm 2 in real time. Numerical solution of $\mathcal{N}(\alpha) = 0$ using the bisection method, numerical integration using the trapezoidal rule.

The estimated number of iterations required to achieve an absolute error ϵ can be determined as follows [95]:

$$\frac{b-a}{2^n} < \epsilon \implies n > \log_2 \frac{b-a}{\epsilon} \quad (3.3)$$

where a and b denote the lower and upper bounds of the interval in which the root of $\mathcal{N}(\alpha)$ lies. Since these values are not known a priori and vary in each guidance cycle, and considering that from Eq. (2.50) α can lie in the interval from $-\pi/2$ to $\pi/2$, the estimated number of iterations required to achieve an absolute error of 10^{-6} is:

$$n > \log_2 \frac{\pi}{10^{-6}} \approx 21.5 \quad (3.4)$$

which is consistent with the results shown in Fig. 3.1b.

Since the required number of iterations depends on the interval length over which the root is searched (this length is at most π , so it will not be further discussed) and on the desired accuracy, the question arises as to what is the maximum allowable value of the absolute error of the solution of equation $\mathcal{N}(\alpha) = 0$, which is the topic of the following analysis.

3.2.2 Solution to the Nonlinear Equation

Table 3.2 summarizes the results obtained using the bisection method for absolute error values ranging from 10^{-1} to 10^{-6} . It can be observed that, for all considered absolute error values, the deviations at the final point are negligibly small, both in terms of coordinates and impact angle. Therefore, one might conclude that an absolute error level of 10^{-1} is acceptable. However, the control input profile differs significantly depending on the adopted accuracy level. Figure 3.2 shows the commanded normal acceleration for the case $\gamma_1 = -60^\circ$ for several absolute error values. It is evident that already at an error level of 10^{-2} , the control input deviates from the nominal one (obtained for $\epsilon = 10^{-6}$), while this effect becomes even more pronounced for larger error values.

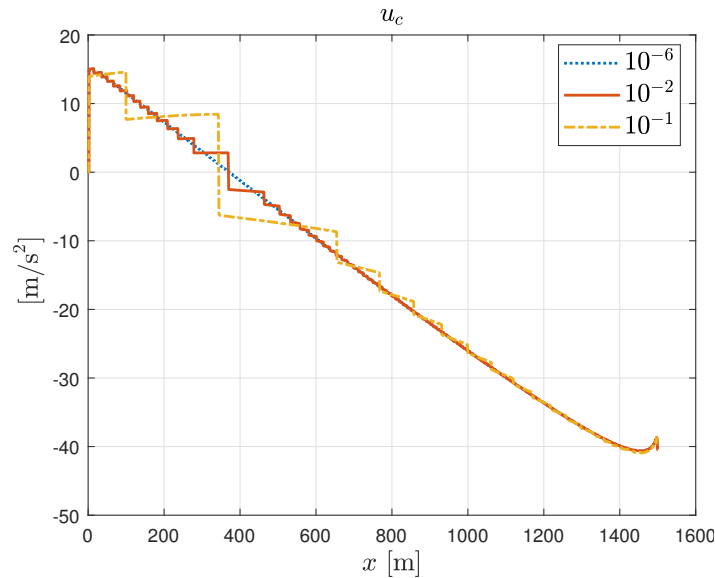


Figure 3.2: Control input for different values of the absolute error of the solution of the equation $\mathcal{N}(\alpha) = 0$, for the scenario described in Table 3.1, $\gamma_1 = -60^\circ$.

Table 3.2: Comparison of results for different values of the absolute error ε of the solution of the equation $\mathcal{N}(\alpha) = 0$.

ε	$\gamma_1 = -30^\circ$				$\gamma_1 = -60^\circ$				$\gamma_1 = -90^\circ$			
	J	Δx	Δy	$\Delta \gamma_1$	J	Δx	Δy	$\Delta \gamma_1$	J	Δx	Δy	$\Delta \gamma_1$
10^{-1}	857	-0.004	+0.002	-0.003	5926	-0.004	+0.006	-0.021	14573	-0.004	+0.004	-0.008
10^{-2}	856	-0.012	+0.007	-0.005	5891	-0.005	+0.007	-0.019	14525	-0.003	+0.012	+0.003
10^{-3}	856	-0.012	+0.007	-0.005	5890	-0.003	+0.003	-0.019	14525	-0.003	+0.015	+0.003
10^{-4}	856	-0.012	+0.007	-0.005	5890	-0.004	+0.006	-0.019	14525	-0.003	+0.003	+0.001
10^{-5}	856	-0.012	+0.007	-0.005	5890	-0.002	+0.003	-0.019	14525	-0.003	+0.004	+0.001
10^{-6}	856	-0.012	+0.007	-0.005	5890	-0.003	+0.004	-0.019	14524	-0.003	+0.002	+0.001

Therefore, determining an acceptable error value cannot be based solely on the final results of the numerical simulation, but must also take into account the behavior of the control input, since the total energy expenditure depends on it. Nevertheless, the control input profile depends on the specific scenario for which the optimal trajectory is computed. Consequently, a general criterion is required for estimating the allowable error of the solution of the equation $\mathcal{N}(\alpha) = 0$.

Expression (2.26) for the optimal control input can be written in the following form:

$$u(\alpha) = [\eta_1(\alpha)\Delta y - \eta_2(\alpha)\Delta x - \eta_3(\alpha)] / (2V) \quad (3.5)$$

where the dependence on the flight parameters x, y, γ has been omitted for simplicity.

If $\hat{\alpha}$ denotes an approximation of the solution α^* , such that

$$\hat{\alpha} = \alpha^* + \epsilon_\alpha \quad (3.6)$$

where ϵ_α is the solution error of the equation $\mathcal{N}(\alpha) = 0$, then, by performing a Taylor series expansion about the solution, the following expression is obtained:

$$u(\hat{\alpha}) \approx u(\alpha^*) + \left(\frac{\partial u}{\partial \alpha} \right)_{\alpha^*} (\hat{\alpha} - \alpha^*) \quad (3.7)$$

Neglecting higher-order terms yields:

$$|u(\hat{\alpha}) - u(\alpha^*)| = \left| \left(\frac{\partial u}{\partial \alpha} \right)_{\alpha^*} \right| \epsilon_\alpha \quad (3.8)$$

The left-hand side of Eq. (3.8) represents the control input error, which is bounded by the allowable value ϵ_u , leading to the following inequality:

$$\left| \left(\frac{\partial u}{\partial \alpha} \right)_{\alpha^*} \right| \epsilon_\alpha < \epsilon_u \quad (3.9)$$

This means that the upper bound of the absolute error of the solution of the equation $\mathcal{N}(\alpha) = 0$ must be limited by the allowable control input error:

$$\epsilon_\alpha < \frac{\epsilon_u}{|(\partial u / \partial \alpha)_{\alpha^*}|} \quad (3.10)$$

To determine the derivative $\partial u / \partial \alpha$, expression (3.5) can conveniently be rewritten in expanded form using expressions (2.17), (2.29), and (2.48):

$$\begin{aligned} u(\alpha) &= -\text{sgn}(\eta_3) \sqrt{V} \sqrt{\eta_1 \cos \gamma_1 + \eta_2 \sin \gamma_1} + (\eta_1 \Delta y - \eta_2 \Delta x) / (2V) = \\ &= -\text{sgn}(\eta_3) \frac{V^2}{|\Delta x|} \sqrt{I_N^2 \cos(\alpha - \gamma_1)} + \frac{V^2}{2(\Delta x)^2} I_N^2 (\cos \alpha \Delta y - \sin \alpha \Delta x) \end{aligned} \quad (3.11)$$

The required derivative has the following form:

$$\begin{aligned} \frac{\partial u}{\partial \alpha} &= -\text{sgn}(\eta_3) \frac{V^2}{|\Delta x|} \frac{1}{\sqrt{\cos(\alpha - \gamma_1)}} \left[\text{sgn}(I_N) \frac{\partial I_N}{\partial \alpha} \cos(\alpha - \gamma_1) - \frac{|I_N|}{2} \sin(\alpha - \gamma_1) \right] + \\ &+ \frac{V^2}{(\Delta x)^2} \left[I_N \frac{\partial I_N}{\partial \alpha} (\cos \alpha \Delta y - \sin \alpha \Delta x) - \frac{I_N^2}{2} (\sin \alpha \Delta y + \cos \alpha \Delta x) \right] \end{aligned} \quad (3.12)$$

When calculating the derivative $\partial u / \partial \alpha$, the corresponding expressions for I_N associated with the specific trajectory convexity case are used, as given in Section 2.2.4, while the expressions for the derivative $\partial I_N / \partial \alpha$ for all convexity cases are provided in Appendix B.1.2.

Figure 3.3a shows the allowable error ϵ_α determined using expressions (3.10) and (3.12) for two values of the control input error ϵ_u . Two characteristic regions can be identified in the figure, where the upper bound of the absolute error assumes small values. One corresponds to the final stage of flight and is not of interest for further analysis, since the error falls below 10^{-3} only when the distance to the target is less than 0.5 m, which is beyond the point where continuing guidance is meaningful, given the closing velocity. The second region, not clearly visible in Fig. 3.3a but evident in Fig. 3.3b, corresponds to the vicinity of the trajectory inflection point. In this region, due to the singularity appearing in the derivative $\partial u / \partial \alpha$, the upper bound of the absolute error tends to zero. It should be noted that the time spent in this region is short and that the minimum value of ϵ_α depends on the adopted value of ϵ_u . For example, for $\epsilon_u = 0.1$, the minimum value

of the upper bound of the solution error is approximately $\epsilon_\alpha \approx 10^{-6}$, whereas for $\epsilon_u = 0.01$, it is approximately $\epsilon_\alpha \approx 10^{-7}$.

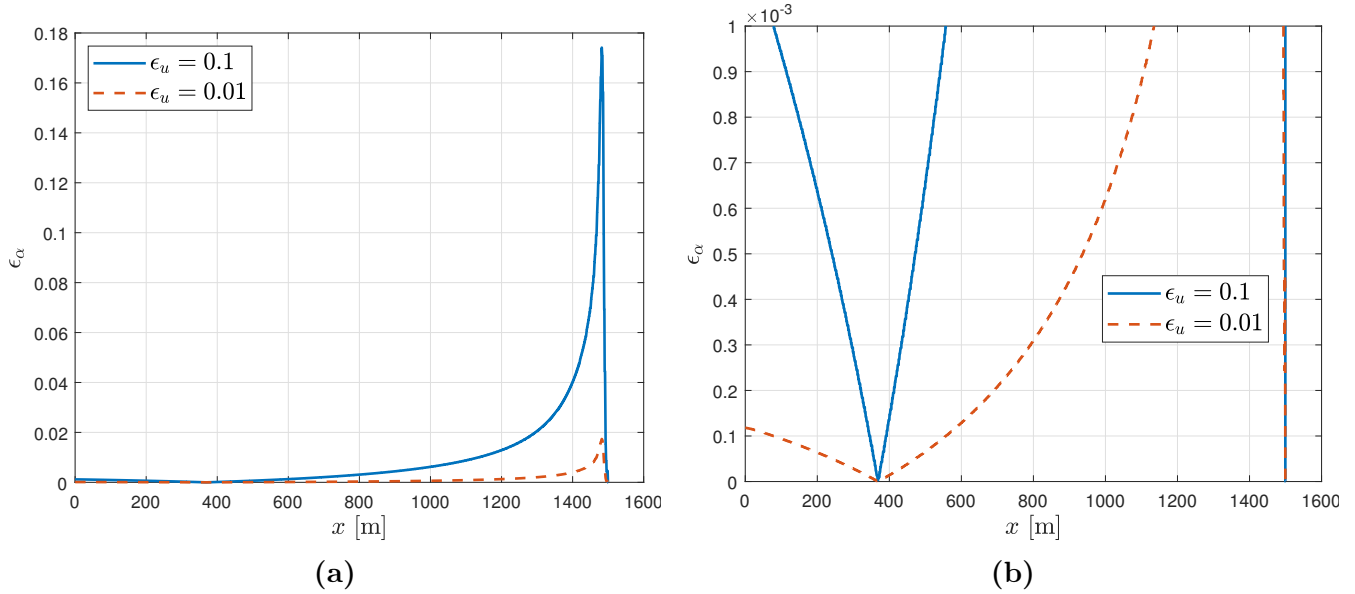


Figure 3.3: Upper bound of the absolute error ϵ_α for the adopted control input error ϵ_u , for the scenario described in Table 3.1. The figure on the right is a magnified view of the left figure.

The allowable control input error should be selected in accordance with the capabilities of the control system. Considering that the proposed algorithm is primarily intended for flight vehicle guidance, the question arises whether commanding normal acceleration with an accuracy of 0.01 m/s^2 is justified. However, regardless of the adopted value of the error ϵ_u for practical applications, it is evident that there exists a point on the trajectory at which the minimum value of the error ϵ_α is required. It is sufficient to demonstrate the existence of a single such trajectory (for one set of initial and terminal conditions) to conclude that the solution of the equation $\mathcal{N}(\alpha) = 0$ must sometimes be determined with maximum accuracy. Since the proposed algorithm is intended for implementation on computing systems with limited numerical precision, and with the goal of enabling real-time execution even on hardware platforms without a dedicated *Floating Point Unit (FPU)*, the solution of the equation $\mathcal{N}(\alpha) = 0$ will, throughout the remainder of this work, be sought with an absolute accuracy of 10^{-6} . This accuracy level approximately corresponds to machine precision when using the *single* floating-point format, making it realistic to expect 6–7 significant digits.

At first glance, this approach may appear unjustified. Namely, Fig. 3.3 shows that 10^{-6} is far below the allowable error value throughout most of the flight. The most rigorous approach would be to determine, in each guidance cycle, the allowable value of ϵ_α for the current cycle using expression (3.10), based on the desired control input accuracy ϵ_u . In this way, the number of iterations required to solve the equation $\mathcal{N}(\alpha) = 0$ would vary from cycle to cycle and would, for most of the flight, be lower than the estimated 22 iterations required to achieve an accuracy of 10^{-6} . However, the drawback of such an approach is that estimating the error requires the value of the derivative $\partial u / \partial \alpha$ at the solution point, which is precisely the quantity being sought. One possible way to address this issue is based on the assumption that the solution in the k -th guidance

cycle, $\alpha(k)$, is close to the solution $\alpha(k-1)$ from the previous cycle, allowing the derivative to be evaluated at the previous solution instead of the current one. Unfortunately, this provides only an approximation of the actual error estimate and may lead to significant inaccuracies at events such as the trajectory inflection point, where the expressions for $I_N(\alpha)$ and $\partial I_N/\partial\alpha$ change form due to a change in trajectory convexity. Another disadvantage of this approach is that, in each guidance cycle, additional computational effort is required to evaluate the derivative (3.12), which consists of functions that are computationally expensive from the standpoint of execution time.

On the other hand, solving the equation $\mathcal{N}(\alpha) = 0$ with a fixed absolute error of 10^{-6} guarantees, at least when using the bisection method, a fixed execution time of the algorithm. This is advantageous during the testing phase because the execution time is always at its maximum value, providing an opportunity to verify compatibility with existing systems in which the flight computer performs numerous functions in addition to vehicle guidance.

Nevertheless, for some real-time algorithms, even approximately 20 iterations may be unacceptable from the standpoint of computational time. For this reason, alternative methods for solving the equation $\mathcal{N}(\alpha) = 0$ were also investigated. The final decision regarding the adopted method will be made after testing on real hardware, as described in Section 3.3.

In the 1979 paper [96], Ridders proposed an iterative method for finding the root of a real-valued function of a single variable. The method is based on linearizing the original function and applying the *regula falsi* method to the resulting linearized function. Although the theoretical convergence of the method is quadratic, two evaluations of the function whose root is sought are required in each iteration. Consequently, the effective order of the method is $\sqrt{2}$ rather than 2, which is nevertheless superlinear [91].

During the 1960s, Dekker and his colleagues developed a method that combines the reliability of the bisection method with the speed of higher-order methods. This method was later improved by Brent [97] and is therefore commonly referred to as Brent's method. During the root-finding process, the method monitors whether superlinear convergence is being achieved. If not, a bisection step is performed to ensure at least linear convergence [91].

When the function whose root is sought is differentiable in the vicinity of the solution, Newton's method (the tangent method) is particularly suitable. This method has a convergence order of at least two and is one of the oldest known methods for solving nonlinear equations of a single variable [95]. For this purpose, it is necessary to determine the derivative of the function $\mathcal{N}(\alpha)$, which is described in Appendix B.1.2. It should be noted that, due to the weaker global convergence properties of Newton's method when applied to arbitrary functions, and in order to increase robustness, the original Newton method should be modified such that a bisection step is performed whenever the derivative-based update produces a point outside the interval containing the root or when the method does not converge at the expected rate.

Figure 3.4 compares the number of iterations required to find the root of the function $\mathcal{N}(\alpha)$ using the aforementioned methods for solving a nonlinear equation of a single variable, for the scenario described in Table 3.1 and for two impact angles: $\gamma_1 = -60^\circ$ and $\gamma_1 = -10^\circ$. These two values were selected to ensure that all trajectory convexity cases are evaluated in real time. For all methods, the root-finding criterion is an absolute accuracy of 10^{-6} , while the maximum number of iterations

is limited to 22, corresponding to the maximum number of iterations required by the bisection method.

As expected, all methods converge to the solution with fewer iterations than the bisection method. In both cases, Ridders' method requires fewer iterations than Brent's method throughout most of the flight. This indicates that this relatively simple¹ procedure is comparable to more complex methods in terms of both reliability and speed [91].

What may initially appear surprising is the number of iterations achieved by Newton's method. Although this method is expected to exhibit the fastest convergence, it should be noted that the function $\mathcal{N}(\alpha)$ does not satisfy all conditions under which convergence of the tangent method is guaranteed. In particular, this refers to the changing sign of the first and second derivatives of the function [95]. As a result, the interval-halving procedure often has to be used as a safeguard within Newton's method, causing the number of iterations at certain stages of flight to become identical to that of the bisection method.

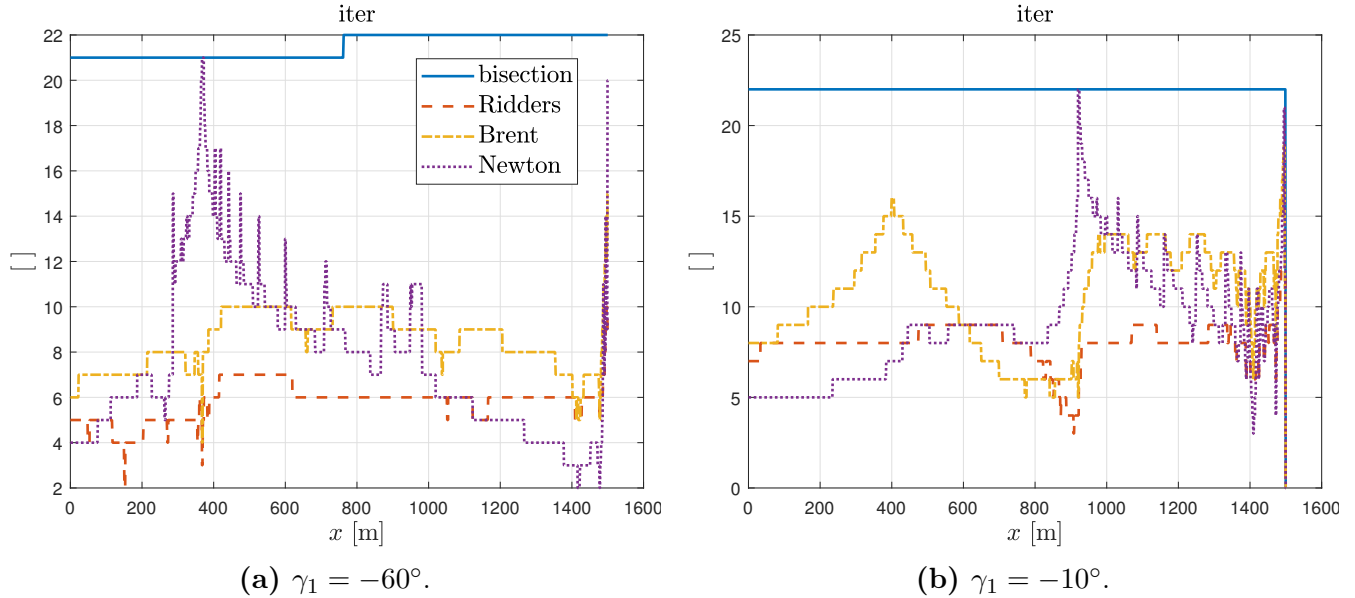


Figure 3.4: Comparison of the number of iterations required to achieve an absolute accuracy of $\epsilon = 10^{-6}$ using different methods for solving the equation $\mathcal{N}(\alpha) = 0$.

Although the interval in which the root of the function $\mathcal{N}(\alpha)$ is sought is defined by Eq. (2.50), the number of iterations required by all of the aforementioned methods can be further reduced if it is possible to estimate a subinterval in which the intersection of $\mathcal{N}(\alpha)$ with the horizontal axis is expected to occur. One approach is to search for the current solution of the equation $\mathcal{N}(\alpha) = 0$ in the vicinity of the previous solution. Here, the current solution refers to the solution obtained in the current guidance cycle, while the previous solution refers to the solution obtained in the preceding guidance cycle. The only exception is the first guidance cycle, for which no information about a previous solution is available. In that case, the root must be searched over the entire interval defined by (2.50).

¹Compared to Dekker's and Brent's methods.

The number of iterations required when using such a subinterval-selection approach depends on how the *vicinity* of the previous solution is defined. The narrower the vicinity, the fewer iterations are required, and vice versa. If the solution from the previous cycle is denoted by $\alpha(k-1)$ and the vicinity of the previous solution by Δ_b , then the subinterval bounds in the k -th guidance cycle are determined as follows:

$$\begin{aligned} l'_b(k) &= \alpha(k-1) - \Delta_b \\ u'_b(k) &= \alpha(k-1) + \Delta_b \end{aligned} \quad (3.13)$$

However, since the interval defined by (2.50) also changes in each cycle, there is no guarantee that the vicinity of the previous solution will actually constitute its subinterval. Figure 3.5 illustrates four possible cases in which this condition is not satisfied.

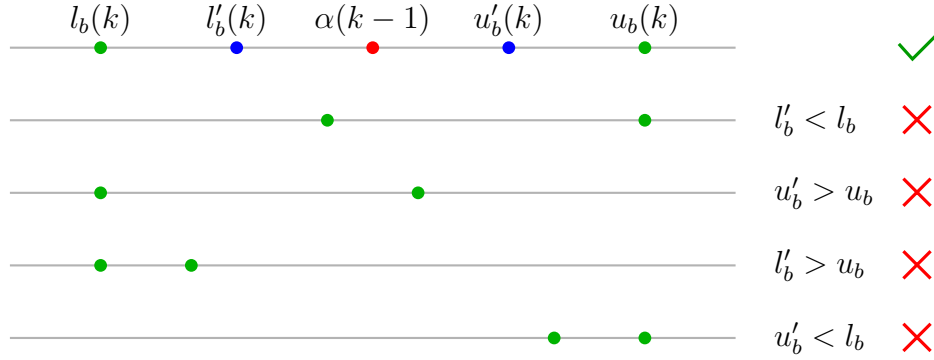


Figure 3.5: Correct subinterval selection and potential errors in its determination.

Therefore, the subinterval bounds must satisfy the following conditions:

$$\begin{aligned} l_b &\leq l'_b < u_b \\ l_b &< u'_b \leq u_b \end{aligned} \quad (3.14)$$

Even with subinterval bounds determined in this manner, there is no guarantee that the solution in the current cycle, $\alpha(k)$, will lie within the selected subinterval. Therefore, it is always necessary to verify whether the function $\mathcal{N}(\alpha)$ crosses the horizontal axis within the considered subinterval:

$$\mathcal{N}(l'_b, s) \mathcal{N}(u'_b, s) < 0, \quad s \in \{A, B, C\} \quad (3.15)$$

If this condition is not satisfied, the root must be searched for in the original interval valid for that cycle, defined by the bounds $l_b(k)$ and $u_b(k)$.

Figure 3.6a shows the solution of the equation $\mathcal{N}(\alpha) = 0$ for the scenario described in Table 3.1. It can be observed that the variation of the solution is small throughout most of the flight, which is even more apparent in the solution variation plot shown in Fig. 3.6b. Except during the final phase of flight, the change in the solution remains below 0.005, and this value can therefore be adopted as the neighborhood of the solution.

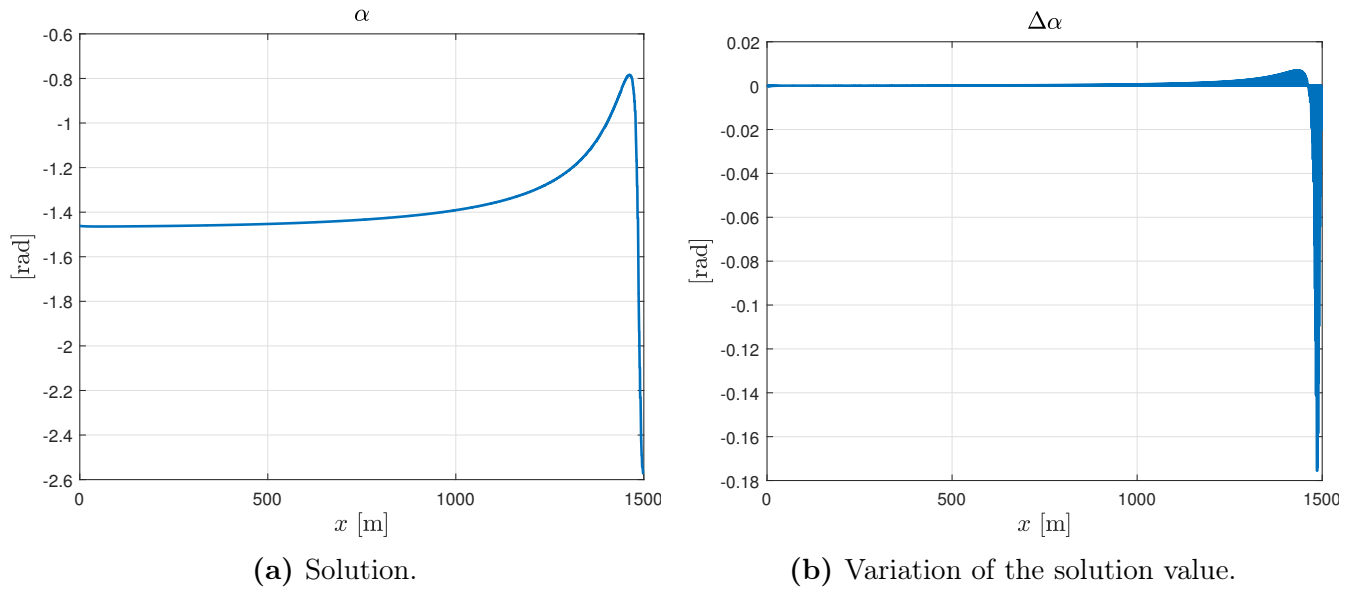


Figure 3.6: Variation of the solution of the equation $\mathcal{N}(\alpha) = 0$ during the numerical simulation for the case $\gamma_1 = -60^\circ$.

Figure 3.7 compares the number of iterations required by different methods when using an adaptive interval with $\Delta_b = 0.005$, for two impact angles: $\gamma_1 = -60^\circ$ and $\gamma_1 = -10^\circ$.

Although the root-finding criterion remains unchanged (absolute accuracy of 10^{-6}), comparison with the results shown in Fig. 3.4 reveals a significant reduction in the number of required iterations. As expected, the bisection method still requires the largest number of iterations, although now fewer than 15 during most of the flight, while for all other methods the number of iterations is below 5 throughout most of the flight.

Figure 3.7 also reveals two regions in which the number of iterations remains relatively high. One region corresponds to the final phase of flight, where, according to Fig. 3.6b, the variation of the solution exceeds the adopted value $\Delta_b = 0.005$. In this case, the solution must again be searched for in the original interval, resulting in a larger number of required iterations.

The second region, visible at approximately 400 m downrange for the case $\gamma_1 = -60^\circ$ and at approximately 900 m downrange for $\gamma_1 = -10^\circ$, corresponds to the portion of flight where the trajectory convexity changes. Figure 3.8 shows the functions $\mathcal{N}(\alpha)$ in the vicinity of the inflection point, which for the case $\gamma_1 = -60^\circ$ occurs in the downrange interval between 300 m and 400 m. Chronologically, the curves $\mathcal{N}(\alpha)$ progress from left to right. That is, they initially have the shape corresponding to case B, shown in red, and subsequently the shape corresponding to case A, shown in blue. It is noticeable that in this region the solution lies very close to the lower interval bound, particularly at instants immediately adjacent to the inflection point. Such cases, in which the solution of the equation is located near one of the interval endpoints, represent a challenging problem regardless of the root-finding procedure employed [91]. Consequently, an increased number of required iterations in these regions is to be expected.

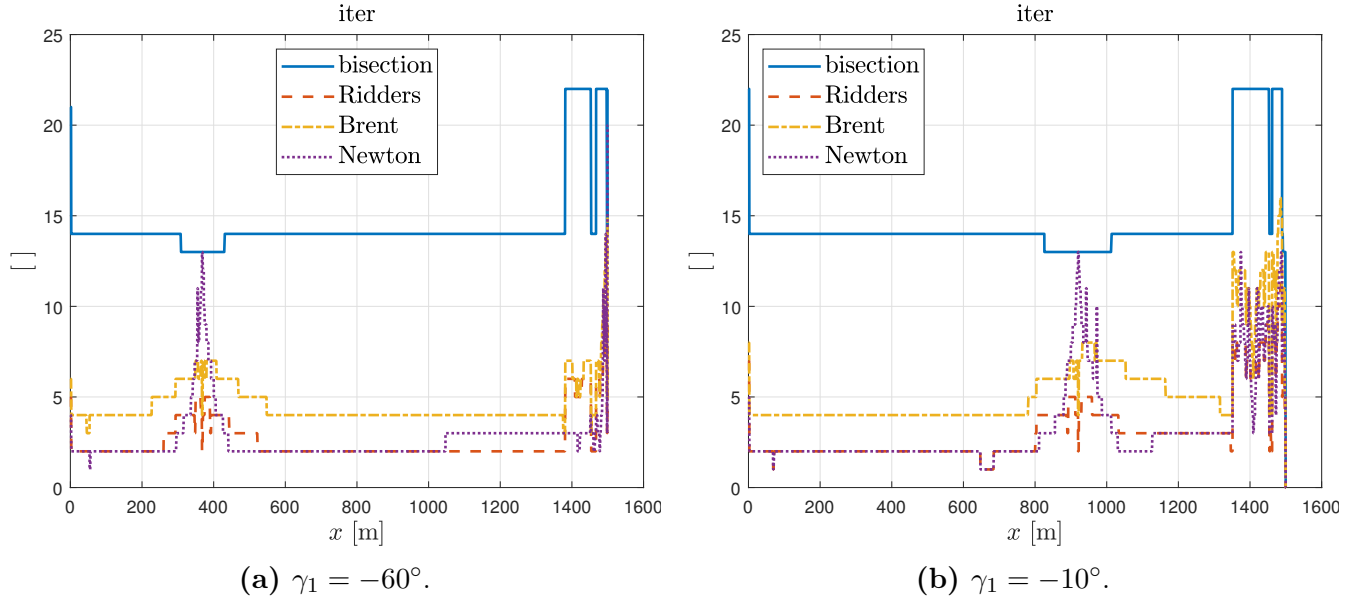


Figure 3.7: Comparison of the number of iterations required by different methods using an adaptive interval.

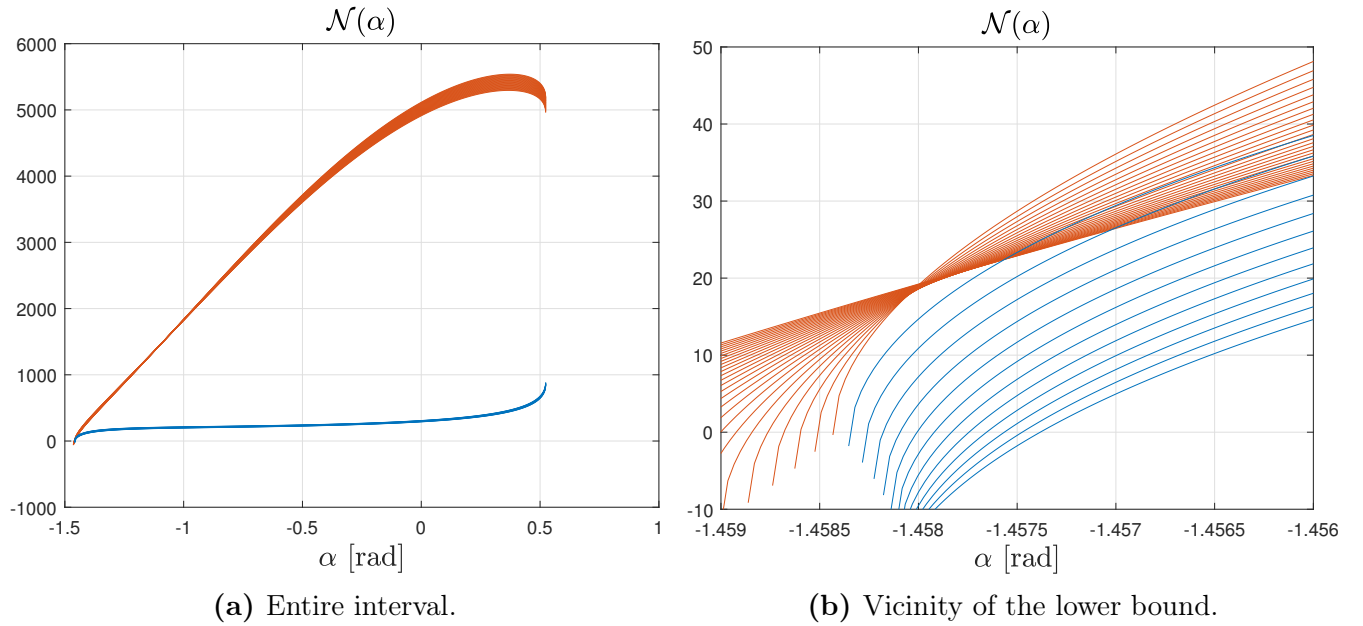


Figure 3.8: Shape of the function $\mathcal{N}(\alpha)$ in the vicinity of the inflection point ($300 \text{ m} < x < 400 \text{ m}$) for the case $\gamma_1 = -60^\circ$.

The previous observation is illustrated more clearly in Fig. 3.9, which shows the position of the solution of the equation $\mathcal{N}(\alpha) = 0$ relative to the original interval (2.50). Comparison with Fig. 3.7 leads to the conclusion that the instants requiring the largest number of iterations correspond to situations in which the solution is located near the interval boundaries.

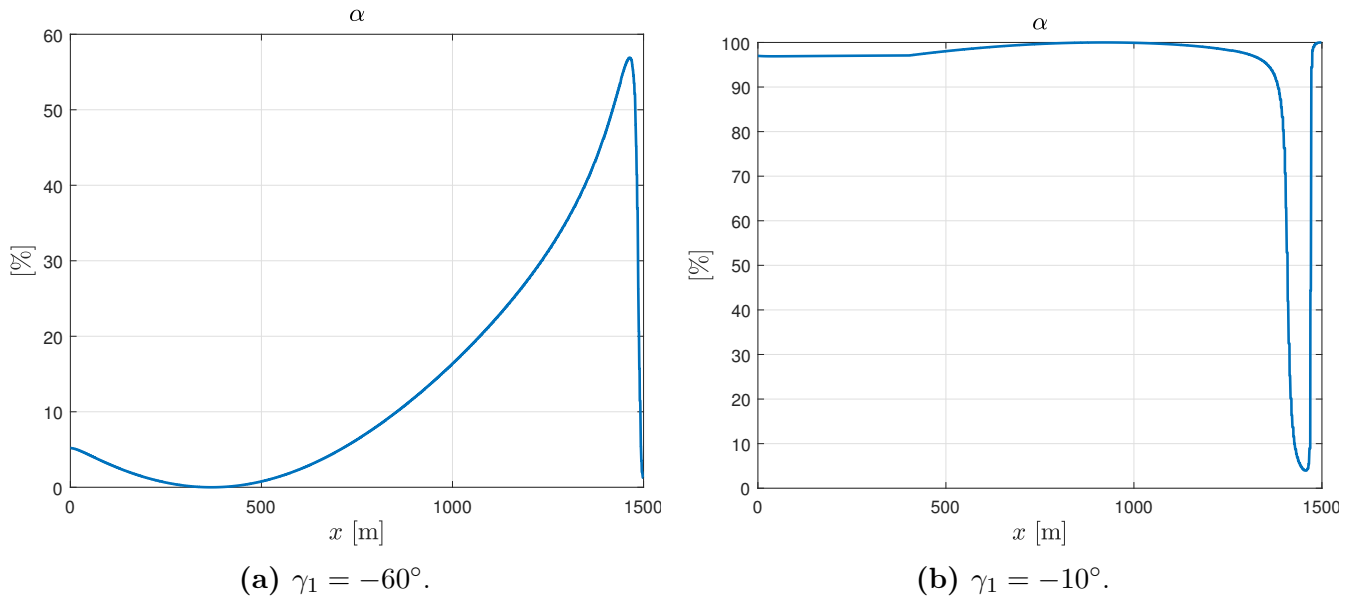


Figure 3.9: Solution of the equation $\mathcal{N}(\alpha) = 0$ expressed as a percentage of the interval (2.50).

Using the proposed variable-interval approach for solving the equation $\mathcal{N}(\alpha) = 0$, the number of required iterations is significantly reduced compared to the original approach. Based on the presented results, it can be concluded that Newton's method and Ridders' method are the most suitable choices, as they achieve the desired accuracy with the smallest number of iterations.

However, for an efficient real-time implementation (in terms of computational efficiency), it is necessary not only to minimize the number of iterations but also to ensure a short execution time for each iteration. The total execution time of a single iteration is influenced by several factors, which will be discussed in more detail in Section 3.3. The most important of these is the number of computational operations that must be performed during each iteration. Depending on the selected method for solving the nonlinear equation of a single variable, both the number and the type of operations, i.e., the functions being evaluated, change. For example, the implementation of the bisection method requires only evaluations of the function $\mathcal{N}(\alpha)$, whereas the implementation of Newton's method requires evaluations of both the function and its derivative. Although the execution time of each iteration can be optimized depending on the hardware platform on which the algorithm is implemented, it is beneficial, before optimizing individual operations, to analyze which operations are computationally expensive (in terms of execution time) and, if possible, replace them with more efficient alternatives.

One operation that is common to all methods for finding the root of the function $\mathcal{N}(\alpha)$ within the framework of Algorithm 2, regardless of the trajectory convexity case, is the computation of the integrals I_1 and I_2 , given by Equations (2.42) and (2.44), which are fundamental components of the function $\mathcal{N}(\alpha)$. The integral I_2 has an analytical solution, so its evaluation requires only direct computation of trigonometric functions. However, the integral I_1 is an elliptic integral and must therefore be evaluated numerically.

3.2.3 Numerical Integration

All results presented thus far, regardless of the method used to solve the nonlinear equation, were obtained using the trapezoidal rule to evaluate the integral I_1 . This method, selected as the baseline approach due to its simplicity, can be expressed as follows [95]:

$$\int_a^b f(x) dx \approx \frac{b-a}{n} \left[\frac{f(x_0)}{2} + \sum_{k=1}^{n-1} f(x_k) + \frac{f(x_n)}{2} \right], \quad (3.16)$$

where n denotes the number of subintervals (hereafter referred to as the number of points).

Since in Chapter 2 the algorithm was executed only once and execution time was not of interest, the number of points used was 10^4 . In contrast, the results presented in this chapter were obtained using 100 points. This raises the question of what the minimum number of points is that can be used with the trapezoidal rule while still achieving satisfactory accuracy. From the approximation formula (3.16), it is evident that reducing the value of n decreases the number of required arithmetic operations and, consequently, the execution time of the algorithm.

The admissible absolute error of the integral I_1 depends on the required accuracy of the solution ϵ_α .

For all three cases of trajectory convexity, the function whose root must be found is given by Eq. (2.53):

$$\mathcal{N}(\Delta x, \Delta y, \gamma_0, \gamma_1, \alpha) = \Delta x I_D(\gamma_0, \gamma_1, \alpha) - \Delta y I_N(\gamma_0, \gamma_1, \alpha) \quad (3.17)$$

where the quantities I_N and I_D are given in compact form by Eq. (2.59).

Equation (3.17), using expressions (2.59), can be rewritten in the following form:

$$\begin{aligned} \mathcal{N}(\Delta x, \Delta y, \gamma_0, \gamma_1, \alpha) = & (\Delta x \cos \alpha + \Delta y \sin \alpha) \mathcal{I}_2(\gamma_0, \gamma_1, \alpha) \\ & + (\Delta x \sin \alpha - \Delta y \cos \alpha) \mathcal{I}_1(\gamma_0, \gamma_1, \alpha) \end{aligned} \quad (3.18)$$

Unlike the notation used for the functions $\mathcal{N}$, I_N , I_D , $\mathcal{I}_1$, $\mathcal{I}_2$ in Section 2.2.4, here the dependence on flight parameters is shown in addition to the dependence on α . The reason is that the focus of the analysis presented in Section 2.2.4 was on determining the optimal value of the control variable at a single specific moment of flight, for which the values Δx , Δy , γ_0 , γ_1 are fixed, so that the only independent variable (at that moment) was α . On the other hand, the topic of this section is the analysis of applying the proposed algorithm over the entire trajectory, for which the variability of all quantities must be taken into account. However, in the interest of conciseness of notation, in the remainder of the analysis the dependence of these functions on flight parameters will be omitted. Since the focus of this part of the work is the analysis of the allowable error in computing the integral I_1 , notation that clearly indicates the presence of this integral will be used, while only the dependence on α will be explicitly emphasized:

$$\mathcal{N}(\alpha) = \mathcal{N}(\alpha, \mathcal{I}_1(\alpha), \mathcal{I}_2(\alpha)) = (\Delta x \cos \alpha + \Delta y \sin \alpha) \mathcal{I}_2(\alpha) + (\Delta x \sin \alpha - \Delta y \cos \alpha) \mathcal{I}_1(\alpha) \quad (3.19)$$

The task of determining the allowable absolute error of the integral I_1 , denoted by ϵ_{I_1} , reduces to determining the allowable error in approximating the quantity $\mathcal{I}_1$, denoted by $\epsilon_{\mathcal{I}_1}$. It is clear that

these two errors are equal only in case A, while in cases B and C the following holds:

$$\epsilon_{I_1} = \frac{\epsilon_{\mathcal{I}_1}}{2} \quad (3.20)$$

If $\hat{\mathcal{I}}_1$ denotes the approximation of the exact value $\mathcal{I}_1$ with absolute error $\epsilon_{\mathcal{I}_1}$:

$$\hat{\mathcal{I}}_1 = \mathcal{I}_1 + \epsilon_{\mathcal{I}_1} \quad (3.21)$$

we can define the function:

$$\hat{\mathcal{N}}(\alpha) \triangleq \mathcal{N}(\alpha, \hat{\mathcal{I}}_1(\alpha), \mathcal{I}_2(\alpha)) \quad (3.22)$$

The function $\hat{\mathcal{N}}$ essentially represents the original function $\mathcal{N}$, where the approximate value $\hat{\mathcal{I}}_1$ is used instead of the exact value $\mathcal{I}_1$. Expanding in a Taylor series yields the following approximation:

$$\hat{\mathcal{N}}(\alpha) = \mathcal{N}(\alpha, \mathcal{I}_1(\alpha) + \epsilon_{\mathcal{I}_1}, \mathcal{I}_2(\alpha)) \approx \mathcal{N}(\alpha, \mathcal{I}_1(\alpha), \mathcal{I}_2(\alpha)) + \frac{\partial \mathcal{N}}{\partial \mathcal{I}_1} \epsilon_{\mathcal{I}_1} \quad (3.23)$$

From expression (3.19), the derivative can be obtained:

$$\frac{\partial \mathcal{N}}{\partial \mathcal{I}_1} = \Delta x \sin \alpha - \Delta y \cos \alpha \quad (3.24)$$

Let α^* be the root of $\mathcal{N}(\alpha)$, and let $\hat{\alpha}$ be the root of $\hat{\mathcal{N}}(\alpha)$. Expanding $\hat{\mathcal{N}}(\hat{\alpha})$ around α^* gives:

$$\hat{\mathcal{N}}(\hat{\alpha}) \approx \hat{\mathcal{N}}(\alpha^*) + \left(\frac{\partial \hat{\mathcal{N}}}{\partial \alpha} \right)_{\alpha^*} (\hat{\alpha} - \alpha^*) = 0 \quad (3.25)$$

From Eq. (3.23) it follows that:

$$\hat{\mathcal{N}}(\alpha^*) \approx \mathcal{N}(\alpha^*, \mathcal{I}_1(\alpha^*), \mathcal{I}_2(\alpha^*)) + \left(\frac{\partial \mathcal{N}}{\partial \mathcal{I}_1} \right)_{\alpha^*} \epsilon_{\mathcal{I}_1} \quad (3.26)$$

Since the first term on the right-hand side is zero, Eq. (3.25) becomes:

$$\left(\frac{\partial \mathcal{N}}{\partial \mathcal{I}_1} \right)_{\alpha^*} \epsilon_{\mathcal{I}_1} + \left(\frac{\partial \hat{\mathcal{N}}}{\partial \alpha} \right)_{\alpha^*} (\hat{\alpha} - \alpha^*) = 0 \quad (3.27)$$

from which it follows:

$$\hat{\alpha} - \alpha^* = - \frac{\left(\frac{\partial \mathcal{N}}{\partial \mathcal{I}_1} \right)_{\alpha^*} \epsilon_{\mathcal{I}_1}}{\left(\frac{\partial \hat{\mathcal{N}}}{\partial \alpha} \right)_{\alpha^*}} \quad (3.28)$$

To complete the previous expression, in addition to the derivative $\partial\mathcal{N}/\partial\mathcal{I}_1$ given in (3.24), we also need:

$$\frac{\partial\hat{\mathcal{N}}}{\partial\alpha} = \frac{\partial}{\partial\alpha}[\mathcal{N}(\alpha, \mathcal{I}_1(\alpha) + \epsilon_{\mathcal{I}_1}, \mathcal{I}_2(\alpha))] \quad (3.29)$$

Using equations (2.53) and (2.59), this can be expanded as:

$$\begin{aligned} \frac{\partial\hat{\mathcal{N}}}{\partial\alpha} = \frac{\partial}{\partial\alpha} & [\Delta x (\cos \alpha \mathcal{I}_2(\alpha) + \sin \alpha \mathcal{I}_1(\alpha) + \sin \alpha \epsilon_{\mathcal{I}_1}) \\ & - \Delta y (\cos \alpha \mathcal{I}_1(\alpha) + \cos \alpha \epsilon_{\mathcal{I}_1} - \sin \alpha \mathcal{I}_2(\alpha))] \end{aligned} \quad (3.30)$$

Collecting terms containing $\epsilon_{\mathcal{I}_1}$ after differentiation with respect to α gives:

$$\begin{aligned} \frac{\partial\hat{\mathcal{N}}}{\partial\alpha} = \frac{\partial}{\partial\alpha} & [\Delta x (\cos \alpha \mathcal{I}_2(\alpha) + \sin \alpha \mathcal{I}_1(\alpha)) \\ & - \Delta y (\cos \alpha \mathcal{I}_1(\alpha) - \sin \alpha \mathcal{I}_2(\alpha))] + (\Delta x \cos \alpha + \Delta y \sin \alpha) \epsilon_{\mathcal{I}_1} \end{aligned} \quad (3.31)$$

The first term on the right-hand side is the derivative of $\mathcal{N}(\alpha)$, so the final form of the derivative is:

$$\frac{\partial\hat{\mathcal{N}}}{\partial\alpha} = \frac{\partial\mathcal{N}}{\partial\alpha} + (\Delta x \cos \alpha + \Delta y \sin \alpha) \epsilon_{\mathcal{I}_1} \quad (3.32)$$

The derivation of $\partial\mathcal{N}/\partial\alpha$ is given in Appendix B.1.2.

Using expressions (3.24) and (3.32), Eq. (3.28) becomes:

$$\hat{\alpha} - \alpha^* = - \frac{\Delta x \sin \alpha^* - \Delta y \cos \alpha^*}{(1/\epsilon_{\mathcal{I}_1}) (\partial\mathcal{N}/\partial\alpha)_{\alpha^*} + (\Delta x \cos \alpha^* + \Delta y \sin \alpha^*)} \quad (3.33)$$

and the following inequality holds:

$$|\hat{\alpha} - \alpha^*| \leq \frac{|\Delta x \sin \alpha^* - \Delta y \cos \alpha^*|}{|(1/\epsilon_{\mathcal{I}_1}) (\partial\mathcal{N}/\partial\alpha)_{\alpha^*} + (\Delta x \cos \alpha^* + \Delta y \sin \alpha^*)|} \leq \epsilon_{\alpha} \quad (3.34)$$

where ϵ_{α} is the allowable error of the solution of $\mathcal{N}(\alpha) = 0$.

For simplicity, the following substitutions are introduced:

$$\mathcal{A} = \Delta x \sin \alpha^* - \Delta y \cos \alpha^* \quad (3.35a)$$

$$\mathcal{B} = \left(\frac{\partial\mathcal{N}}{\partial\alpha} \right)_{\alpha^*} \quad (3.35b)$$

$$\mathcal{C} = \Delta x \cos \alpha^* + \Delta y \sin \alpha^* \quad (3.35c)$$

$$\chi = \frac{1}{\epsilon_{\mathcal{I}_1}} \quad (3.35d)$$

so inequality (3.34) becomes:

$$\frac{|\mathcal{A}|}{|\mathcal{B}\chi + \mathcal{C}|} \leq \epsilon_{\alpha} \quad (3.36)$$

i.e.:

$$|\mathcal{B}\chi + \mathcal{C}| \geq \mathcal{K} \quad (3.37)$$

where:

$$\mathcal{K} = \frac{|\mathcal{A}|}{\epsilon_\alpha} \quad (3.38)$$

This inequality is satisfied if:

$$\mathcal{B}\chi + \mathcal{C} \geq \mathcal{K} \quad \vee \quad \mathcal{B}\chi + \mathcal{C} \leq -\mathcal{K} \quad (3.39)$$

which can be written in general form as:

$$\chi \leq \chi_{\text{ub}} \quad \vee \quad \chi \geq \chi_{\text{lb}} \quad (3.40)$$

where the bounds depend on the sign of $\mathcal{B}$:

$$\chi_{\text{ub}} = \begin{cases} -\frac{\mathcal{K} + \mathcal{C}}{\mathcal{B}}, & \mathcal{B} > 0 \\ \frac{\mathcal{K} - \mathcal{C}}{\mathcal{B}}, & \mathcal{B} < 0 \end{cases} \quad (3.41a)$$

$$\chi_{\text{lb}} = \begin{cases} \frac{\mathcal{K} - \mathcal{C}}{\mathcal{B}}, & \mathcal{B} > 0 \\ -\frac{\mathcal{K} + \mathcal{C}}{\mathcal{B}}, & \mathcal{B} < 0 \end{cases} \quad (3.41b)$$

Using substitution (3.35d), inequality (3.40) becomes:

$$\frac{1}{\epsilon_{\mathcal{I}_1}} \leq \chi_{\text{ub}} \quad \vee \quad \frac{1}{\epsilon_{\mathcal{I}_1}} \geq \chi_{\text{lb}} \quad (3.42)$$

The first inequality has no physical meaning since it defines a lower bound on the error $\epsilon_{\mathcal{I}_1}$, while the upper bound is given by:

$$\epsilon_{\mathcal{I}_1} \leq \frac{1}{\chi_{\text{lb}}} \quad (3.43)$$

provided that $\chi_{\text{lb}} > 0$. If $\chi_{\text{lb}} < 0$, the right-hand inequality is satisfied for any positive $\epsilon_{\mathcal{I}_1}$.

Inequality (3.43) finally defines the allowable error of the integral I_1 . As in the case of the analysis of the solution error ϵ_α , determining this upper bound requires knowledge of the solution α^* , which is unknown at the beginning of each guidance cycle. A possible real-time approach for determining the allowable error of the integral I_1 is based on using expressions (3.35a)–(3.35c), (3.38), and (3.41b), where the true solution α^* is replaced by its approximation obtained in the current iteration of the nonlinear equation solver $\mathcal{N}(\alpha) = 0$. In this way, during the iterative solution process, as the estimate converges to the true value α^* , the estimate of the allowable integration error I_1 is also progressively refined.

This approach implies a variable accuracy requirement for the numerical method used to compute I_1 , and therefore a variable program structure (for example, in the trapezoidal rule: higher accuracy requires more subintervals, leading to more computational operations). The second approach assumes a fixed-structure numerical method, with a prior verification of whether such a method can achieve the required accuracy over the entire trajectory.

In both cases, the methodology for determining allowable errors can be schematically described by Fig. 3.10. Starting from the allowable control error ϵ_u , expression (3.10) is used to determine the required accuracy for solving $\mathcal{N}(\alpha) = 0$. Based on this, using expressions (3.35a)–(3.35c), (3.38), (3.41b), and (3.43), the allowable error for the numerical approximation of the integral I_1 is determined.

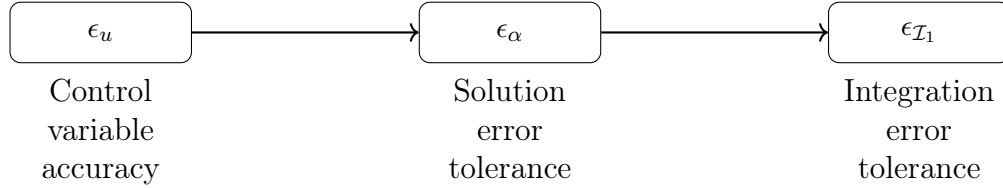


Figure 3.10: The allowable error of the control variable determines the allowable error of numerical procedures.

Figure 3.11 shows the allowable value of the absolute error ϵ_{I_1} determined by the described procedure, for two values of the control variable error ϵ_u , for the scenario described in Table 3.1.

It can be observed that in both cases, in the flight segment $x < 1400$ m, the allowable error is relatively high. Around 1420 m, a discontinuity appears in the graph, corresponding to the case when $\chi_{lb} > 0$, in which case the right-hand inequality in (3.42) is satisfied for any ϵ_{I_1} . After this region, the allowable error decreases, most significantly in the last 10 m of flight for the case $\epsilon_u = 0.1$, and in the last 20 m for the case $\epsilon_u = 0.01$. In both cases, the allowable error is then on the order of 10^{-6} .

Here, as in the case of the analysis of the error ϵ_α , the question can be raised regarding the justification of such a low allowable error as $\epsilon_u = 0.01$. This value was chosen in order to demonstrate that a high accuracy of the control variable can be achieved even with a relatively large error in the approximation of the integral I_1 . However, since in the final moments of flight there are drastically lower values of the allowable error ϵ_{I_1} , the further work will focus on finding a method that achieves the maximum level of accuracy with a minimal number of computational operations.

This approach is consistent with the tendency adopted in this work that, if there exists a moment in which the algorithm requires maximum complexity (e.g., the maximum number of iterations in an iterative procedure, or maximum accuracy of an approximation method), then the algorithm should be tested in its most complex form over the entire trajectory.

Although at first glance one might conclude that this leads to unnecessary consumption of computational resources (computation time), since Fig. 3.11 shows that an error of about 10^{-6} is far below the allowable error level over most of the trajectory, this approach guarantees that the algorithm remains applicable even if, during flight, the required accuracy reaches the level of single-precision floating-point format.

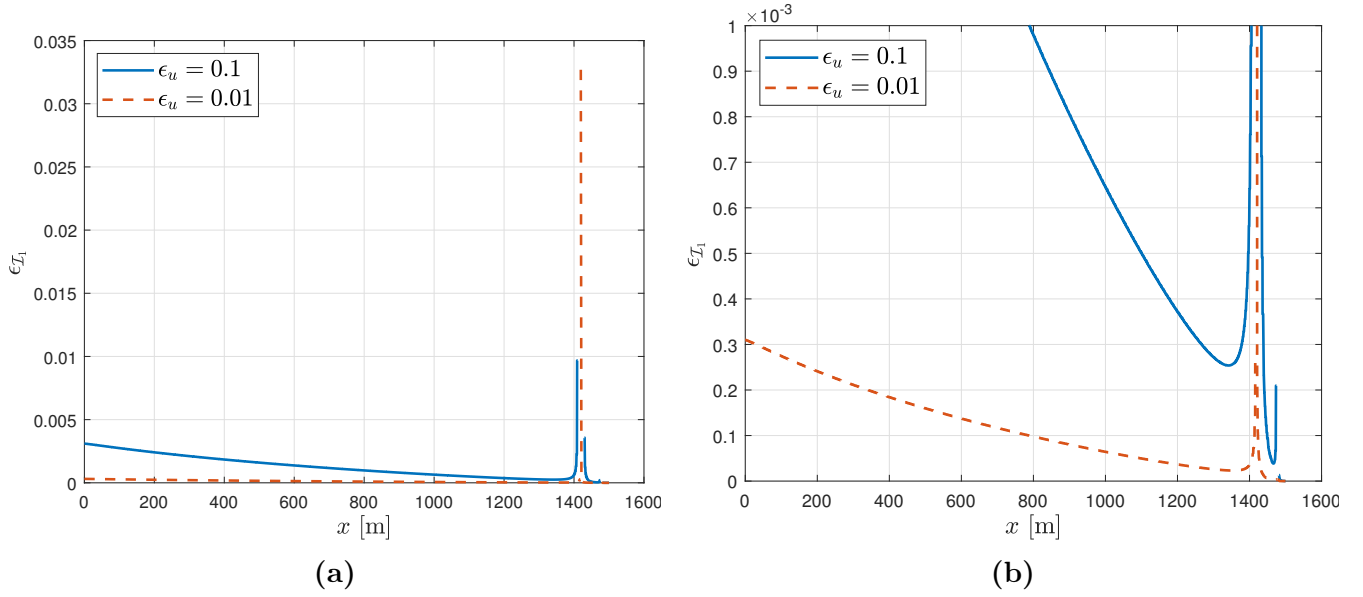


Figure 3.11: Allowable error ϵ_{I_1} for the adopted value of the control variable error ϵ_u , for the scenario described in Table 3.1. The right figure is a zoomed-in version of the left one.

Figure 3.12 shows the dependence of the absolute error in the evaluation of the integral I_1 using the trapezoidal rule for different numbers of subintervals n . The integral value was computed for limits in the range $a, b \in [-\pi/2, \pi/2]$, but due to symmetry only the portion of the surface for $a < b$ is shown. The absolute error was obtained by comparing with the integral value computed using the trapezoidal rule with 10000 subintervals.

Regardless of the chosen number of subintervals, it is noticeable that the absolute error increases when one of the integration limits is close to the value $\pm\pi/2$, which is expected given the shape of the integrand function (shown in Fig. 3.13) near the ends of the interval.

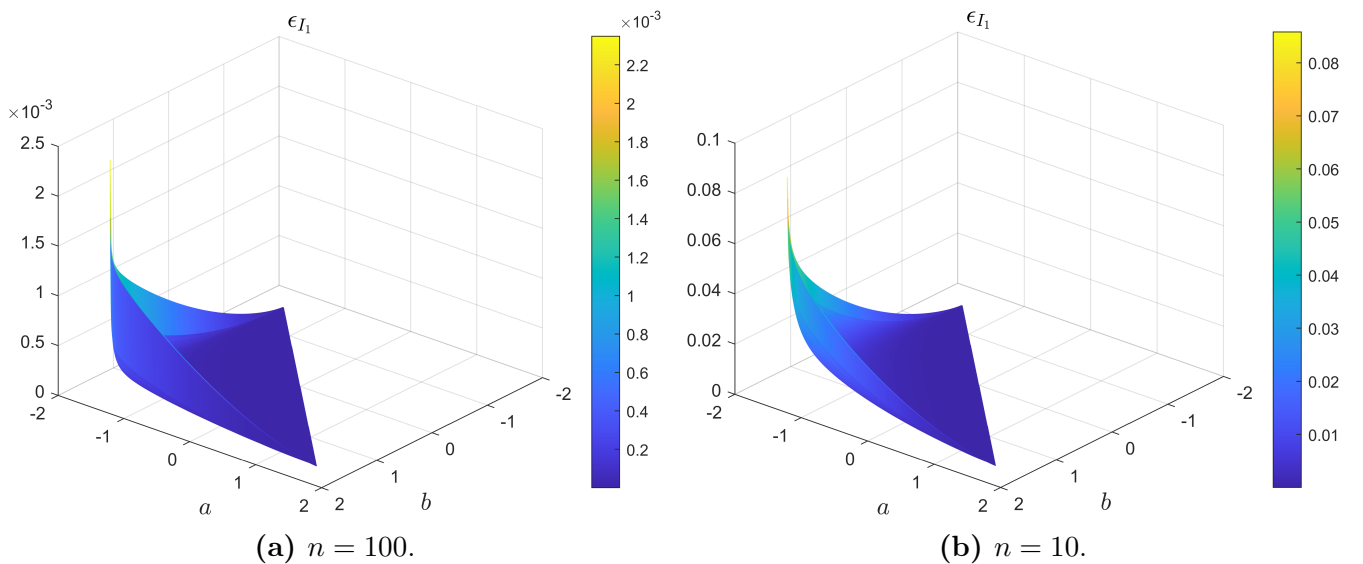


Figure 3.12: Absolute error in the evaluation of the integral I_1 using the trapezoidal rule for different numbers of subintervals (n).

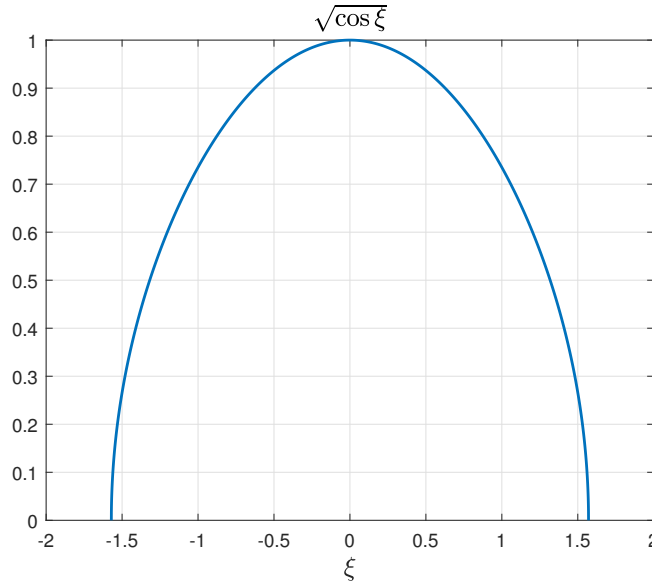


Figure 3.13: Integrand of the integral I_1 , given by (2.42), over the interval $\xi \in [-\pi/2, \pi/2]$.

Since the application of the trapezoidal rule with tens or even hundreds of subintervals would not be efficient from the standpoint of the number of computational operations required for a single evaluation of the integral I_1 , the *Gauss-Legendre (GL)* quadrature formula [98] was adopted in the remainder of this work because of the accuracy achieved with a relatively small number of points. The method can be described as follows:

$$\int_{-1}^1 f(x) dx \approx \sum_{k=1}^n w_k f(x_k), \quad (3.44)$$

where x_k are the nodes, obtained as the roots of the Legendre polynomial, and w_k are the quadrature weights. The procedure for computing the nodes and weights of this method, which has an algebraic degree of precision equal to $2n - 1$, is described in Appendix B.2.1.

For the implementation of this method, the nodes and weights need to be determined only once, which can be done in advance, before the mission itself. However, prior to each evaluation of the integral, a transformation of its integration limits must be performed, since the approximation given by (3.44) is defined for the interval $[-1, 1]$. The procedure for transforming the integration limits is also described in Appendix B.2.1.

Figure 3.14 shows the absolute error in the evaluation of the integral I_1 using the Gauss-Legendre quadrature formula with $n = 10$ nodes. As in Fig. 3.12, the absolute error was obtained by comparison with the integral value computed using the trapezoidal rule with 10000 subintervals.

By comparing Figs. 3.14 and 3.12, it can be observed that the Gauss-Legendre quadrature formula with only 10 nodes achieves a lower maximum error than the trapezoidal rule with as many as 100 subintervals. Nevertheless, the behavior of the error near the ends of the interval remains the same.

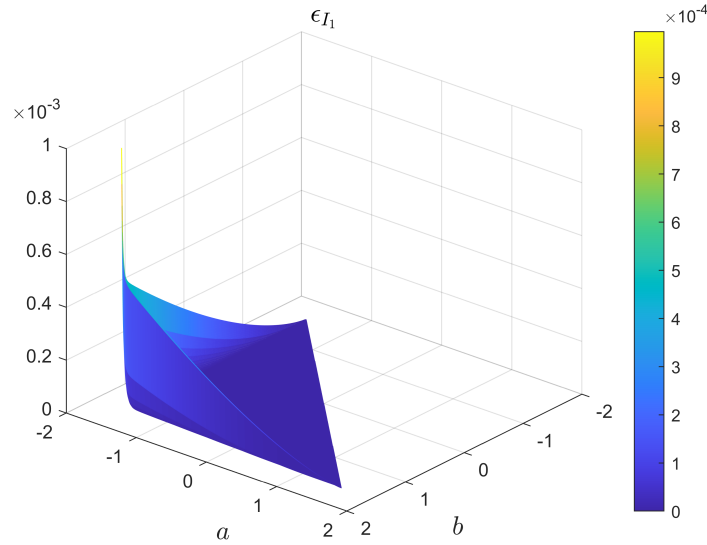


Figure 3.14: Absolute error in the evaluation of the integral I_1 using the Gauss-Legendre quadrature formula with 10 nodes.

Unfortunately, a combination of integration limits for I_1 in which one of the limits is close to $\pm\pi/2$ cannot be dismissed as an unlikely case. Indeed, from (2.60a) it is evident that in cases B and C, the upper integration limit is always equal to $\pm\pi/2$. Moreover, there is no guarantee that such a combination of limits cannot also occur in case A.

Therefore, instead of increasing the number of quadrature nodes—which would improve accuracy at the cost of a larger number of computational operations—an algorithm for the computation of $\mathcal{I}_1$, which is composed of the integrals I_1 , is proposed in the remainder of this work so as to achieve a minimal error even near the ends of the interval. As before, the emphasis is placed on fixed-structure algorithms in order to ensure a fixed execution time in every guidance cycle.

Figure 3.15a shows the absolute error in the approximation of the integral I_1 using the same quadrature formula as in Fig. 3.14, but for a fixed upper integration limit $b = \pi/2$. In other words, the graph shown in Fig. 3.15a corresponds to the portion of the surface in Fig. 3.14 closest to the axis labeled a .

The graph clearly illustrates the increase in error as the lower integration limit moves away from the upper limit, i.e., when integration is performed over a large interval.

Since the integrand shown in Fig. 3.13 is an even function, it is sufficient in the following analysis to consider only the right half-plane (RHP). In that case, the maximum error occurs when integrating over the entire half-interval, i.e., for $a = 0$ and $b = \pi/2$. However, the value of the integral corresponding to this combination of limits can be determined in advance with high accuracy:

$$I_{\text{RHP}} = I_1(0, \pi/2) = \int_0^{\pi/2} \sqrt{\cos \xi} \, d\xi \quad (3.45)$$

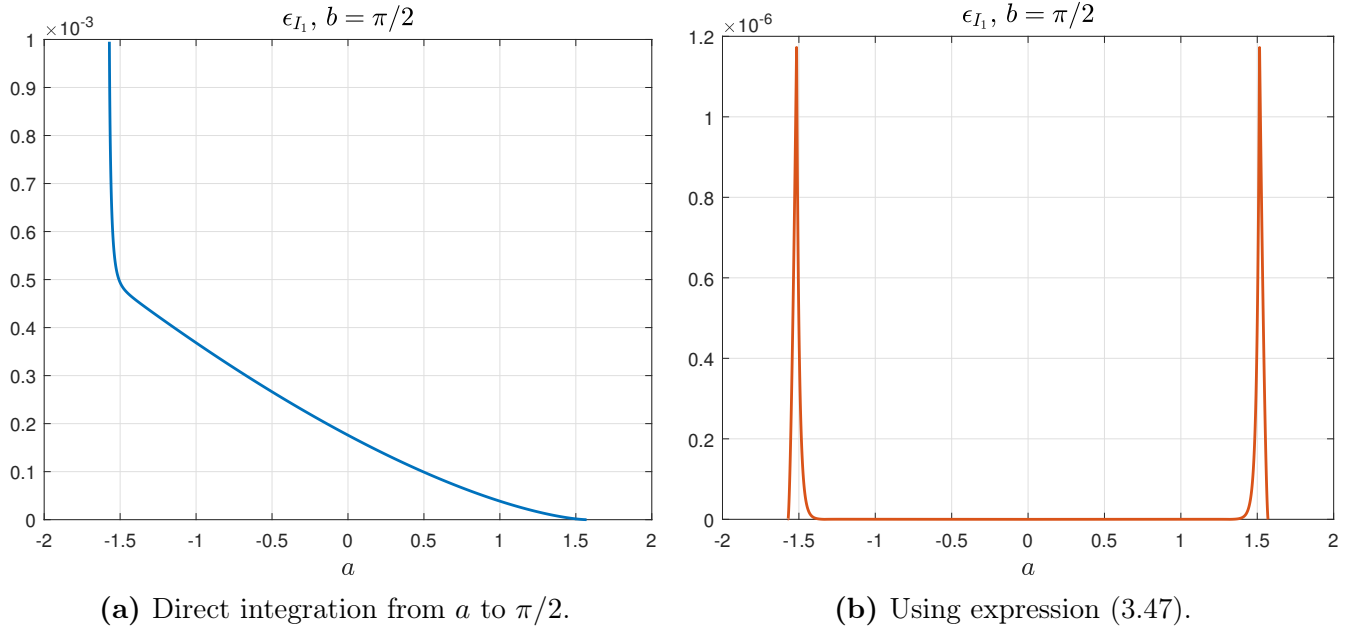


Figure 3.15: Absolute error in the evaluation of the integral I_1 using the Gauss-Legendre quadrature formula with 10 nodes.

The integration in an arbitrary case ($a > 0$) can then be performed as follows:

$$I_1(a, \pi/2) = \int_a^{\pi/2} \sqrt{\cos \xi} \, d\xi = I_{\text{RHP}} - \int_0^a \sqrt{\cos \xi} \, d\xi \quad (3.46)$$

In this way, the integration is always performed starting from zero, which is a region where the integrand is easier to approximate than near the ends of the interval. Nevertheless, if the lower limit is also close to the end of the interval, the error will again become large.

Figure 3.16 shows the absolute errors obtained when approximating the integral in the right half-plane using two methods. The blue curve corresponds to the graph from Fig. 3.15a, while the red curve represents the error obtained using (3.46). In both cases, a quadrature formula with 10 nodes was used, and the error was determined by comparison with a reference integral value obtained using a quadrature formula with 1000 nodes. The quantity I_{RHP} was also precomputed using a quadrature formula with 1000 nodes.

Based on the presented errors, it is clear that for the computation of $I_1(a, \pi/2)$ it is preferable to use expression (3.46) rather than performing direct integration from a to $\pi/2$. The exception is the region close to the upper interval boundary, where a smaller error is achieved by direct integration over the required limits.

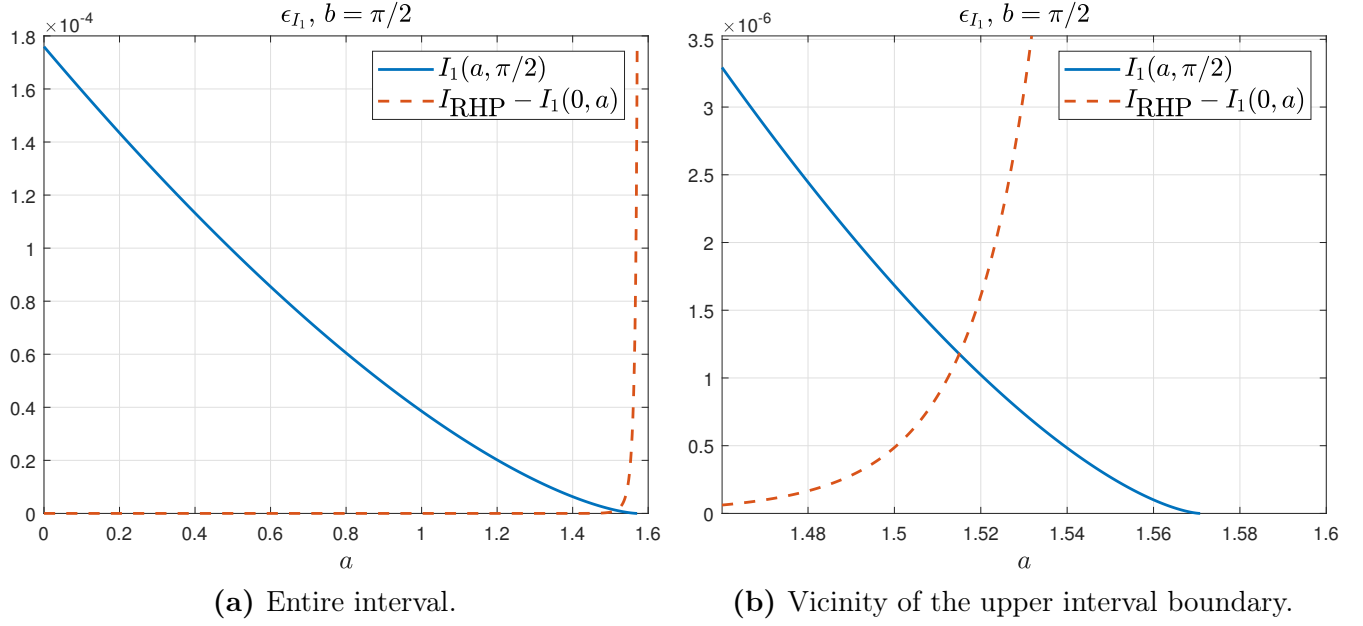


Figure 3.16: Absolute error in the evaluation of the integral I_1 using the Gauss-Legendre quadrature formula with 10 nodes, computed in two different ways. The figure on the right is a zoomed-in version of the figure on the left.

If a^* denotes the value of the lower limit a at which the two curves in Fig. 3.16b intersect, then, taking into account the part of the interval for $a < 0$, the computation of the integral I_1 for case B can be represented by the following expression:

$$I_1(a, \pi/2) = \begin{cases} I_1(a, \pi/2), & a \geq a^* \\ I_{\text{RHP}} - I_1(0, a), & 0 < a < a^* \\ I_{\text{RHP}}, & a = 0 \\ I_{\text{RHP}} + I_1(0, |a|), & -a^* < a < 0 \\ 2I_{\text{RHP}} - I_1(|a|, \pi/2), & a \leq -a^* \end{cases} \quad (3.47)$$

The absolute error obtained using (3.47) over the entire integration interval is shown in Fig. 3.15b. The maximum error, which occurs at the point a^* , is approximately 10^{-6} , which is within the prescribed error tolerance.

Since in case C the upper integration limit is always equal to $-\pi/2$, the proposed procedure is also valid for that case, with the remark that the integration limits must then be exchanged and the sign of the final integral value reversed:

$$I_1(a, -\pi/2) = -I_1(-\pi/2, a) \quad (3.48)$$

In this way, the desired accuracy in the computation of the integral I_1 has been achieved for cases B and C. These two cases are characterized by a fixed upper integration limit, $b = \pm\pi/2$, and by the fact that, according to (2.59) and (2.60), two evaluations of the integral I_1 are always required. In case A, on the other hand, both integration limits are free, and the integral must be evaluated only once.

Nevertheless, the phenomena already analyzed for case B are also present in case A. Since the upper and lower limits may now take arbitrary values, integration sometimes has to be performed over a large interval. This is less problematic near zero than near the ends of the interval, where the integrand is difficult to approximate. Therefore, a similar methodology can be applied in this case as well in order to achieve satisfactory accuracy.

At first glance, this may appear to complicate the simplest case: instead of computing a single integral $I_1(a,b)$, it becomes necessary to determine two integrals—one involving only the lower limit and one end of the interval (e.g., $I_1(0,a)$), and another involving only the upper limit (e.g., $I_1(b,\pi/2)$). However, in this way, case A is algorithmically reduced to cases B and C, since the evaluation of $\mathcal{I}_1$, given by (2.60), will require two evaluations of the integral I_1 .

The advantage of this approach lies in the fixed execution time of the numerical procedure used for evaluating the integral, regardless of which case is currently active. Furthermore, when the lower and upper limits define a large interval, splitting the integral into two parts while retaining the same quadrature formula results in improved accuracy.

If $\mathcal{G}(c,d)$ denotes the function that approximates the integral $I_1(c,d)$ using the Gauss-Legendre quadrature formula:

$$I_1(c,d) \approx \mathcal{G}(c,d) \quad (3.49)$$

then the general procedure for computing this approximation can be described by Algorithm 3.

For a clearer visualization of the algorithm, Fig. 3.17 indicates the possible positions of the upper integration limit depending on the point c^* . For each possible case, the figure also indicates the region in which the lower integration limit may be located.

The point c^* represents a generalization of the point a^* identified during the analysis of the integral in case B. It denotes the point beyond which, for an arbitrary value of c , the integral $I_1(0,c)$ can be approximated more accurately as $I_{\text{RHP}} - I_1(c,\pi/2)$ than by direct integration from 0 to c .

Figure 3.18 shows the absolute error in the approximation of the integral I_1 obtained using Algorithm 3 and the Gauss-Legendre quadrature formula with 10 nodes. The same value of c^* as in the previous analysis was adopted.

It can be observed that over the entire integration interval the absolute error remains below 10^{-6} , except in the vicinity of three characteristic regions. These correspond to combinations of integration limits for which a is close to c^* from the left, while b is close to c^* from the right, resulting in a doubling of the error associated with the point c^* . Consequently, the maximum expected error is equal to twice the maximum error shown in Fig. 3.15b, which is approximately 2×10^{-6} . One such example is shown in Fig. 3.19, for the upper integration limit value $b = 1.523$.

Algorithm 3 Calculation of $I = I_1(a, b)$ given by Eq. (2.42)

```

1: if  $a = b$  then
2:    $I \leftarrow 0$ ; return
3: end if
4: if  $a > b$  then
5:    $I \leftarrow -I_1(b, a)$ ; return
6: end if
7:  $c^* \leftarrow 1.515032499555405$ 
8:  $I_{rhp} \leftarrow 1.198140234937978$ 
9: if  $b \geq c^*$  then
10:   $I_b \leftarrow \mathcal{G}(b, \pi/2)$ 
11:  if  $a \geq c^*$  then
12:     $I_a \leftarrow \mathcal{G}(a, \pi/2)$ 
13:     $I \leftarrow I_a - I_b$ 
14:  elseif  $a \geq 0$  then
15:     $I_a \leftarrow \mathcal{G}(0, a)$ 
16:     $I \leftarrow I_{rhp} - (I_a + I_b)$ 
17:  elseif  $a > -c^*$  then
18:     $I_a \leftarrow \mathcal{G}(0, |a|)$ 
19:     $I \leftarrow I_{rhp} - I_b + I_a$ 
20:  else
21:     $I_a \leftarrow \mathcal{G}(|a|, \pi/2)$ 
22:     $I \leftarrow 2I_{rhp} - (I_a + I_b)$ 
23:  end if
24: elseif  $b \geq 0$  then
25:   $I_b \leftarrow \mathcal{G}(0, b)$ 
26:  if  $a \geq 0$  then
27:     $I_a \leftarrow \mathcal{G}(0, a)$ 
28:     $I \leftarrow I_b - I_a$ 
29:  elseif  $a > -c^*$  then
30:     $I_a \leftarrow \mathcal{G}(0, |a|)$ 
31:     $I \leftarrow I_a + I_b$ 
32:  else
33:     $I_a \leftarrow \mathcal{G}(|a|, \pi/2)$ 
34:     $I \leftarrow I_{rhp} - I_a + I_b$ 
35:  end if
36: elseif  $b > -c^*$  then
37:   $I_b \leftarrow \mathcal{G}(0, |b|)$ 
38:  if  $a > -c^*$  then
39:     $I_a \leftarrow \mathcal{G}(0, |a|)$ 
40:     $I \leftarrow I_a - I_b$ 
41:  else
42:     $I_a \leftarrow \mathcal{G}(|a|, \pi/2)$ 
43:     $I \leftarrow I_{rhp} - (I_a + I_b)$ 
44:  end if
45: else
46:   $I_b \leftarrow \mathcal{G}(|b|, \pi/2)$ 
47:   $I_a \leftarrow \mathcal{G}(|a|, \pi/2)$ 
48:   $I \leftarrow I_b - I_a$ 
49: end if

```

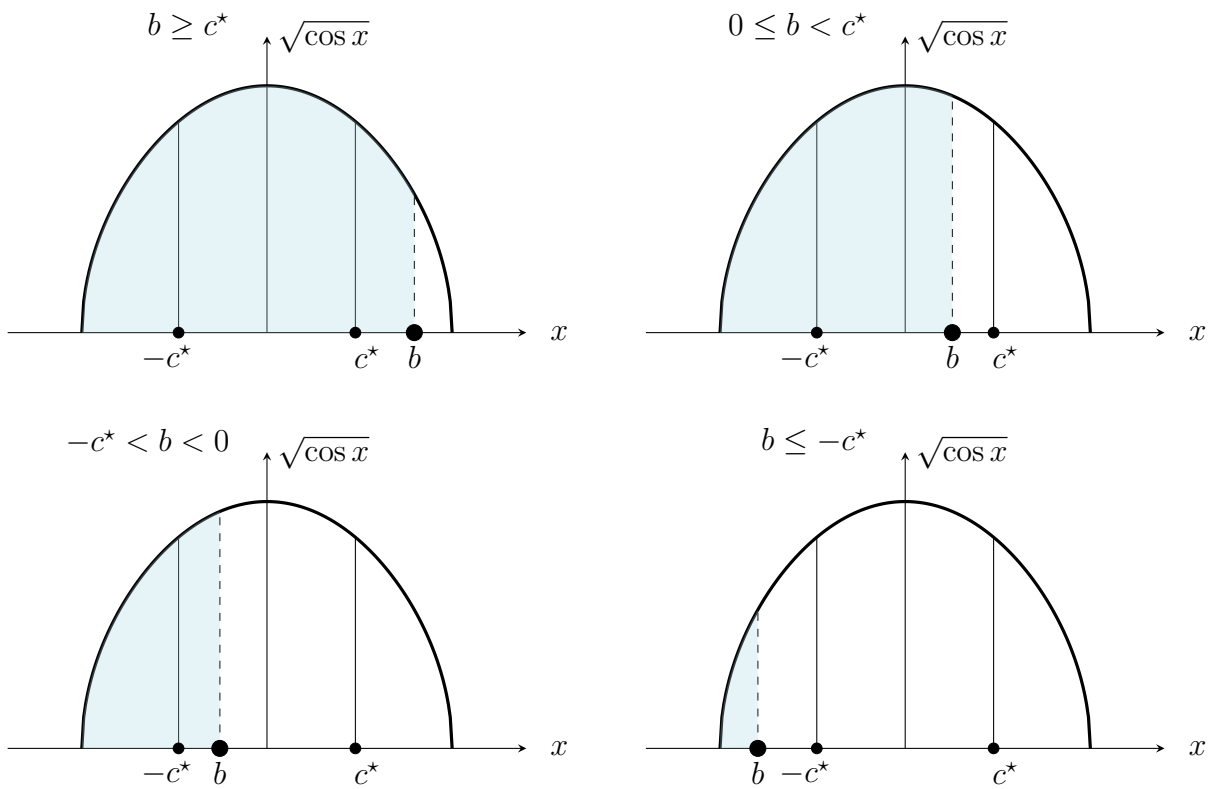


Figure 3.17: Spatial partitioning of integration limits: possible positions of upper limit b relative to threshold c^* and the resulting domains (colored area) for lower limit a .

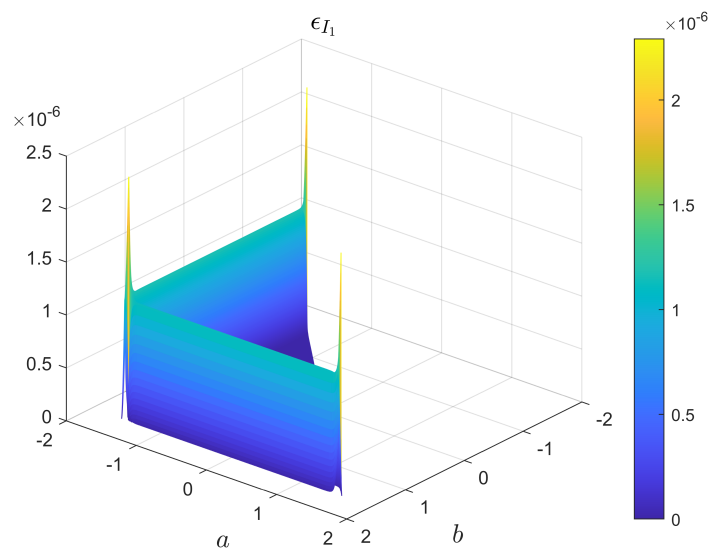


Figure 3.18: Absolute error in the evaluation of the integral I_1 using Algorithm 3 and a quadrature formula with 10 nodes.

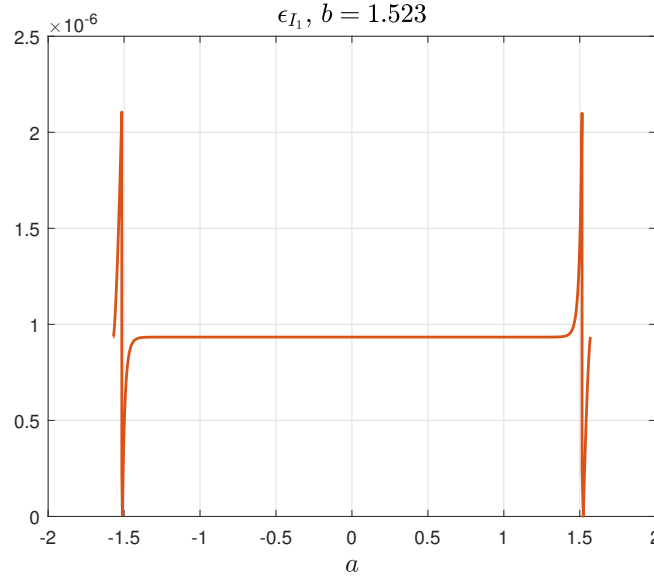


Figure 3.19: Absolute error in the evaluation of the integral I_1 using Algorithm 3 and a quadrature formula with 10 nodes, for the upper integration limit value at which the maximum error may occur.

Although it is difficult to discern from Fig. 3.18, the error level for cases B and C, i.e., for the upper integration limits $b = \pm\pi/2$, still follows the profile shown in Fig. 3.15b. This means that the proposed algorithm for evaluating the integral I_1 achieves a maximum error of approximately 10^{-6} in cases B and C, whereas in case A the maximum error reaches 2×10^{-6} . In all cases, the maximum error occurs near the ends of the interval, while throughout the remainder of the interval the error is several orders of magnitude smaller than its maximum value.

Based on the conducted analysis, it can be concluded that the maximum error values are acceptable and consistent with the analysis presented in the previous section of this work (Fig. 3.11), and that Algorithm 3 can be used for the numerical evaluation of the integral I_1 given by (2.42).

Cruise Missile Climb Phase

In Section 2.2.5, a special case of the original optimization problem was considered, corresponding to the climb phase of a cruise missile and the transition to level flight. The guidance law (2.85) is given in closed form in terms of the state variables:

$$u(y, \gamma) = -\frac{V^2 \sqrt{\sin \gamma}}{\Delta y} I_S(\gamma) \quad (3.50)$$

where:

$$I_S(\gamma) = \int_0^\gamma \sqrt{\sin \xi} d\xi \quad (3.51)$$

Although the commanded normal acceleration in this phase is computed directly, without the application of an iterative procedure for solving a nonlinear equation, the integral (3.51) does not

possess an analytical solution and must therefore be evaluated numerically. For this reason, the analysis of the procedure used for its computation is included in this section.

Let $\hat{I}_S$ denote an approximation of the integral I_S such that:

$$\hat{I}_S = I_S + \epsilon_{I_S} \quad (3.52)$$

where ϵ_{I_S} is the absolute approximation error. By expanding (3.50) into a Taylor series about the exact value of the control variable, the following expression is obtained:

$$u(y, \gamma, \hat{I}_S) \approx u(y, \gamma, I_S) + \frac{\partial u}{\partial I_S} (\hat{I}_S - I_S) \quad (3.53)$$

Neglecting higher-order terms yields the following equality:

$$|u(y, \gamma, \hat{I}_S) - u(y, \gamma, I_S)| = \left| \frac{\partial u}{\partial I_S} \right| \epsilon_{I_S} \quad (3.54)$$

The left-hand side of (3.54) represents the control error, which is bounded by the prescribed error tolerance ϵ_u . Therefore, the following inequality holds:

$$|u(y, \gamma, \hat{I}_S) - u(y, \gamma, I_S)| = \left| \frac{\partial u}{\partial I_S} \right| \epsilon_{I_S} < \epsilon_u \quad (3.55)$$

This implies that the upper bound on the absolute error of the integral I_S must be limited by the allowable control error:

$$\epsilon_{I_S} < \frac{\epsilon_u}{|\partial u / \partial I_S|} \quad (3.56)$$

Differentiating (3.50) with respect to I_S gives

$$\frac{\partial u}{\partial I_S} = -\frac{V^2 \sqrt{\sin \gamma}}{\Delta y} \quad (3.57)$$

and therefore inequality (3.56) becomes

$$\epsilon_{I_S} < \frac{\Delta y}{V^2 \sqrt{\sin \gamma}} \epsilon_u \quad (3.58)$$

As shown in Section 2.2.5, the use of the guidance law (3.50) during the climb phase results in a smooth transition to horizontal flight, i.e., a zero value of the control variable at the instant when the desired altitude is reached. Consequently, from the kinematic equations describing the motion:

$$\dot{y} = V \sin \gamma \quad (3.59a)$$

$$\dot{\gamma} = \frac{u}{V} \quad (3.59b)$$

it follows that, during the final moments of the climb phase:

$$\dot{y} \approx V\gamma \implies \Delta y(t) \approx V \int_t^{t_1} \gamma(\tau) d\tau \quad (3.60a)$$

$$\gamma(t) = -\frac{1}{V} \int_t^{t_1} u(\tau) d\tau \quad (3.60b)$$

In the above expressions, t_1 denotes the instant at which the desired altitude is reached, i.e., the final instant of the climb phase, for which $\gamma(t_1) = 0$.

If it is assumed that the control variable during the final moments of the climb phase is proportional to a power of the remaining time:

$$u(t) \sim (t_1 - t)^p, \quad p > 0 \quad (3.61)$$

then, based on expressions (3.60), the state variables evolve according to:

$$\gamma(t) \sim (t_1 - t)^{p+1} \quad (3.62a)$$

$$\Delta y(t) \sim (t_1 - t)^{p+2} \quad (3.62b)$$

which yields:

$$\Delta y \sim \gamma^{\frac{p+2}{p+1}} \quad (3.63)$$

indicating that Δy decreases faster than γ .

From inequality (3.58), it is clear that near the end of the climb phase, for a prescribed control error ϵ_u , the allowable approximation error of the integral I_S is proportional to:

$$\epsilon_{I_S} \sim \frac{\Delta y}{\sqrt{\gamma}} \quad (3.64)$$

Using expression (3.63), it follows that:

$$\frac{\Delta y}{\sqrt{\gamma}} \sim \gamma^{\frac{p+2}{p+1} - \frac{1}{2}} = \gamma^{\frac{p+3}{2(p+1)}} \quad (3.65)$$

Since:

$$\frac{p+3}{2(p+1)} > 0 \quad \forall p > 0 \quad (3.66)$$

the allowable approximation error tends to zero immediately before the transition to horizontal flight:

$$\gamma \rightarrow 0 \implies \epsilon_{I_S} \rightarrow 0 \quad (3.67)$$

Figure 3.20 shows the allowable error ϵ_{I_S} determined by expression (3.58) for two values of the control error ϵ_u , as a function of altitude. The same scenario as in Section 2.2.5 was considered, whose trajectory is shown in Fig. 2.34. The predicted behavior can clearly be observed: the allowable error approaches zero during the final moments of the climb phase regardless of the adopted value of ϵ_u .

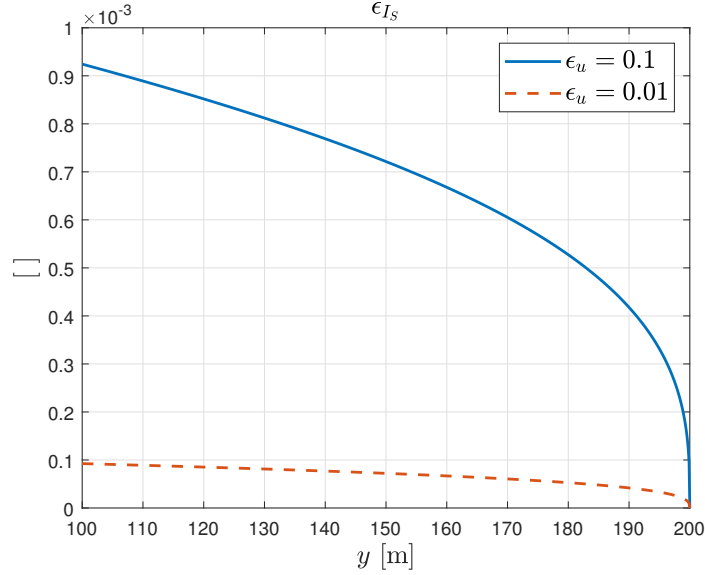


Figure 3.20: Allowable approximation error of integral (3.51) for different values of the control error ϵ_u .

Although the conducted analysis and the diagram in Fig. 3.20 show that high accuracy is required only during the final moments of the climb phase, the same approach used in the previous section will also be adopted here when formulating the numerical integration method. Namely, if there exists a point along the trajectory at which maximum accuracy is required, the numerical procedure employed throughout the entire flight will be designed according to the minimum-error requirements at that point. In this way, the algorithm used for evaluating the integral I_S will have a fixed structure and a fixed execution time.

In Fig. 3.21a, the blue curve represents the absolute error obtained when evaluating integral (3.51) using a Gauss–Legendre quadrature formula with 10 nodes. The interval of interest is from 0 to $\pi/2$, since this is the range of the flight-path angle γ during the climb phase.

It can be observed that the error increases as the lower integration limit moves away from zero, reaching approximately 2×10^{-4} at the end of the interval. This behavior is expected considering the shape of the integrand over the considered interval, shown in Fig. 3.22.

Since the integrand is difficult to approximate in the vicinity of zero, similarly to the integrand $\sqrt{\cos \xi}$ analyzed in the previous section but near the interval endpoints, a dedicated algorithm will also be employed here depending on the value of the upper integration limit.

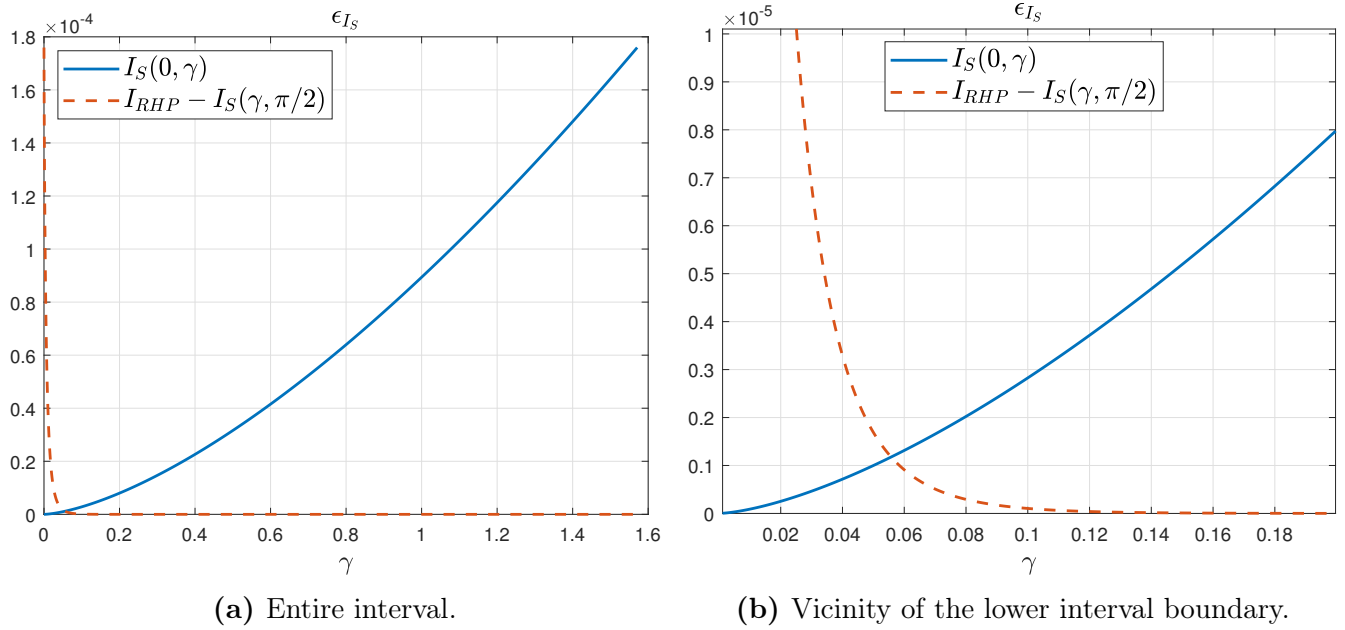


Figure 3.21: Absolute error in the evaluation of the integral (3.51) using the Gauss-Legendre quadrature formula with 10 nodes, computed in two different ways. The figure on the right is a zoomed-in version of the figure on the left.

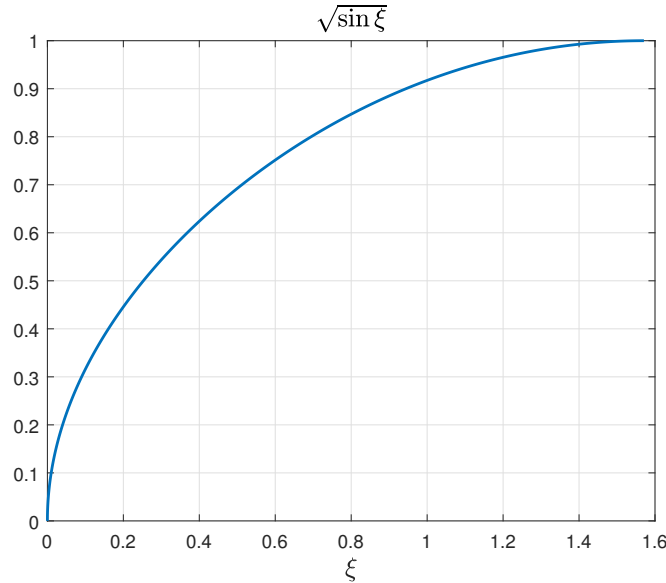


Figure 3.22: Integrand of the integral I_S , given by expression (3.51), over the interval $\xi \in [0, \pi/2]$.

In Fig. 3.21a, in addition to the error obtained by direct evaluation of $I_S(0, \gamma)$, the dashed curve represents the error obtained when approximating integral (3.51) using the following expression:

$$I_S(0, \gamma) = I_{RHP} - I_S(\gamma, \pi/2) \quad (3.68)$$

where I_{RHP} is a precomputed value of the integral that can be determined with high accuracy:

$$I_{\text{RHP}} = I_S(0, \pi/2) = \int_0^{\pi/2} \sqrt{\sin \xi} \, d\xi \quad (3.69)$$

Expectedly, evaluating integral (3.51) using expression (3.68) yields a small absolute error over most of the interval. The exception is the vicinity of zero, where numerical integration must be performed over a large interval.

It is therefore evident that the minimum error can be achieved by combining the advantages of both approaches, i.e., by performing direct integration when the upper limit is smaller than the intersection point shown in Fig. 3.21b, and by evaluating the integral using expression (3.68) when the upper limit is greater.

If $\mathcal{G}'(c, d)$ denotes the function used to approximate the integral:

$$I'_S(c, d) = \int_c^d \sqrt{\sin \xi} \, d\xi \quad (3.70)$$

using the Gauss–Legendre quadrature formula:

$$I'_S(c, d) \approx \mathcal{G}'(c, d) \quad (3.71)$$

then the general procedure for evaluating this approximation can be described by Algorithm 4.

Figure 3.23 shows the absolute error obtained when evaluating integral (3.51) using Algorithm 4. The presented curve is a combination of portions of the curves shown in Fig. 3.21.

It is concluded that the proposed algorithm for the numerical evaluation of integral (3.51) achieves satisfactory accuracy at the level of machine precision.

Algorithm 4 Calculation of $I = I_S(0, \gamma)$, given by Eq. (3.51)

```

1: if  $\gamma = 0$  then
2:    $I \leftarrow 0$ ; return
3: end if
4:  $c^* \leftarrow 0.055768846485867$ 
5:  $I_{rhp} \leftarrow 1.198140234937978$ 
6: if  $\gamma < c^*$  then
7:    $I \leftarrow \mathcal{G}'(0, \gamma)$ 
8: else
9:    $I_g \leftarrow \mathcal{G}'(\gamma, \pi/2)$ 
10:   $I \leftarrow I_{rhp} - I_g$ 
11: end if
```

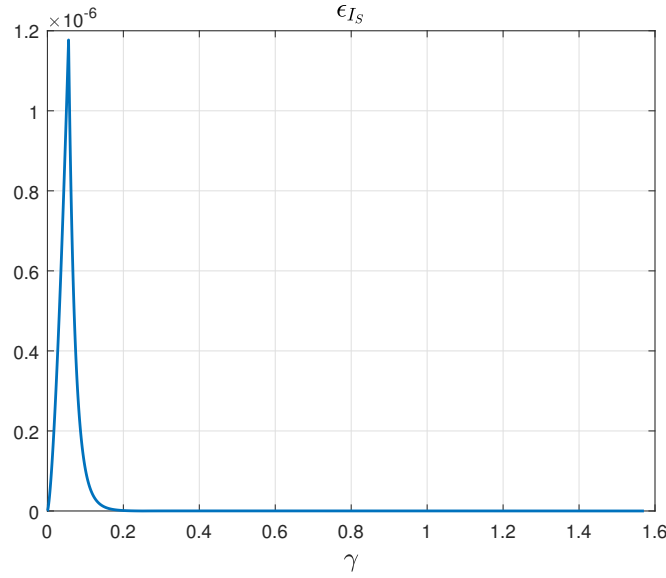


Figure 3.23: Absolute error in the evaluation of the integral I_S using a Gauss–Legendre quadrature formula with 10 nodes and the combined method.

3.2.4 Influence of Variable Velocity

So far, the application analysis of Algorithm 2 has assumed a constant missile velocity. Although this is the assumption under which the optimal guidance law (3.1) was derived, such a case is rarely expected in practice. Since the objective of this work has been to develop an algorithm capable of determining all parameters of the guidance law (3.1) in real time, velocity variations do not violate the problem formulation because every flight instant (i.e., every instant at which a guidance command must be generated) is treated as an initial condition. In this manner, closed-loop control is achieved as a function of the state variables that are available during every guidance cycle.

Nevertheless, in order to analyze the influence of variable velocity, a linear variation of velocity with time was introduced into the numerical simulation:

$$V(t) = 160 \pm 2t \quad (3.72)$$

where 160 m/s is the nominal velocity about which the variation was simulated. All other parameters of the numerical simulation correspond to those described in Table 3.1. Figure 3.24 shows the velocity profiles for the nominal case, as well as for the cases of increasing and decreasing velocity during flight.

The variable velocity affects only the kinematic equations (2.91), whereas two strategies were adopted for the velocity V appearing in the optimal guidance law (3.1).

The first strategy assumes that, despite the variable flight velocity, the nominal value $V = 160$ m/s is used within the guidance law. The corresponding control-input profile is shown in Fig. 3.25a.

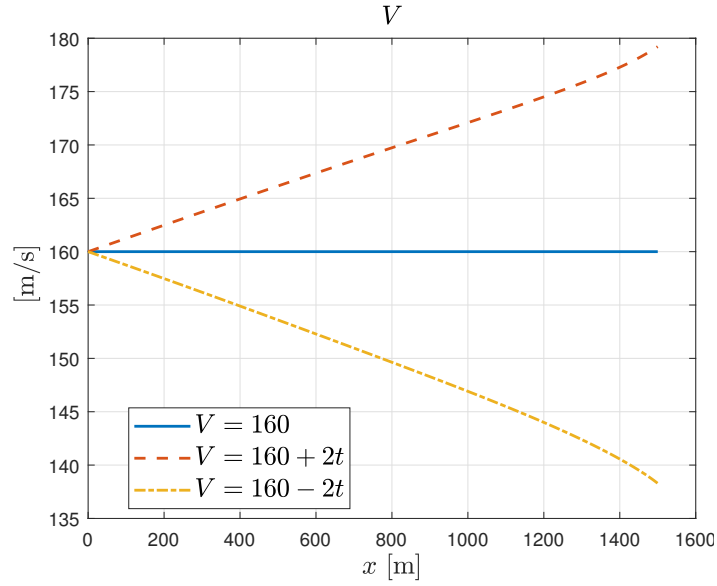


Figure 3.24: Different actual velocity profiles for the scenario described in Table 3.1.

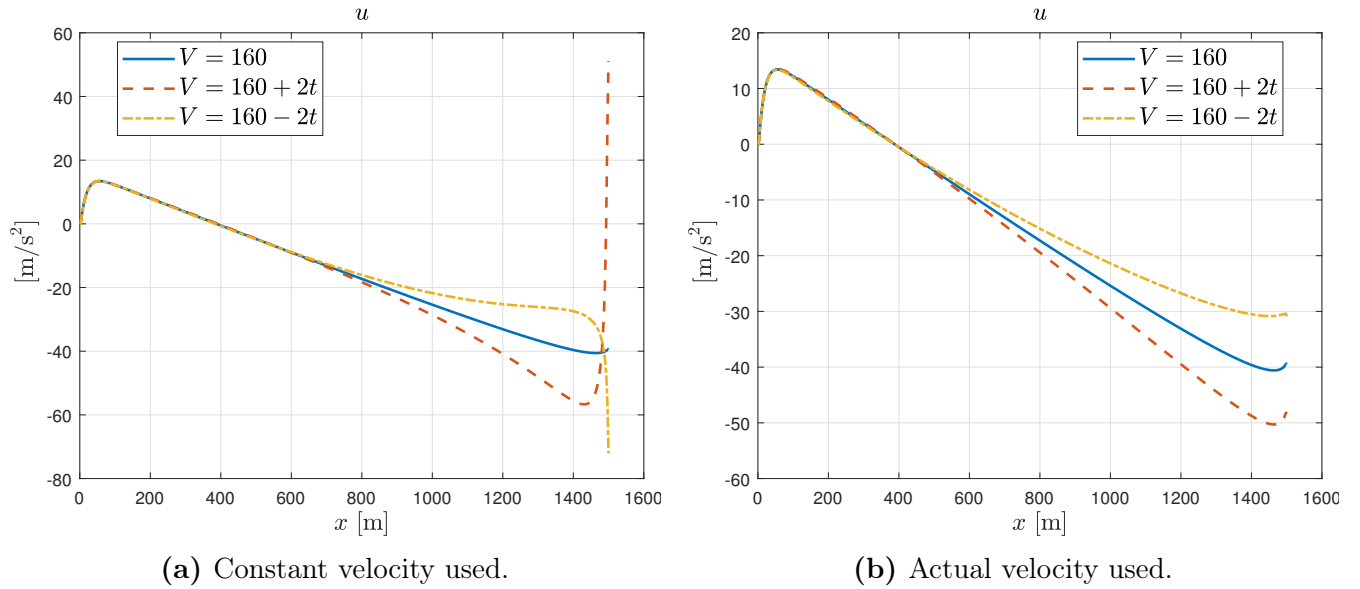


Figure 3.25: Control-input profiles for different choices of the velocity used in the optimal guidance law, under variable actual velocity conditions.

The second strategy computes the optimal command using the actual velocity value, which varies throughout the flight. The corresponding control-input profile is shown in Fig. 3.25b.

By comparing the presented results, it is evident that in the first case a significant increase in the control input occurs when the actual velocity begins to deviate substantially from the nominal value. This is a consequence of the fact that the computed command no longer corresponds to the actual flight conditions.

On the other hand, when information about the actual velocity is incorporated into the guidance law, the resulting control-input profile is consistent with the real flight conditions and does not lead to excessive control effort.

Since no significant differences in the achieved trajectories are observed in either case, only the variations in the control input are presented in this analysis. Based on these results, it can be concluded that the optimal guidance law should be implemented using the actual velocity value, i.e., the velocity available during each guidance cycle.

3.2.5 Influence of Delay in Updating Optimal Guidance Parameters

The commanded normal acceleration computed by the optimal guidance law (3.1) can be written in a more compact form:

$$u = u(\mathbf{x}, \boldsymbol{\eta}) \quad (3.73)$$

where:

$$\mathbf{x} = (x, y, \gamma), \quad \boldsymbol{\eta} = (\eta_1, \eta_2) \quad (3.74)$$

are the state vector and the optimal guidance parameter vector, respectively.

The guidance law (3.73) represents feedback control with respect to the state variables x , y , and γ , whose values are available from the navigation system. Since the quantities x and y appear explicitly in expression (3.1) through the relative missile-to-target distances Δx and Δy , the command can be computed immediately after these values are updated.

However, in order to compute the optimal guidance parameters η_1 and η_2 (and consequently η_3 based on (2.17)), a certain computational time is required to execute Algorithm 2. A detailed analysis of the required computational time for the implementation of Algorithm 2 is given in Section 3.3, while in this section the influence of delay in updating the parameters η_1 and η_2 is investigated.

Here, *delay* refers to the time required after updating the state variables x , y , and γ to compute new values of the optimal guidance parameters. Since navigation data are updated at a fixed sampling rate, this delay is also expressed as the number of cycles required to compute new values of η_1 and η_2 , so that the guidance law (3.73) can be written in discrete form as:

$$u_k = u(\mathbf{x}_k, \boldsymbol{\eta}_{k-d}) \quad (3.75)$$

The quantity d in the previous expression denotes the delay, i.e., the number of cycles required for updating the guidance parameters.

In order to analyze the influence of this delay on the entire trajectory, in the numerical simulation the values of η_1 and η_2 from previous cycles were used for each guidance update (16 ms), depending on the adopted delay level d . All other quantities required for the implementation of the optimal guidance law were updated as before, and the simulation parameters are given in Table 3.1.

Figure 3.26 shows the control-input profile for several delay levels d . As in the case of the variable velocity analysis, no significant differences in the achieved trajectories are observed here either,

so only the control-input variations are presented, specifically in the first and last 100 m of the trajectory, where the changes are most pronounced. Based on the presented results, it can be concluded that the delay does not have a significant influence on the overall trajectory (at least for delay levels up to 5 cycles).

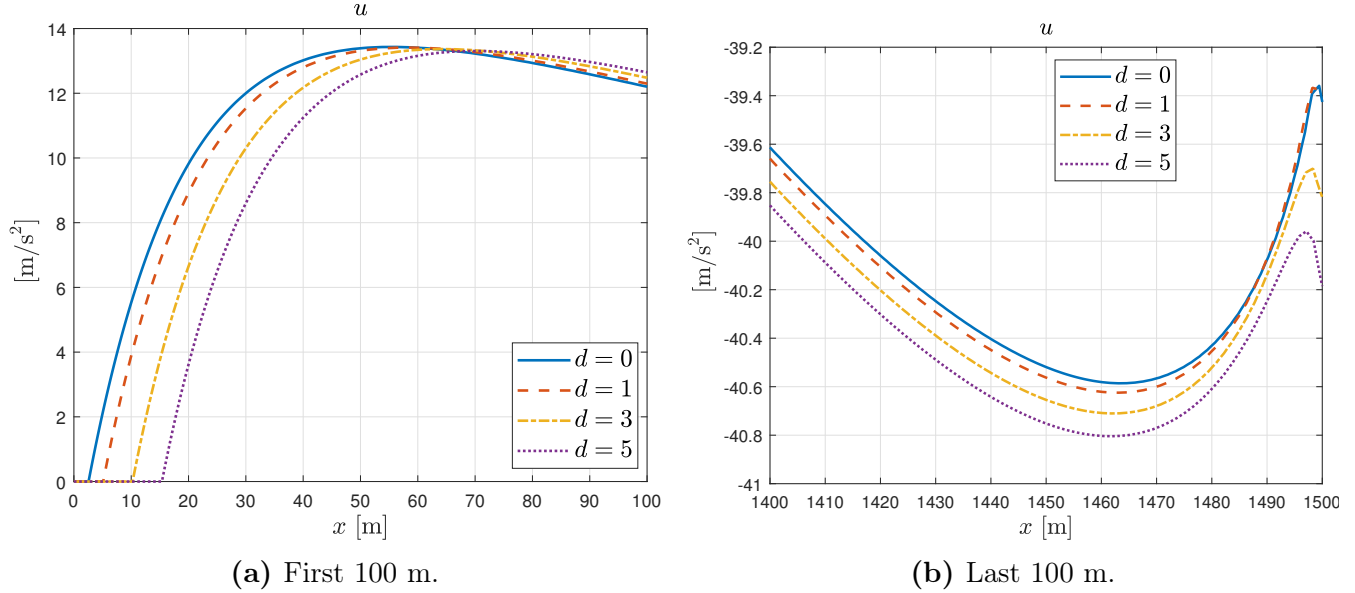


Figure 3.26: Control-input profile for different delay levels d in the optimal guidance law (3.75).

This conclusion is not unexpected, considering the nature of the variation of the optimal guidance parameters, which for the considered scenario is shown in Fig. 3.27. Namely, these parameters remain approximately constant during most of the flight. The exception is the final part of the trajectory, where more significant changes occur, typically within 10–20 m from the target. The onset of this region depends on the specific scenario: for more demanding maneuvers, the rapid variation occurs earlier in the flight.

Nevertheless, demanding maneuvers are not only a consequence of larger impact angles and smaller relative distances at which terminal guidance begins, but also of an imperfect control system. Any delay in executing the commanded input will lead to a deviation of the actual trajectory from the nominal one. If the required normal acceleration is larger, the influence of delay on this deviation is also greater. A larger deviation leads to a larger command, and thus a positive feedback effect may arise, resulting in continuously increasing required normal acceleration and trajectory deviation.

In the numerical simulation used in this chapter, the non-ideal behavior of the control system is represented by a first-order transfer function (3.2) with time constant τ . Figure 3.28 shows the variation of optimal guidance parameters for the previously considered scenario, but for different values of the control-system time constant. It can be observed that larger variations occur earlier than in the previous case ($\tau = 0.1$ s for the nominal case), which is consistent with the conclusions drawn earlier.

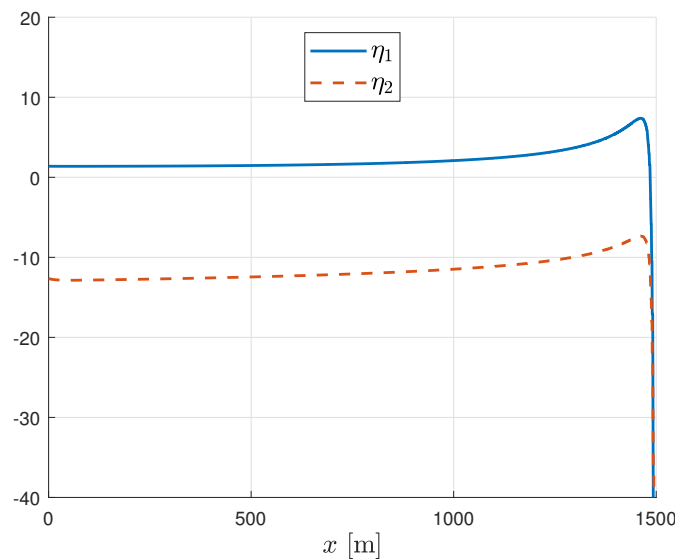


Figure 3.27: Variation of optimal guidance parameters along the trajectory for the scenario described in Table 3.1.

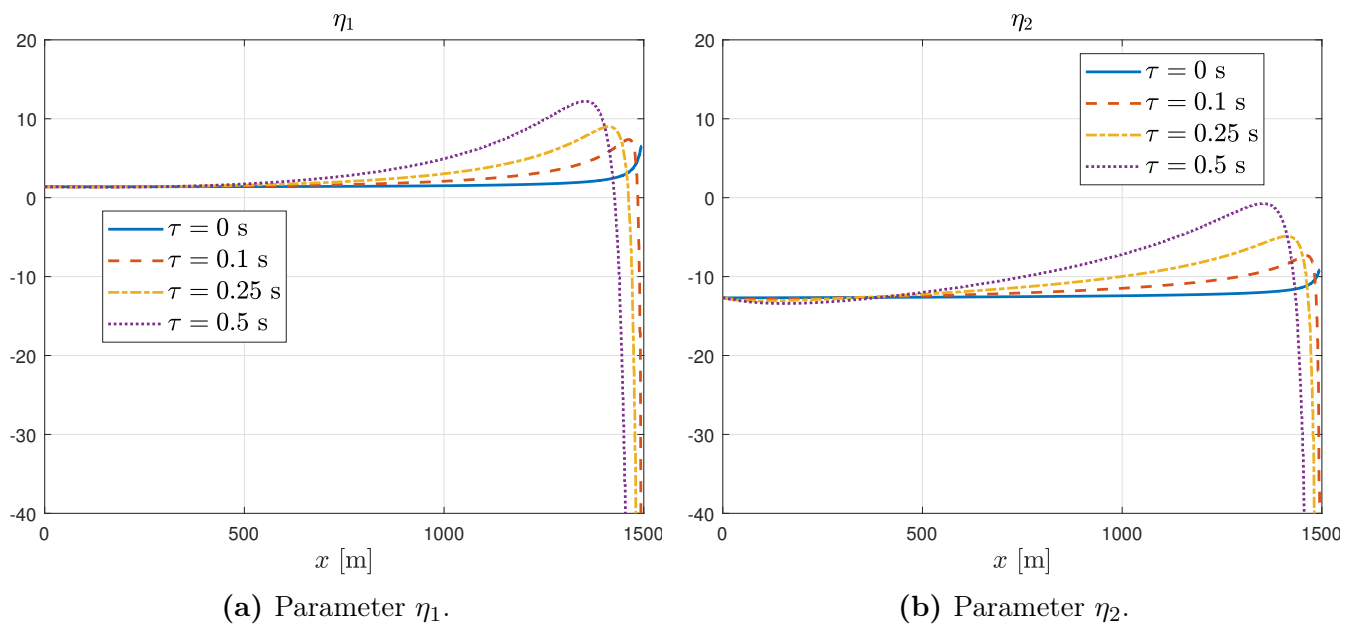


Figure 3.28: Variation of optimal guidance parameters for different values of the time constant τ in the control-system transfer function.

Figure 3.29 shows the control-input profile for the considered values of the time constant τ . By comparison with Fig. 3.26, it is evident that delay in command execution has a much greater influence on the overall trajectory than delay in updating the parameters η_1 and η_2 . This is expected, since the guidance cycle is only 16 ms, so a delay of a few cycles does not significantly affect the nominal behavior.

For achieving optimal trajectories, it is much more important to properly design the control system in order to reduce the delay in tracking the commanded input, which is generally valid regardless of the adopted guidance law.

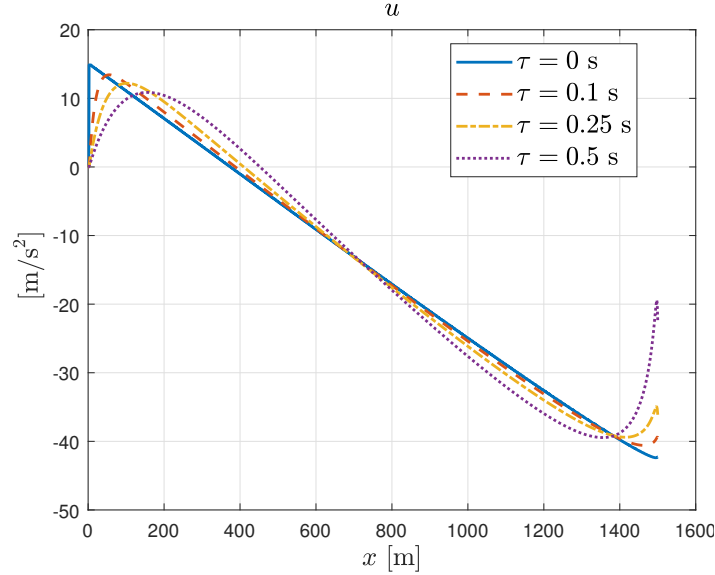


Figure 3.29: Control-input profile for different values of the time constant τ in the control-system transfer function.

3.3 Hardware Implementation

In this part of the dissertation, the applicability of the previously presented algorithms on real hardware is analyzed. The existing systems for which the proposed algorithms are primarily intended employ flight computers based on 32-bit LPC1768 microcontrollers with an ARM Cortex-M3 processor architecture. Consequently, all preliminary investigations discussed in this section were performed on a development board equipped with this microcontroller, shown in Fig. 3.30.

The aforementioned microcontroller does not include a specialized module for performing floating-point arithmetic operations (*Floating Point Unit - FPU*); therefore, all computations are performed using single-precision (*single*) arithmetic.

Since the proposed optimal guidance algorithm is executed during flight using updated initial conditions at each guidance cycle, its implementation does not involve the risk of error accumulation due to numerical rounding (as is the case with navigation algorithms). The only operation where such a risk may be present is the iterative solution of the nonlinear equation $\mathcal{N}(\alpha) = 0$, whose accuracy is primarily affected by the accuracy of the integral I_1 .

Figure 3.31 shows the absolute error in determining the integral I_1 using Algorithm 3 and a 10-node Gauss–Legendre quadrature formula, but employing single-precision (*single*) arithmetic. The error was obtained by comparison with the value of the integral calculated using a 1000-node quadrature formula and double-precision (*double*) arithmetic.

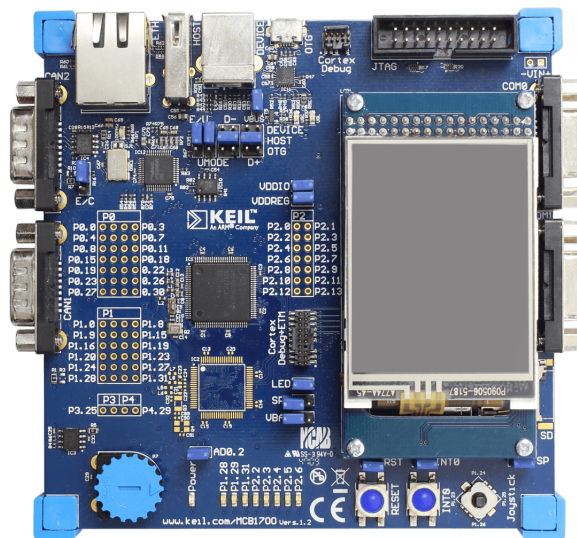


Figure 3.30: MCB1700 development board with an LPC1768 microcontroller.

A comparison between Figs. 3.31 and 3.18 reveals no significant loss of accuracy; therefore, it is justified to continue the subsequent analysis using single-precision arithmetic.

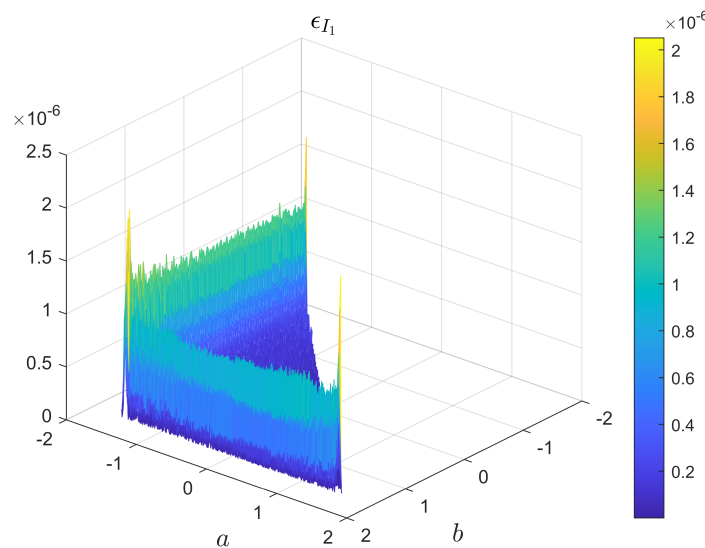


Figure 3.31: Absolute error in determining the integral I_1 using Algorithm 3, a 10-node quadrature formula, and single-precision arithmetic.

In addition to the number of arithmetic operations, the total execution time of the algorithm is influenced by the execution time of an individual operation. Since the iterative procedure for solving the nonlinear equation accounts for the majority of the computational effort, estimating the overall execution time requires estimating the execution time of a single iteration.

In Section 3.2.2, several methods for solving a nonlinear equation of a single variable were analyzed, and the bisection method was ultimately adopted. The reason for this choice is that the bisection method requires the largest number of iterations. Furthermore, it was shown that other methods

may, during certain flight intervals, require the same number of iterations as the bisection method. Therefore, successful implementation of the bisection method guarantees the feasibility of other, more efficient methods that require fewer iterations.

During a single iteration of the bisection method, only one evaluation of the function whose root is sought is required. The procedure for evaluating the function $\mathcal{N}(\alpha)$ is given in Algorithm 5. The input argument `alg` specifies the trajectory convexity case:

$$\text{alg} \in \{A, B, C\} \quad (3.76)$$

and is used when computing the integral I_2 according to the procedure described by Algorithm 6.

Algorithm 5 Evaluation of the function $N = \mathcal{N}(\alpha, \text{alg}, \Delta x, \Delta y, \gamma_0, \gamma_1)$ according to (2.53), (2.59), (2.60)

```

1:  $a \leftarrow \gamma_0 - \alpha$ 
2:  $b \leftarrow \gamma_1 - \alpha$ 
3: if alg = A then
4:    $\mathcal{I}_1 \leftarrow I_1(a, b)$ 
5:    $\mathcal{I}_2 \leftarrow I_2(a, b, \text{alg})$ 
6: else if alg = B then
7:    $\mathcal{I}_1 \leftarrow I_1(a, \pi/2) + I_1(b, \pi/2)$ 
8:    $\mathcal{I}_2 \leftarrow I_2(a, \pi/2, \text{alg}) + I_2(b, \pi/2, \text{alg})$ 
9: else
10:   $\mathcal{I}_1 \leftarrow I_1(a, -\pi/2) + I_1(b, -\pi/2)$ 
11:   $\mathcal{I}_2 \leftarrow I_2(a, -\pi/2, \text{alg}) + I_2(b, -\pi/2, \text{alg})$ 
12: end if
13:  $c_\alpha \leftarrow \cos(\alpha)$ 
14:  $s_\alpha \leftarrow \sin(\alpha)$ 
15:  $I_N \leftarrow c_\alpha \cdot \mathcal{I}_1 - s_\alpha \cdot \mathcal{I}_2$ 
16:  $I_D \leftarrow c_\alpha \cdot \mathcal{I}_2 + s_\alpha \cdot \mathcal{I}_1$ 
17:  $N \leftarrow \Delta x \cdot I_D - \Delta y \cdot I_N$ 

```

Algorithm 6 Evaluation of the integral $I = I_2(a, b, \text{alg})$ given by (2.44)

```

1:  $a \leftarrow \gamma_0 - \alpha$ 
2: if alg = A then
3:    $b \leftarrow \gamma_1 - \alpha$ 
4:    $I \leftarrow 2 \left( \sqrt{\cos(a)} - \sqrt{\cos(b)} \right)$ 
5: else
6:    $I \leftarrow 2\sqrt{\cos(a)}$ 
7: end if

```

Since the integral I_2 possesses an analytical solution, its evaluation is straightforward and requires only a small number of calls to standard mathematical library functions. The execution time of the function for evaluating the integral I_2 is approximately 25 μs for Case A and approximately 13 μs for Cases B and C. Considering that, according to Algorithm 5, two evaluations of the integral I_2 are required in Cases B and C, the execution time is approximately the same for all trajectory convexity cases.

This implies that the execution time of a single iteration is dominated by the procedure for evaluating the integral I_1 , described by Algorithm 3.

For the numerical evaluation of the integral I_1 using the adopted 10-node Gauss–Legendre quadrature formula, ten evaluations of the integrand $\sqrt{\cos(x)}$ are required. Table 3.3 presents the execution times of the elementary functions required for evaluating this integrand. The functions `cosf(x)` and `sqrtof(y)` belong to the standard mathematical library for the selected processor.

Table 3.3: Execution times of the elementary functions used in the evaluation of the integral I_1 . $x = 1.23456f$, $y = 0.32993874f$.

Function	Execution time [μs]	Relative error
<code>cosf(x)</code>	11	4.8236×10^{-8}
<code>fast_cosf_6(x)</code>	6	3.9848×10^{-4}
<code>fast_cosf_8(x)</code>	7	6.7344×10^{-6}
<code>fast_cosf_10(x)</code>	9	1.3059×10^{-7}
<code>sqrtof(y)</code>	2	-

It can be observed that the execution time of the `sqrtof(y)` function is negligible, whereas the execution time of `cosf(x)` is considerable, particularly given that this function must be called ten times for a single evaluation of the integral I_1 . Therefore, Table 3.3 also includes, in addition to the standard library functions, the execution times of functions that approximate the original cosine function by means of polynomials designed to minimize the maximum approximation error over a specified interval (minimax polynomials). A more detailed description of this approximation technique is provided in Appendix B.3.

In addition to execution time, the table also presents the relative error obtained when evaluating the function $\cos(x)$ using different methods for the value $x = 1.23456$. The error was calculated relative to the value obtained using double-precision arithmetic.

Different polynomial orders result in different execution times. Lower-order polynomials yield shorter execution times but larger relative errors. The final polynomial order must therefore be selected as a compromise between the conflicting requirements of high accuracy and low processor load. Figure 3.32 shows the absolute error in determining the integral I_1 using Algorithm 3, a 10-node quadrature formula, single-precision arithmetic, and minimax approximations of different polynomial orders.

It can be observed that, as in the previous analyses, the overall error profile remains unchanged. Over most of the integration interval, the error remains acceptable even when a sixth-order approximation polynomial is used. However, near the interval boundaries, the error becomes more than 100 times greater than the error shown in Fig. 3.31. For the eighth-order polynomial, the maximum error is on the order of 10^{-5} , which is consistent with the permissible error level presented in Fig. 3.11. Using a tenth-order polynomial further reduces the maximum absolute error over the considered interval, but the execution time becomes comparable to that of the standard `cosf(x)` function. For this reason, the eighth-order minimax polynomial, implemented as the function `fast_cosf_8(x)`, was adopted for evaluating $\cos(x)$ in the remainder of this work.

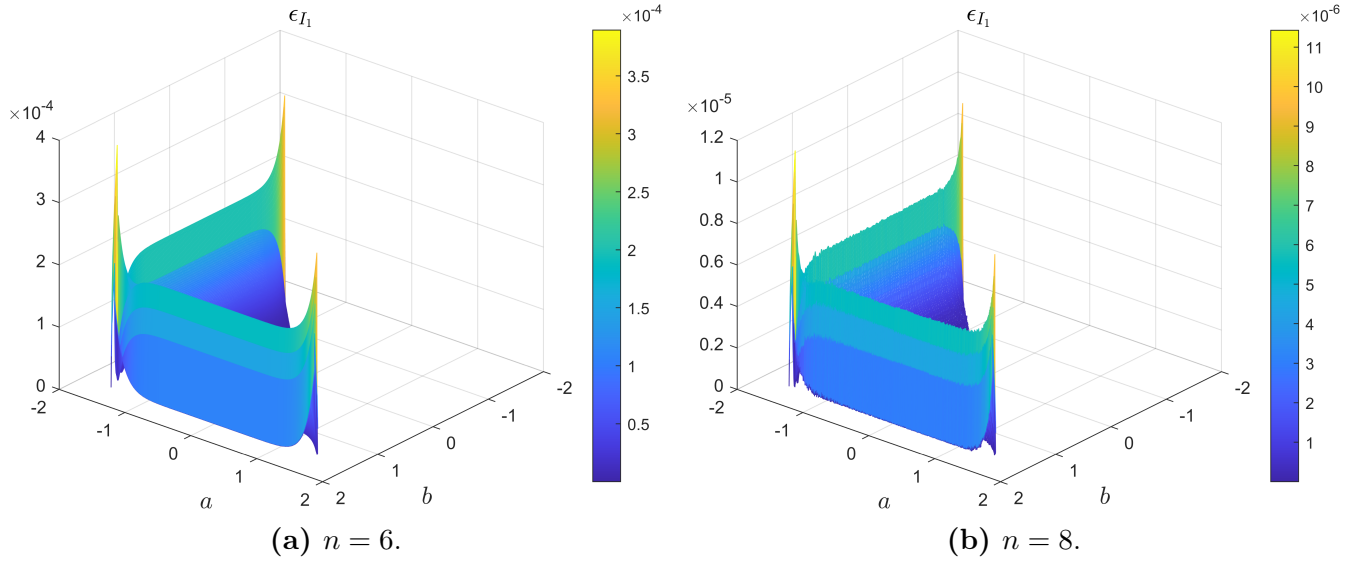


Figure 3.32: Absolute error in determining the integral I_1 using Algorithm 3, a 10-node quadrature formula, single-precision arithmetic, and minimax polynomial approximations of different orders n .

An additional advantage over standard library functions lies in avoiding unnecessary checks typically performed by such functions, as well as avoiding argument mapping to a predefined interval. Since, during the numerical evaluation of the integral I_1 , the argument of the integrand is always confined to the interval $[-\pi/2, \pi/2]$, these additional operations are not required.

Table 3.4 compares the execution times of the function used to evaluate the integral I_1 via Algorithm 3, employing either the standard `cosf(x)` function or the eighth-order minimax polynomial. The reported function calls correspond to Case A, where integration is performed over an arbitrary interval $I_1(a, b)$, as well as Cases B and C, where two integral evaluations are required. The reported execution times are consistent with the measurements presented in Table 3.3: a single evaluation of the integral I_1 requires ten executions of the functions `fast_cosf_8(x)` and `sqrtf(y)`, resulting in an execution time of approximately 100 μs .

Table 3.4: Execution time of the function used to evaluate the integral I_1 by Algorithm 3. Integration limits: $a = -1.5$, $b = 1.5$.

Function	Execution time [μs]	
	<code>cosf(x)</code>	<code>fast_cosf_8(x)</code>
<code>integral_I1(a,b)</code>	221	202
<code>integral_I1(a,PIover2)</code>	112	107
<code>integral_I1(b,PIover2)</code>	111	107
<code>integral_I1(-PIover2,a)</code>	114	105
<code>integral_I1(-PIover2,b)</code>	114	104

Finally, Table 3.5 presents the execution time of the function `numeratorFunction(alpha,alg)`, which evaluates the function $\mathcal{N}(\alpha, \text{alg})$ according to the procedure described in Algorithm 5. For simplicity of notation, the dependence on the remaining flight parameters (Δx , Δy , γ_0 , γ_1) is omitted here, since they do not affect the structure of the function and are common to all cases for

a given value of the argument `alg`. Execution times are shown for the three different trajectory convexity cases, and in all three cases the execution time is similar: approximately 300 μs when the standard `cosf(x)` function is used, and approximately 260 μs when the function `fast_cosf_8(x)` is employed.

Table 3.5: Execution time of the function $\mathcal{N}(\alpha)$, defined by Algorithm 5, for `alpha` = 1.2217.

Function	Execution time [μs]	
	<code>cosf(x)</code>	<code>fast_cosf_8(x)</code>
<code>numeratorFunction(alpha,alg_A)</code>	297	258
<code>numeratorFunction(alpha,alg_B)</code>	303	264
<code>numeratorFunction(alpha,alg_C)</code>	301	263

A comparison of the execution times obtained using the standard `cosf(x)` function and `fast_cosf_8(x)` may lead to the misleading conclusion that the reduction in computational time is relatively small—only about 40 μs . However, it should be noted that this time corresponds to a single evaluation of the function used to determine the value of $\mathcal{N}(\alpha)$. During the solution of the nonlinear equation $\mathcal{N}(\alpha) = 0$ by the bisection method, 22 iterations are required to achieve the desired accuracy, i.e., 22 evaluations of the function $\mathcal{N}(\alpha)$. Consequently, this difference represents a significant saving in computational time.

Nevertheless, it would also be incorrect to conclude that the computation of the optimal guidance command requires only $22 \times 260 \mu\text{s} \approx 5.7 \text{ ms}$ and that this value can therefore be adopted as the guidance system update period. The proposed algorithm is intended for application in real systems whose flight computers, in addition to the guidance function, perform numerous other tasks required for successful mission execution. Examples include execution of the navigation algorithm, computation of actuator commands, communication with other subsystems, data acquisition and transmission, and so forth. Therefore, successful implementation of the guidance algorithm requires consideration of the main flight-computer operating algorithm.

Since the guidance system requires information about both the missile and the target, i.e., their relative position, to compute the guidance command, it is most reasonable to synchronize the guidance system update rate with that of the navigation system. In other words, the guidance command should be updated immediately after the missile position estimate has been updated. The inertial navigation algorithm for cruise and ballistic missiles, for which the guidance algorithm developed in this dissertation is primarily intended, is described in detail in [99] and [100]. The algorithm is divided into two parts: the first operates at a frequency of 500 Hz and is responsible for determining velocity and angle increments, while the second operates eight times more slowly, at 62.5 Hz, and is responsible for determining the missile velocity and position. [99]

This means that the main flight-computer cycle is 2 ms, and that eight such cycles are required to complete one full navigation and guidance cycle. Each of these eight intervals is allocated to the execution of different functions, either in their entirety or in part, with the objective of achieving a balanced processor workload within the flight computer [100].

To verify the feasibility of implementing the proposed algorithm in real time on the existing hardware platform, it is first necessary to estimate the available computational resources of the

current systems. For this purpose, the execution time of the main flight-computer algorithm must be measured in an environment that reproduces flight conditions as closely as possible.

For laboratory evaluation of the Guidance, Navigation, and Control (GNC) section, a group of tests is employed in which all GNC subsystems are used in their final configuration (immediately prior to installation in the missile), while flight conditions are simulated using auxiliary equipment. For example, actuator subsystem testing is performed using specialized test benches that subject the actuator subsystem to the same loads encountered during flight; the angular motion of the missile, i.e., the section containing inertial sensors and/or the seeker, is reproduced using three-axis motion platforms; the relative angular motion between the missile and the target is simulated using five-axis platforms; and so on.

Because these tests employ the actual hardware that will ultimately be installed in the vehicle, they belong to the category of *Hardware-In-the-Loop (HIL)* testing. A special case within this category involves testing only the flight computer, without sensors, control subsystems, or any physical motion. Since this approach evaluates the functionality of the flight-computer algorithms, i.e., its software², such tests belong to the category of *Software-In-the-Loop (SIL)* testing.

A common characteristic of both testing approaches (*SIL* and *HIL*) is that, in addition to the basic subsystems required for mission initialization and launch (ground station), a real-time missile flight simulation and data-exchange system is also required.

Figure 3.33 schematically illustrates the network of devices used for real-time testing of the proposed algorithm. Serial communication between the onboard computer (OBC) and the ground station is performed via the *RS232* protocol. This communication is required only for mission initialization and launch and is an integral part of the system even under operational conditions.

Missile flight simulation is performed on a computer specifically designed for real-time operation, on which a six-degree-of-freedom flight simulation is executed. Data exchange with the flight computer is performed bidirectionally via the *Ethernet* protocol at a frequency of 500 Hz:

- The flight computer receives acceleration and angular-rate data (quantities normally measured by the inertial measurement unit), altitude data (normally measured by the altimeter), and geographical position data (optionally, normally obtained from a satellite-navigation receiver);
- The flight computer transmits the computed control-surface deflection commands.

Finally, the transmission of data required for analysis to the data-acquisition computer is also performed via the *Ethernet* protocol at a frequency of 500 Hz. During the measurement of the execution time of the main flight-computer algorithm, the time required for transmitting these data was not included, since this operation is not performed under actual flight conditions. As in previous measurements, the built-in timer was used for timing purposes, and the measured execution time for each cycle was transmitted with every new data packet.

²Naturally, under the assumption that the flight-computer hardware itself has already been verified.

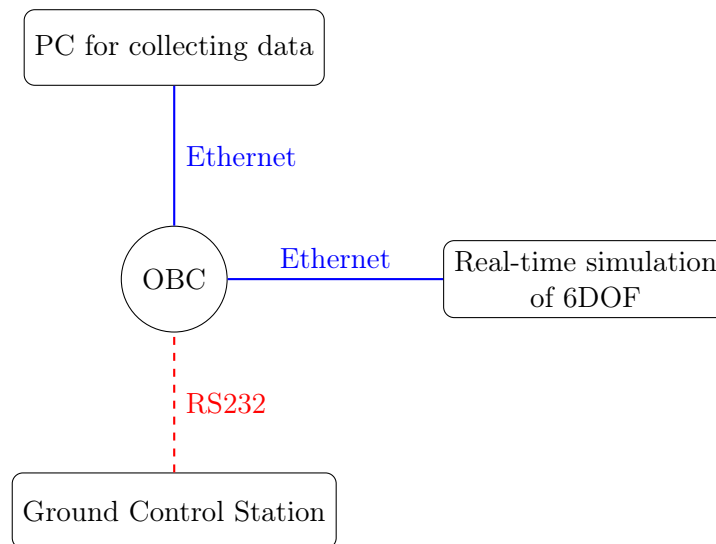


Figure 3.33: Schematic representation of the device network used during *SIL* testing.

For the purposes of the analysis presented in this section, a cruise missile was first considered that, after launch, transitions to horizontal flight at a predefined altitude and remains in cruise flight until the distance to the target becomes less than 800 m. At that point, the missile enters the terminal guidance phase, where proportional navigation is employed as the baseline guidance method. For a missile without a seeker, all quantities required for the implementation of this guidance law are calculated from information available from the navigation system.

Figure 3.34 shows the measured execution time of the main flight computer algorithm for a target engagement scenario with a range of 6400 m. This execution time is approximately constant throughout the entire flight, amounting to about 450 μs , and does not differ significantly between the cruise phase and the terminal guidance phase.

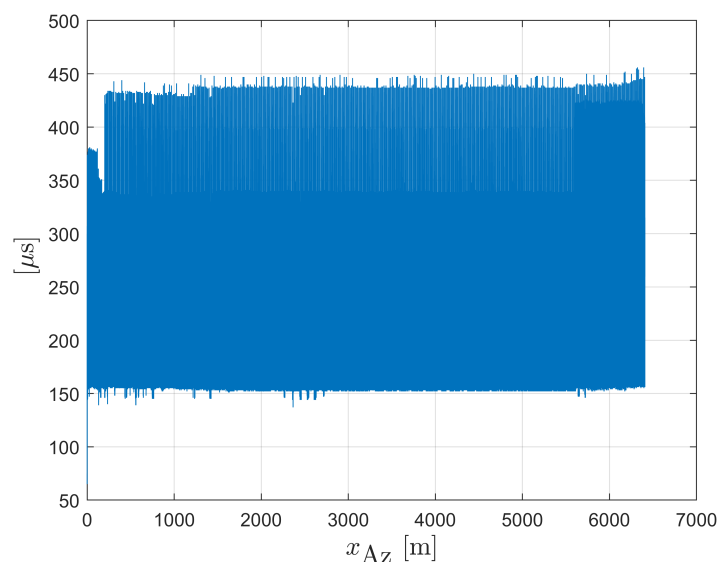


Figure 3.34: Execution time of the main flight computer algorithm when proportional navigation is used as the terminal guidance method of a cruise missile.

Considering that the execution period of the main flight computer algorithm is 2 ms, the measured execution time indicates that approximately 1.5 ms is available for the implementation of additional functions. However, the measured time refers only to the main part of the algorithm, i.e., the functions associated with navigation, guidance, and control, and does not include interrupt service routines or communication with other missile subsystems. Therefore, to improve robustness, it is necessary to maintain a computational time margin so that all operations are completed before the next execution cycle of the main algorithm.

Since the procedure for determining the optimal guidance command proposed in this work is iterative, it is desirable to utilize as much computational time as possible during each cycle in order to obtain the command as quickly as possible. Nevertheless, there is no benefit in calculating the guidance command before new information about the missile position becomes available, which is determined by the update rate of the navigation system. Consequently, the optimal guidance command must be updated every 16 ms, corresponding to the navigation data update frequency of 62.5 Hz. This means that eight cycles of the main algorithm are available for implementing the new guidance algorithm.

It was previously shown that solving the nonlinear equation iteratively requires 21–22 iterations. Therefore, to determine the guidance command in real time, it is necessary to perform three iterations per cycle, ensuring that the root of the function $\mathcal{N}(\alpha)$ is found within eight cycles. Furthermore, performing three iterations per cycle does not require the entire duration of all eight cycles, making it possible to use the first cycle for determining the trajectory convexity case, which also requires evaluations of the function $\mathcal{N}(\alpha)$ and therefore additional execution time.

Figure 3.35 schematically illustrates the timing of the main algorithm around the k -th cycle. For simplicity, all quantities determined by the navigation algorithm are represented by the state vector $\mathbf{x}$, while the parameters of the optimal guidance law (3.1) are represented by the vector $\boldsymbol{\eta}$. On the horizontal axis representing time, the 2 ms main clock cycles are labeled with indices 0–7, after which a 16 ms cycle repeats. The earliest cycle within one period at which position information becomes available depends on the specific navigation algorithm. For the case considered here, this occurs during the fourth cycle, and therefore the updated navigation information, denoted by $\mathbf{x}_k$, is shown at that instant. During the same cycle, it is immediately possible to calculate the guidance command using the currently available state information but with the previously computed guidance parameters: $u(\mathbf{x}_k, \boldsymbol{\eta}(\mathbf{x}_{k-1}))$. At the same time, the computation of the new values of $\boldsymbol{\eta}$ begins. These parameters will be used during the $(k+1)$ -th guidance cycle, immediately after the new state values $\mathbf{x}_{k+1}$ become available, since one complete update cycle of the optimal guidance parameters lasts exactly eight clock cycles.

In this way, the command is always computed in real time using the latest position information, while the optimal guidance parameters are updated with a delay of only one cycle. Since the computation of their new values does not require the entire eight-cycle interval, the control command could alternatively be calculated after the parameters $\boldsymbol{\eta}$ have been updated within the same cycle, resulting in a control law of the form $u(\mathbf{x}_k, \boldsymbol{\eta}(\mathbf{x}_k))$. However, it was previously demonstrated in Section 3.2.5 that even a delay of several cycles has a negligible effect. Therefore, it is not justified to postpone the guidance command update while the parameters $\boldsymbol{\eta}$ are being computed, since the command can already be determined using the current missile position and the previously calculated guidance parameters while preserving the real-time operation property.

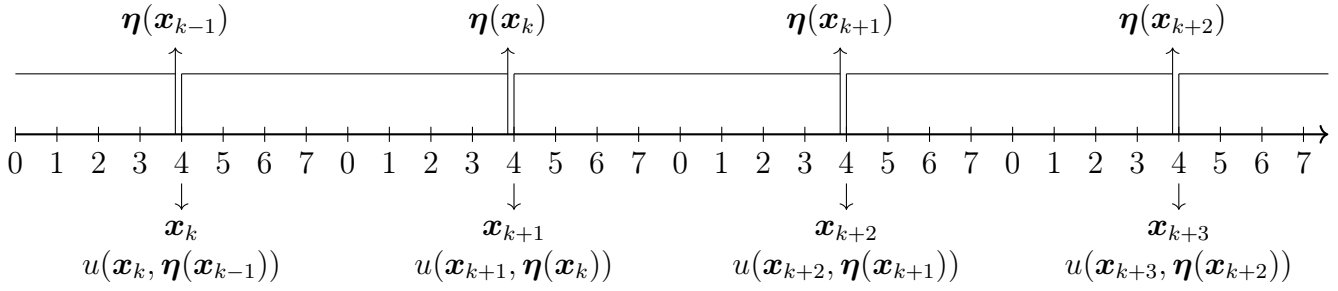


Figure 3.35: Schematic flow of the main algorithm with update of guidance parameters using three iterations per basic cycle.

Figure 3.36a presents the measured execution time of the main flight computer algorithm for the previously analyzed scenario when the optimal guidance method proposed in this work is employed, using three iterations per basic cycle. The optimal guidance phase begins at a distance of 3 km from the target, at which point the execution of Algorithm 2 starts. It can be observed that the execution time during the terminal guidance phase is now approximately 1230 μs , which is consistent with the measured execution times summarized in Table 3.5: the baseline 450 μs is increased by three evaluations of the function $\mathcal{N}(\alpha)$, each requiring approximately 260 μs .

This approach provides a computational time margin of nearly 800 μs , which is sufficient for all additional flight computer functions as well as potential future extensions of system functionality.

Additional processor load reduction can be achieved by decreasing the number of iterations performed during a single cycle of the main flight computer algorithm. In that case, solving the equation $\mathcal{N}(\alpha) = 0$ will take longer, but the available computational time margin within each cycle will increase. For the systems considered in this work, this is not necessary because a satisfactory margin has already been achieved. Nevertheless, due to its advantages, such an approach is also presented here.

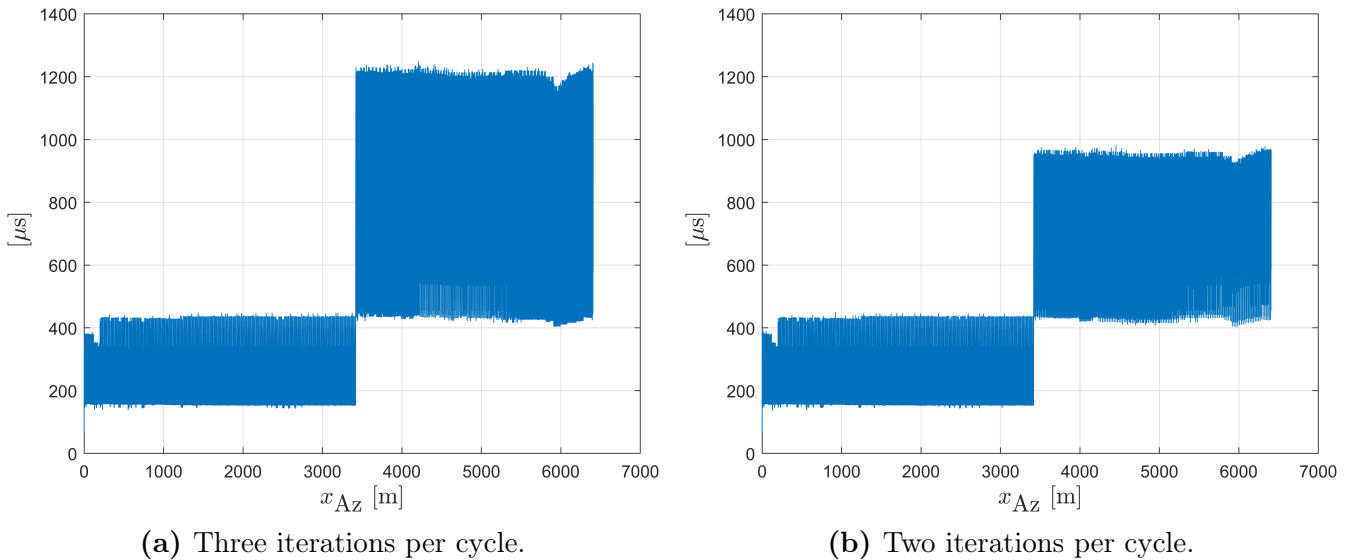


Figure 3.36: Comparison of the execution time of the main flight computer algorithm when optimal guidance is implemented with different numbers of iterations per main algorithm cycle.

Figure 3.37 shows the timing of the main algorithm when two iterations are performed per cycle. In this case, 11 cycles are required to find the root of the function $\mathcal{N}(\alpha)$, while one additional cycle is allocated for the complete update of the guidance parameters, as in the previous case. Since these 12 cycles exceed the duration of a single navigation cycle, the guidance command is computed using values of the parameters $\boldsymbol{\eta}$ that were updated more than one cycle earlier. Specifically, after the parameters have been updated twice, the entire sequence repeats beginning from the fourth cycle of a given navigation period, and the guidance law uses parameter values that were updated no more than three cycles earlier:

$$u(\mathbf{x}_k, \boldsymbol{\eta}(\mathbf{x}_{k-d})), \quad \max(d) = 3 \quad (3.77)$$

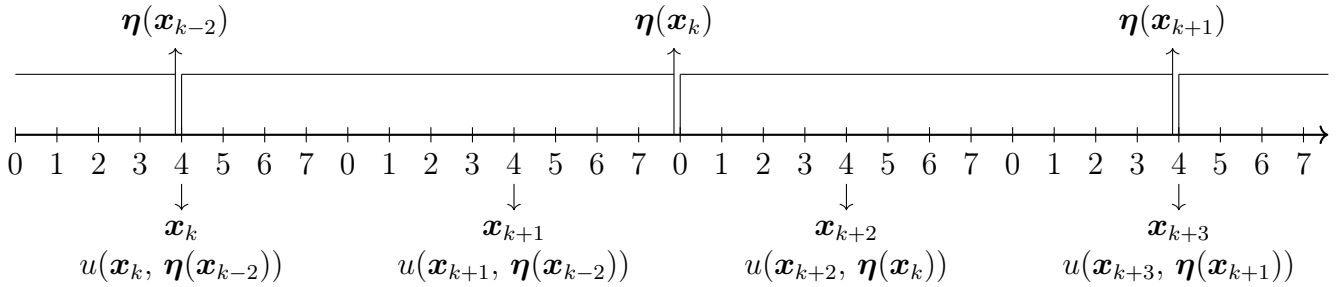


Figure 3.37: Schematic flow of the main algorithm with update of guidance parameters using two iterations per basic cycle.

Figure 3.36b shows the measured execution time of the main flight computer algorithm for the previously analyzed scenario when the optimal guidance procedure is implemented using two iterations per basic step. The execution time during the terminal guidance phase is now approximately 970 μs , which is consistent with the measured values summarized in Table 3.5: in addition to the baseline 450 μs , two evaluations of the function $\mathcal{N}(\alpha)$ are performed, each requiring approximately 260 μs .

In this way, the computational time margin is increased to more than 1 ms, which corresponds to half of the main flight computer basic cycle, thereby providing the possibility of doubling the execution rate of the main algorithm. This increase comes at the cost of a larger delay in updating the optimal guidance parameters, which in the worst case amounts to three cycles. However, it was already shown in Section 3.2.5 that even a delay of five cycles in updating these parameters has no significant influence, and that the greatest advantage of the proposed guidance approach is that the guidance command is calculated in real time using the current missile position.

The same analysis was also performed on the flight computer of a ballistic missile, which uses a similar main algorithm structure based on a 2 ms clock period and navigation data updates every eighth cycle. Compared with the previously considered cruise missile, the ballistic missile flight computer communicates with a smaller number of sensors and subsystems, and employs a different guidance law based on *Impact Point Prediction (IPP)*. Consequently, the execution time of the main algorithm, shown in Fig. 3.38a, differs from that presented in Fig. 3.34.

When the optimal guidance algorithm is implemented using two iterations per basic cycle, the execution time increases to approximately 920 μs , as shown in Fig. 3.38b. This result is again

consistent with the values given in Table 3.5: in addition to the baseline $400 \mu\text{s}$, two evaluations of the function $\mathcal{N}(\alpha)$ are required, each taking approximately $260 \mu\text{s}$. In this case, optimal guidance is applied from the very beginning of the flight, unlike the cruise missile scenario in which terminal guidance begins only when the missile reaches a predefined distance from the target.

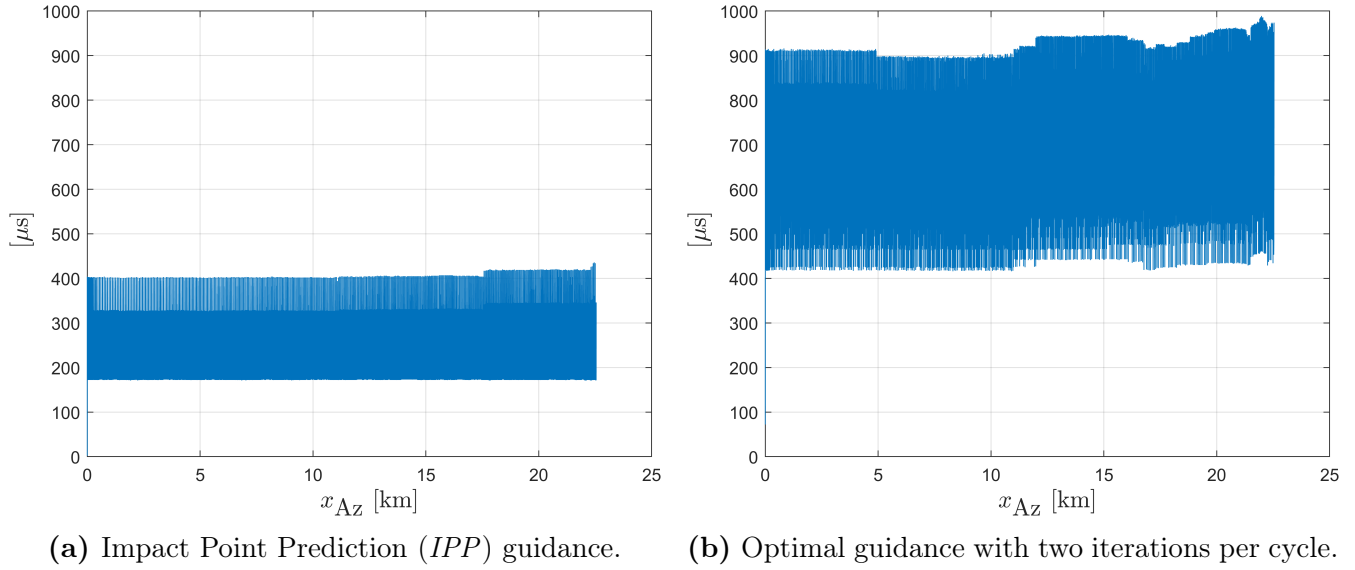


Figure 3.38: Execution time of the main flight computer algorithm of a ballistic missile when different guidance methods are employed.

In both considered examples, optimal guidance was implemented only in the vertical plane, while the guidance command in the horizontal plane was determined using the standard procedure adopted for the particular missile. Since the guidance law in the horizontal plane may have a mixed structure (for a cruise missile, for example, predefined path following during the cruise phase and proportional navigation during the terminal phase), the execution time of the main algorithm may vary throughout the flight. This effect is more noticeable in the ballistic missile example (Fig. 3.38b) than in the cruise missile case (Fig. 3.36), but it is important to emphasize that this variability is not caused solely by the optimal guidance algorithm, which has an approximately fixed execution time (the same number of iterations and approximately the same computational effort in each iteration).

The objective of this part of the research was to investigate the feasibility of implementing the proposed optimal guidance algorithm on flight computers of real systems. The evaluation was performed in a laboratory environment using actual hardware, while the operating conditions were simulated through a six-degree-of-freedom flight simulation executed on a dedicated real-time computer. Although the algorithm was successfully validated in such an environment, other SIL test results (trajectory, commanded and achieved acceleration, velocity, etc.) were not presented here. Now that the final structure of the algorithm has been established, it is possible to perform its verification once again³ through numerical simulations and statistical analysis. This is the subject of the next section, where all quantities relevant to flight performance assessment will be considered.

³The initial verification was carried out using two-degree-of-freedom flight simulations.

3.4 Numerical Simulation of Missile Flight with Six Degrees of Freedom

In this part of the dissertation, the proposed optimal guidance algorithm is analyzed through six-degree-of-freedom (6DOF) missile flight simulations.

Two types of guided missiles are considered: a cruise missile and a ballistic missile. In both cases, the optimal guidance law is implemented only in the vertical plane. Guidance in the horizontal plane is performed using the guidance law already employed by the respective missile.

The mathematical model of six-degree-of-freedom missile flight and the coordinate systems used are described in Appendix C. Since the previous analyses in this dissertation employed a two-degree-of-freedom kinematic model, the optimal guidance command must be adapted to the new model, which is applicable during actual flight.

Specifically, the control variable selected in this work is the missile normal acceleration, denoted by u in Fig. 3.39. The missile, represented by the x and $\tilde{z}$ axes of the aeroballistic coordinate frame, may assume an arbitrary orientation relative to the velocity vector V . This orientation is defined by the angle of attack α , which can be determined from the flight-path angle γ and the pitch angle θ as:

$$\alpha = \theta - \gamma \quad (3.78)$$

The estimated values of the angles θ and γ are available from the navigation algorithm; therefore, the angle of attack can also be estimated during flight using the above expression.

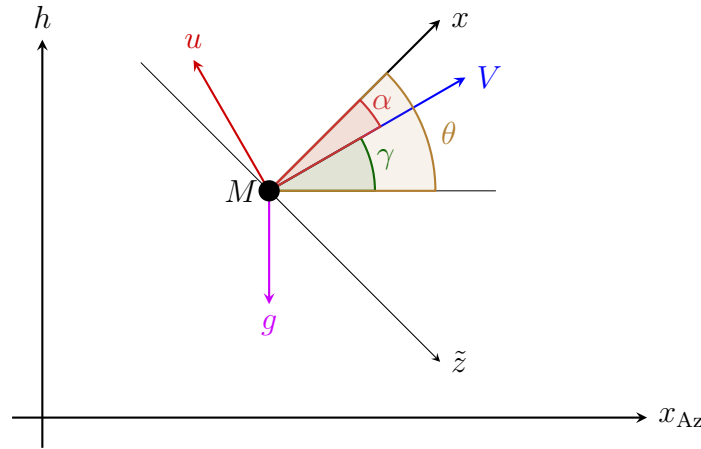


Figure 3.39: Kinematic quantities relevant to guidance in the vertical plane.

In addition to the control input obtained by solving the optimization problem, the commanded specific force in the vertical plane must also include compensation for gravitational acceleration. Its projection onto the $\tilde{z}$ -axis of the aeroballistic coordinate frame is therefore:

$$f_{\tilde{z},\text{cmd}} = -(u \cos(\theta - \gamma) + g \cos \theta) \quad (3.79)$$

If the projection of the commanded specific force in the horizontal plane onto the $\tilde{y}$ -axis of the aeroballistic frame is denoted by $f_{\tilde{y},\text{cmd}}$, then the projections of the commanded specific force onto the body frame axes are:

$$f_{z,\text{cmd}} = f_{\tilde{z},\text{cmd}} \cos \phi - f_{\tilde{y},\text{cmd}} \sin \phi \quad (3.80a)$$

$$f_{y,\text{cmd}} = f_{\tilde{z},\text{cmd}} \sin \phi + f_{\tilde{y},\text{cmd}} \cos \phi \quad (3.80b)$$

where ϕ is the missile roll angle.

Both considered missiles are roll-stabilized, i.e., the commanded roll angle is zero and is regulated by the roll autopilot. The commands calculated using (3.80) are applied as input signals to the pitch and yaw autopilots. The autopilots, as well as the overall control system implementation, are described in Appendix C.2.

Unless stated otherwise, all quantities required for the implementation of the guidance law are assumed to be known exactly. The influence of navigation algorithm errors is investigated separately, and in those cases it is explicitly indicated which quantity is affected by estimation error. Furthermore, whenever information about the missile velocity is required for the implementation of the guidance law, the velocity component projected onto the plane to which the corresponding guidance command applies is used.

To assess optimality, the following performance index, representing the control effort, is employed:

$$J = \int_{t_0}^{t_1} f_z^2(t) dt \quad (3.81)$$

where the time instants t_0 and t_1 are selected according to the specific scenario under consideration (and will be specified where appropriate).

The performance index is defined in terms of the specific non-gravitational force rather than the total normal acceleration (which also includes the contribution of gravity) in order to isolate only the control effort generated by the vehicle itself.

3.4.1 Cruise Missile

In this section, the application of the proposed optimal guidance algorithms to a cruise missile is analyzed.

A typical mission profile of this missile consists of three phases, clearly visible in Fig. 3.40:

- Launch and climb to the desired cruise altitude;
- Cruise phase / tracking of a predefined trajectory;
- Terminal guidance toward the target.

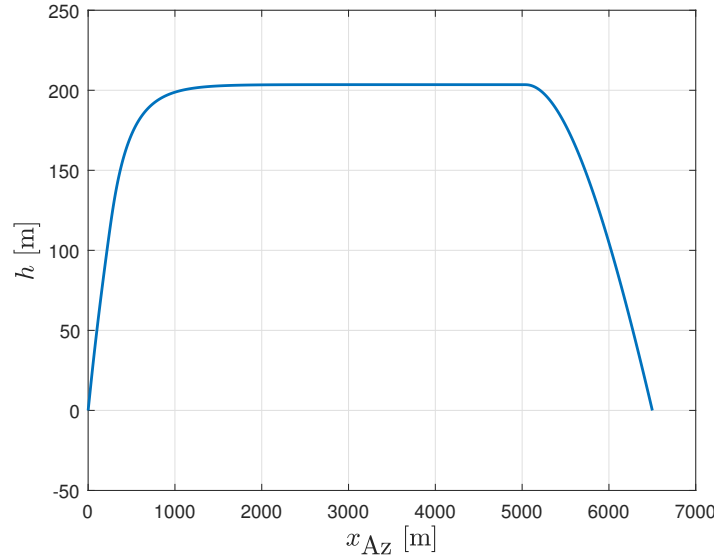


Figure 3.40: Typical trajectory of a cruise missile engaging a target at a range of 6.5 km.

Since the cruise phase, i.e., predefined trajectory tracking (Trajectory Tracking – TT), is not the focus of this dissertation, only the initial and terminal portions of the trajectory are analyzed in detail. The guidance law employed during the cruise phase was presented in [101], while its application to a cruise missile was analyzed in greater detail in [99], including examples involving tracking trajectories of various shapes. For the special case of straight-line tracking, the guidance law reduces to the following form:

$$u = K_d d + K_{\dot{d}} \dot{d} \quad (3.82)$$

where d denotes the distance to the desired trajectory, as illustrated in Fig. 3.41, while K_d and $K_{\dot{d}}$ are the guidance-law gains. The values of these gains used in the flight simulations, for both the vertical and horizontal planes, are given in Table 3.6.

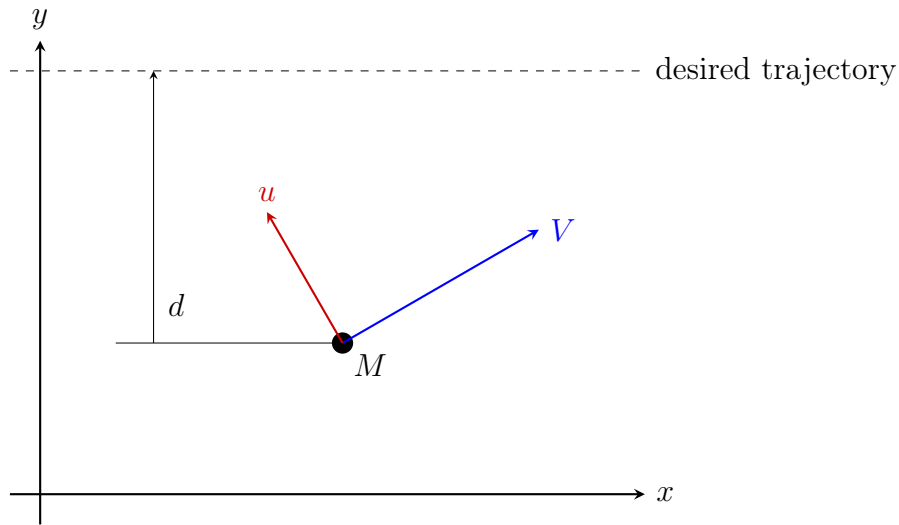


Figure 3.41: Kinematic model of straight-line tracking used for implementation of the guidance law (3.82).

Table 3.6: Values of the gains in the guidance law (3.82) used in the numerical simulations.

Gain	Vertical plane	Horizontal plane
K_d	0.4	0.15
K_i	1.2	0.75

The commanded specific force during the cruise phase, expressed through its projections onto the axes of the aeroballistic frame, is given by

$$(f_{\bar{z},\text{cmd}})_{\text{TT}} = -[(0.4\Delta h - 1.2\dot{h}) \cos(\theta - \gamma) + g \cos \theta] \quad (3.83a)$$

$$(f_{\bar{y},\text{cmd}})_{\text{TT}} = -0.15y_{\text{Az}} - 0.75\dot{y}_{\text{Az}} \quad (3.83b)$$

where Δh denotes the difference between the desired cruise altitude and the current missile altitude:

$$\Delta h = h_{\text{des}} - h \quad (3.84)$$

while $\dot{h}$ represents the rate of change of altitude.

In (3.83b), which defines the commanded specific force in the horizontal plane, it is assumed that the desired trajectory is the straight line connecting the launch point and the target. In other words, the objective during the cruise phase is for the missile to remain "on course". Since the launch-point-to-target direction is defined by the x_{Az} -axis of the azimuth coordinate system, deviations from this direction are measured by the coordinate y_{Az} .

The correction of the commanded specific force due to the misalignment between the missile body axis and the velocity vector, caused by the angle of attack in the vertical plane and the sideslip angle in the horizontal plane, is taken into account only in (3.83a). This approach was adopted in order to maintain consistency between the cruise-phase guidance law and the guidance laws used during the initial climb and terminal guidance phases, where achieving a desired velocity-vector orientation at a specific point along the trajectory is important. Since no such requirement exists in the horizontal plane, and the missile response characteristics (in terms of the time required to return to the desired course after a disturbance) are not of primary importance, the basic form of the guidance law given by (3.82) is used in that plane.

Climb Phase

Based on the optimal guidance law for the climb phase proposed in Section 2.2.5, given by Eq. (2.85), the optimal commanded specific force in the vertical plane is:

$$(f_{\bar{z},\text{cmd}})_{\text{OPT}} = \left[\frac{V^2 \sqrt{\sin \gamma}}{h_{\text{des}} - h} I_S(\gamma) \right] \cos(\theta - \gamma) - g \cos \theta \quad (3.85)$$

where $I_S(\gamma)$ is the integral defined by expression (3.51), computed using Algorithm 4.

During the climb phase, the same guidance law form as in the cruise phase, given by expression (3.83b), was used in the horizontal plane.

Figure 3.42 compares the results obtained using the trajectory-tracking guidance law (3.83a) and the optimal guidance law (3.85) for a launch elevation angle of 35° and a climb toward a cruise altitude of $h_{\text{des}} = 200$ m. Only the portion of the trajectory up to a range of 1.5 km is shown, corresponding to the climb phase and the initial part of the horizontal-flight phase. Both guidance laws are activated once the missile altitude exceeds 100 m, while the commanded specific force is zero before that point.

When the trajectory-tracking (TT) guidance law is used, the commanded specific force is calculated in the same manner throughout the first two flight phases. Since the optimal guidance law (3.85) is intended only for the climb phase, it is necessary to switch guidance laws at some point during the first phase and begin trajectory tracking. In the presented example, the termination condition for the climb phase was chosen as a deviation of 5 m from the desired altitude. This transition occurred at a range of approximately $x_{\text{Az}} \approx 592$ m, when the flight-path angle was about 3° . On the specific-force diagram, this transition is visible as a change in the slope of the curve, which had been approximately linear up to that point, consistent with the results previously presented in Section 2.2.5.

It can also be observed that the optimal guidance law produces a smaller initial command, thereby reducing the maneuverability demands on the missile during the initial portion of the flight. The reduced maneuvering effort is also reflected in the values of the control-effort performance index, defined by (3.81), which for the considered trajectory segment are:

$$J_{\text{TT}} = 1169 \text{ m}^2/\text{s}^3, \quad J_{\text{OPT}} = 1058 \text{ m}^2/\text{s}^3 \quad (3.86)$$

The larger values of commanded specific force during the initial phase when using the trajectory-tracking law (3.83a) are a consequence of the large initial altitude error Δh . As the desired cruise altitude increases, this error also increases, making the advantage of optimal guidance even more pronounced.

For this reason, when comparing these guidance laws for higher cruise altitudes, the trajectory-tracking guidance law is activated only after the altitude error decreases below 100 m, whereas optimal guidance begins as soon as the missile altitude exceeds 100 m. Such a comparison is shown in Fig. 3.43 for a desired cruise altitude of 300 m. All other parameters remain identical to those used in the example shown in Fig. 3.42.

Although, in the example of climbing to 300 m, trajectory-tracking guidance (TT) is activated only after the altitude error becomes smaller than 100 m, the control effort is still more than twice that obtained with optimal guidance due to the initial jump in commanded specific force:

$$J_{\text{TT}} = 505 \text{ m}^2/\text{s}^3, \quad J_{\text{OPT}} = 216 \text{ m}^2/\text{s}^3 \quad (3.87)$$

Therefore, for larger cruise altitudes there is little value in further comparing the guidance laws (3.83a) and (3.85). Instead, it is advantageous to use them in a combined manner: optimal guidance during the climb phase and trajectory tracking during the cruise phase.

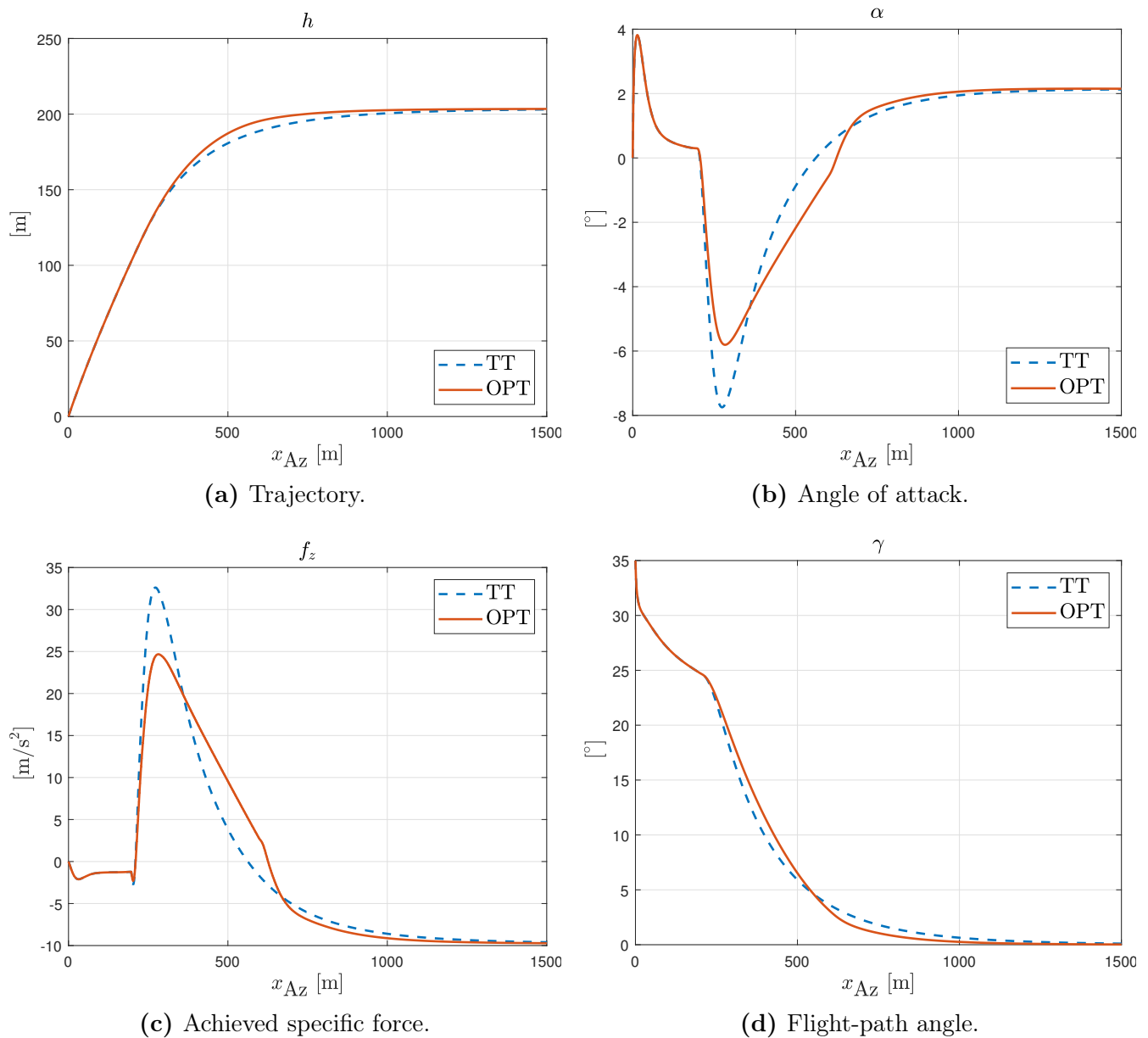


Figure 3.42: Comparison of results obtained using the trajectory-tracking guidance law (3.83a) (TT) and the optimal guidance law (3.85) (OPT) for a desired cruise altitude of $h_{\text{des}} = 200$ m.

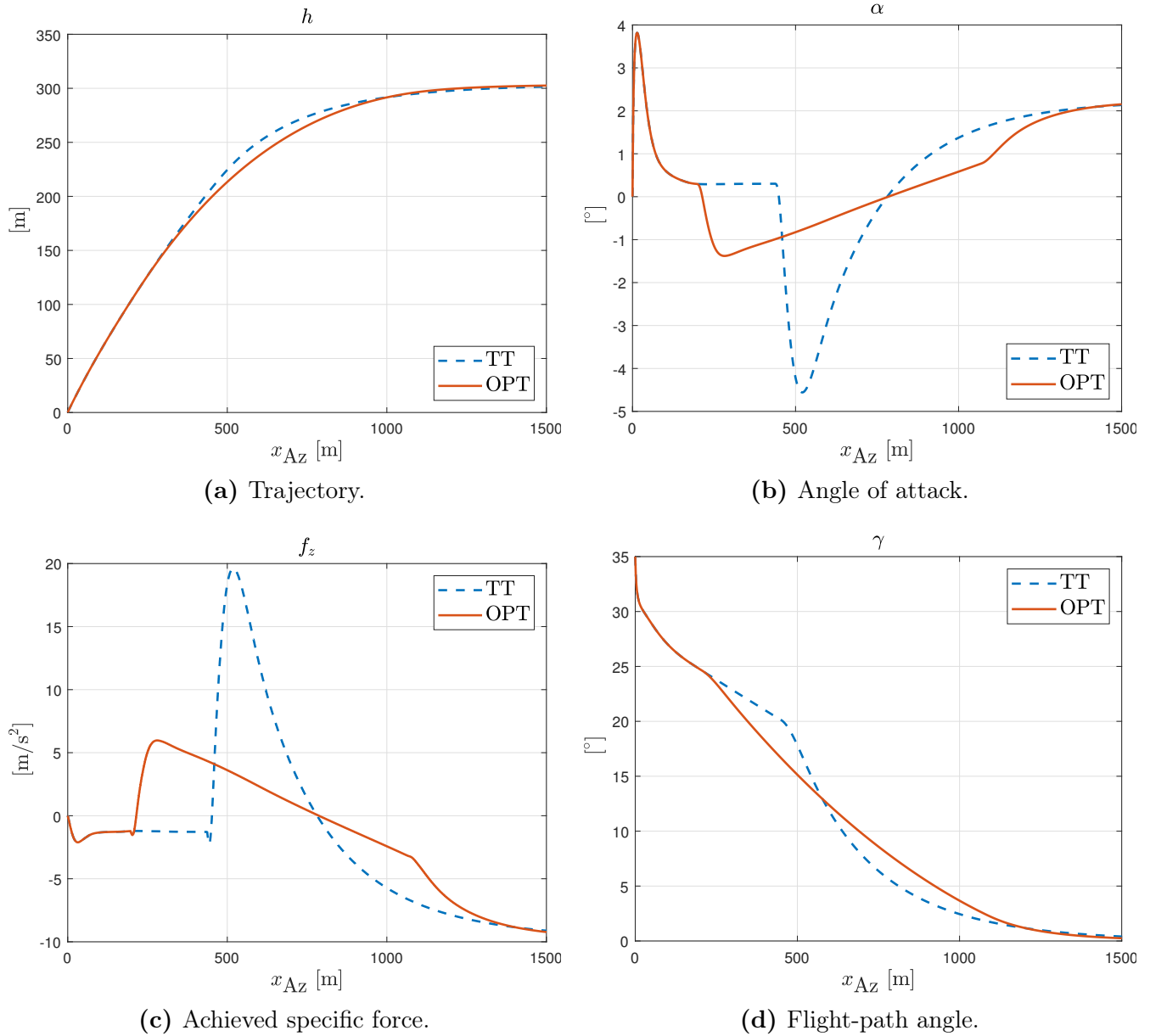


Figure 3.43: Comparison of results obtained using the trajectory-tracking guidance law (3.83a) (TT) and the optimal guidance law (3.85) (OPT) for a desired cruise altitude of $h_{\text{des}} = 300$ m.

Figure 3.44 compares the results obtained using the optimal guidance law (3.85) during climbs to different cruise altitudes. The initial elevation angle is 45° for all shown trajectories; therefore, increasing the desired cruise altitude leads to a reduction in the control effort (since the selected elevation angle is well suited to this set of altitudes):

$$J_{400} = 1073 \text{ m}^2/\text{s}^3, \quad J_{500} = 707 \text{ m}^2/\text{s}^3, \quad J_{600} = 510 \text{ m}^2/\text{s}^3, \quad J_{700} = 399 \text{ m}^2/\text{s}^3 \quad (3.88)$$

However, for a cruise altitude of 400 m, a smaller elevation angle would certainly be more appropriate, since the specific-force diagram reveals a more pronounced command at the beginning

of the guidance phase. Nevertheless, selecting the optimal elevation angle is not the subject of this analysis, whose purpose is to demonstrate the applicability of the proposed algorithm under different boundary conditions.

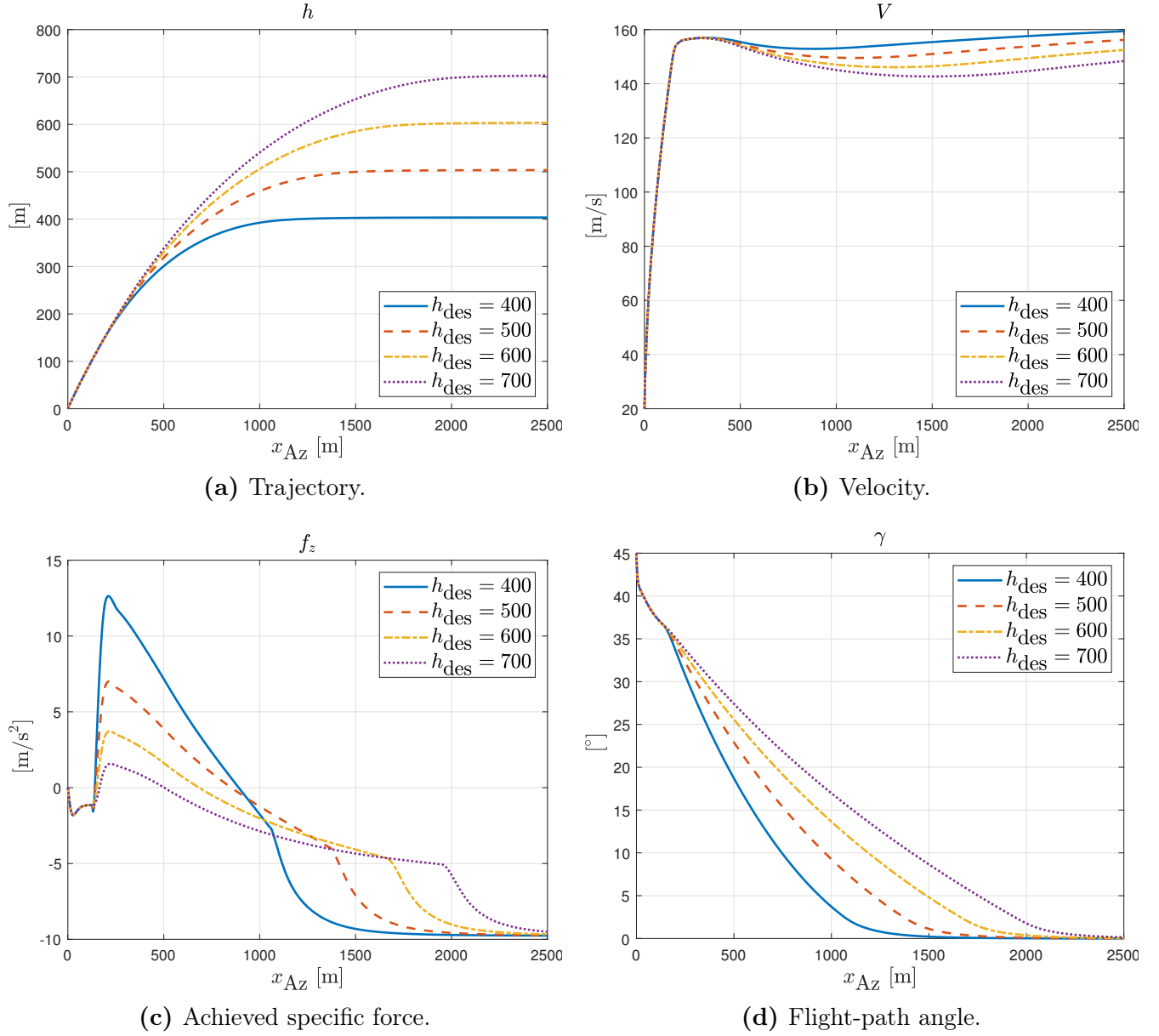


Figure 3.44: Comparison of results obtained during optimal climbs to different cruise altitudes.

In addition to the elevation angle, another parameter of the initial flight phase that must be selected is the altitude deviation at which the guidance algorithm switches from optimal guidance (3.85) to trajectory tracking (3.83a). One possibility is to perform the transition exactly when the desired altitude is reached. However, considering that the proposed method is intended for real systems, imperfect tracking of the commanded input may prevent the missile from reaching the exact desired altitude. It is therefore safer to define a transition zone around the cruise altitude, such that trajectory tracking begins once the missile enters this zone.

Figure 3.45 presents the results for two values of this altitude deviation. It can be observed that a smaller deviation leads to a more pronounced change in specific force and causes the missile to pitch up during the transition. Since such behavior is undesirable when transitioning to horizontal flight, the final condition adopted for terminating the optimal climb phase is $\Delta h_{\text{TT}} = 5$ m.

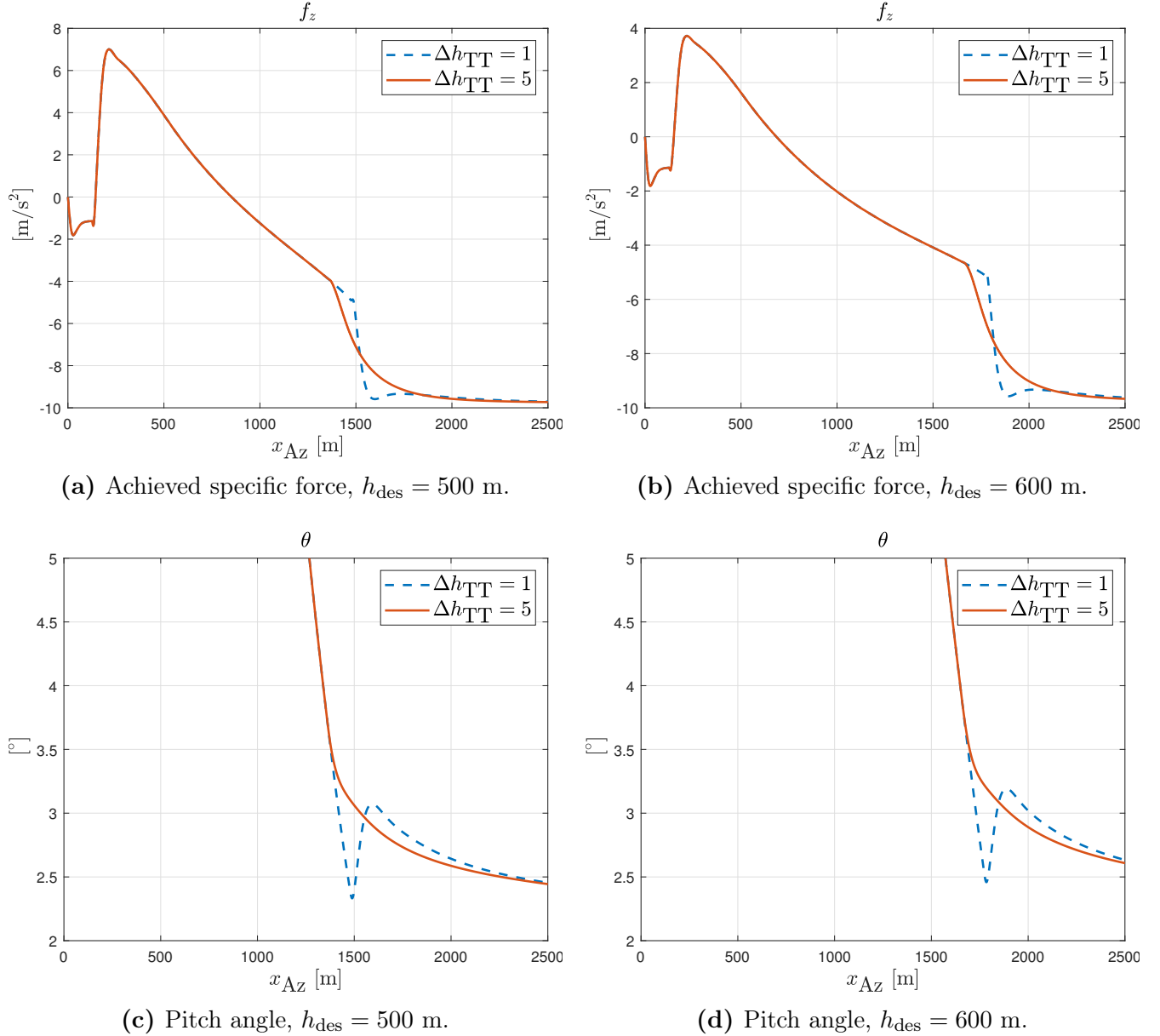


Figure 3.45: Comparison of results obtained during optimal climbs to different cruise altitudes h_{des} and different values of the altitude deviation Δh_{TT} used for switching to trajectory tracking.

In all examples presented so far, the exact value of the flight-path angle γ was used for implementing the optimal climb guidance law (3.85). This angle is determined from the velocity components in the local (*NED*) coordinate system:

$$\gamma = -\arcsin \frac{v_D}{V} \quad (3.89)$$

and its accuracy depends on the accuracy of the navigation data. Therefore, Fig. 3.46 presents results obtained using optimal guidance but with an erroneous estimate of the flight-path angle, varying according to:

$$\hat{\gamma} = \gamma \mp \frac{5}{8}t \quad (3.90)$$

This variation was selected so that, at the moment when the desired altitude is reached under nominal conditions ($t \approx 8$ s), the error in the estimated flight-path angle equals 5° .

Results are shown for both negative and positive errors. It can be observed that when the estimated flight-path angle is smaller than the actual one, the missile still succeeds in reaching the desired altitude. The difference relative to the nominal trajectory is the presence of altitude overshoot. This is expected because applying the guidance law (3.85) with a flight-path angle smaller than the true value causes the missile to attempt to increase this angle, resulting in an approach to the desired altitude with a steep flight-path angle and subsequent overshoot during the transition to horizontal flight.

On the other hand, if the estimated flight-path angle is larger than the actual one, the guidance law (3.85) generates commands that reduce the true flight-path angle, causing the trajectory to level off prematurely. In that case, the missile may fail to reach the altitude required to initiate trajectory tracking and may even begin descending, which is precisely the situation shown in Fig. 3.46d.

The linear error variation given by (3.90) was adopted to represent the case in which the flight-path angle must be estimated by a navigation algorithm whose error grows with flight time, as is typical of an inertial navigation system. If another source for determining velocity components in the local coordinate system is available (for example, satellite navigation), a different error model, such as a constant error, would be more appropriate. Nevertheless, it is clear that regardless of the error magnitude, the mission can be jeopardized only when the estimated flight-path angle is greater than the actual one.

Although an error of 5° is unrealistically large for the relatively short duration of the climb phase, this example demonstrates that a protective mechanism should be incorporated when implementing the guidance law (3.85) on real systems in order to prevent unsuccessful transitions into the cruise phase.

The simplest and most reliable way to protect against large errors in the estimated flight-path angle is to impose a time limit on the application of the optimal guidance law. Namely, it is known that the climb phase, regardless of the guidance law employed, cannot last longer than a certain time under nominal conditions. The nominal climb duration increased by a specified margin can therefore be used as a criterion for switching to the trajectory-tracking guidance law, regardless of whether the originally defined criterion has been satisfied.

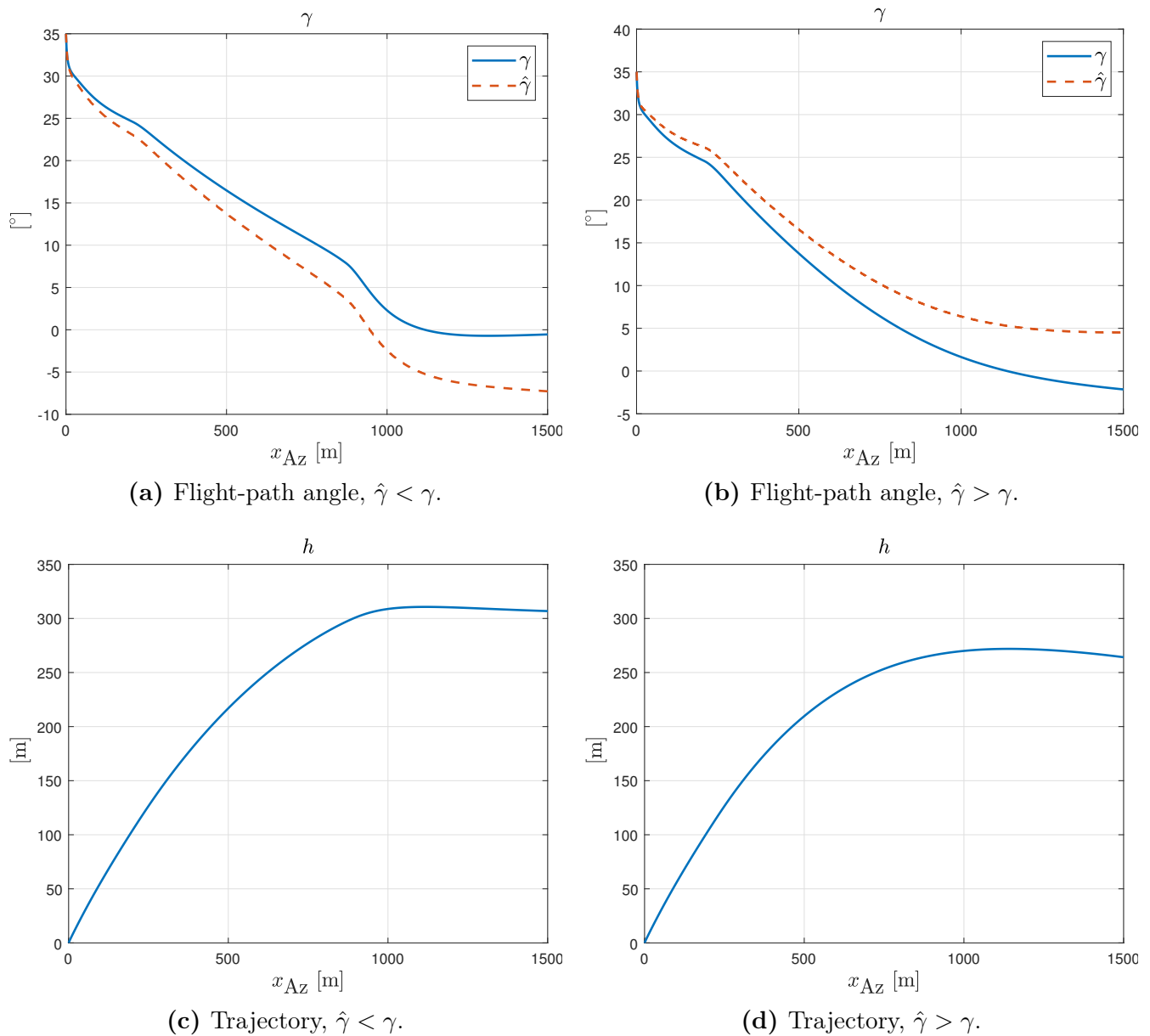


Figure 3.46: Comparison of results obtained during optimal climb in the presence of flight-path-angle estimation errors.

Figure 3.47 presents a comparison of the results obtained using the optimal guidance law during the climb phase, in the presence of a positive error in the flight-path-angle estimate (the case $\hat{\gamma} > \gamma$), with and without a time-based criterion for switching to trajectory tracking. Since the climb duration under nominal conditions is approximately 8 s, the moment after which the application of the optimal guidance law is terminated regardless of the achieved altitude is selected as $t = 10$ s.

In this way, even after the missile begins to lose altitude, switching to the trajectory-tracking guidance law (3.83a), which does not depend on the accuracy of the flight-path-angle estimate, enables the desired cruise altitude to be reached and the mission to continue.

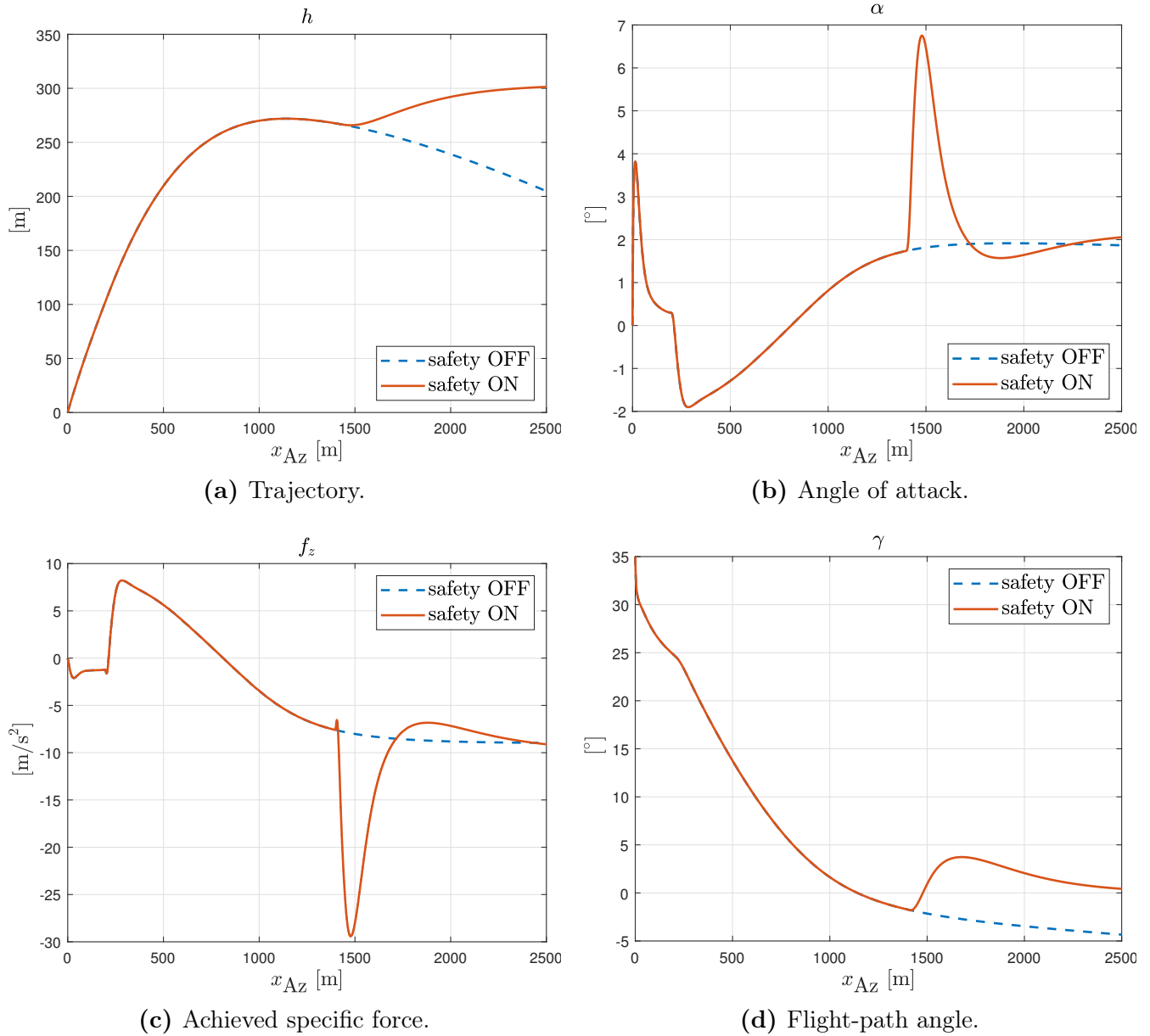


Figure 3.47: Comparison of the results obtained using the guidance law (3.85) in the presence of a positive flight-path-angle estimation error ($\hat{\gamma} > \gamma$), with (safety ON) and without (safety OFF) a time-based criterion for switching to trajectory tracking.

Another approach for preventing adverse effects caused by large errors in the flight-path-angle estimate is to use the pitch angle instead of the flight-path angle in the optimal guidance law (3.85). Figure 3.48 compares the results obtained using the exact value of the flight-path angle and its approximation by the pitch angle. The differences in the resulting trajectories are practically negligible, which is confirmed by the similar values of the control effort:

$$J_{\hat{\gamma}=\gamma} = 216 \text{ m}^2/\text{s}^3, \quad J_{\hat{\gamma}=\theta} = 225 \text{ m}^2/\text{s}^3 \quad (3.91)$$

Such an approach is justified when the pitch angle is known with greater accuracy than the flight-path angle. The reliability of this approximation is reflected in the fact that, under optimal guidance, the angle of attack, shown in Fig. 3.43b, remains small.

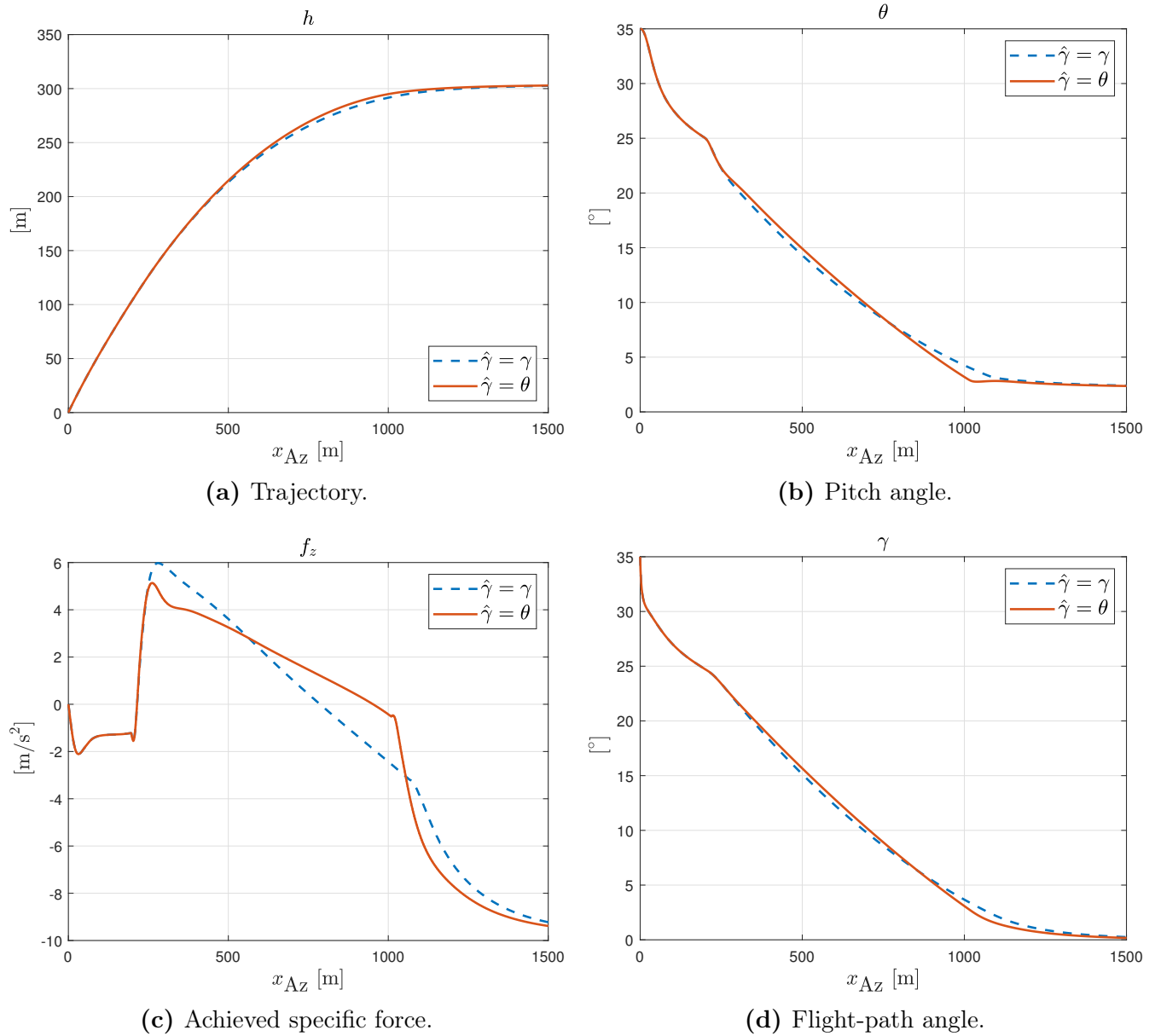


Figure 3.48: Comparison of optimal climb results obtained using the exact flight-path angle and its approximation by the pitch angle.

Terminal Guidance

The baseline guidance law employed on the considered cruise missile during the terminal guidance phase is proportional navigation, given by Eq. (1.63). Therefore, the commanded specific force in the two planes is given by:

$$(f_{\ddot{z},\text{cmd}})_{\text{PN}} = -N'V\dot{\lambda}_{\text{ver}}\cos(\theta - \gamma) - g\cos\theta \quad (3.92a)$$

$$(f_{\ddot{y},\text{cmd}})_{\text{PN}} = N'V\dot{\lambda}_{\text{hor}} \quad (3.92b)$$

In the above equations, $\dot{\lambda}_{\text{ver}}$ and $\dot{\lambda}_{\text{hor}}$ denote the vertical and horizontal components of the line-of-sight rate, respectively, which are calculated using Eq. (2.93). Since only stationary targets are considered, these components are computed as follows:

$$\dot{\lambda}_{\text{ver}} = \frac{\dot{x}_{\text{Az}}\Delta h - \dot{h}\Delta x}{(\Delta x)^2 + (\Delta h)^2} \quad (3.93a)$$

$$\dot{\lambda}_{\text{hor}} = \frac{-\dot{x}_{\text{Az}}y_{\text{Az}} - \dot{y}_{\text{Az}}\Delta x}{(\Delta x)^2 + y_{\text{Az}}^2} \quad (3.93b)$$

where Δx and Δh denote the relative missile position with respect to the target in the azimuth coordinate system:

$$\Delta x = (x_{\text{Az}})_{\text{T}} - x_{\text{Az}} \quad (3.94a)$$

$$\Delta h = (h)_{\text{T}} - h \quad (3.94b)$$

The adopted navigation constant for command generation in both planes is $N' = 3.3$.

The optimal guidance law proposed in this dissertation is given by Eq. (2.26), and the corresponding optimal command of specific force in the vertical plane is:

$$(f_{\ddot{z},\text{cmd}})_{\text{OPT}} = -\frac{1}{2V}(\eta_1\Delta h - \eta_2\Delta x - \eta_3)\cos(\theta - \gamma) - g\cos\theta \quad (3.95)$$

while the baseline proportional-navigation law (3.92b) is retained in the horizontal plane.

The parameters η_1 , η_2 , and η_3 in Eq. (3.95) are determined during each guidance cycle using Algorithm 2 and the procedures described in Section 3.3.

Figure 3.49 compares the nominal trajectory corresponding to a range of 6.5 km and a cruise altitude of 300 m obtained using proportional navigation (3.92a) with the trajectory generated by the optimal guidance law (3.95) for the desired impact angle $\gamma_1 = -45^\circ$. In both cases, terminal guidance begins at a distance of 2 km from the target.

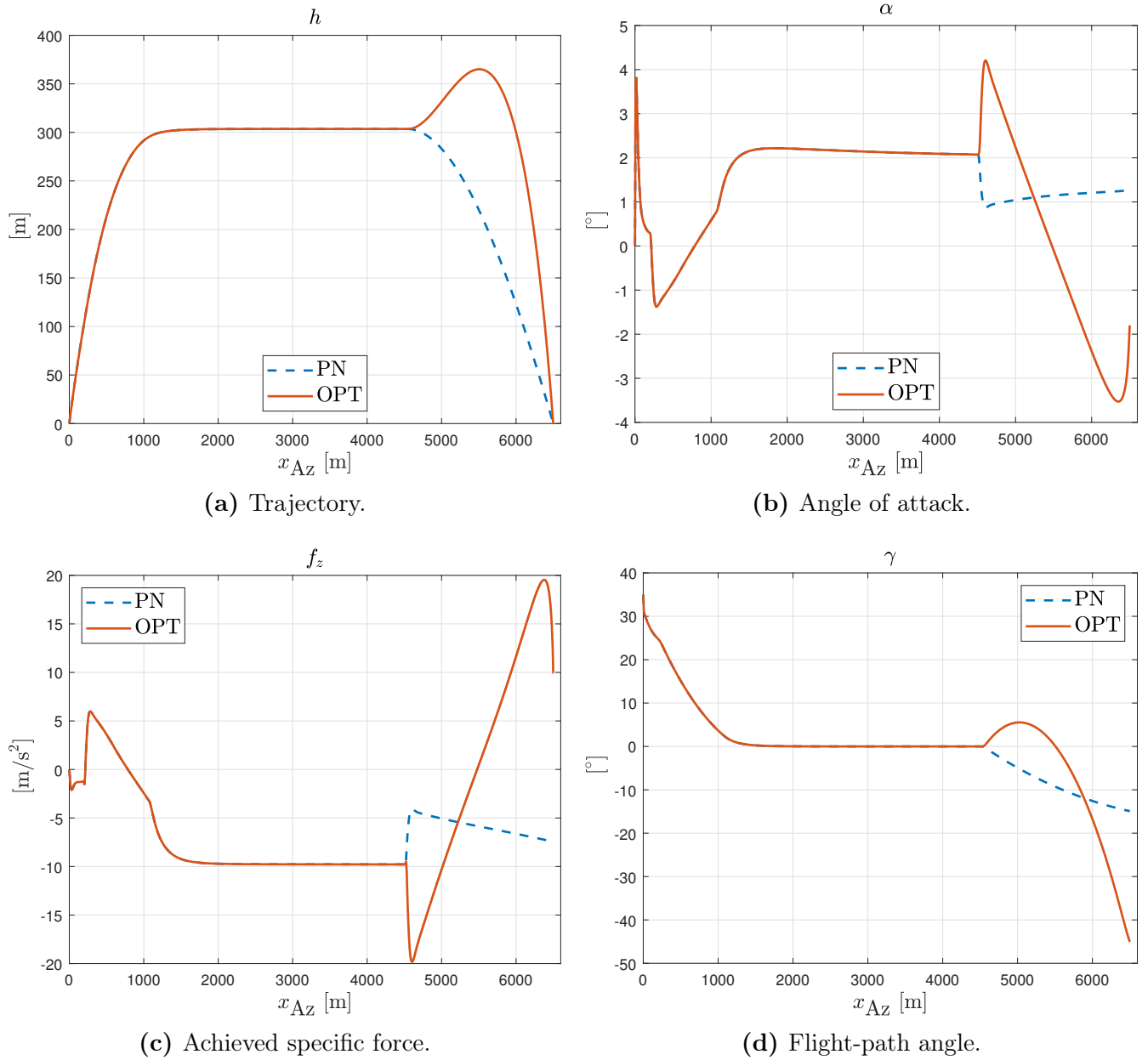


Figure 3.49: Comparison of the trajectory obtained using proportional navigation (3.92a) and optimal guidance (3.95) for the desired impact angle $\gamma_1 = -45^\circ$.

It is evident that proportional navigation cannot achieve an arbitrary impact angle. Therefore, the previous example should be viewed only as an illustrative comparison between the nominal trajectory and the trajectory obtained using the algorithms proposed in this dissertation. Further analysis will include only guidance law specifically designed to achieve prescribed impact angles.

For this purpose, three impact-angle values from the upper group of considered angles will be examined: -45° , -65° , and -85° . Achieving smaller impact-angle magnitudes is not of interest in this work, and Fig. 3.49d has already shown that such values can be achieved using the baseline guidance law.

The optimal guidance law (3.95) was obtained as the unique solution of the optimization problem (2.6), in which the missile position relative to the target and the desired impact angle are predefined. Under real operating conditions, however, the relative missile-target position at the start of terminal guidance depends on the constraints and requirements of the particular mission. The cruise altitude and the distance from the target at which terminal guidance begins may be dictated by terrain characteristics or enemy-defense capabilities. Nevertheless, when such constraints are absent, the question of selecting these mission parameters optimally naturally arises.

Figures 3.50, 3.51, and 3.52 compare the results obtained using the optimal guidance law (3.95) for different cruise altitudes h_{cruise} . In all cases, the same distance from the target at which terminal guidance begins was adopted ($\Delta x_{\text{start}} = 2$ km), in order to isolate the influence of a single mission parameter.

In all cases, successful target interception was achieved with the desired velocity-vector orientation at the final instant of flight. It can be observed that, for the same impact angle, increasing the altitude from which terminal guidance begins reduces the maneuverability requirements. This is particularly evident in Fig. 3.50 for $\gamma_1 = -45^\circ$, where the highest altitude results in an almost imperceptible change in trajectory convexity, although such a change is generally unavoidable for the considered range of impact angles.

Table 3.7 presents the values of the control effort for the considered examples. The performance index (3.81) was evaluated from the instant when the missile was 100 m before the point at which terminal guidance begins until target interception.

Table 3.7: Control effort (3.81) for different cruise altitudes, $\Delta x_{\text{start}} = 2$ km.

h_{cruise}	J		
	$\gamma_1 = -45^\circ$	$\gamma_1 = -65^\circ$	$\gamma_1 = -65^\circ$
200 m	2564	5765	8916
300 m	1969	5104	8457
400 m	1426	4399	7841
500 m	964	3703	7132

The previous part of this dissertation also demonstrated that, with an appropriately selected launch elevation angle, higher cruise altitudes can be achieved with lower control effort. Based on the results presented thus far, it may therefore be concluded that higher cruise altitudes are generally desirable. However, climbing to a higher altitude comes at the cost of a larger velocity loss during the initial phase of flight, resulting in lower cruise speeds, as can be clearly seen in Figs. 3.50e, 3.51e, and 3.52e. On the other hand, during the terminal phase of flight, less altitude gain is required to achieve the desired impact angle, and consequently the velocity loss is smaller. As a result, the missile retains greater maneuverability at the moments when the required maneuvers are the most demanding, which justifies the lower cruise speed whenever mission constraints permit.

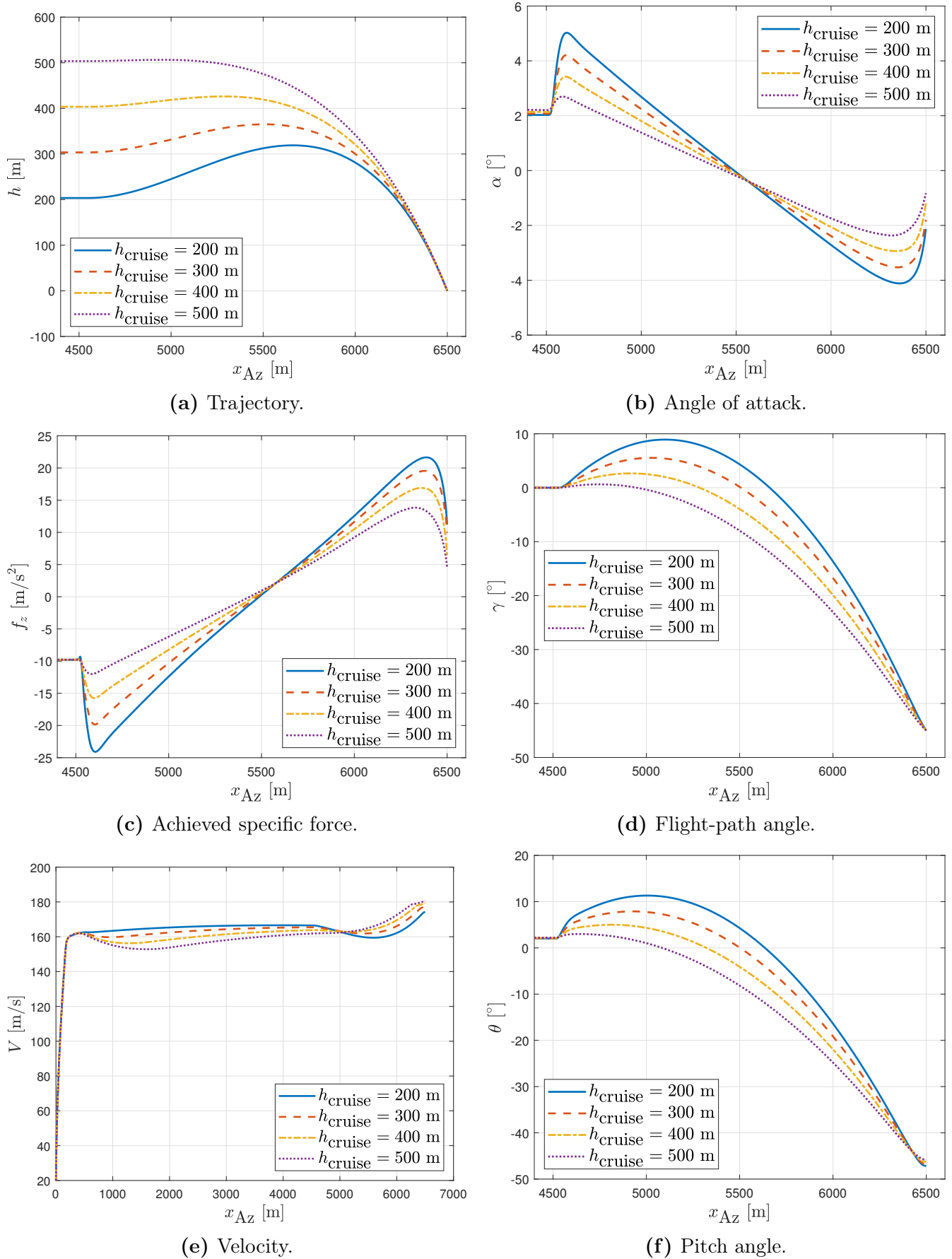


Figure 3.50: Influence of the cruise altitude h_{cruise} at which terminal guidance begins, $\gamma_1 = -45^\circ$.

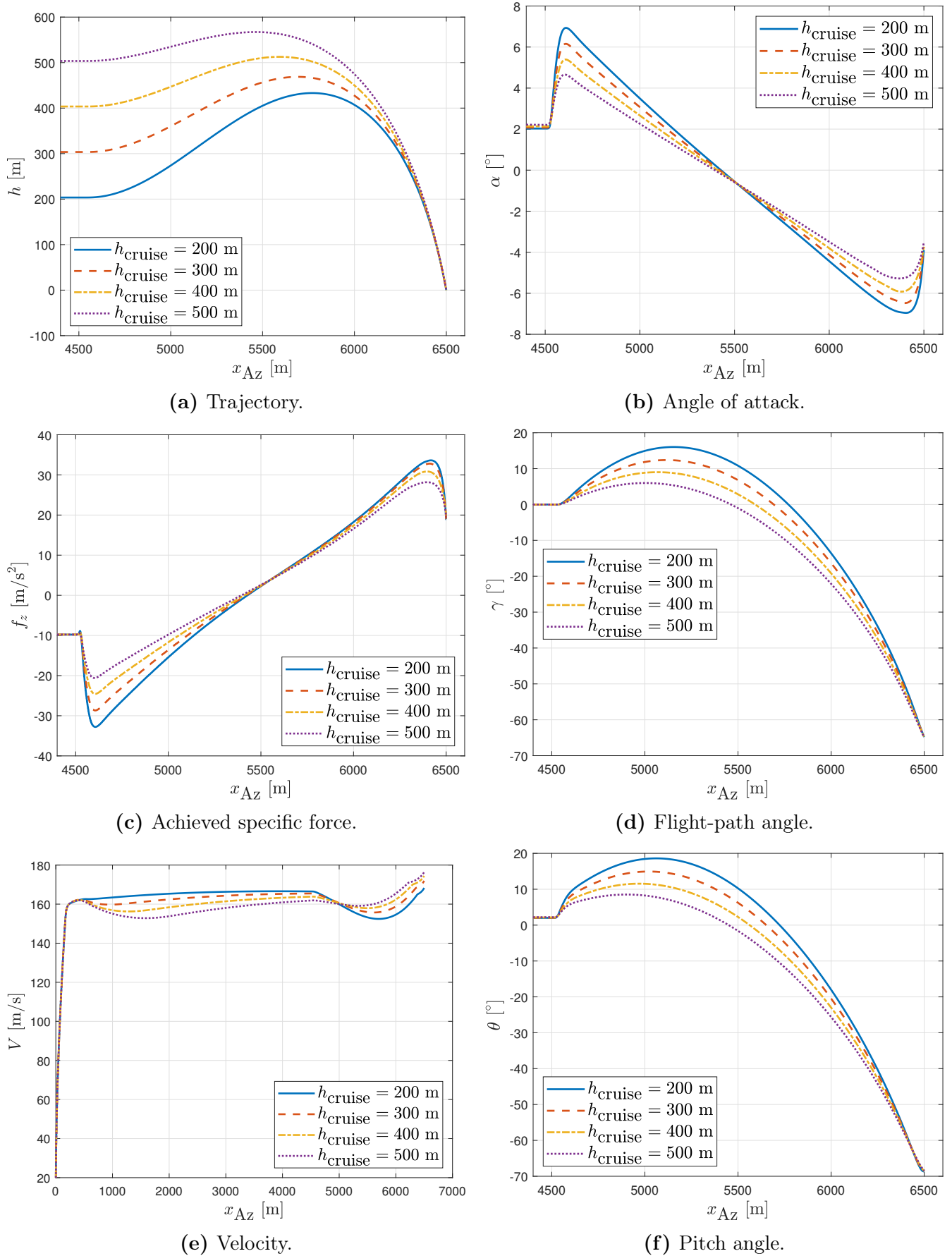
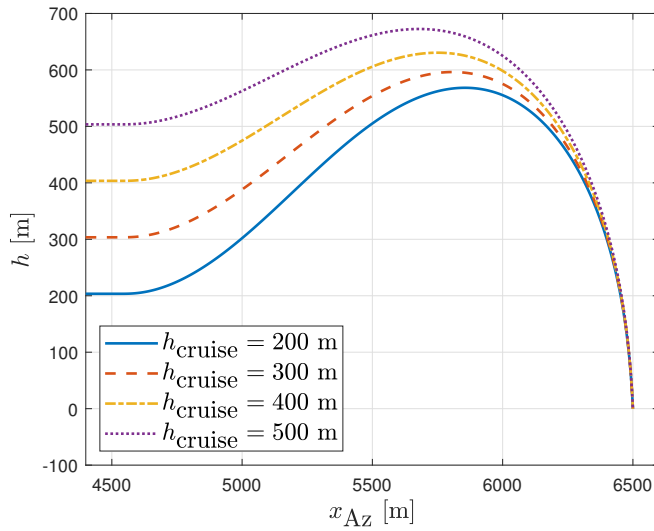
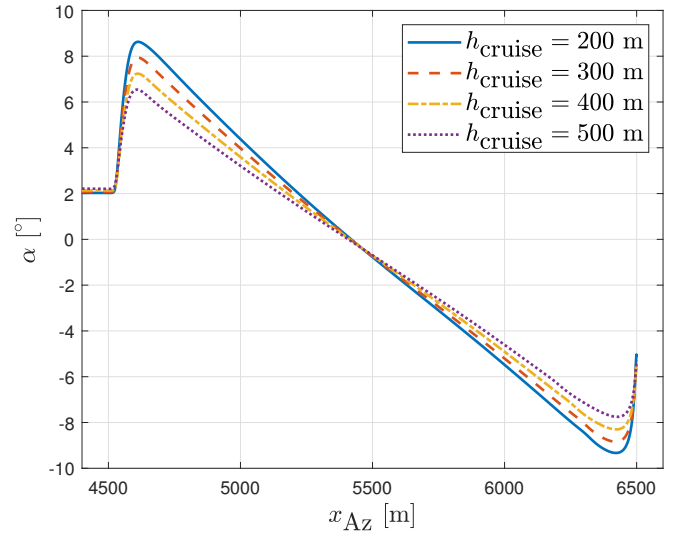


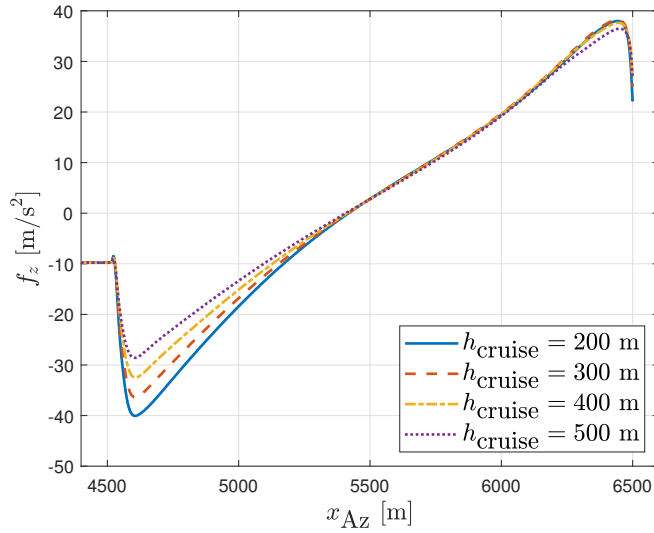
Figure 3.51: Influence of the cruise altitude h_{cruise} at which terminal guidance begins, $\gamma_1 = -65^\circ$.



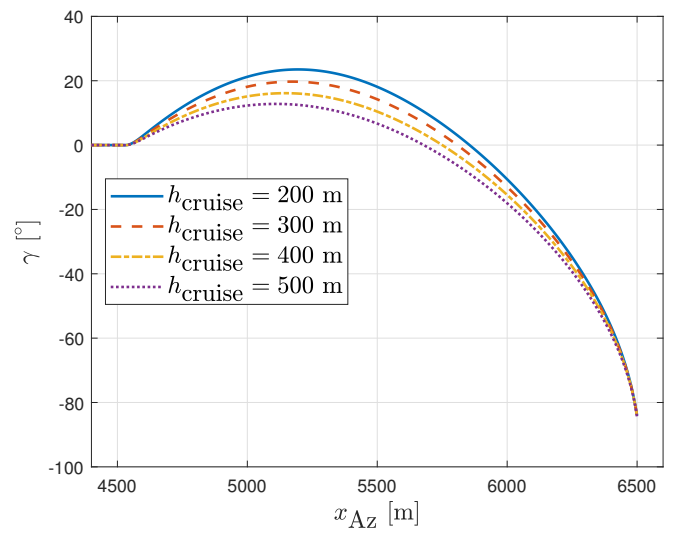
(a) Trajectory.



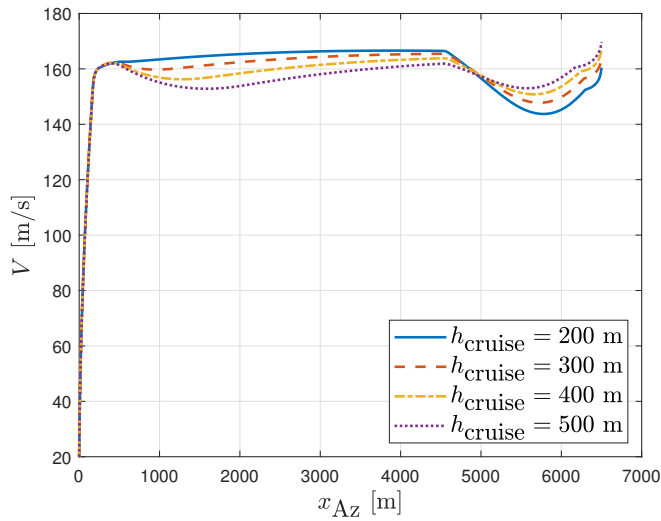
(b) Angle of attack.



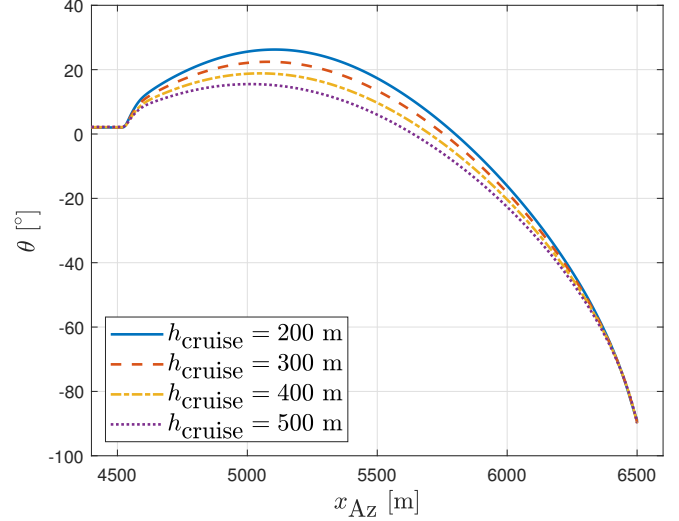
(c) Achieved specific force.



(d) Flight-path angle.



(e) Velocity.



(f) Pitch angle.

Figure 3.52: Influence of the cruise altitude h_{cruise} at which terminal guidance begins, $\gamma_1 = -85^\circ$.

In Figs. 3.53, 3.54, and 3.55, a comparison of the results obtained using the optimal guidance law (3.95) is presented for different distances from the target Δx_{start} at which terminal guidance begins. In all cases, the same cruise altitude is assumed ($h_{\text{cruise}} = 300$ m) in order to isolate the influence of a single mission parameter.

It is observed that, as the distance from the target at which optimal guidance is initiated increases, the maximum flight altitude also increases, in some cases reaching values that are almost three to four times higher than the cruise altitude. As expected, the required maneuvers are then smaller; however, this raises the question of the practical feasibility of such trajectories, especially considering the significant decrease in velocity during the terminal phase of flight shown in Figs. 3.53e, 3.54e, and 3.55e.

On the other hand, as the start of terminal guidance is moved closer to the target, the required maneuvers increase, becoming particularly pronounced in the second part of the final phase of flight, after the change in trajectory convexity. In the limiting case where the initial point is 1 km from the target and the required impact angles are large, the missile fails to achieve the desired impact angle and even fails to hit the target, which is already noticeable for $\gamma_1 = -65^\circ$ in Figs. 3.54a and 3.54d, and becomes clearly visible for $\gamma_1 = -85^\circ$ in Figs. 3.55a and 3.55d. This is a consequence of saturation of the commanded specific force shown in Figs. 3.54c and 3.55c, which in the considered examples is limited to 40 m/s^2 . In the most critical case, when the required impact angle is -85° and the terminal phase begins at 1 km from the target, oscillations in the achieved specific force also occur, leading to an additional increase in miss distance.

Table 3.8 presents the values of the achieved control effort for the considered cases. Although the results in Figs. 3.53, 3.54, and 3.55 are shown for the last 4 km of flight, the performance index (3.81) is computed from the instant when the missile is 100 m before the point at which terminal guidance begins, up to target interception. Therefore, for example, in the case $\gamma_1 = -45^\circ$, a higher value of the performance index is obtained when terminal guidance starts at a distance of 4 km from the target than at 1 km, even though larger specific force values are then achieved. For larger impact angles, the situation is reversed: although the flight duration is longer at larger distances, higher control effort is achieved at smaller distances due to significantly larger maneuvers.

Table 3.8: Control effort (3.81) for different distances from the target at which terminal guidance begins, $h_{\text{cruise}} = 300$ m.

Δx_{start}	J		
	$\gamma_1 = -45^\circ$	$\gamma_1 = -65^\circ$	$\gamma_1 = -85^\circ$
1 km	1744	5222	7481
2 km	1969	5104	8457
3 km	1974	4004	6144
4 km	2150	3558	5030

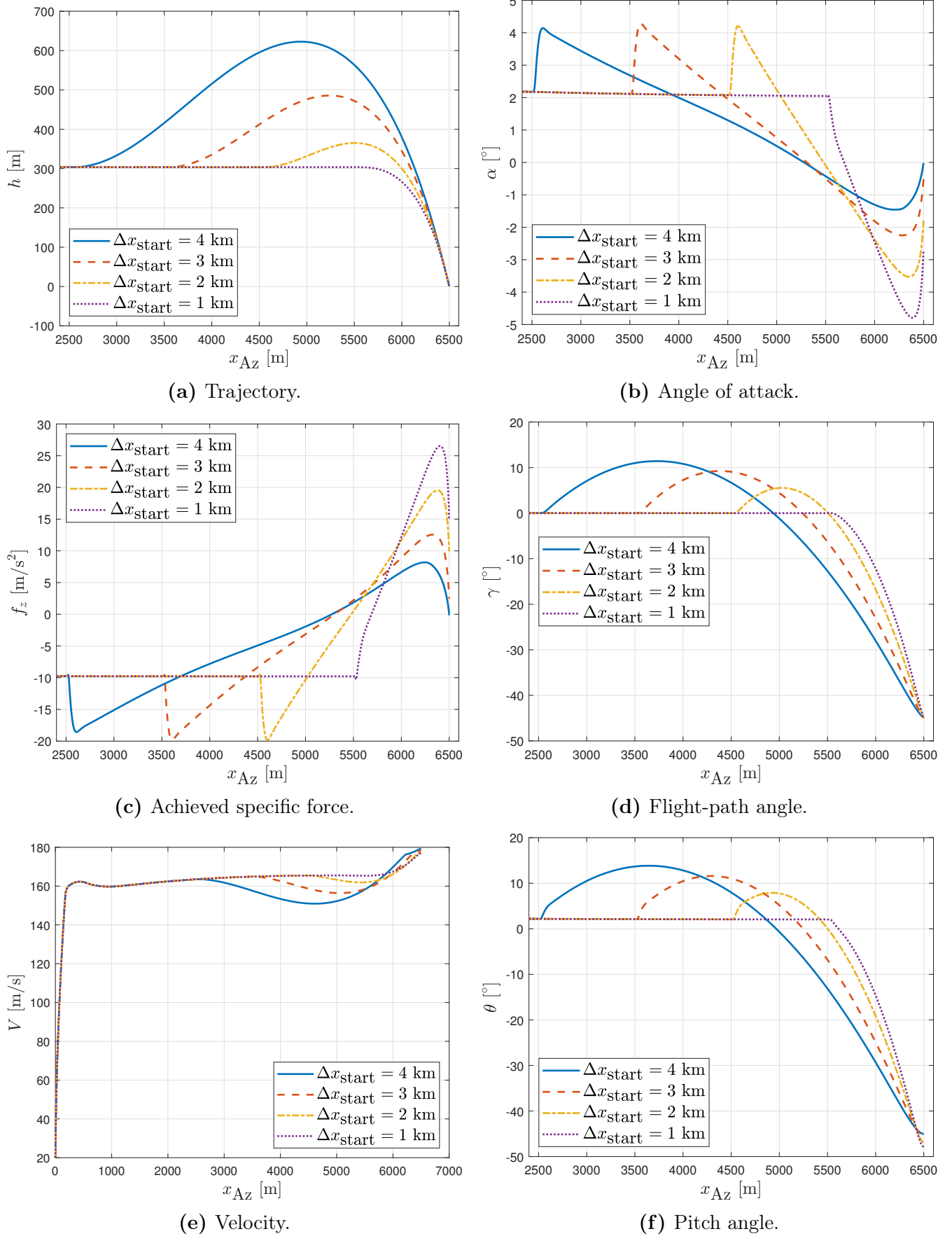
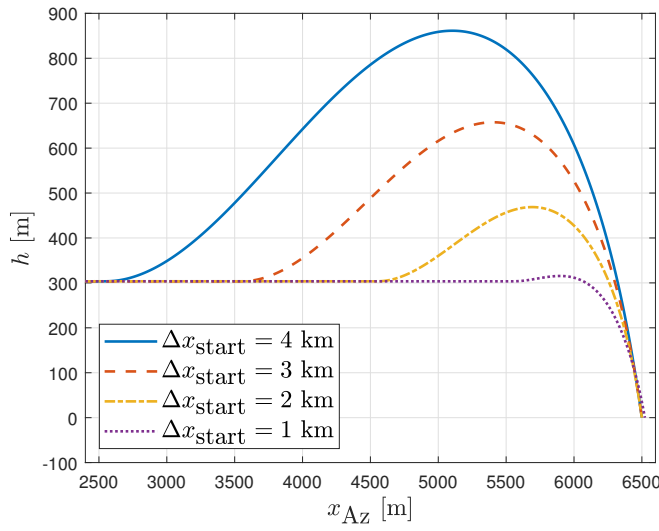
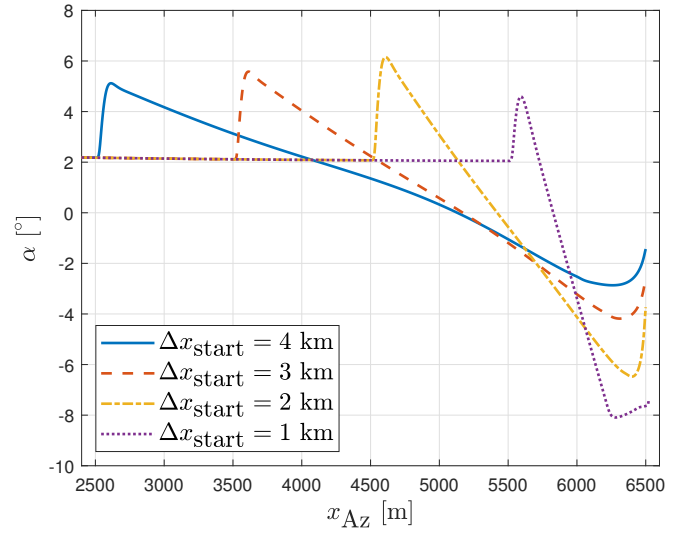


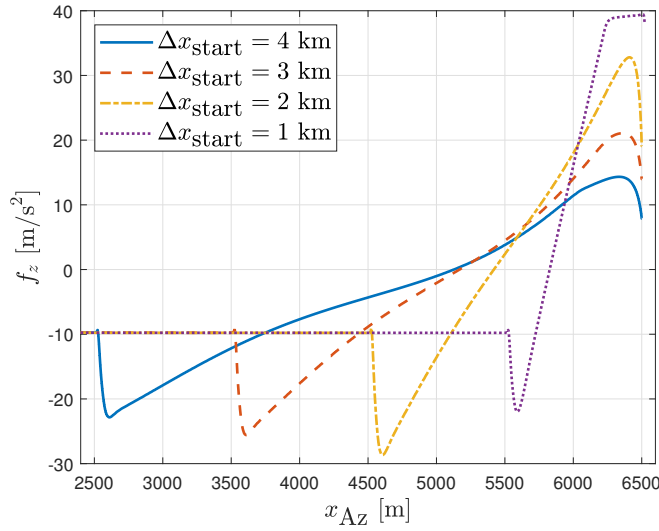
Figure 3.53: Influence of the distance Δx_{start} at which terminal guidance begins, $\gamma_1 = -45^\circ$.



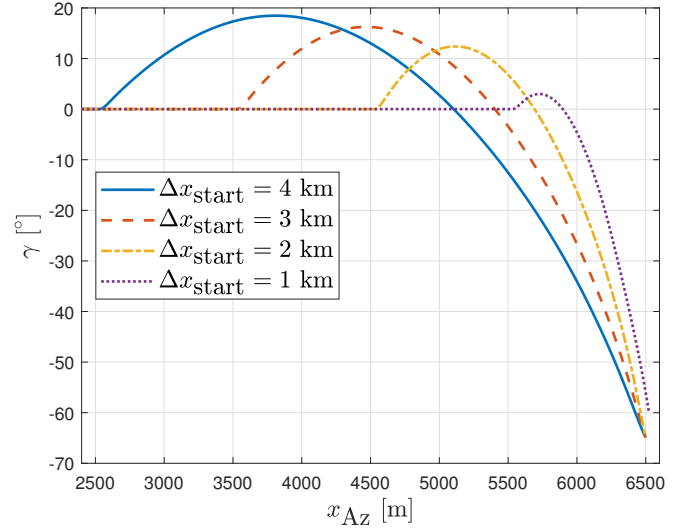
(a) Trajectory.



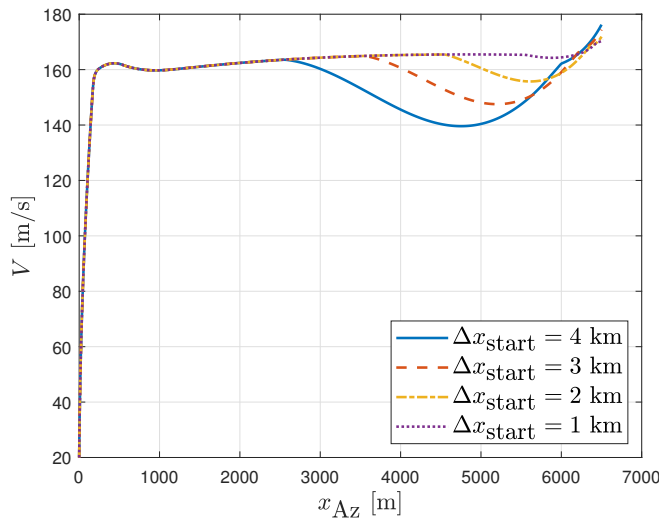
(b) Angle of attack.



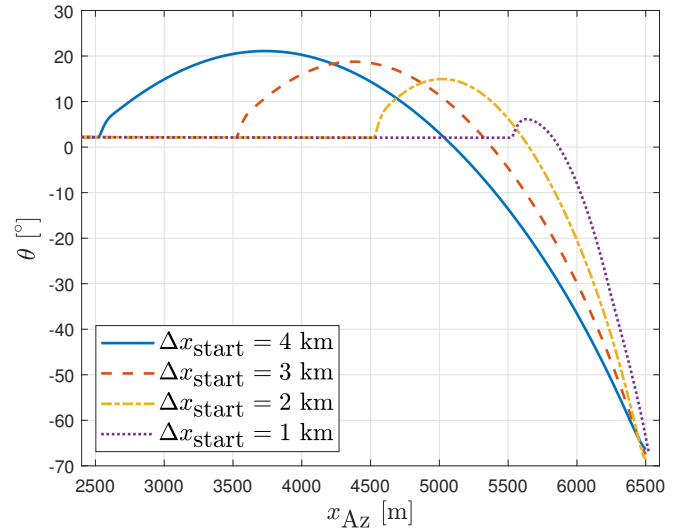
(c) Achieved specific force.



(d) Flight-path angle.

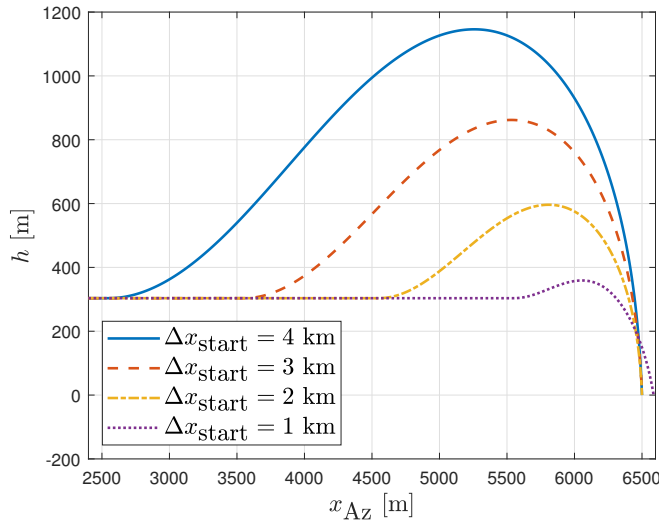


(e) Velocity.

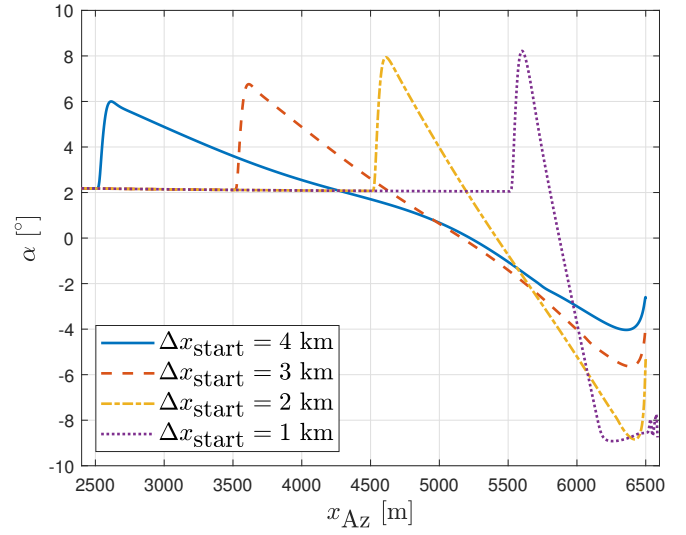


(f) Pitch angle.

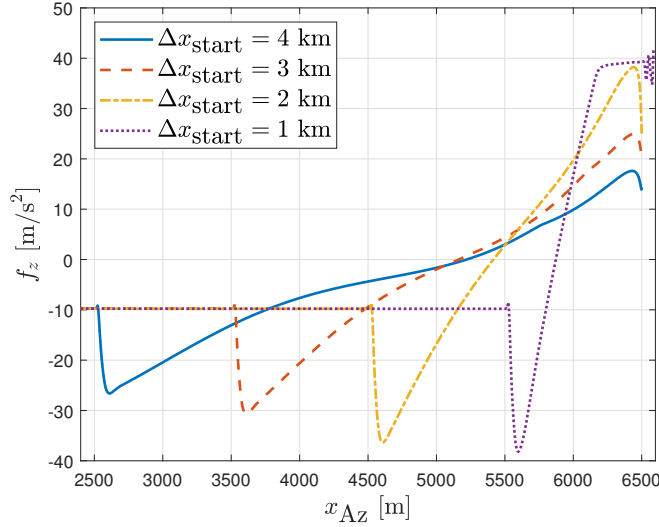
Figure 3.54: Influence of the distance Δx_{start} at which terminal guidance begins, $\gamma_1 = -65^\circ$.



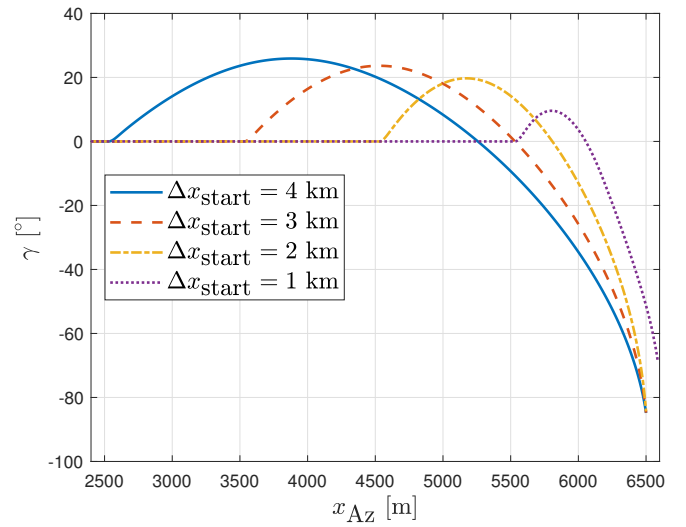
(a) Trajectory.



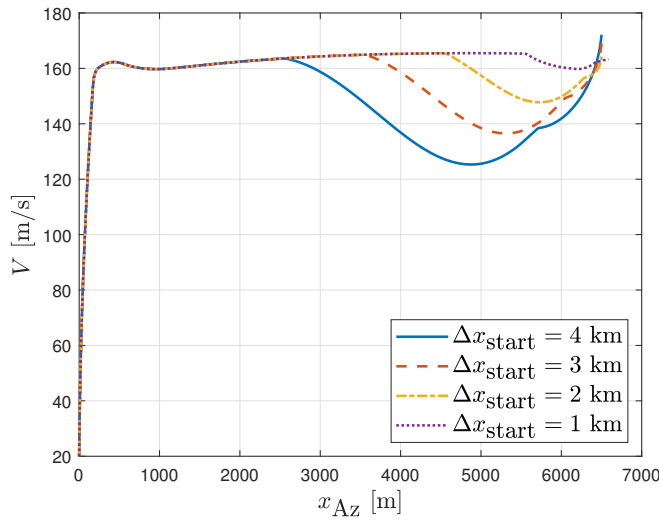
(b) Angle of attack.



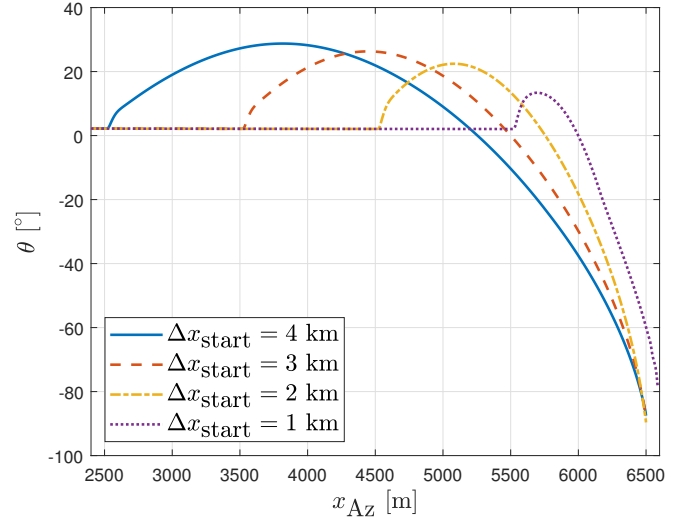
(c) Achieved specific force.



(d) Flight-path angle.



(e) Velocity.



(f) Pitch angle.

Figure 3.55: Influence of the distance Δx_{start} at which terminal guidance begins, $\gamma_1 = -85^\circ$.

In the previous examples, the influence of varying only a single mission parameter on the resulting trajectory was analyzed. Based on the conducted analysis, it is clear that, in order to evaluate the overall system performance, one must consider quantities that characterize the fulfillment of the main mission requirements: target interception and achieved impact angle, as well as quantities related to control effort. Naturally, flight speed is also an important factor that directly affects missile maneuverability, and it has been shown that the reduction in speed is a consequence of a large altitude increase during the final phase of flight. Therefore, in the following analysis, the following quantities are used as performance criteria:

- Miss distance: $|\Delta x_1| = |x_1 - x(t_1)|$;
- Error in achieved impact angle: $|\Delta \gamma_1| = |\gamma_1 - \gamma(t_1)|$;
- Maximum value of specific force: $\max(f_z)$;
- Achieved control effort given by the performance index (3.81);
- Maximum flight altitude;
- Minimum velocity after the cruise phase.

Unlike the previous analysis, here the performance index (3.81) is computed for all cases starting from the instant when the missile is 2 km from the launch point up to the end of flight, in order to ensure a consistent definition of control effort regardless of the start of terminal guidance.

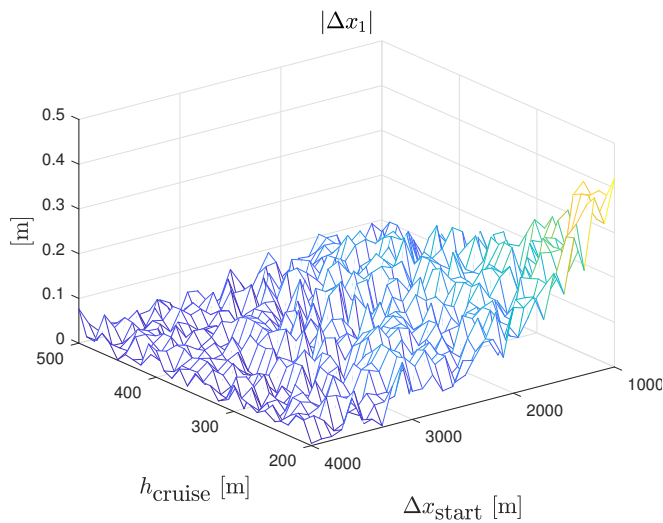
Figures 3.56, 3.57, and 3.58 show the quantities of interest as functions of the varied parameters: cruise altitude h_{cruise} and the distance from the target Δx_{start} at which terminal guidance begins, for three values of the impact angle. The mission parameters are varied within intervals chosen according to the considered 6.5 km range:

$$h_{\text{cruise}} \in [200, 500] \text{ m}, \quad \Delta x_{\text{start}} \in [1000, 4000] \text{ m}$$

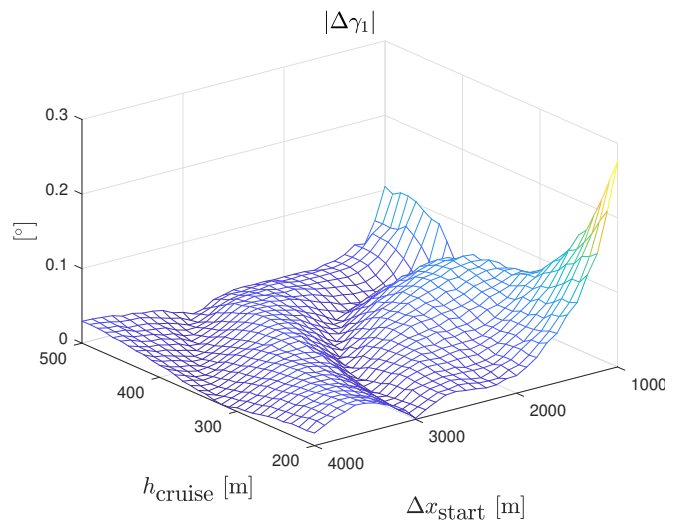
For the impact angle of -45° , a successful target interception with the desired impact angle is achieved in all cases, as clearly seen from the small errors shown in Figs. 3.56a and 3.56b. As the terminal guidance start distance decreases, the control effort shown in Fig. 3.56d increases, as a consequence of larger achieved specific forces, particularly at lower cruise altitudes, as shown in Fig. 3.56c.

This effect becomes more pronounced at larger impact angles. For $\gamma_1 = -65^\circ$, as the terminal guidance start distance decreases and for lower cruise altitudes, there is a drastic increase in both the impact angle error and miss distance shown in Figs. 3.57b and 3.57a, respectively. These large errors are a consequence of control saturation, and in some cases also increased oscillations, which is why regions can be observed in Fig. 3.57c where the achieved specific force exceeds the limit of 40 m/s^2 .

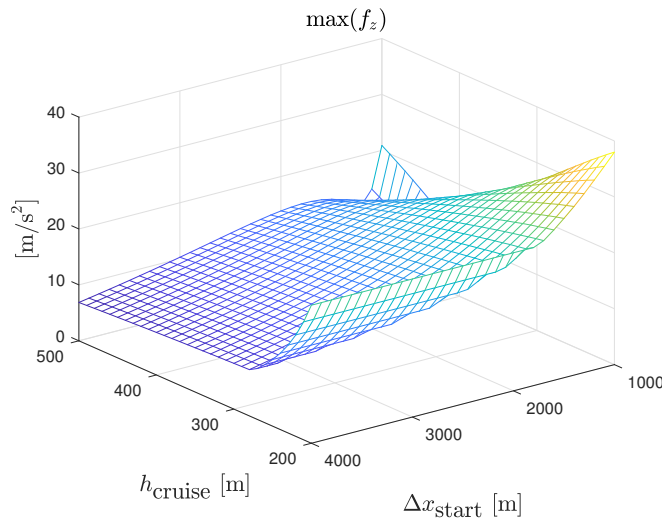
For the desired impact angle $\gamma_1 = -85^\circ$, it becomes impossible to hit the target for $\Delta x_{\text{start}} < 2 \text{ km}$, regardless of the cruise altitude from which terminal guidance begins.



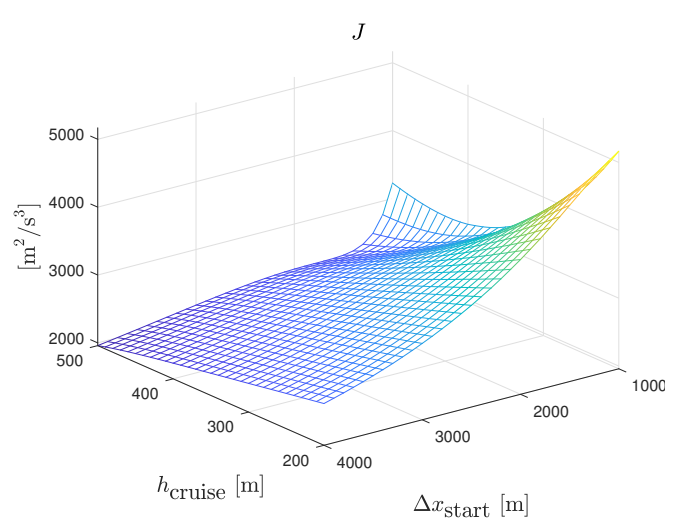
(a) Miss distance.



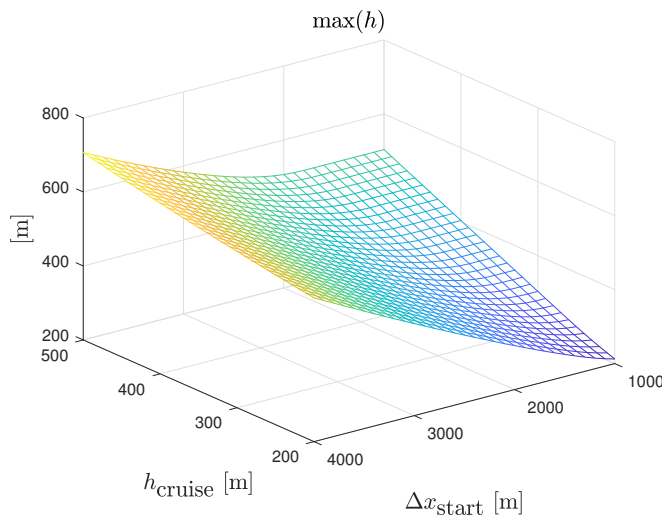
(b) Impact angle error.



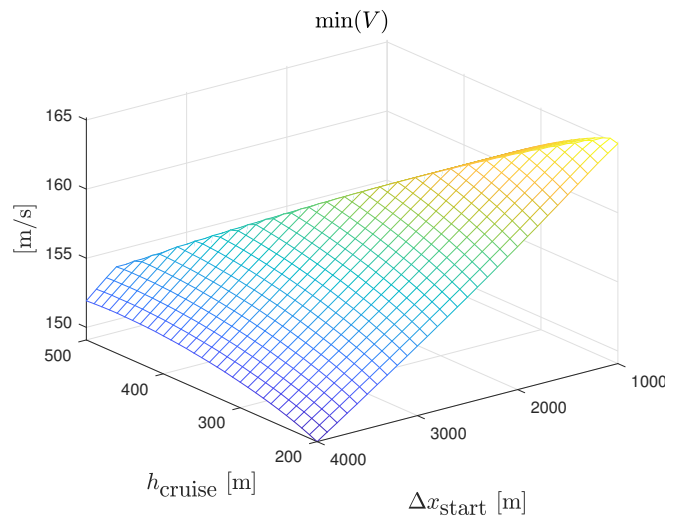
(c) Maximum specific force.



(d) Control effort.

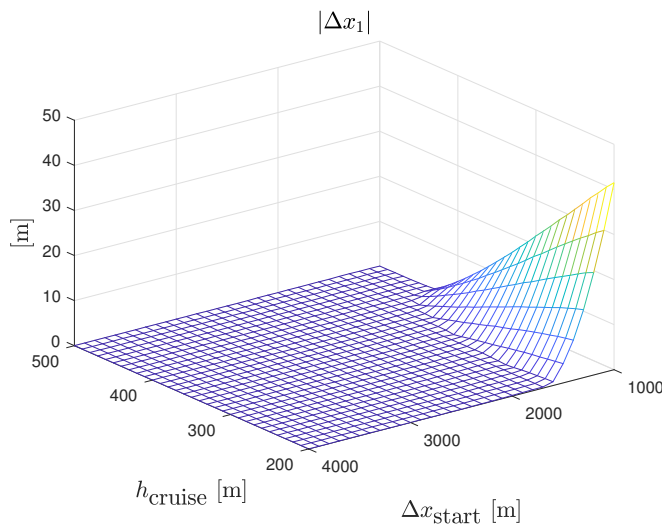


(e) Maximum altitude.

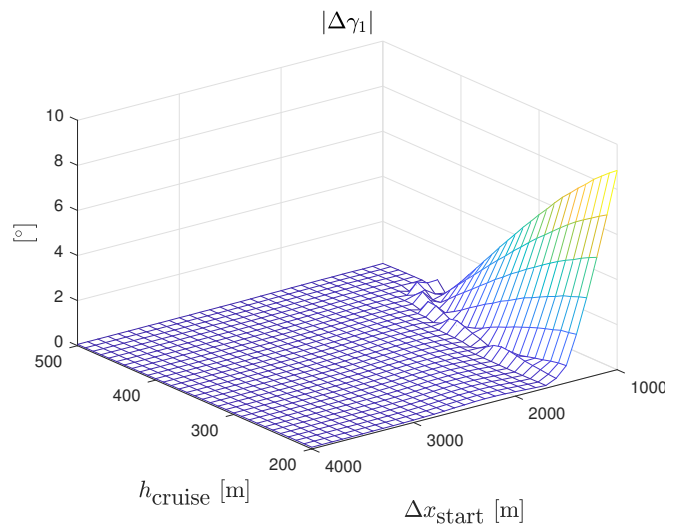


(f) Minimum velocity.

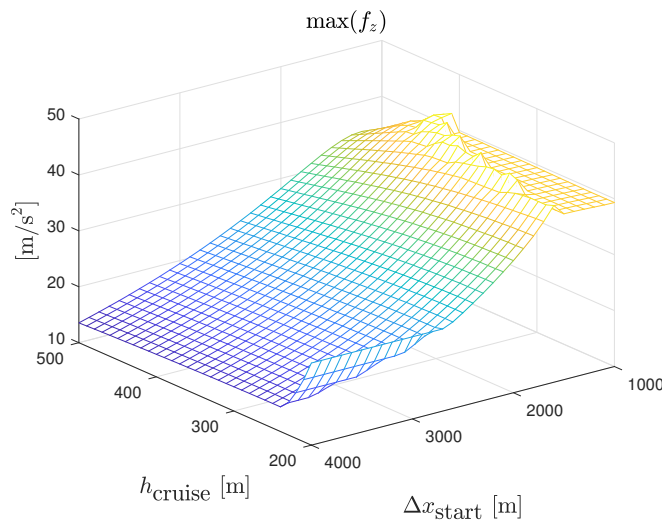
Figure 3.56: Effect of mission parameters on final results, $\gamma_1 = -45^\circ$.



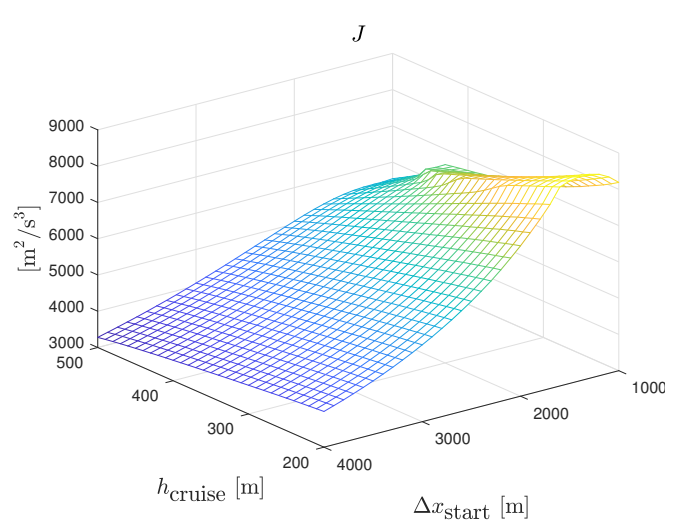
(a) Miss distance.



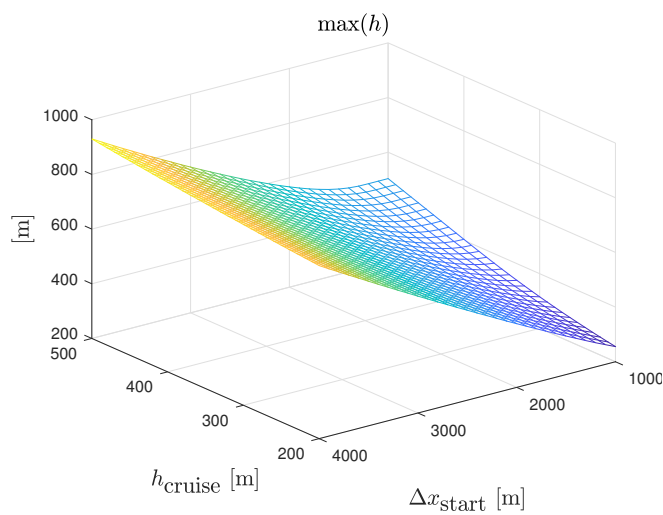
(b) Impact angle error.



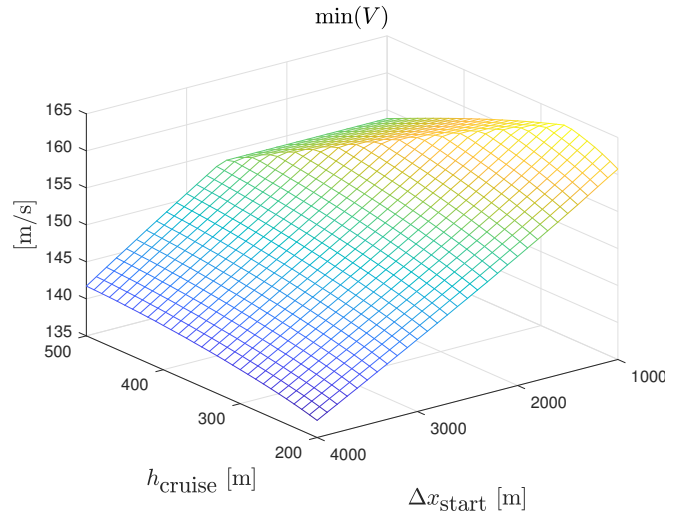
(c) Maximum specific force.



(d) Control effort.

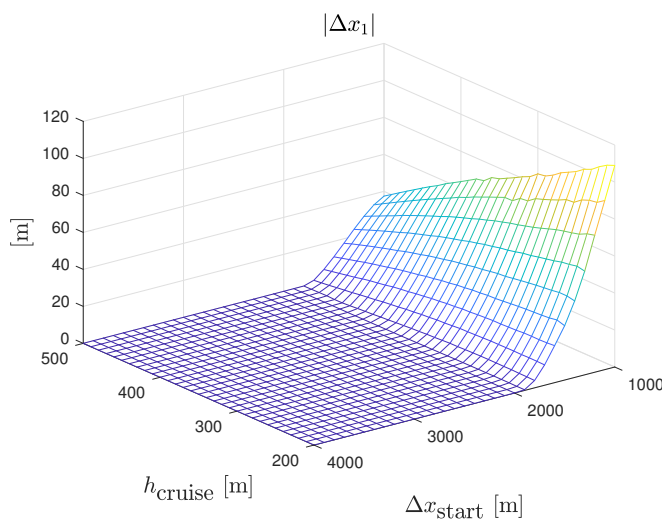


(e) Maximum altitude.

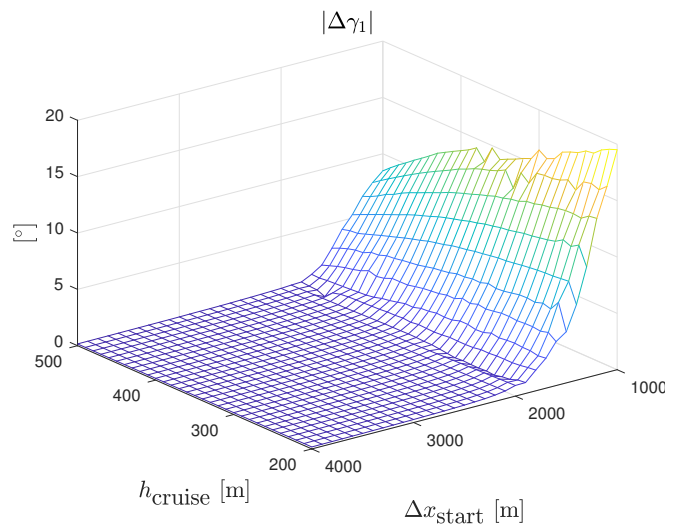


(f) Minimum velocity.

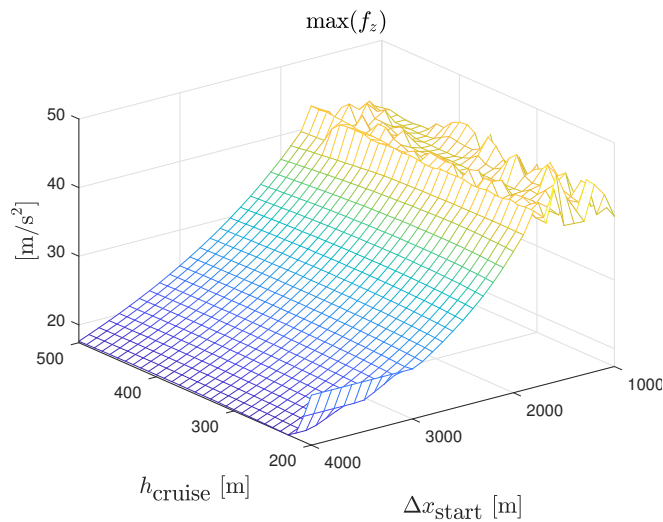
Figure 3.57: Effect of mission parameters on final results, $\gamma_1 = -65^\circ$.



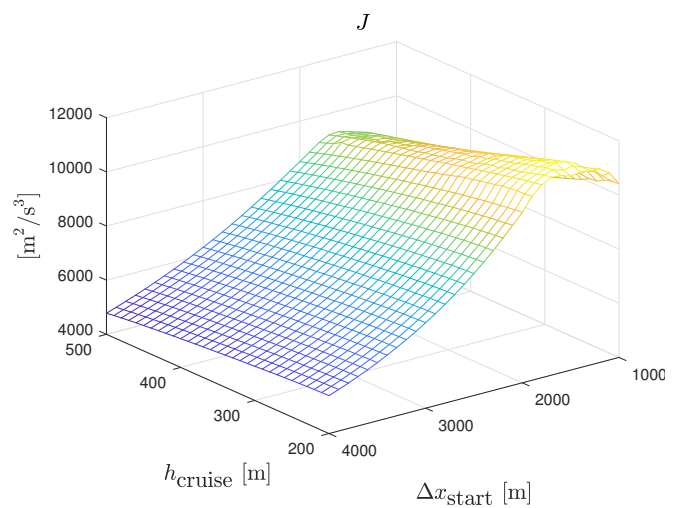
(a) Miss distance.



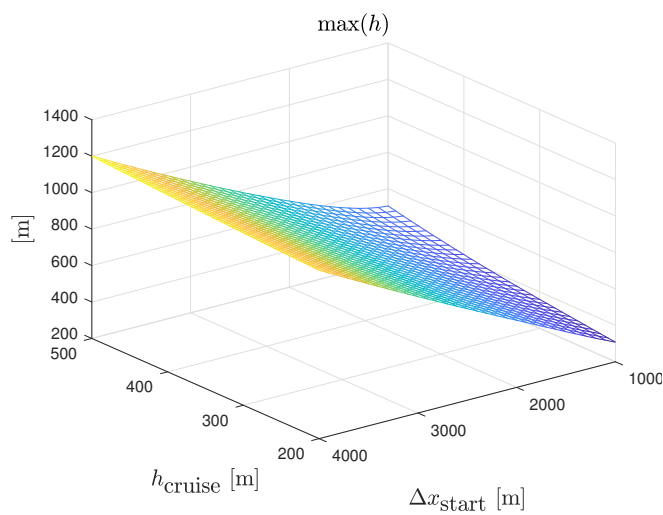
(b) Impact angle error.



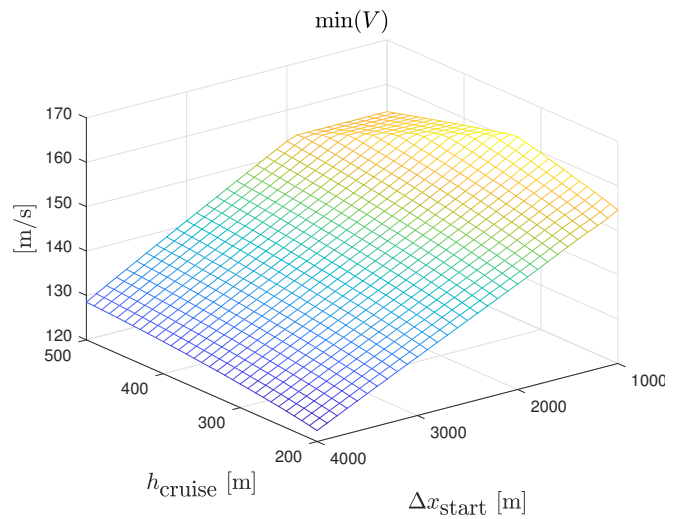
(c) Maximum specific force.



(d) Control effort.



(e) Maximum altitude.



(f) Minimum velocity.

Figure 3.58: Effect of mission parameters on final results, $\gamma_1 = -85^\circ$.

Based on the conducted analysis, it can be concluded that achieving larger impact angles is favorable when the terminal guidance is initiated as far from the target as possible and from the highest possible altitude. However, both of these factors influence the increase of the maximum flight altitude, as shown in Figs. 3.56e, 3.57e, and 3.58e, and this altitude may be constrained by mission requirements or by individual subsystems of the vehicle. For example, cruise missiles in most cases employ air-breathing propulsion systems, for which optimal operation may impose a limitation on the maximum flight altitude. It has already been shown that premature transition to the terminal guidance phase leads to a longer climb phase, due to the need to align the approach direction to the target, which also results in an increased maximum altitude and a larger velocity drop, clearly visible in Figs. 3.56f, 3.57f, and 3.58f.

It should also be noted that all surfaces exhibit a more pronounced gradient in the direction of changing the distance from the target at which terminal guidance begins, meaning that cruise altitude has a smaller influence on the considered criteria, which can also be observed from the results shown in Figs. 3.50–3.55.

For all the above reasons, cruise altitude and the distance to the target at which terminal guidance starts—if their selection is not otherwise constrained—should be chosen as a compromise between maximum flight altitude and required maneuverability. For this purpose, multi-parameter optimization methods can be used. However, selecting optimal mission parameters for a specific vehicle is the task of the fire-control system and is not the primary objective of this work. Therefore, in the following, only one possible approach for their selection will be presented.

If the quantities of interest for selecting optimal mission parameters are chosen as maximum specific force, maximum flight altitude, and minimum velocity during the terminal phase, it is clear that the optimal pair $(h_{\text{cruise}}^*, \Delta x_{\text{start}}^*)$ can be found by minimizing the first two and maximizing the third. Since these quantities do not represent the same physical variables and have different units and ranges, they must first be normalized. For maximum specific force and flight altitude, denoted as f_3 and f_5 ⁴, the normalized values are computed as:

$$f_{i,\text{norm}} = \frac{f_i - \min(f_i)}{\max(f_i) - \min(f_i)} \quad i \in \{3, 5\} \quad (3.96)$$

while the normalized value of minimum flight velocity, denoted as f_6 , is:

$$f_{6,\text{norm}} = \frac{\max(f_6) - f_6}{\max(f_6) - \min(f_6)} \quad (3.97)$$

The cost function can be formed as a linear combination of normalized values:

$$\mathcal{C}(h_{\text{cruise}}, \Delta x_{\text{start}}) = \sum_{i \in \{3, 5, 6\}} w_i f_{i,\text{norm}} \quad (3.98)$$

where w_i are weighting coefficients used to adjust the relative influence of each quantity on the cost function.

⁴The indices are chosen according to the order in which these quantities are presented in Figs. 3.56–3.58.

The optimal mission parameters are obtained by minimizing the cost function:

$$(h_{\text{cruise}}^*, \Delta x_{\text{start}}^*) = \arg \min_{(h_{\text{cruise}}, \Delta x_{\text{start}}) \in \mathcal{D}} \mathcal{C}(h_{\text{cruise}}, \Delta x_{\text{start}}) \quad (3.99)$$

Figure 3.59 shows the shape of the cost function (3.98) for two groups of weighting coefficients, for the case $\gamma_1 = -65^\circ$. The first group, where w_3 and w_5 are equal and set to $1/2$, corresponds to minimizing maximum specific force and altitude. The optimal point is marked in red in Fig. 3.59a, and the corresponding cruise altitude and distance to the target at which terminal guidance begins are:

$$h_{\text{cruise}} = 210 \text{ m}, \quad \Delta x_{\text{start}} = 3000 \text{ m}$$

The second group of weighting coefficients is chosen such that, in addition to minimizing maximum specific force and altitude, velocity loss is also equally penalized, and the optimal mission parameters are:

$$h_{\text{cruise}} = 410 \text{ m}, \quad \Delta x_{\text{start}} = 2000 \text{ m}$$

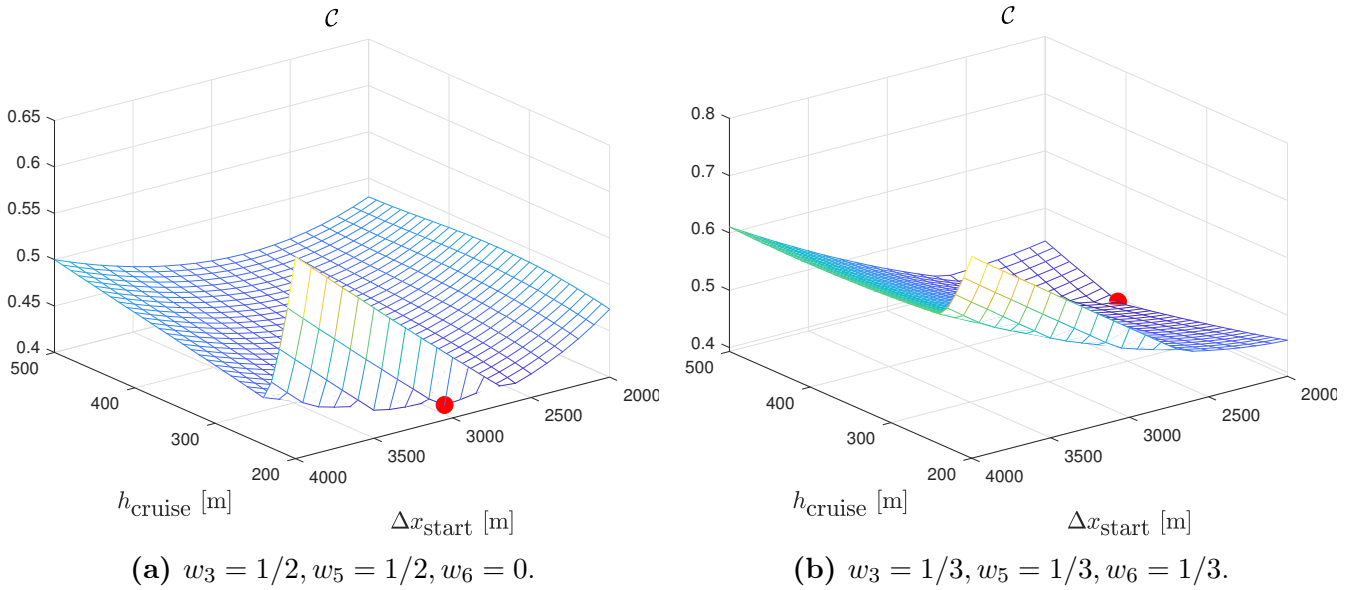


Figure 3.59: Shape of the cost function (3.98) for different values of weights w_i . $\gamma_1 = -65^\circ$.

The trajectories corresponding to the optimal mission parameters obtained by minimizing different cost functions are compared in Fig. 3.60. It can be observed that maneuvering demands are lower in the case where maximum specific force and altitude are minimized, which is expected since terminal guidance starts at a greater distance from the target. However, due to the threefold increase in maximum altitude compared to cruise altitude, the maximum velocity drop is around 20 m/s. On the other hand, when velocity loss is included in the cost function, the maximum specific force during the terminal phase is about 10 m/s^2 higher, but the velocity drop is almost negligible due to the small difference between the trajectory apex and cruise altitude.

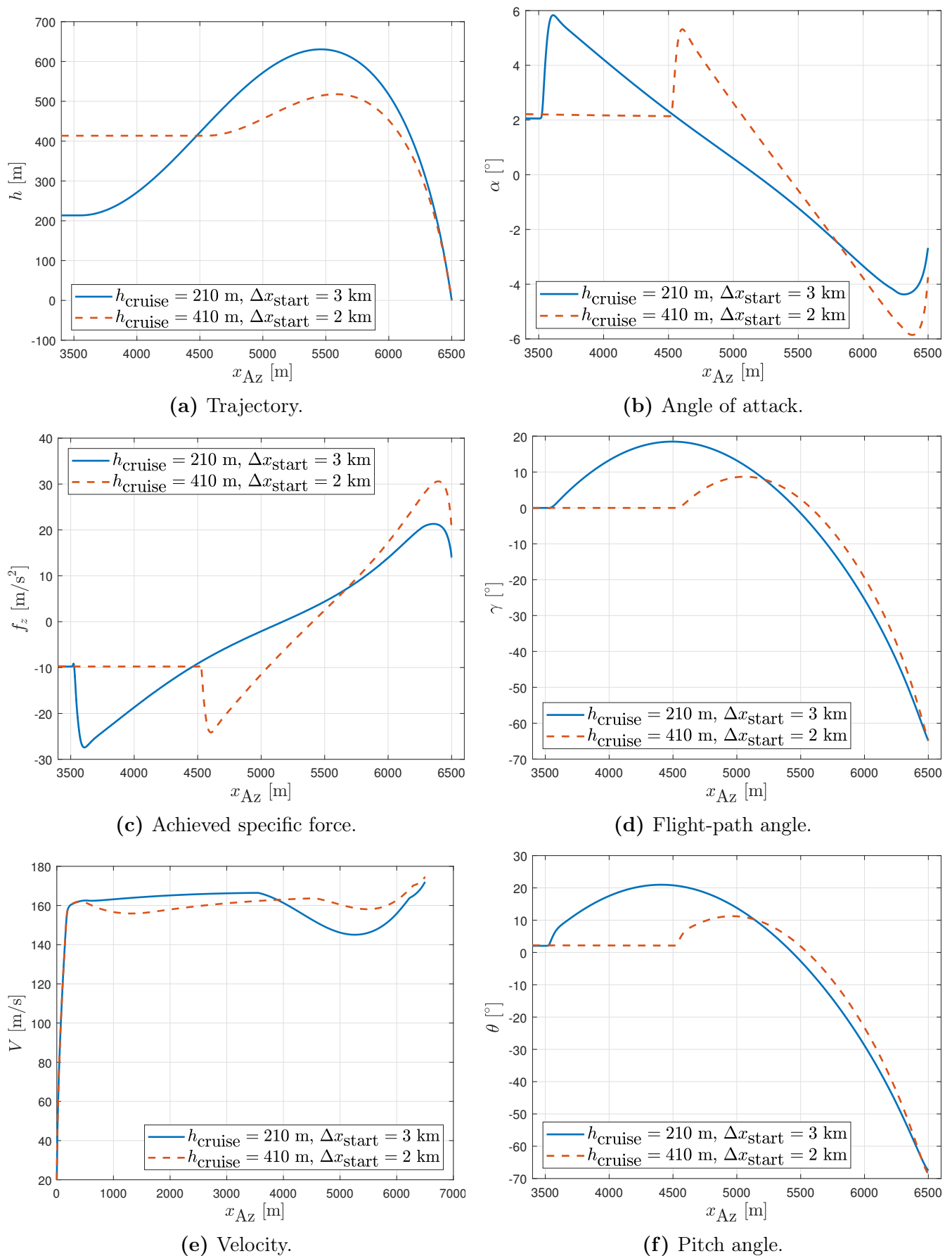


Figure 3.60: Comparison of trajectories for optimal mission parameter selection, $\gamma_1 = -65^\circ$.

Another example of multi-parameter optimization that arises during system preparation can be found in field testing, where safety zones impose constraints. In particular, the maximum allowed flight altitude indirectly limits the desired impact angle, since achieving larger impact angles, given realistic maneuverability limits, requires higher flight altitudes. Therefore, a possible task is the selection of the impact angle and the distance to the target at which terminal guidance begins, for a fixed cruise altitude.

Based on the conducted analysis, it is clear that the number of free parameters depends on the specific mission, and that optimal values must be selected by considering constraints and requirements that may change during the system development phase. In the following, the influence of navigation errors on the final results will be investigated, for the nominal trajectory previously identified as optimal in terms of minimizing velocity loss, maximum specific force, and altitude:

$$(h_{\text{cruise}}^*, \Delta x_{\text{start}}^*) = (410 \text{ m}, 2000 \text{ m})$$

This trajectory was selected due to its smaller altitude increase during the terminal phase and consequently smaller velocity drop, as well as the higher cruise altitude which is favorable in terms of reduced drag during cruise and lower control effort during the climb phase.

Up to now, in the implementation of the proposed guidance law given by (3.95), exact values of all quantities obtained from the navigation algorithm were used. In all terminal guidance examples, the target was successfully hit with the desired impact angle, provided that the mission parameters (cruise altitude and distance at which terminal guidance begins) were appropriately selected. It is clear that in the case of erroneous estimation of the relative position to the target (Δx and Δy), a different point would be hit, i.e., the target would be missed to the extent that the navigation system incorrectly estimates the current position of the vehicle. Since the magnitude of this error depends on the navigation algorithm, sensor quality, availability of satellite navigation aiding, flight duration, etc., the influence of position estimation errors will not be further analyzed. The influence of velocity estimation errors has already been covered in Section 3.2.4, meaning that among the navigation data required for implementation of the optimal guidance law, only the influence of errors in the estimated velocity-vector angle γ and pitch angle θ remains unexamined.

Figure 3.61 shows a comparison of trajectories obtained with a constant error of $\pm 10^\circ$ in the estimated velocity-vector angle $\hat{\gamma}$ during the terminal phase of flight. The erroneous value of this angle was used only in the guidance law (3.95), in order to isolate the influence of this error on the guidance system. Although in both cases the target miss is less than 5 m, the resulting trajectories differ significantly. Specifically, when the estimated value of the flight-path angle is higher than the true one, the missile reaches its final flight direction earlier, resulting in an impact angle smaller than the desired one. On the other hand, when the estimated value is lower than the true one, a greater altitude increase occurs during the terminal phase, leading to a larger impact angle.

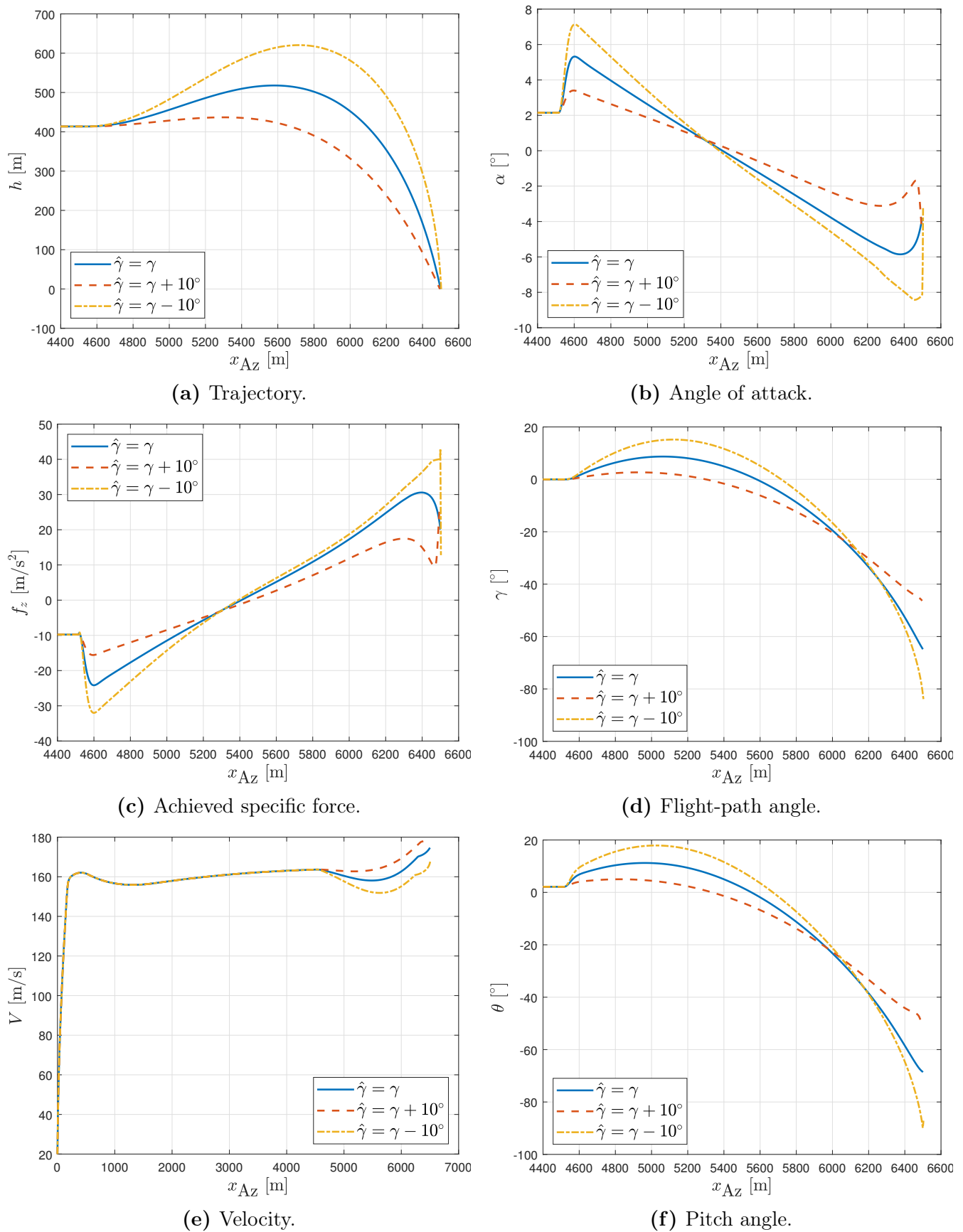


Figure 3.61: Effect of errors in the estimated flight-path angle, $\gamma_1 = -65^\circ$.

Figure 3.62 presents the results of an analysis similar to that shown in Fig. 3.61, except that the exact value of the flight-path angle γ was used in the guidance law (3.95), while the estimated pitch angle was corrupted by a constant error of $\pm 10^\circ$. Unlike the previous analysis, where a significant influence of errors in the estimated angle γ was observed, deviations of the trajectories from the nominal one due to errors in the estimated pitch angle are minimal. This phenomenon results from the fact that the pitch angle θ in the guidance law (3.95) is used only to determine the projection of the commanded normal acceleration onto the body-fixed coordinate system, whereas the velocity-vector angle γ is involved in the computation of the optimal-guidance parameters η_1 , η_2 , and η_3 , thereby directly affecting the shape of the optimal trajectory.

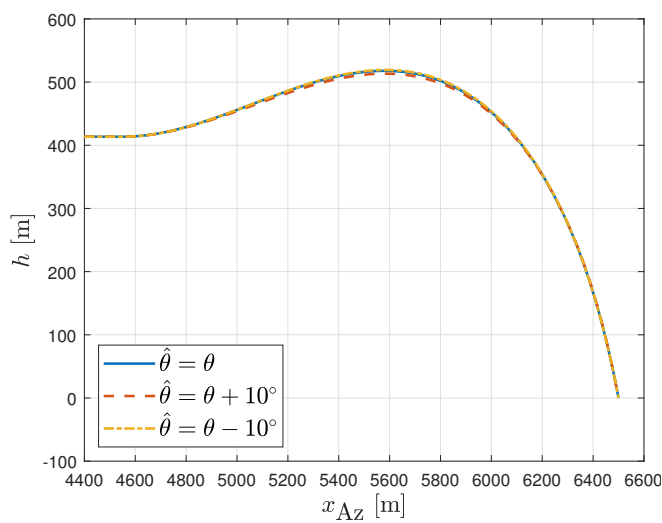
This effect is illustrated even more clearly in the limiting case where the estimated pitch angle is assumed to be zero throughout the entire terminal-guidance phase. The results of such an analysis are presented in Fig. 3.63, from which it can be seen that the deviation from the nominal trajectory becomes significant only during the final moments of flight. The demonstrated robustness is a consequence of the fact that the guidance system essentially represents a closed-loop system with respect to the states (missile position and velocity-vector orientation). Therefore, even under non-optimal operation of the inner loop (the control system), caused by an incorrect projection of the commanded signal, the resulting miss distance remains below 10 m even for extreme values of errors in the estimated pitch angle.

In the previous two examples (Figs. 3.61 and 3.62), the individual effects of errors in the estimated velocity-vector angle and pitch angle were investigated separately. However, in order to provide a complete analysis, Fig. 3.64 shows the dependence of the quantities of interest on errors in the estimated values of these angles. Both errors were varied within the interval from -10° to 10° .

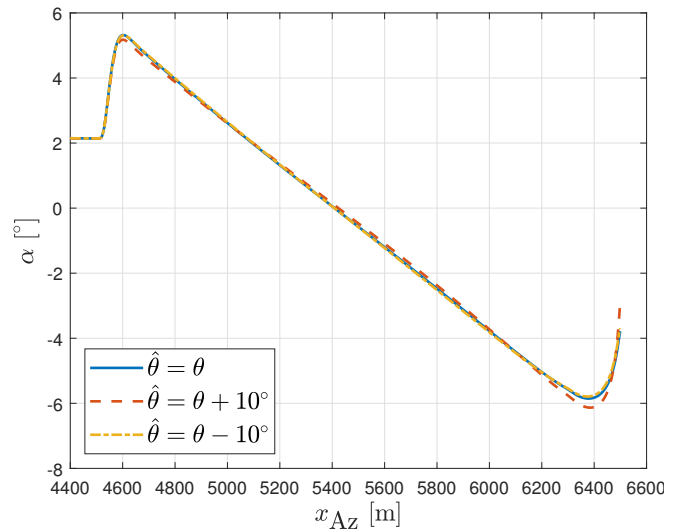
A negligible influence of errors in the estimated pitch angle can be observed on all surfaces. The only exception is the surface shown in Fig. 3.64c, where an increase in the maximum value of the specific force can be observed as the error in the estimated pitch angle increases, which is consistent with the results shown in Fig. 3.62c.

The effect of errors in the estimated flight-path angle on miss distance is symmetric, and the maximum miss distances within the selected error interval do not exceed 5 m. Errors in the achieved impact angle are also symmetric, but more pronounced than the miss distance errors, and a linear dependence on errors in the estimated flight-path angle can be observed. The remaining quantities exhibit an asymmetric behavior, which is expected since the sign of the error affects the trajectory shape: negative errors result in a "higher" trajectory, a larger velocity loss, and greater control effort.

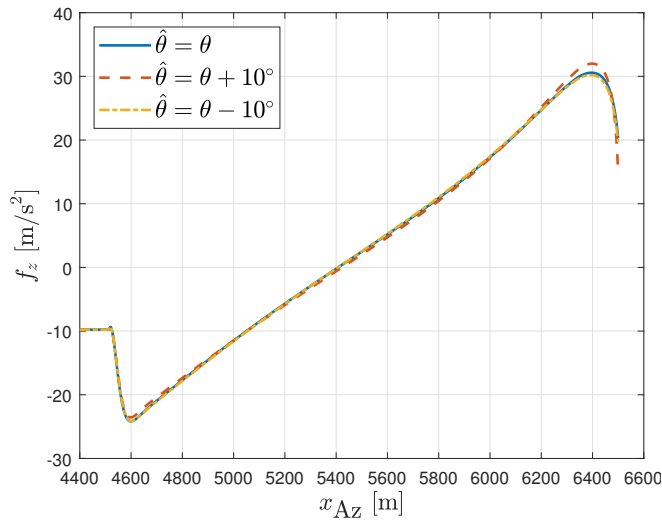
Based on the conducted analysis, it can be concluded that the optimal guidance law given by Eq. (3.95) is more sensitive to errors in the estimated flight-path angle than to errors in the estimated pitch angle. Even for extremely large values of these errors, the target miss distance remains small, although the errors in the achieved impact angle become more significant. Nevertheless, deviations of the impact angle from the desired value cannot jeopardize mission success to the same extent as significant target misses, which are unacceptable. Furthermore, these impact-angle deviations are proportional to the errors in the estimated flight-path angle, making it possible to determine the expected level of impact-angle error through a simple analysis based on the accuracy of the navigation data.



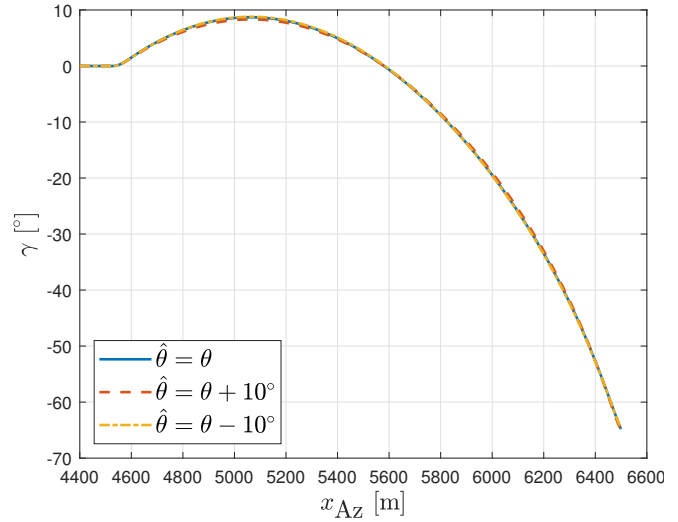
(a) Trajectory.



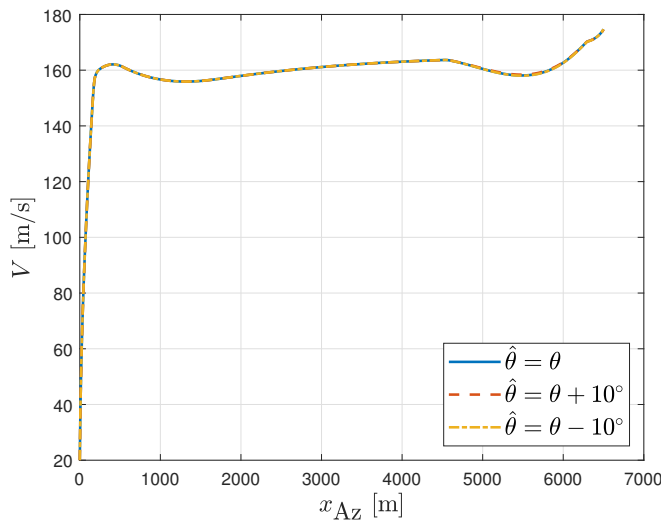
(b) Angle of attack.



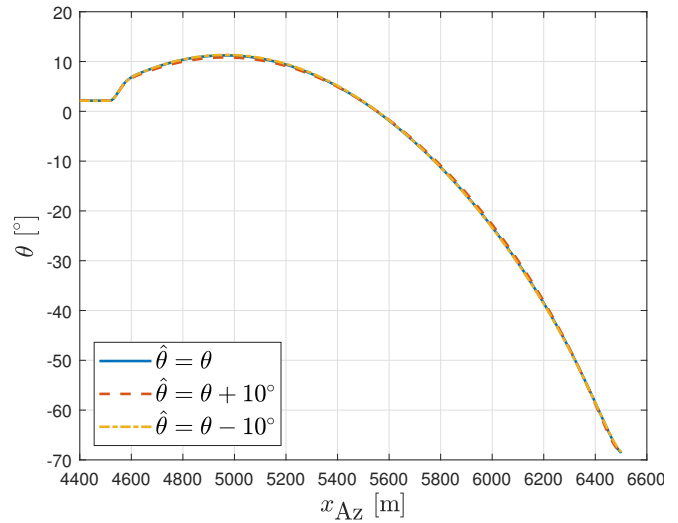
(c) Achieved specific force.



(d) Flight-path angle.

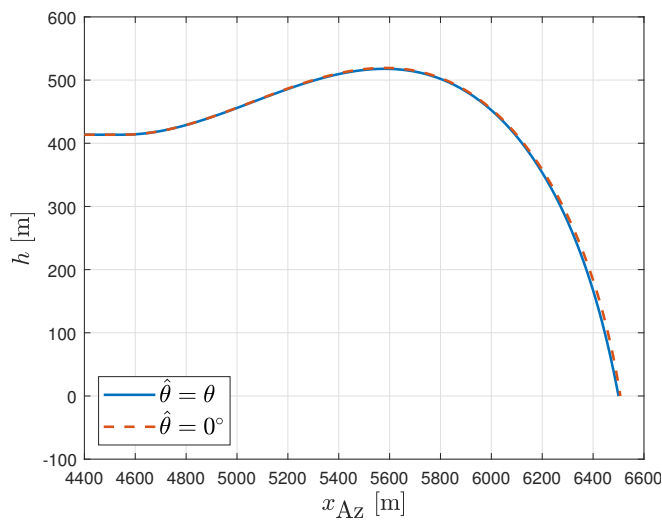


(e) Velocity.

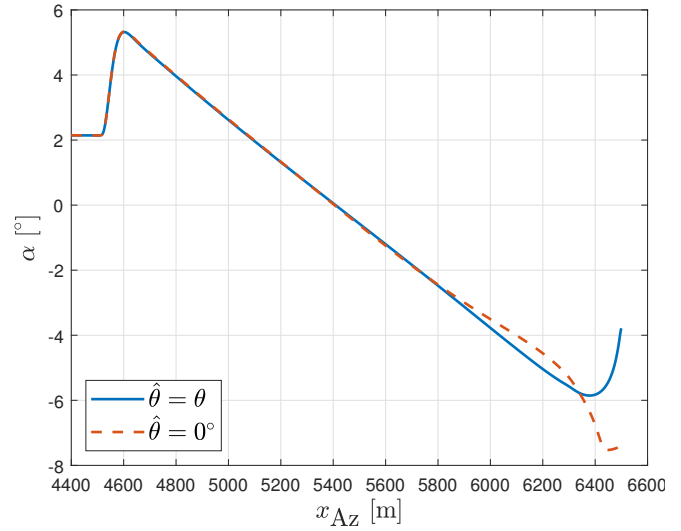


(f) Pitch angle.

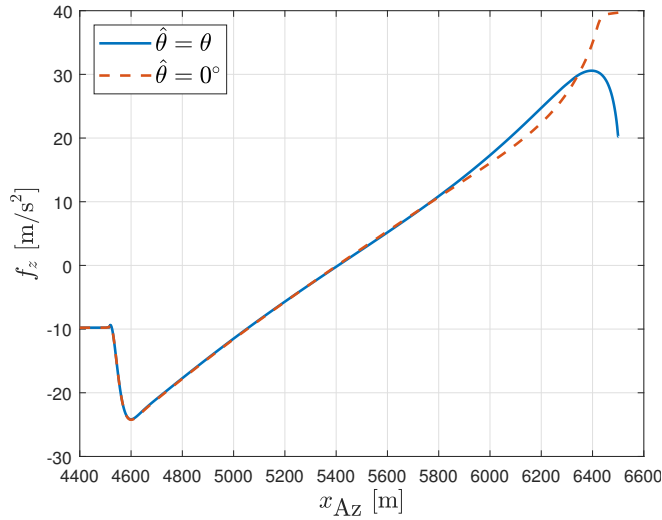
Figure 3.62: Effect of errors in the estimated pitch angle, $\gamma_1 = -65^\circ$.



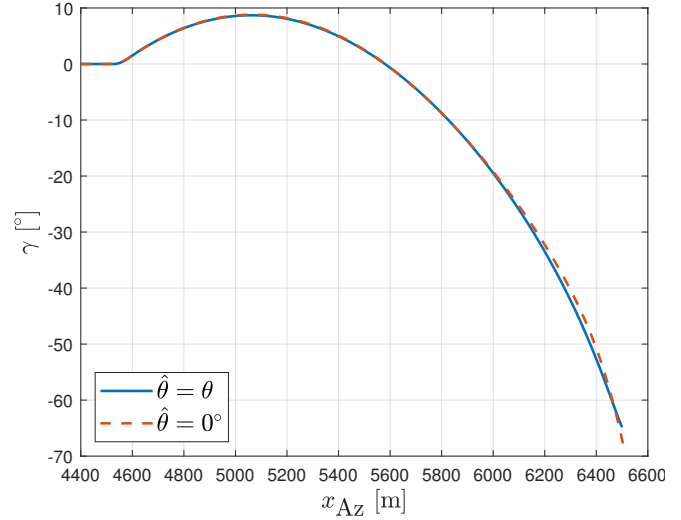
(a) Trajectory.



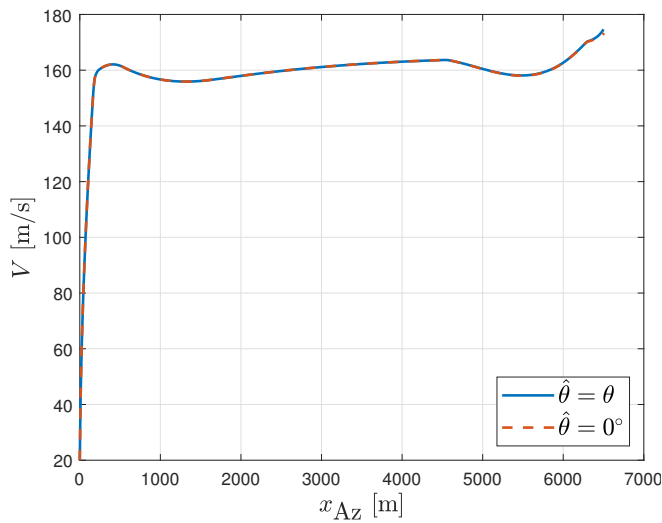
(b) Angle of attack.



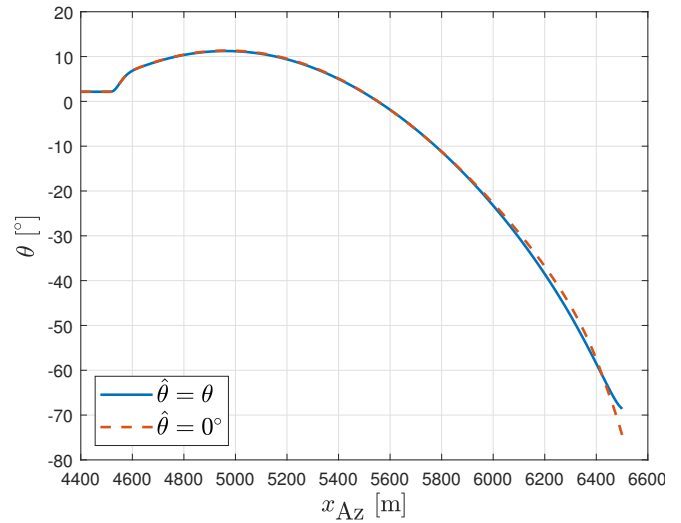
(c) Achieved specific force.



(d) Flight-path angle.



(e) Velocity.



(f) Pitch angle.

Figure 3.63: Effect of erroneous estimated pitch angle, $\gamma_1 = -65^\circ$.

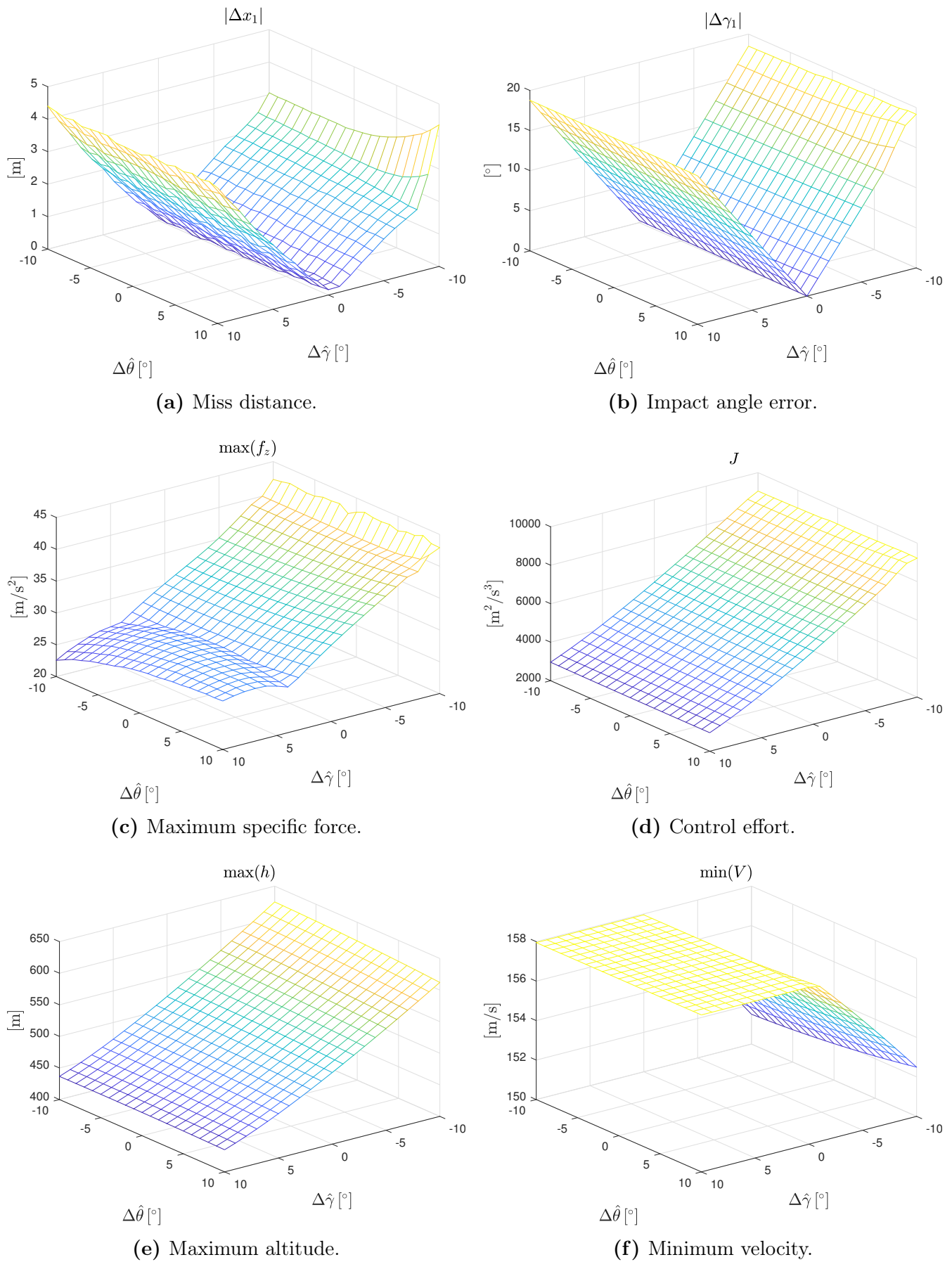


Figure 3.64: Effect of errors in estimated angles on the final results, $\gamma_1 = -65^\circ$.

3.4.2 Ballistic Missile

In this section, the performance of the proposed algorithm is evaluated in a short-range ballistic missile scenario.

The analysis presented in this section is limited to the vertical plane; therefore, no results for the horizontal plane are shown. For all examples considered herein, it is assumed that the launch azimuth has been selected appropriately, while the guidance logic in the horizontal plane remains unchanged from the standard missile configuration. The missile is roll-stabilized, allowing two independent guidance and control channels to be employed. To focus exclusively on the performance of the vertical guidance channel, neither roll disturbances nor disturbances in the horizontal plane are considered.

Figure 3.65 shows the nominal trajectory for a range of 180 km. The initial pitch angle was 46.32° , resulting in a terminal flight-path angle of -49.25° . No guidance was employed in this example. The velocity profile, shown in Fig. 3.65b, illustrates that the missile experiences a wide range of flight speeds, spending most of its flight in the supersonic regime. Consequently, this example provides a suitable test case for demonstrating the robustness of the closed-loop guidance formulation proposed in this work.

Among the various disturbances that can contribute to the dispersion of artillery rockets during launch and flight, deviations in the total impulse of the rocket motor are particularly significant when range accuracy is of interest. Therefore, Fig. 3.66 presents comparative results for the unguided rocket case with total impulse deviations of $\pm 2\%$. This deviation is implemented by modifying the thrust level while maintaining a constant motor burn time in all cases.

As can be seen from the trajectory diagram in Fig. 3.66a, total impulse deviations result in considerable miss distances, reaching approximately 12 km in both directions, as also summarized in Table 3.9. The velocity profiles shown in Fig. 3.66b are modified accordingly due to the increased or decreased available energy. However, no significant differences are observed in the impact angle, as shown in Fig. 3.66d.

Clearly, deviations of this type must be compensated for by an appropriate guidance system. In this example, it is demonstrated that the optimal guidance method proposed in this work can also be successfully applied to ballistic missile scenarios. Although impact-angle control is rarely of practical interest for such systems (one exception being the intentional increase of the impact angle to reduce sensitivity to navigation errors), the following examples also demonstrate the ability of the proposed method to achieve prescribed impact angles.

The guidance logic is implemented in the same manner as described in Section 3.4.1 for the cruise missile case, with the vertical plane corresponding to the vertical plane of the azimuth coordinate frame. The same algorithm (Algorithm 2) is employed. The only difference is that, in the present examples, optimal guidance is activated from the early phase of flight.

There are two reasons for this choice. First, the considered ballistic missile reaches significantly greater altitudes than the cruise missile, and its control effectiveness decreases substantially at high altitudes, particularly because it is controlled exclusively through aerodynamic control surfaces. Second, disturbances caused by rocket-motor performance variations must be compensated for

early in the flight, before the trajectory deviates significantly from the nominal path and before control effectiveness is reduced at high altitude. Another potential application of the optimal guidance approach developed in this dissertation for ballistic missile scenarios will be discussed in the concluding chapter as a topic for future research.

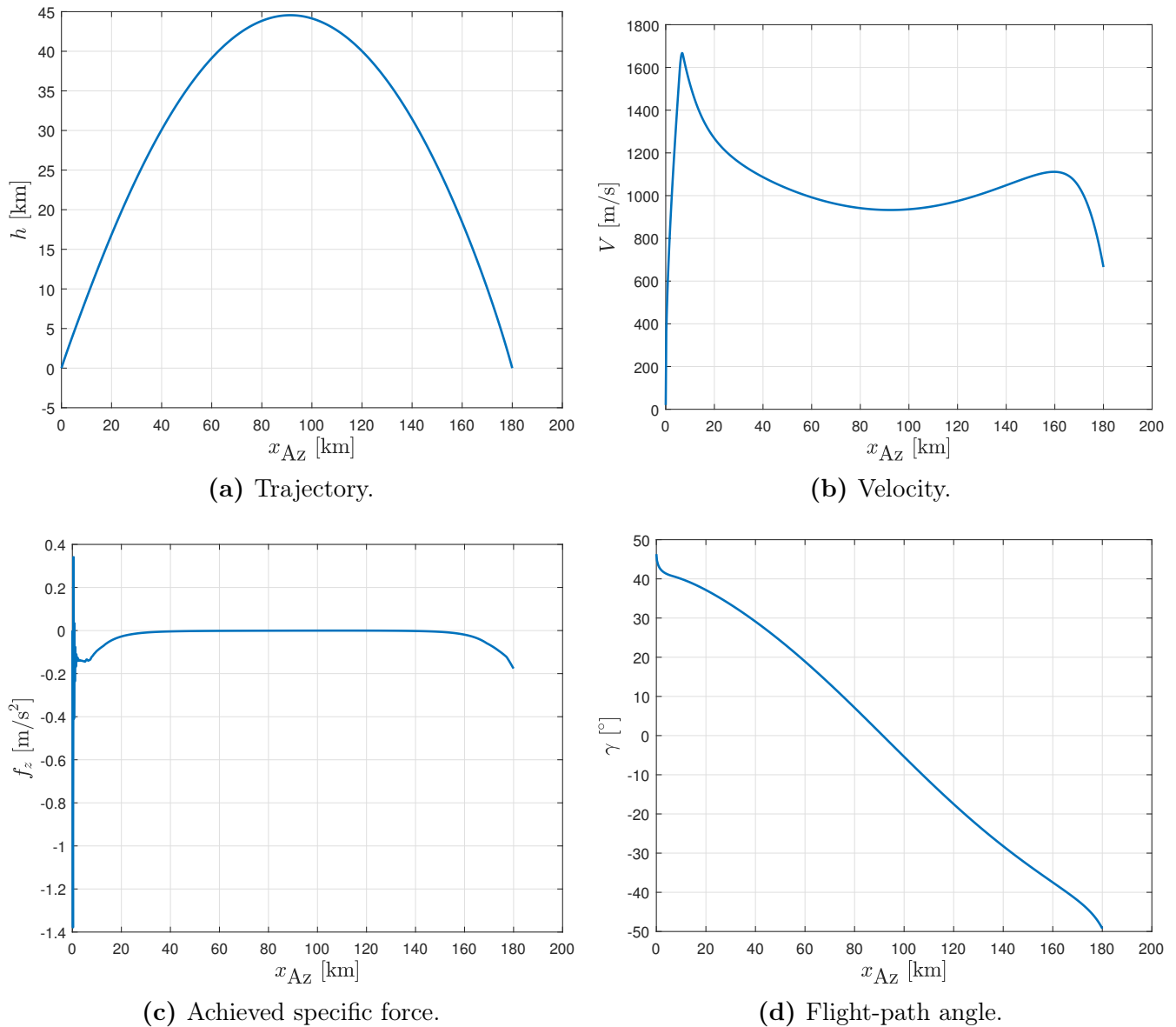


Figure 3.65: Nominal ballistic trajectory of an unguided artillery rocket at a range of 180 km.

Figure 3.67 presents the results for the same total impulse deviations considered previously, but now with optimal guidance active throughout the entire flight. It should be noted that the trajectory labeled as *nominal* corresponds to the nominal total impulse case without guidance, since the objective is to reproduce its range and impact-angle characteristics. It is therefore included only as a reference trajectory.

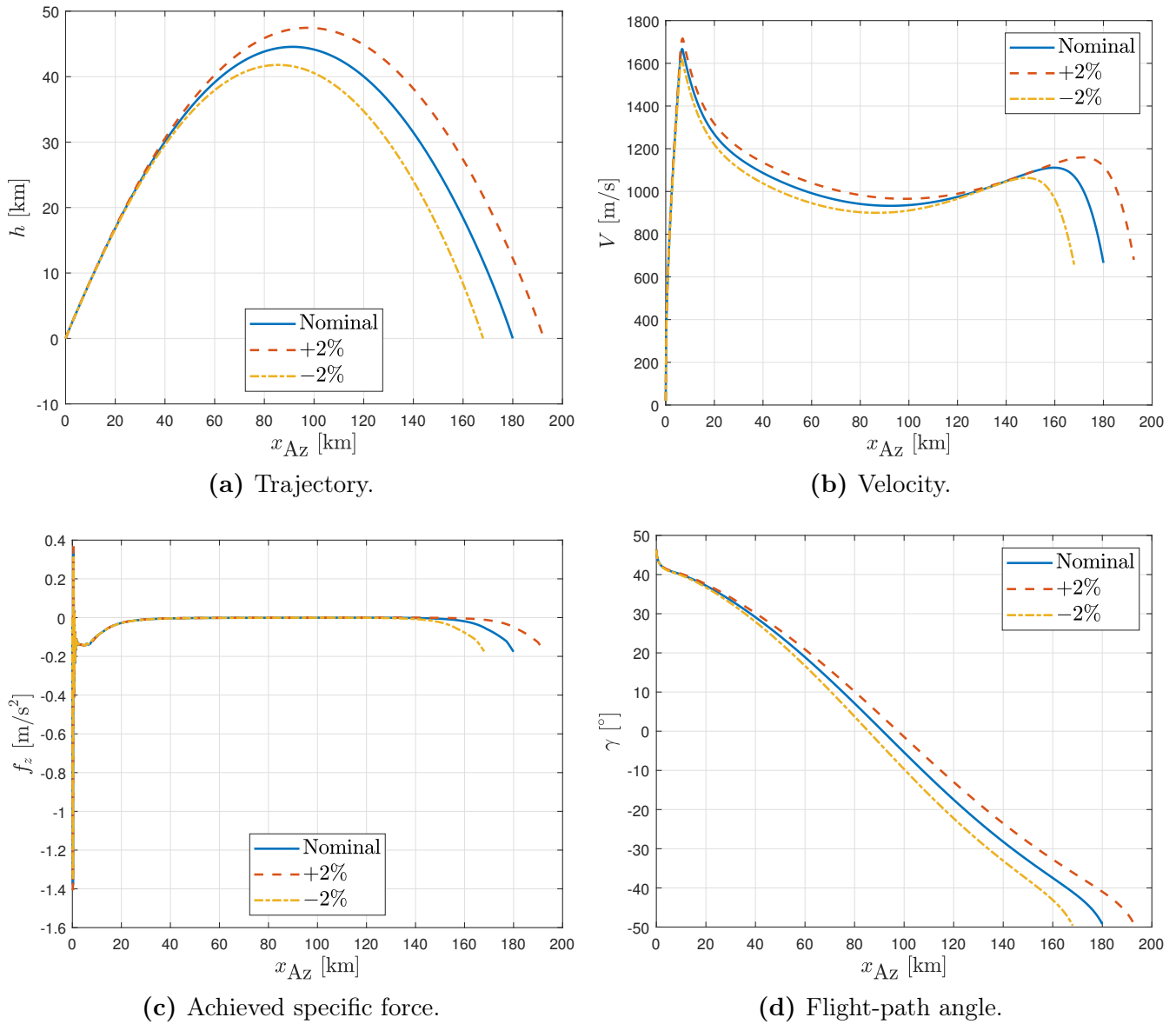


Figure 3.66: Comparison of results for an unguided rocket with total impulse deviations.

Table 3.9: Final results for an unguided rocket with total impulse deviations.

Total Impulse Deviation	$x(t_1)$ [m]	$\gamma(t_1)$ [°]
/	180000	-49.25
+2%	192505	-48.89
-2%	168051	-49.65

As can be seen from Fig. 3.67a and Table 3.10, even in the presence of severe system deviations, the target is intercepted with negligible errors in both miss distance and impact angle. Interestingly, although of little practical significance because the errors are already extremely small, both objectives are achieved with slightly greater accuracy in the case of reduced total impulse.

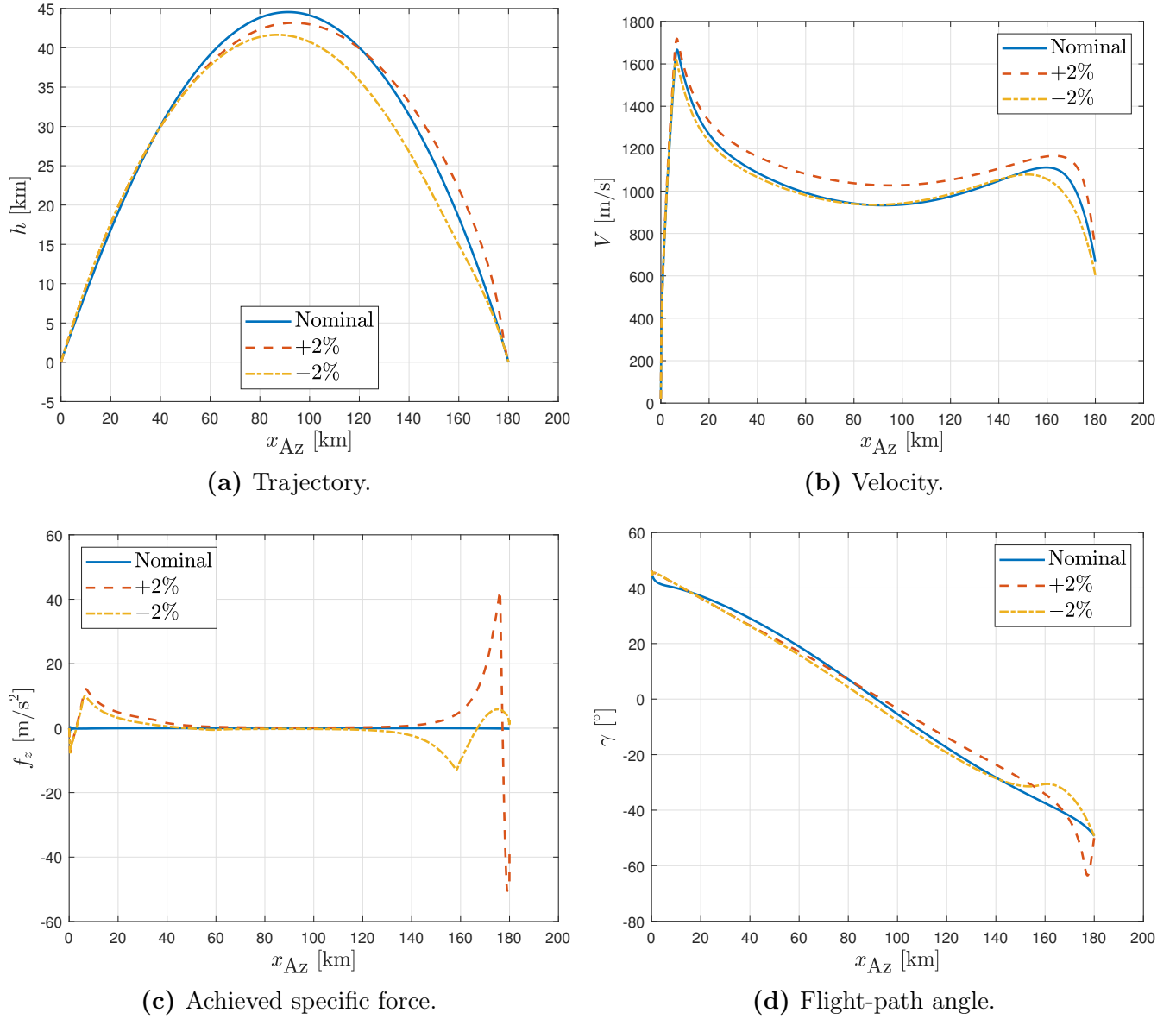


Figure 3.67: Comparison of results obtained using optimal guidance (Algorithm 2) with total impulse deviations, $\gamma_1 = -49.25^\circ$.

Table 3.10: Final results obtained using optimal guidance (Algorithm 2) with total impulse deviations, $\gamma_1 = -49.25^\circ$.

Total Impulse Deviation	$x(t_1)$ [m]	$\gamma(t_1)$ [°]
/	180000	-49.25
+2%	179992	-49.82
-2%	180000	-49.28

This behavior is a consequence of the substantially different trajectories obtained in the two cases. In both scenarios, the most pronounced trajectory shaping occurs during the initial climb phase

and during the final re-entry into denser atmospheric layers, as evident from the specific-force profiles shown in Fig. 3.67c. In both cases, the greatest maneuvering effort is required during the terminal phase.

For the -2% total impulse deviation, this maneuvering is necessary to prolong the trajectory toward the target, which is also evident from the flight-path-angle profile shown in Fig. 3.67d. In contrast, for the $+2\%$ total impulse deviation, the terminal maneuvering is required to manage excess energy. The temporary decrease of the flight-path angle below the desired value during the final phase of flight is a consequence of the delayed recovery of control effectiveness. The missile must initially pitch down to ensure target interception and subsequently pitch up to achieve the desired impact angle.

The desired impact angle of -49.25° was selected to match the impact angle achieved by the nominal unguided trajectory without total impulse deviations. In practice, however, the desired impact angle should be selected according to mission requirements. As shown in Fig. 3.67c, significant maneuvering is required in the $+2\%$ total impulse case, despite the successful target interception. Consequently, it is of interest to determine a more suitable impact angle that remains achievable under the expected range of system deviations.

Figure 3.68 therefore presents results for a desired impact angle of -45° . In this case, the trajectory labeled as *nominal* corresponds to a guided trajectory generated using Algorithm 2 with nominal total impulse, but with the modified desired impact angle.

As can be seen from Fig. 3.68a and Table 3.11, precise target interception is achieved in all cases, with essentially zero error in both miss distance and impact angle. Similar trajectory-shaping mechanisms can be observed as in the previous example. However, for the -2% total impulse deviation, increased maneuvering is required during the terminal phase to extend the trajectory toward the target, whereas for the $+2\%$ case, the guidance system performs more aggressive trajectory shaping during the early climb phase to compensate for excess energy. Despite these differences, the required specific force remains well within the maneuverability limits of the missile in all three cases.

Clearly, the impact angle of -45° is better suited to the considered scenario and the expected range of system deviations.

A study similar to the multi-parameter optimization analysis presented for the cruise missile case could also be performed here in order to determine the optimal impact angle. However, such an analysis is beyond the scope of the present work and is more closely related to mission planning and fire-control system design. The objective of this section is instead to demonstrate the applicability of Algorithm 2 across substantially different flight regimes, ranging from low-altitude cruise flight to high-dynamic ballistic re-entry.

It is also important to note that, in the ballistic missile examples, the proposed guidance method remained effective under highly variable velocity conditions. The velocity variations shown in Figs. 3.67b and 3.68b differ significantly from the constant-speed assumption adopted during the derivation of the proposed guidance law. Nevertheless, the closed-loop nature of the method, in which the optimal command is recomputed during every guidance cycle using instantaneous navigation estimates of missile position and velocity, proved highly effective even under the strongly

varying speed conditions characteristic of supersonic ascent and re-entry. These results confirm that the proposed algorithm maintains full functionality as the missile transitions through the wide range of speeds and altitudes characteristic of ballistic trajectories.

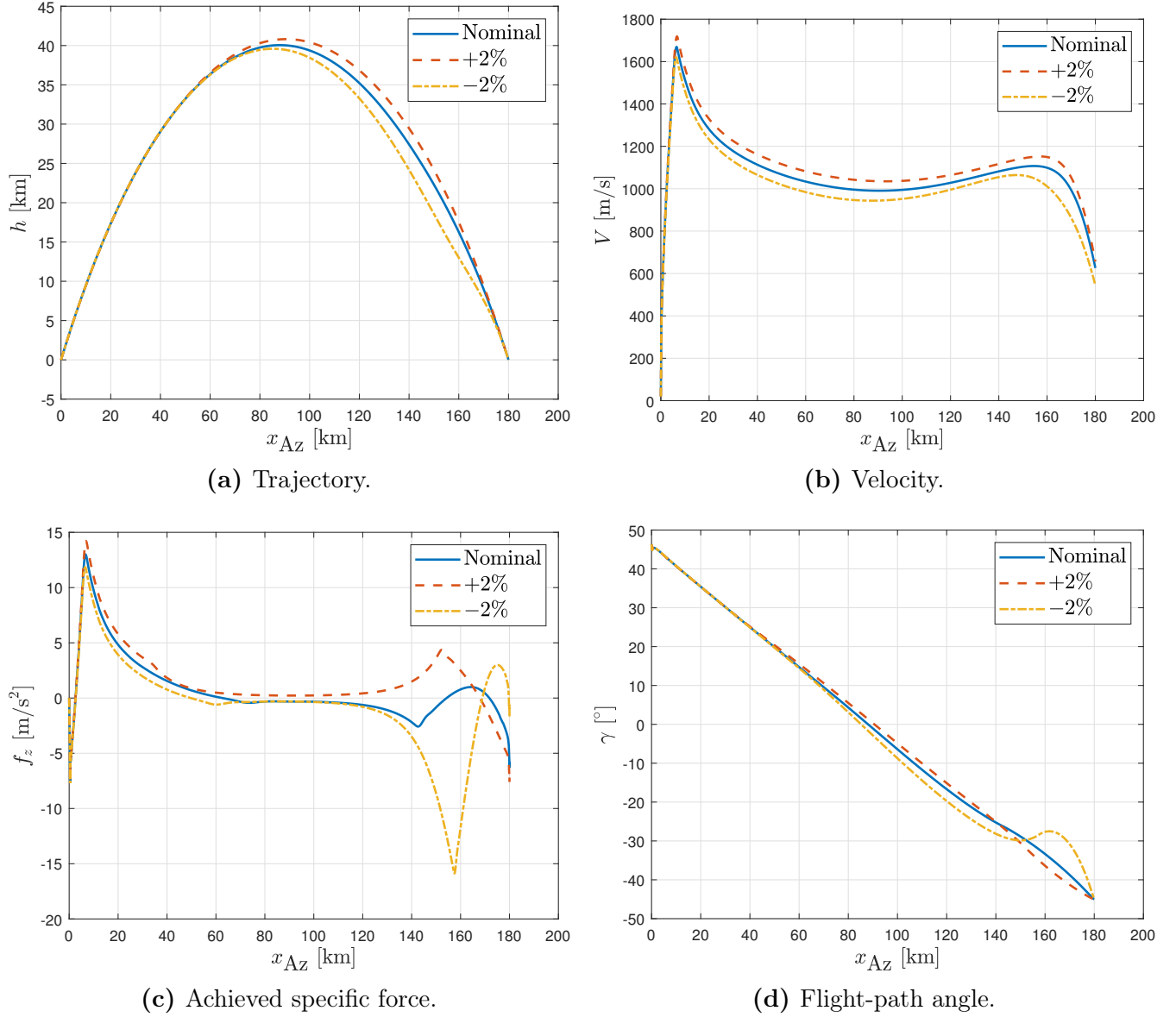


Figure 3.68: Comparison of results obtained using optimal guidance (Algorithm 2) with total impulse deviations, $\gamma_1 = -45^\circ$.

Table 3.11: Final results obtained using optimal guidance (Algorithm 2) with total impulse deviations, $\gamma_1 = -45^\circ$.

Total Impulse Deviation	$x(t_1)$ [m]	$\gamma(t_1)$ [°]
/	180000	-45
+2%	180000	-45
-2%	180000	-45.02

3.5 Hardware-in-the-Loop Validation and Results

This section presents the validation of the proposed guidance algorithm through high-fidelity six-degree-of-freedom (6DOF) simulations conducted in a hardware-in-the-loop (HIL) laboratory environment. A part of the practical implementation of the proposed algorithm was already addressed in Section 3.3, where the focus was placed on implementation aspects and verification through a Software-in-the-Loop (SIL) environment. In the present section, the same onboard computer (OBC) and the same high-fidelity 6DOF flight simulation are employed; however, the OBC is mounted on a three-axis motion platform and uses real inertial measurement unit (IMU) data from physical gyroscopes, together with real actuators operating within the control loop.

The HIL architecture, shown in Fig. 3.69, integrates the onboard computer with a 6DOF real-time simulator and physical actuators. The OBC is mounted on a three-axis platform that receives attitude commands from the simulator via UDP. Initial mission parameters are uploaded from the ground control station through an RS-232 interface. Flight data and acceleration measurements are provided to the OBC via Ethernet. During flight, the OBC transmits control surface commands to real actuators through the CAN protocol. The resulting physical deflections, measured by high-resolution encoders, are fed back into the 6DOF simulation, thereby closing the control loop. All data are simultaneously broadcast by the OBC via UDP and recorded on a dedicated workstation.

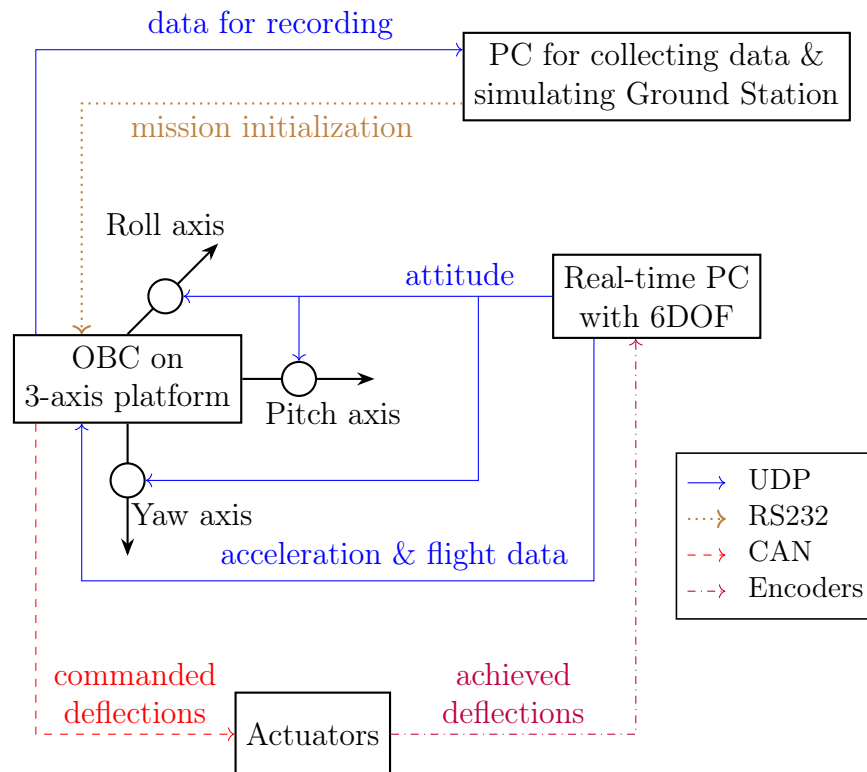


Figure 3.69: Schematic of the HIL architecture and communication interfaces.

The results presented in this section were obtained during preparations for flight testing of the A50 missile. Both this system and the specific test will be discussed in more detail in Section 3.6.

The mission profile considered here is similar to that investigated in the 6DOF numerical simulation study presented in Section 3.4. It consists of an initial climb to the cruise altitude, a cruise segment toward the target vicinity, and a terminal guidance phase leading to target interception. The 6DOF model executed on the real-time simulator differs from the one used in Section 3.4, as it was adapted to the characteristics of the A50 missile.

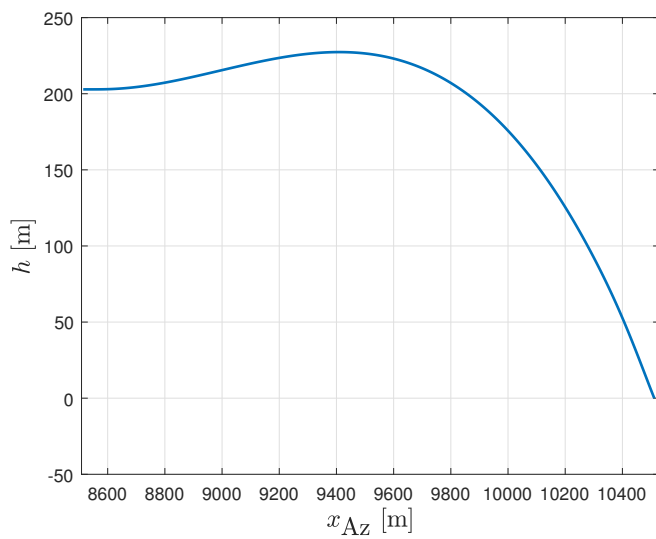
Figure 3.70 presents the HIL results for the nominal trajectory at a range of 10.5 km. All diagrams correspond to the terminal guidance phase, which begins when the missile is 2 km from the target in downrange distance, except for the velocity profile shown in Fig. 3.70e, which is presented over the entire flight to better illustrate the overall speed evolution. A detachable booster rocket motor is used immediately after launch to accelerate the missile, while a small turbojet sustainer engine maintains the flight speed during the cruise phase. Consequently, the missile velocity remains nearly constant at approximately 140 m/s throughout most of the cruise segment.

The desired impact angle was selected as -25° . Although this value belongs to the group of relatively small impact angles, it was chosen for two reasons. First, these tests were conducted during an early verification phase of the missile system, and proving-ground safety regulations required the impact angle to remain moderate in order to avoid excessive altitude gain during the terminal phase. Second, the three-axis motion platform used in laboratory testing has limitations regarding the achievable angular positions. Since the primary objective of this development phase was to verify the required guidance accuracy throughout the flight envelope while incorporating real subsystems into the loop, the impact angle of -25° was considered appropriate. Larger impact angles, which require greater maneuverability, will be investigated during later development stages.

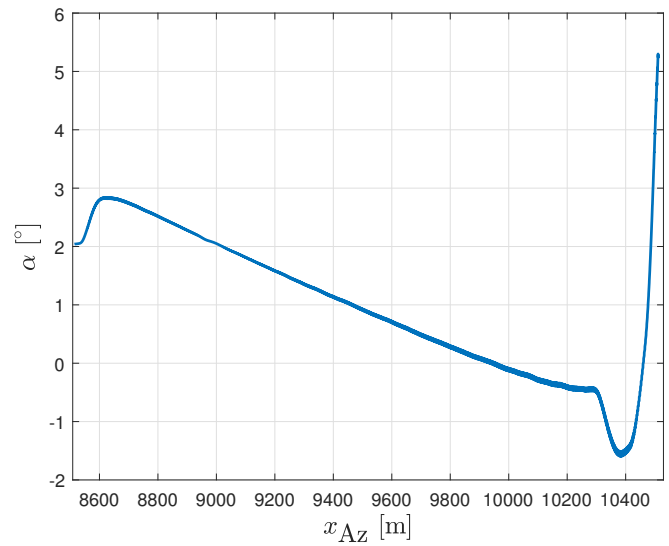
As a consequence, the achieved specific force, shown in Fig. 3.70c, remains well below the maneuverability limits of the missile. The resulting trajectory, presented in Fig. 3.70a, exhibits only a modest increase in altitude during terminal engagement, with a maximum climb of approximately 25 m above the cruise altitude of 200 m.

For the nominal trajectory, the actual downrange miss distance was approximately 6 m, while the achieved terminal flight-path angle error was less than 0.5° . The presence of a small miss distance is primarily related to practical implementation aspects of the guidance, navigation, and control system. During testing, a real inertial navigation system (INS) was employed, using physical gyroscope measurements and accelerometer data synthesized from the 6DOF simulation. Consequently, the guidance algorithm operated on realistic navigation estimates rather than exact simulation states. For the nominal trajectory, the final INS position error was approximately 1.5 m, rather than the actual 6 m miss distance.

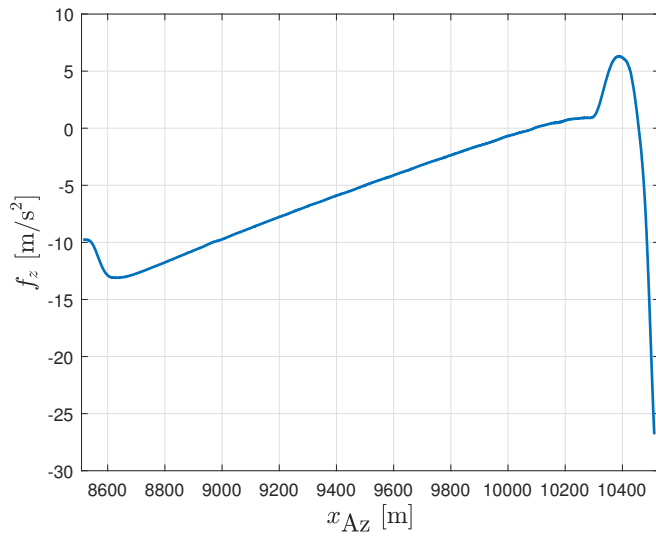
Furthermore, the navigation solution was not based solely on the INS. During the experiment, satellite navigation aiding was employed using synthetic GNSS measurements generated by the real-time simulator. However, to mimic real-world scenarios where the quality of satellite-based corrections degrades in the vertical channel, this aiding was intentionally disabled during the larger terminal pitching-down maneuver. In addition, a simulated altimeter signal was introduced to improve altitude estimation during the flight. This aiding was also disabled below an altitude of 100 m to account for the various ground-level parasitic effects that typically degrade physical measurement quality.



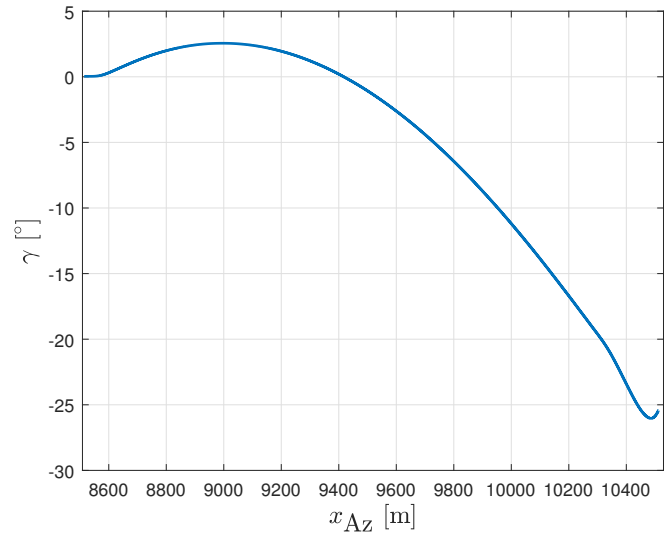
(a) Trajectory.



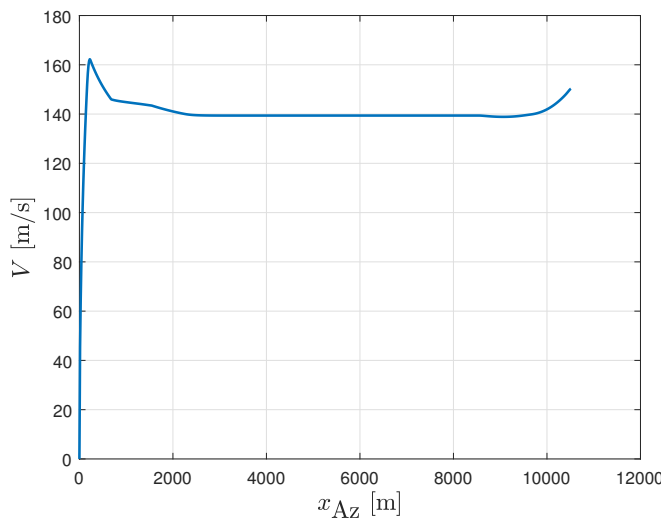
(b) Angle of attack.



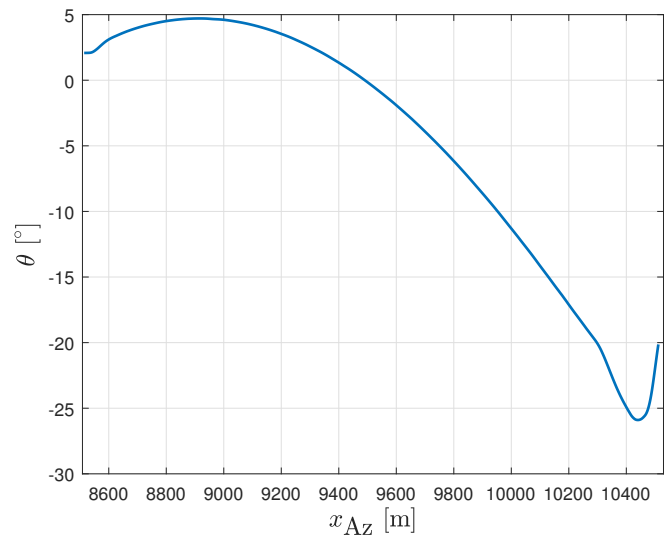
(c) Achieved specific force.



(d) Flight-path angle.



(e) Velocity.



(f) Pitch angle.

Figure 3.70: HIL results for the nominal case, $\gamma_1 = -25^\circ$.

The transition in navigation logic can be observed directly in the recorded data. The sudden change in normal acceleration visible in Fig. 3.70c, approximately 200 m before target impact, occurs because altitude estimation becomes dependent solely on the INS after the altimeter aiding is disabled. In the considered example, the INS-based altitude estimate was slightly higher than the aided estimate, resulting in an increased command intended to bend the trajectory downward. Consequently, a short time later, a maneuver in the opposite direction becomes necessary in order to recover the desired impact angle. This final maneuver is clearly visible in Figs. 3.70c, 3.70d, and 3.70f, where it appears as a pitching-up motion near the end of the flight.

It is also instructive to briefly analyze the operation of the optimal guidance algorithm itself. As established in previous sections, the core of the algorithm consists of solving a single nonlinear equation. The chosen numerical method is the bisection method, for which the expected number of iterations is 22. Consequently, the computational burden is fixed and has already been quantified and analyzed.

From the algorithmic perspective, however, it is interesting to examine which trajectory-convexity case is active during different portions of the terminal phase. Figure 3.71 therefore presents the same trajectory and flight-path-angle profile, but with regions indicating the active case (A, B, or C). These cases were introduced in Section 2.2.4 and correspond to different trajectory convexity conditions.

As expected, the terminal phase begins in Case B, indicating that the missile must initially gain altitude in order to achieve the desired impact angle. After passing the inflection point, the algorithm transitions to Case A. Under ideal conditions, the trajectory would remain within this case until target interception, continuously decreasing the flight-path angle toward the desired terminal value. However, due to practical imperfections—including non-ideal control-system tracking and navigation-related effects such as the sensor-fusion transition discussed previously—Case C also appears during a portion of the flight. This occurs precisely when the INS-altimeter fusion is disabled, forcing the trajectory to first reduce altitude and subsequently increase the flight-path angle again to satisfy the impact-angle requirement.

In practice, the final portion of the trajectory may involve various combinations of convexity cases depending on the specific flight conditions. Nevertheless, for the beginning of terminal guidance, the sequence B-A-... is observed in almost all practically relevant scenarios, except for very small impact angles that are generally of limited operational interest.

Finally, Table 3.12 summarizes the test descriptions along with the primary performance indicators—the miss distance and impact-angle error—for a subset of the standard HIL qualification tests performed for the considered missile. The corresponding results are presented in Fig. 3.72. These tests include the simulation of manufacturing imperfections such as booster-motor misalignment and roll asymmetry, as well as environmental disturbances including wind and heading errors.

Although many of these disturbances primarily affect the initial phase of flight, they provide an effective means of generating diverse initial conditions prior to terminal guidance, thereby broadening the range of scenarios in which the proposed guidance algorithm must demonstrate satisfactory performance.

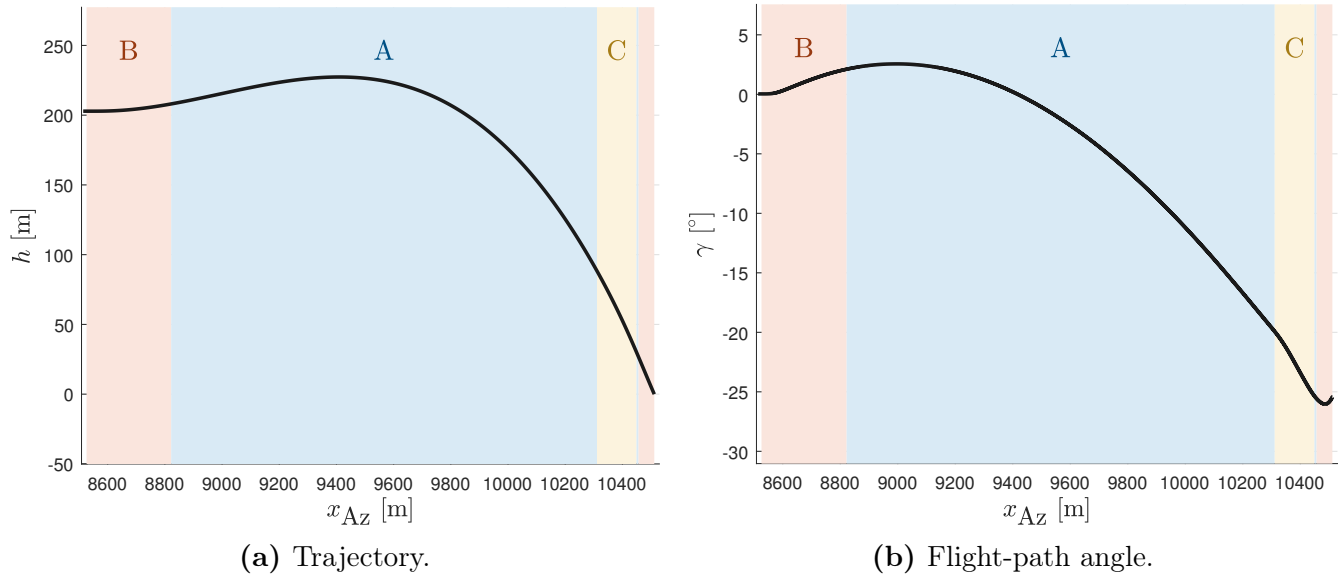


Figure 3.71: Results for the nominal case, highlighting the trajectory-convexity cases.

Table 3.12: Subset of standard HIL test cases and corresponding results.

Test No.	Test Description	Results		
		Δx_1 [m]	$(\Delta x_1)_{\text{INS}}$ [m]	$\Delta \gamma_1$ [°]
0	Nominal case	6.3	-1.5	0.4
1	Positive roll asymmetry and $+0.1^\circ$ pitch booster misalignment	6.1	-0.5	0.5
2	Double negative roll asymmetry and -0.1° pitch booster misalignment	6.2	-2.3	0.5
3	Negative roll asymmetry with crosswind and turbulence	6.4	-1.1	0.4
4	Double positive roll asymmetry with severe crosswind and turbulence	6.2	-0.5	0.5
5	Positive roll asymmetry, $+1^\circ$ azimuth error, and $+0.1^\circ$ yaw booster misalignment	6.7	-0.2	0.5
6	Positive roll asymmetry, -1° azimuth error, and -0.1° yaw booster misalignment	6.6	-1.5	0.4

The differences between the trajectories shown in Fig. 3.72 are relatively small, indicating consistent behavior across the considered scenarios. The results summarized in Table 3.12 show a maximum INS-estimated miss distance of 2.3 m, with a mean value of approximately 1 m. In contrast, the actual miss distance is close to 6 m for all tests, indicating the presence of a systematic navigation error associated with the flight duration and sensor configuration employed in this study.

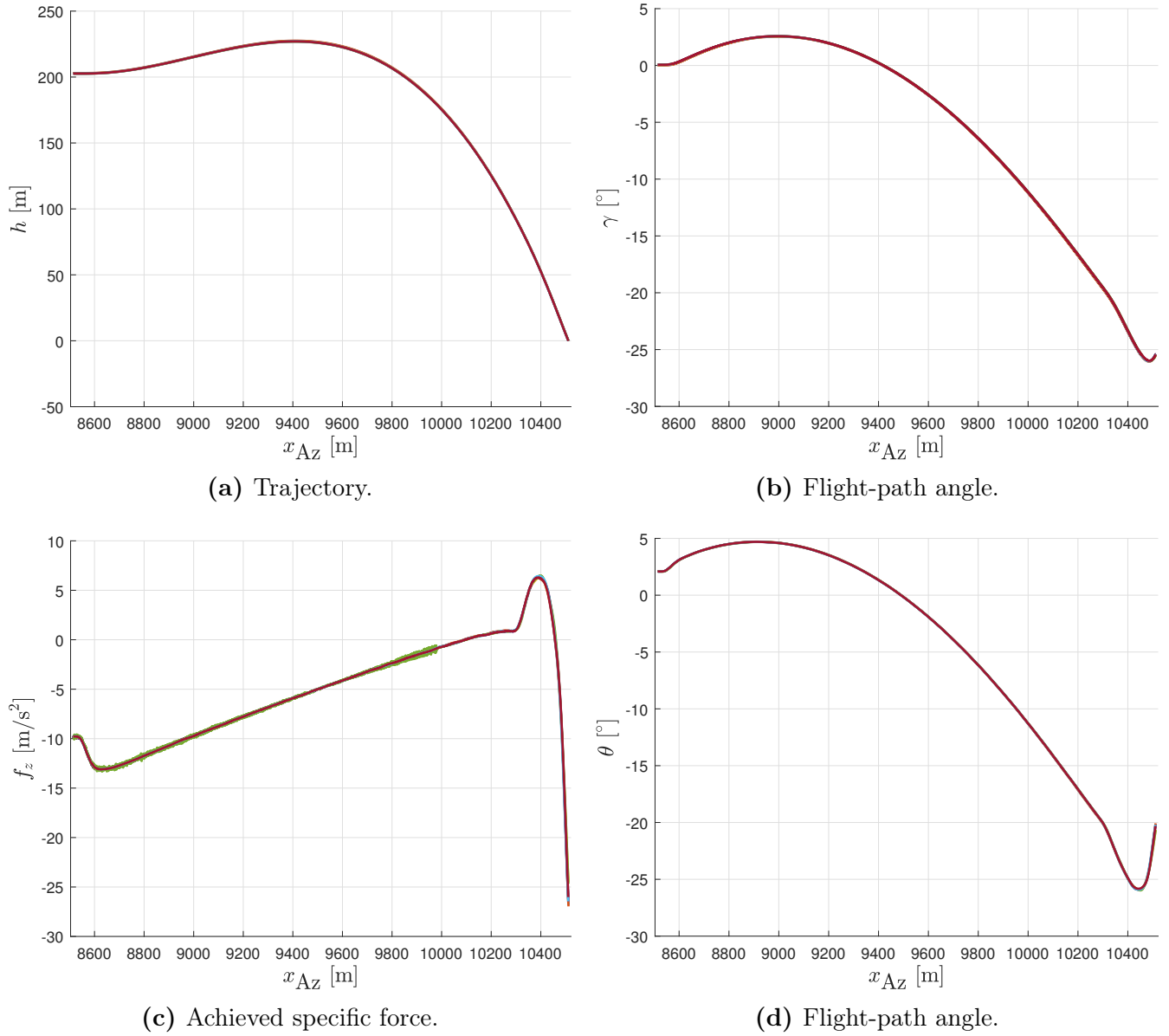


Figure 3.72: Results for the six HIL test cases summarized in Table 3.12.

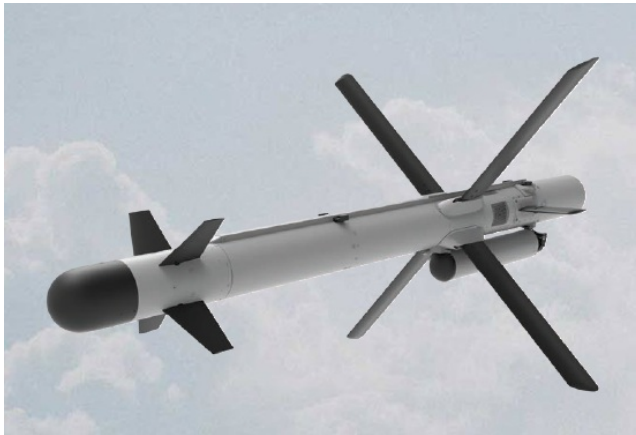
Naturally, different mission scenarios and flight durations would result in different miss distances, which is a well-known characteristic of INS-based navigation systems. Nevertheless, the achieved impact-angle accuracy is excellent. In all cases, the terminal flight-path angle differs from the desired value by no more than 0.5° . This observation is consistent with the analysis presented in Section 3.4.1, which showed that navigation errors primarily manifest themselves as position errors and therefore directly affect the miss distance, whereas accurate estimation of the relevant angular states leads to excellent impact-angle performance.

3.6 Flight Test

This section presents the results obtained during field testing of the A50 missile, in which the guidance method proposed in this dissertation was implemented.

The A50, shown in Fig. 3.73a, is a fifth-generation non-line-of-sight (NLOS) cruise missile with a range of 50 km that is currently under development. Similar to other cruise missiles intended for the same class of missions, such as the ALAS-XX missile [102], currently in service with the Serbian Armed Forces and shown in Fig. 3.73b, the A50 is launched using a detachable booster rocket motor, after which the cruise phase is sustained by a turbojet engine that maintains an approximately constant flight speed. The overall mission profile consists of the launch phase, cruise flight toward the target vicinity using waypoint guidance, and terminal engagement.

Two operational scenarios arise naturally for this type of missile. The first relies on an onboard seeker, either electro-optical (EO) or imaging infrared (IIR), together with a radio data link to the ground control station for target detection and tracking. This mode is primarily intended for engaging armored vehicles and other moving targets. The second scenario is intended for strikes against stationary ground targets and reflects the ongoing evolution of light cruise missiles toward extended-range land-attack capabilities. This latter scenario corresponds exactly to the application for which the guidance method proposed in this dissertation was developed.



(a) A50 model. [103]



(b) ALAS-XX launch. [104]

Figure 3.73: Fifth-generation cruise missiles for land attacks.

Consequently, the proposed guidance method was implemented on the onboard computer of the A50 missile and evaluated during a flight test conducted over a range of 21.4 km, representing the first flight of this missile beyond 10 km. The primary objectives of the test were related to the first long-range flight of the A50 model, including verification of the turbojet sustainer engine, the electrical power subsystem, and other newly introduced components. At the same time, the proposed guidance algorithm was evaluated, since the missile was not equipped with a seeker and navigation relied exclusively on onboard navigation data. All telemetry and flight data were transmitted and recorded via the onboard radio data link for subsequent analysis.

The missile was launched at an elevation angle of 35° , with a commanded cruise altitude of 200 m. The optimal guidance method proposed in this dissertation was applied only in the vertical plane, with a desired terminal flight-path angle of -35° and terminal guidance initiated 2 km from the target. Since the commanded cruise altitude was relatively low, the trajectory-tracking guidance law described in Section 3.4.1 was retained during the initial climb phase, while the proportional navigation law was used in the horizontal channel. The missile was equipped with an inertial measurement unit (IMU) providing inputs to the navigation algorithm, while the INS was aided by both a GNSS receiver and an altimeter.

Unfortunately, because of the non-ideal installation of the radio-link receiver antenna on the launch test vehicle, two interruptions in telemetry data occurred during the flight. The first interruption took place during the cruise phase, where no significant events occurred. The second occurred near the end of the terminal phase, when the missile was already approaching the target with a steep negative pitch angle.

Figure 3.74 presents the flight results in the horizontal plane of the azimuth coordinate frame. The azimuth-frame coordinates were computed from the missile geographic position obtained through GPS/INS integration. As can be seen, the horizontal guidance channel performed successfully throughout the entire flight, maintaining only small deviations from the desired course and achieving a negligible lateral miss distance.

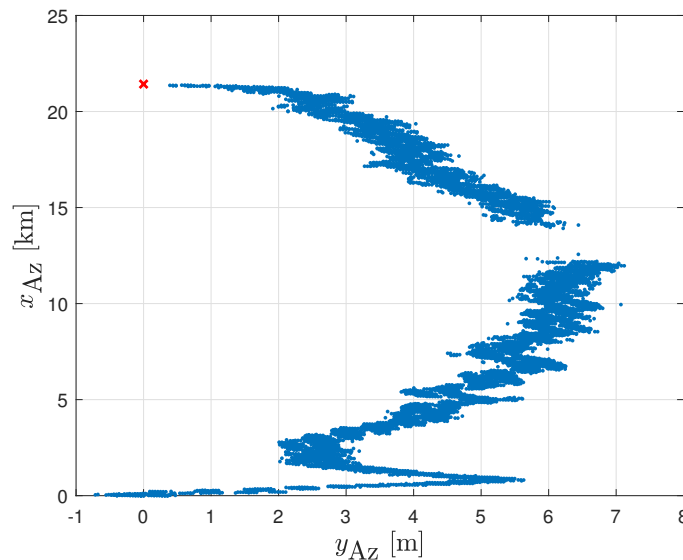
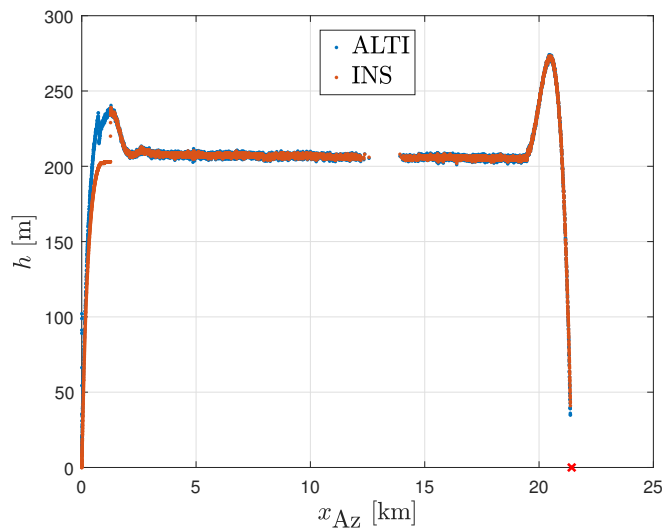
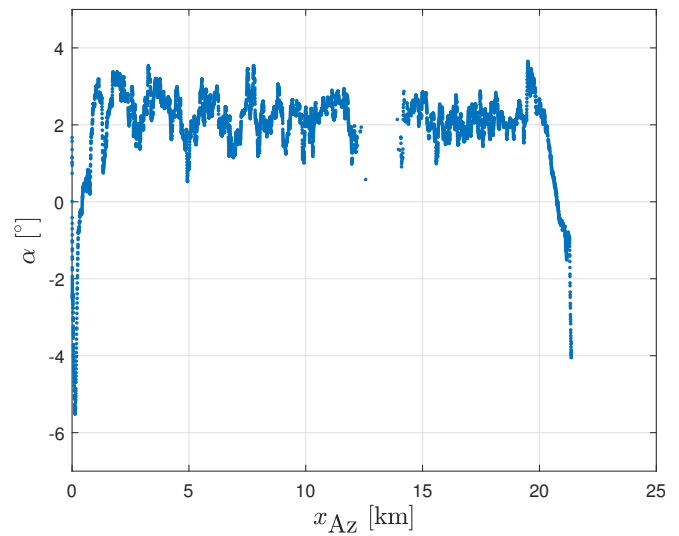


Figure 3.74: Missile trajectory in the horizontal plane.

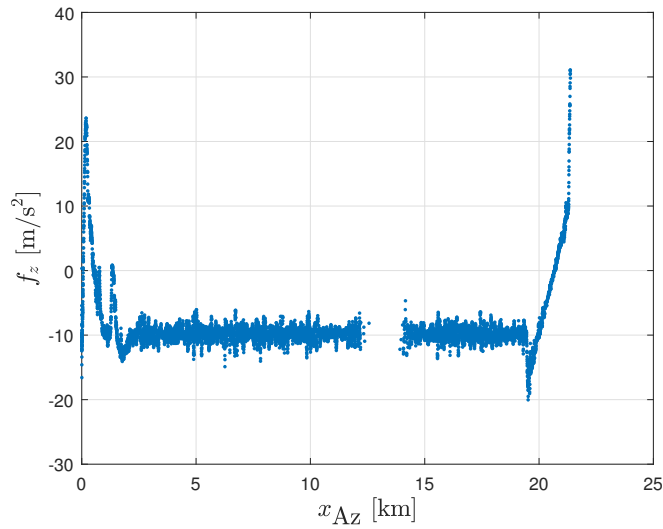
Figure 3.75 presents the flight variables of interest for the vertical channel over the entire flight. The cruise phase was completed successfully before the initiation of terminal guidance, while the cruise speed remained approximately constant at 140 m/s. Figure 3.75a compares the altitude measured directly by the altimeter with the altitude estimated by the integrated INS/altimeter solution. The instant at which altitude aiding becomes active is clearly visible, occurring approximately when the missile reaches the commanded cruise altitude.



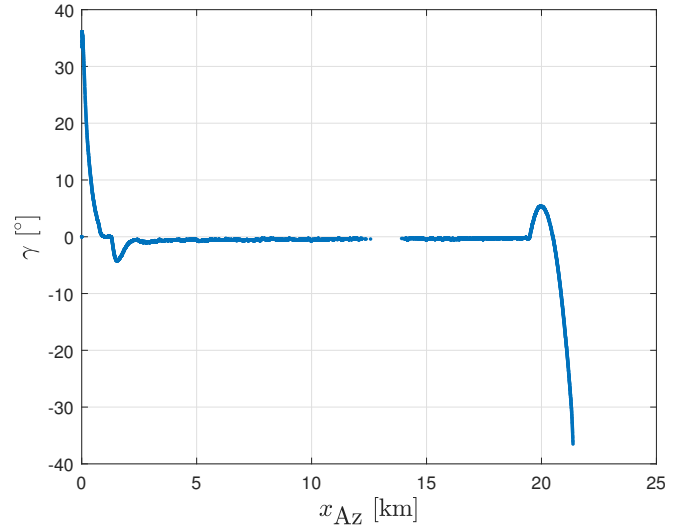
(a) Trajectory.



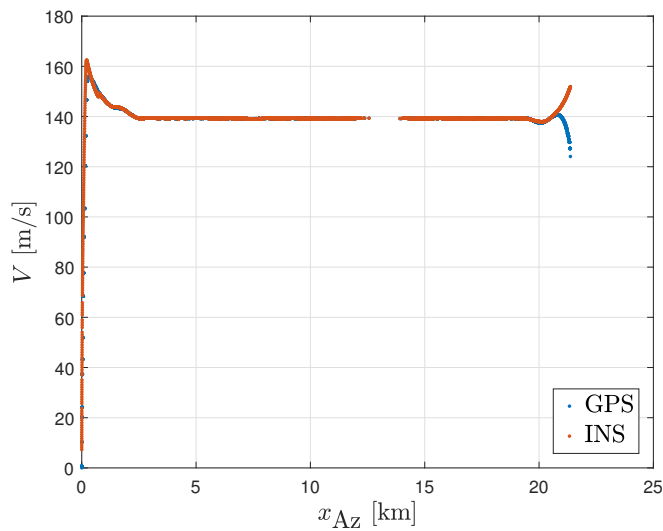
(b) Angle of attack.



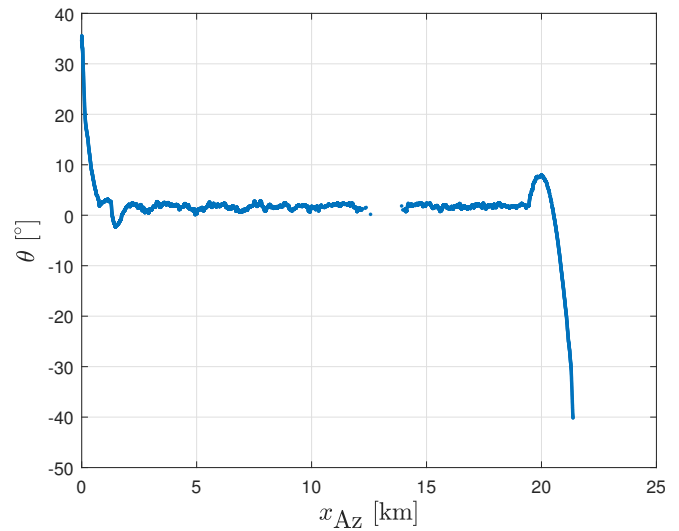
(c) Achieved specific force.



(d) Flight-path angle.



(e) Velocity.



(f) Pitch angle.

Figure 3.75: Flight test results (whole trajectory), $\gamma_1 = -35^\circ$.

The corresponding results for the terminal phase alone are presented in Fig. 3.76, where the quantities most relevant to this dissertation can be observed in greater detail. First, the trajectory shown in Fig. 3.76a is fully consistent with the numerical results presented throughout this dissertation. For the commanded terminal flight-path angle of -35° , the missile reaches a maximum altitude of approximately 270 m. This altitude was intentionally limited by proving-ground safety regulations, and therefore larger terminal climb maneuvers, which would have enabled even steeper impact angles, were not considered during this phase of development.

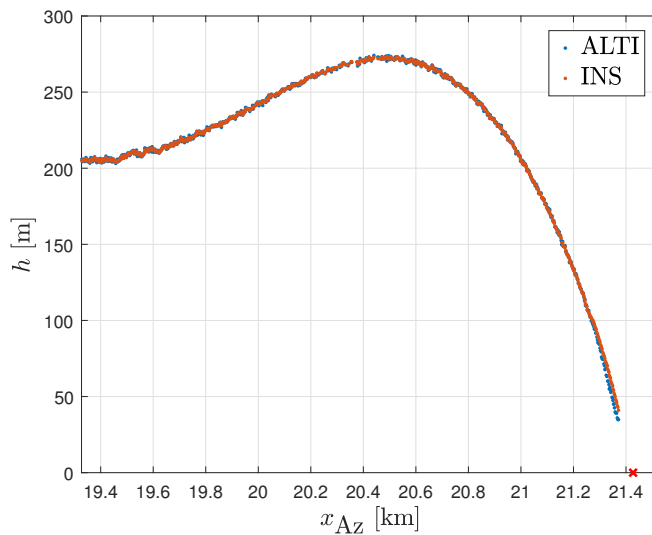
Unfortunately, the loss of the radio link during the final stage of flight prevented recording of the last approximately 50 m of the trajectory. Nevertheless, several important conclusions can still be drawn from the available measurements. Figures 3.76a and 3.76d clearly indicate that the mission objectives were successfully achieved. Extrapolation of the recorded trajectory leads directly to the target location, while the final recorded estimated flight-path angle was -36.5° , corresponding to a pitch angle of approximately -40° , shown in Fig. 3.76f. The measured specific-force profile, obtained from the onboard accelerometers, remained well within the missile maneuverability limits.

The final change in the slope of the specific-force profile, already discussed in Section 3.5, is more pronounced during the flight test because of real-world navigation effects. Specifically, vertical INS aiding using altimeter measurements is intentionally disabled once the missile descends below 100 m above ground level. This behavior is clearly visible in Fig. 3.77, which enlarges the final portion of Fig. 3.75a. Furthermore, because the missile is approaching the target at a steep negative pitch angle, GNSS aiding is also disabled owing to the reduced reliability of satellite-based corrections in the vertical channel. This effect is visible in Fig. 3.76e, where the speed estimated from GNSS begins to deviate from the INS-derived estimate.

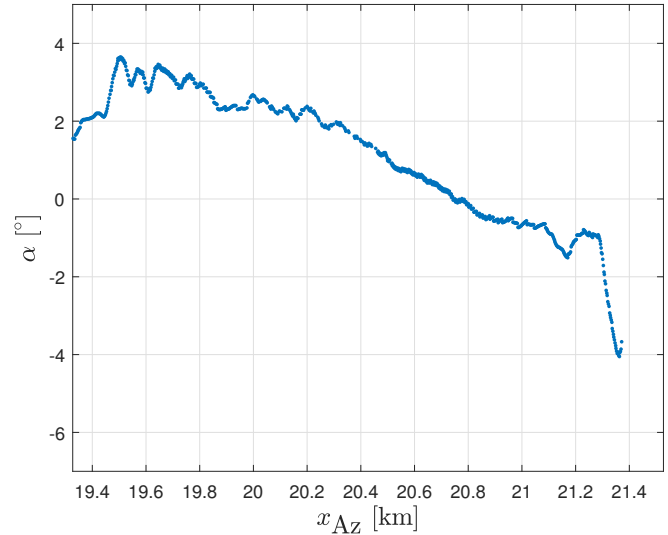
As a consequence, once altimeter aiding is disabled, the missile begins to overestimate its altitude. The guidance algorithm therefore commands a positive normal acceleration to ensure target interception. This additional pitching-down maneuver temporarily produces a flight-path angle slightly steeper than the desired value, after which the flight-path angle would naturally increase again as the missile converges toward the desired terminal conditions. This desirable property of the proposed optimal guidance method—that guaranteeing target interception takes precedence over exact impact-angle tracking—has already been emphasized in previous sections.

Since no additional flight data are available beyond this point, the final impact angle cannot be determined directly. Nevertheless, behavior similar to that shown in Fig. 3.72c, where the commanded specific force changes sign during the final moments of flight, is expected. Given the very short remaining distance to the target, only negligible changes could occur, leading to the conclusion that the desired impact angle was achieved with only a small error.

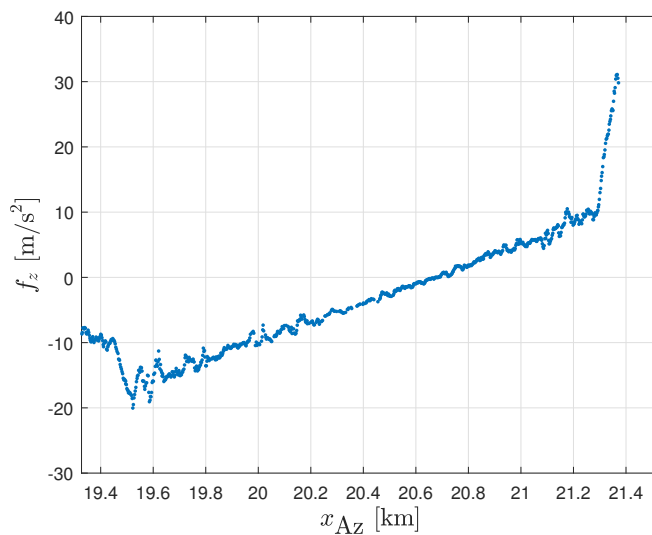
The previously described behavior is further illustrated in Fig. 3.78, where the regions corresponding to the different trajectory convexity cases, introduced in Section 2.2.4, are highlighted using the guidance algorithm status recorded onboard the missile. As expected, terminal guidance begins in case B, since an altitude increase is initially required to achieve the desired impact angle. After the trajectory inflection point, the active case switches to case A, indicating a continuous decrease of the flight-path angle. Under nominal conditions, this case would remain active until impact. However, after altimeter aiding is disabled, the active convexity case changes to case C, indicating that the guidance algorithm temporarily requires an additional decrease in the flight-path angle before subsequently increasing it again to recover the desired terminal condition.



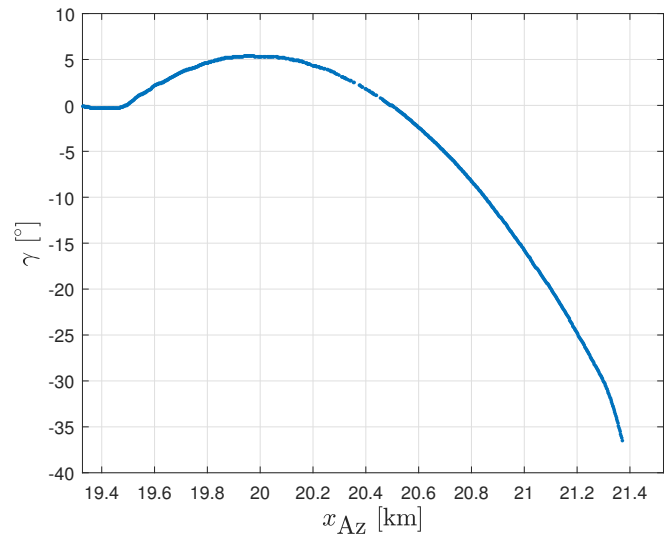
(a) Trajectory.



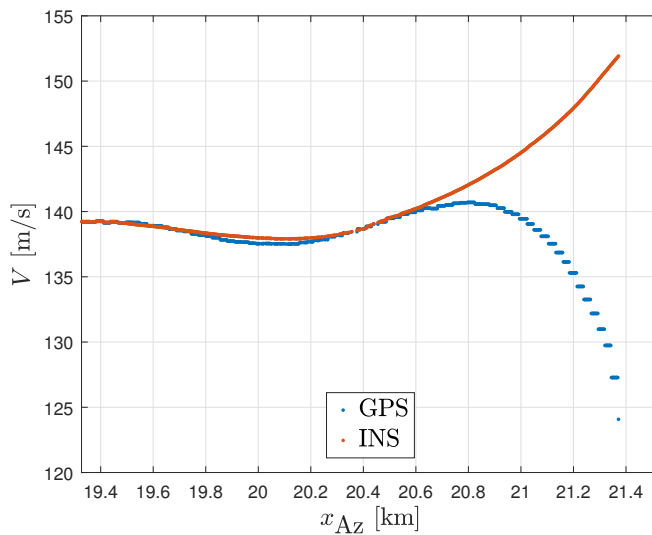
(b) Angle of attack.



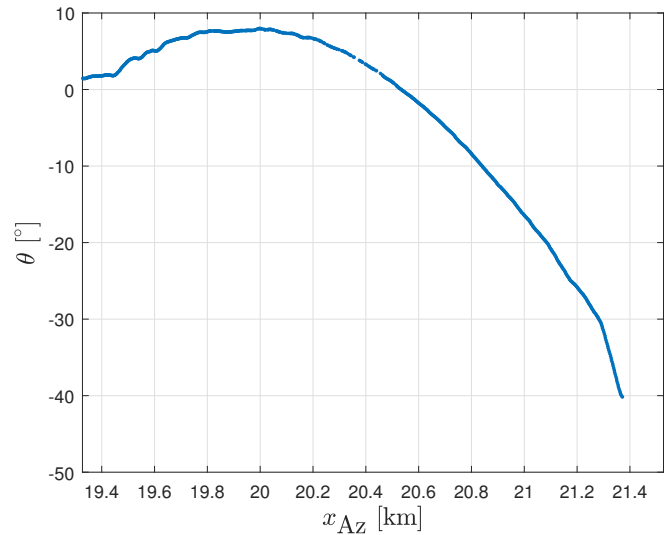
(c) Achieved specific force.



(d) Flight-path angle.



(e) Velocity.



(f) Pitch angle.

Figure 3.76: Flight test results (terminal phase), $\gamma_1 = -35^\circ$.

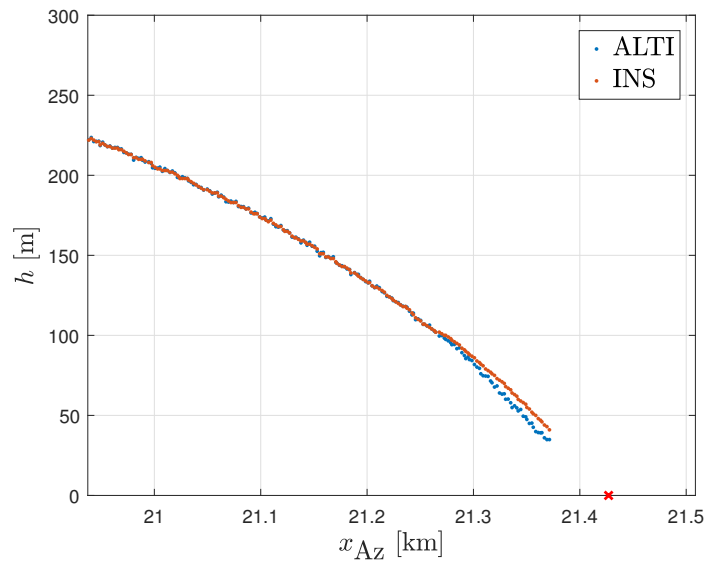


Figure 3.77: Difference between the estimated altitude profiles during the final phase of flight.

Successful mission completion was additionally confirmed by observers at the proving ground, who reported that the missile impacted within a 5 m radius of the target. Furthermore, the missile was equipped with a fixed forward-looking nose camera, shown in Fig. 3.79, whose purpose was to provide a real-time video stream and to support post-flight analysis. Several frames extracted from the final segment of the recorded video are presented in Fig. 3.80, providing additional confirmation of the successful target interception.

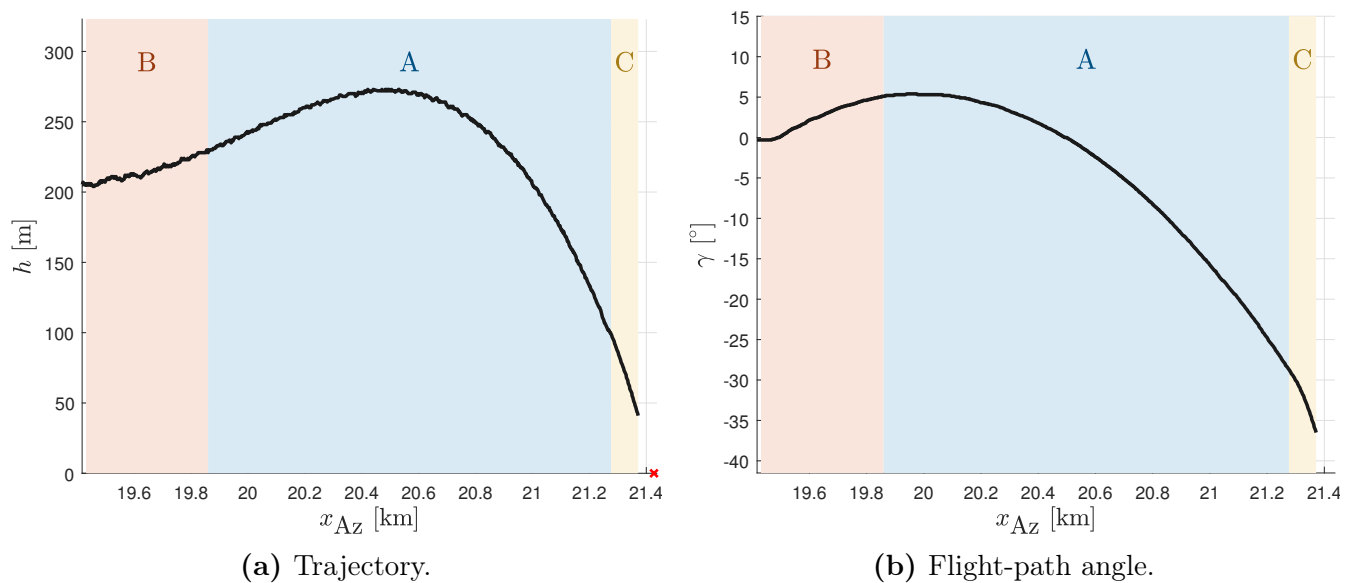


Figure 3.78: Flight test results, highlighting the trajectory-convexity cases.



Figure 3.79: Fixed forward-looking nose camera used for real-time flight video transmission.



(a)



(b)



(c)



(d)

Figure 3.80: Selected video frames captured during the final phase of flight. The number in the upper-left corner indicates the remaining downrange distance to the target.

Chapter 4

Conclusion

In this dissertation, an optimal guidance method has been proposed for flight vehicles required to satisfy a prescribed flight-path angle at the target point. Although the method was primarily developed and analyzed for tactical missile applications, it is formulated in a general manner and is applicable to any flight vehicle regardless of its type, mission, or control mechanism.

A comprehensive review of the existing literature revealed several important observations. The largest class of guidance methods capable of satisfying both terminal position and flight-path angle constraints originates from the missile guidance community. Besides terminal accuracy, many practical applications require the optimization of an additional performance index, such as control effort, flight time, terminal velocity, or quantities introduced to softly constrain variables such as the control input or miss distance. Optimal control theory has become the predominant mathematical framework for solving such problems.

Most existing studies focus on the terminal engagement phase against stationary or moving targets, although the underlying formulations are often applicable to other phases of flight as well. However, the overwhelming majority of available guidance laws rely on linearization of the engagement geometry about the collision triangle, where small-angle approximations are valid. Such assumptions are acceptable only when a preceding guidance phase ensures that the engagement reaches the vicinity of the linearization point before terminal guidance begins. In addition to introducing approximation errors, these methods typically require information beyond the standard navigation states, most commonly the remaining time of flight (time-to-go) and the line-of-sight rate. Although efficient approximations for the time-to-go exist, their accuracy deteriorates as the actual trajectory deviates from the assumed linearized geometry, which is precisely the case when large terminal flight-path angles are required. Several nonlinear guidance methods have also been reported in the literature; however, they generally involve a computational complexity that prevents practical implementation on resource-constrained embedded systems.

These observations naturally motivated the primary objective of this dissertation: to develop a guidance method based on the nonlinear engagement model, free of linearization assumptions, while remaining suitable for real-time implementation on operational flight hardware.

Consistent with the intended application to cruise and ballistic missiles engaging stationary targets, the selected performance index was the control effort. Since the targets are stationary, minimizing the flight time provides no practical benefit and would unnecessarily degrade the truly minimum-effort solution. Furthermore, minimizing the control effort introduces a soft limitation on the control input, thereby avoiding the additional complexity associated with explicitly constrained optimal control problems. Two standard assumptions were adopted during the derivation of the guidance law: constant vehicle speed and planar motion. Both assumptions were subsequently analyzed and shown to be justified for the considered class of systems.

Chapter 2 addressed the derivation of the optimal guidance law and reduced the original optimal control problem to a form suitable for efficient numerical solution. Although the appearance of elliptic integrals makes a completely closed-form analytical solution impossible, the developed formulation minimizes the required numerical effort. In particular, the entire optimization problem is reduced to the solution of a single nonlinear equation in a single unknown.

The resulting guidance law is expressed in closed-loop feedback form solely in terms of the vehicle position and velocity, which represent the standard navigation data available on virtually all flight vehicles. The solution of the nonlinear equation is required only to determine the guidance law parameters. This property later proved particularly advantageous for real-time implementation because the navigation variables and the optimal guidance parameters do not need to be updated at the same frequency, allowing the computationally demanding parameter update to be performed less frequently without degrading guidance performance.

In addition to the general terminal flight-path angle problem, a special case corresponding to the climb phase of a cruise missile toward its cruising altitude was investigated. For this practically important scenario, a simplified guidance law was derived in the form of a single analytical expression, eliminating the need for iterative root-finding. Although the solution still requires the numerical evaluation of a single integral, its computational cost is significantly lower than that of an iterative procedure involving elliptic integrals.

The proposed guidance law was finally compared with the most representative guidance methods available in the literature. The obtained results demonstrated that the proposed approach generates unique trajectories that are optimal with respect to the selected performance index while simultaneously satisfying the prescribed terminal conditions.

Chapter 3 was devoted to the practical implementation of the proposed guidance algorithm. Efficient numerical procedures were developed for all computationally demanding components, accompanied by a detailed error analysis. It was shown that the allowable error of the guidance command uniquely determines the permissible errors of all numerical procedures involved. Consequently, the computational burden can be systematically adjusted to satisfy the requirements of a particular flight control system during the offline design phase.

Despite establishing several more efficient numerical procedures, the subsequent implementation analysis deliberately employed the computationally least efficient alternative, namely the bisection method for solving the nonlinear equation. This choice ensured a fixed and predictable computational burden throughout the entire flight, thereby enabling a conservative validation of the algorithm under worst-case conditions. More efficient root-finding algorithms were also analyzed,

together with specialized numerical integration procedures for the elliptic integrals appearing in the nonlinear equation.

Before addressing the final implementation aspects, the influence of two important practical effects was investigated: variable vehicle speed and delayed updating of the guidance parameters. The analysis demonstrated that the closed-loop structure of the proposed guidance law largely compensates for deviations from the constant-speed assumption adopted during the derivation. Consequently, the guidance command should be recomputed at every guidance cycle using the latest navigation measurements, including the current vehicle speed. Furthermore, it was shown that delaying the update of the optimal guidance parameters by several guidance cycles has a negligible influence on the resulting trajectory. This finding is particularly important from an implementation perspective, as it provides sufficient computational margin for solving the nonlinear equation while maintaining real-time operation.

The practical implementation of the proposed guidance method on resource-constrained embedded hardware was investigated in Section 3.3. Efficient algorithms were developed for evaluating all intermediate functions required by the guidance law, with particular emphasis on the nonlinear function whose root determines the guidance-law parameters. Execution-time measurements were performed throughout the implementation process, allowing the final algorithmic structure to be selected based on computational efficiency rather than theoretical considerations alone.

The resulting implementation was subsequently evaluated in a software-in-the-loop environment, demonstrating that the computational burden is well within the available real-time constraints. Two implementation strategies were considered, differing in the number of iterations of the nonlinear equation solver performed during each execution cycle of the flight computer. In both cases, substantial timing margins were obtained, leaving sufficient computational resources for additional onboard functionality. In configuration where two nonlinear solver iterations are executed during each guidance cycle, the total execution time remained below one half of the flight computer cycle (2 ms). Consequently, the proposed algorithm can be directly extended to three-dimensional guidance by executing an identical instance of the algorithm for the horizontal guidance channel. Similar timing margins were obtained for both cruise and ballistic missile applications, confirming that the proposed guidance method is suitable for real-time implementation on operational embedded systems.

After establishing the final implementation, the algorithm was validated through comprehensive six-degree-of-freedom simulations of the considered flight vehicles.

For the cruise missile climb phase, the proposed guidance law achieved a smooth transition to level flight at the prescribed cruise altitude while producing a smooth specific-force profile throughout the maneuver. The simulations further showed that, for every initial flight-path angle, there exists an optimal cruise altitude that minimizes the required control effort, indicating that cruise altitude selection remains an important task of the fire-control system even when optimal guidance is employed. Consequently, the proposed guidance method is particularly advantageous when higher cruise altitudes are desired, although it remains applicable over the entire practical operating range. It was additionally demonstrated that switching to trajectory-tracking guidance shortly before reaching the desired altitude improves robustness against navigation errors and model uncertainties. Finally, two practical safety mechanisms were proposed to ensure reliable operation in the presence of inaccurate flight-path-angle measurements: one based on replacing the flight-path

angle with the measured pitch angle whenever appropriate, and another based on switching to trajectory-tracking guidance whenever the climb duration exceeds a prescribed threshold.

For the terminal phase of cruise missile flight, the proposed guidance method successfully achieved prescribed impact angles provided that the mission parameters were selected appropriately. In particular, the cruise altitude and the distance from the target at which terminal guidance is initiated were identified as the dominant mission parameters governing the feasibility and efficiency of the terminal maneuver. Under realistic mission planning, these parameters would naturally be selected by the fire-control system, allowing Algorithm 2 to generate smooth and control-effort-optimal trajectories.

The analysis further showed that unsuccessful interception at the desired impact angle occurs only when terminal guidance is initiated from extremely low altitudes or insufficient distances, situations that would compromise mission success regardless of the selected guidance law because of the missile's finite maneuverability. Increasing the cruise altitude proved beneficial, as it reduced the altitude gain required during the terminal maneuver, thereby reducing velocity loss while simultaneously exploiting the aerodynamic advantages of high-altitude cruise. Higher cruise altitudes also enabled steeper impact angles to be achieved with lower control effort. Conversely, increasing the terminal engagement distance reduced maneuver aggressiveness at the expense of larger altitude excursions and corresponding velocity loss. Determining the optimal combination of these mission parameters therefore naturally becomes a higher-level optimization problem belonging to the fire-control system. Several examples of such optimization were presented for different performance criteria.

The influence of navigation errors was also investigated. Position errors were shown to affect the terminal impact point directly, as expected for any guidance law, whereas errors in the flight-path-angle measurement primarily affected the achieved impact angle while producing only negligible miss-distance errors provided that the position estimate remained accurate. This observation is particularly important from a practical standpoint because position estimates can often be significantly improved through integration with satellite navigation or terrain-referenced measurements, whereas accurate estimation of the flight-path angle in the vertical plane is considerably more difficult.

The proposed guidance method was also evaluated for ballistic missile applications under severe total-impulse deviations. The results demonstrated that the method is capable of achieving both target interception and prescribed impact angles even under extreme propulsion uncertainties that would otherwise produce substantial miss distances for long-range artillery rockets. Owing to the characteristic ballistic trajectory, the guidance algorithm was initiated immediately after launch, with the desired terminal conditions specified directly at the target. Appropriate selection of the desired impact angle was additionally shown to produce smoother maneuvering profiles throughout the flight.

An alternative implementation strategy for ballistic missiles was also proposed. Instead of continuously guiding toward the final target, the guidance law could successively track intermediate waypoints with flight-path-angle references extracted from the nominal trajectory. Such an approach would require only waypoint coordinates and nominal flight-path-angle values to be stored prior to launch, resulting in substantially lower memory requirements than methods based on

impact-point prediction. Investigation of this guidance strategy represents a promising direction for future research.

Finally, the proposed guidance method was evaluated during the preparation and execution of flight testing of the A50 cruise missile. The verification process first included extensive hardware-in-the-loop experiments, representing the final validation stage prior to flight testing of an operational missile system. These experiments combined the advantages of software-in-the-loop implementation and high-fidelity six-degree-of-freedom simulation, providing the most comprehensive laboratory validation of the proposed guidance method. The presented results covered representative operational disturbances and system imperfections, demonstrating successful achievement of all guidance objectives. The experiments also confirmed the theoretical predictions regarding the individual trajectory convexity cases discussed in Chapter 2. In all tested scenarios, the terminal miss distance remained determined almost exclusively by inertial navigation errors, while the prescribed impact angles were achieved with excellent accuracy. Finally, the guidance method was successfully implemented on the onboard computer of the new A50 cruise missile and verified during its first long-range flight test. Despite real-world imperfections, including temporary telemetry interruptions and navigation aiding limitations during the final stage of flight, the recorded flight data confirmed the predicted trajectory profile, successful target interception, and accurate achievement of the prescribed terminal flight-path angle.

Future research should primarily focus on extending the proposed methodology toward practical operational applications. For cruise missile guidance, the theoretical framework should be generalized to engagements against moving targets and missiles equipped with terminal seekers. Such an extension is particularly relevant for other members of the land attack missile systems intended for engaging mobile ground targets or achieving pinpoint accuracy against lightly armored vehicles. The resulting guidance algorithm should preserve the computational efficiency achieved in this dissertation while providing clear operational advantages over existing guidance methods.

For ballistic missile applications, future work should investigate the waypoint-based implementation, where prescribed flight-path angles are tracked at selected control points along the nominal trajectory. Such an approach has the potential to further reduce maneuvering requirements while providing a natural foundation for alternative control concepts, such as flight-path steering.

Finally, for both application domains, future research should incorporate the vehicle dynamics directly into the optimization model. Such an extension would naturally lead toward more advanced optimization techniques, including methods based on convex optimization, thereby providing a foundation for future generations of guidance systems operating under increasingly demanding mission requirements.

Bibliography

- [1] K. E. Tsiolkovsky, “Исследование Мировых Пространств Реактивными Приборами (The Exploration of Cosmic Space by Means of Reaction Devices),” *Научное Обозрение (Scientific Review)*, vol. 5, pp. 1–75, 1903.
- [2] U.S. Army, White Sands Missile Range, and Applied Physics Laboratory, “First Photograph of Earth from Space.” Public domain photograph, 1946. Accessed: 4 June 2026.
- [3] No. 5 Army Film & Photographic Unit and Morris (Sgt), “The V2 Rocket.” Imperial War Museums photograph BU 11149, 1945. A German V-2 rocket at the moment of launch during Allied tests in Germany, 10 October 1945. Accessed: 4 June 2026.
- [4] National Aeronautics and Space Administration (NASA), “Soyuz MS-23 Docking to the International Space Station.” NASA TV video broadcast, 6 April 2023. Frame captured from video showing the Soyuz MS-23 spacecraft docking to the Prichal module of the International Space Station.
- [5] SpaceX, “Starship Mars Atmospheric Entry.” Illustration of a SpaceX Starship entering the Martian atmosphere and decelerating aerodynamically.
- [6] National Archives and Records Administration, “Extended Range Anti-Armor Munition (ERAM) Warhead Test.” Combined Military Service Digital Photographic Files, 1981. Photograph showing the formation of a self-forging fragment during an ERAM warhead test at Otis Air National Guard Base, Massachusetts, USA.
- [7] Kenneth Moll and Reuters, “Tomahawk Cruise Missile Launch from USS Cape St. George.” Reuters news photograph, 2003. Photograph of a Tomahawk cruise missile launch from USS Cape St. George on 23 March 2003.
- [8] A. A. Markov, “Несколько Примеров Решения Особого Рода Задач о Наибольших и Наименьших Величинах (Some Examples of the Solution of a Special Kind of Problem on Greatest and Least Quantities),” *Сообщения Харьковского математического общества (Soobscenija Charkovskogo Matematiceskogo Obscestva)*, vol. 2, pp. 250–276, 1889.
- [9] L. E. Dubins, “On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents,” *American Journal of Mathematics*, vol. 79, pp. 497–516, 1957.
- [10] R. Isaacs, “Games of Pursuit,” *RAND Report, Santa Monica, CA: RAND Corporation*, 1951.

-
- [11] J. Shinar and D. Steinberg, “Analysis of Optimal Evasive Maneuvers Based on a Linearized Two-Dimensional Kinematic Model,” *Journal of Aircraft*, vol. 14, pp. 795–802, 1977.
 - [12] E. Zermelo, “Über das Navigationsproblem bei ruhender oder veränderlicher Windverteilung,” *Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM) - Journal of Applied Mathematics and Mechanics*, vol. 11, Issue 2, pp. 114–124, 1931.
 - [13] E. Bakolas and P. Tsiotras, “Minimum-Time Paths for a Small Aircraft in the Presence of Regionally-Varying Strong Winds,” *AIAA Infotech at Aerospace 2010, Session: Intelligent Flight Planning and Guidance II*, Atlanta, Georgia, CA, 2010.
 - [14] B. Or, J. Z. Ben-Asher, and I. Yaesh, “Pursuit Evasion Games with Imperfect Information Revisited,” in *Proceedings of the 58th Israel Annual Conference on Aerospace Sciences*, Tel-Aviv and Haifa, Israel, 2018.
 - [15] B. Or, *Integrated Guidance / Estimation in Linear Quadratic Differential Games*. MSc Thesis, Technion—Israel Institute of Technology, Israel, 2018.
 - [16] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. Courier Corporation, 1999.
 - [17] I. M. Ross, R. Proulx, and M. Karpenko, “Unscented Optimal Control for Orbital and Proximity Operations in an Uncertain Environment: A New Zermelo Problem,” *AIAA/AAS Astrodynamics Specialist Conference, Session: Asteroid Missions III*, San Diego, CA, 2014.
 - [18] E. Bakolas and P. Tsiotras, “Optimal Synthesis of the Zermelo–Markov–Dubins Problem in a Constant Drift Field,” *Journal of Optimization Theory and Applications*, vol. 156, pp. 469–492, 2012.
 - [19] R. P. Anderson, E. Bakolas, D. Milutinović, and P. Tsiotras, “Optimal Feedback Guidance of a Small Aerial Vehicle in a Stochastic Wind,” *Journal of Guidance, Control, and Dynamics*, vol. 36, 2013.
 - [20] A. Wolek and C. Woolsey, “Feasible Dubins Paths in Presence of Unknown, Unsteady Velocity Disturbances,” *Journal of Guidance, Control, and Dynamics*, vol. 38, 2015.
 - [21] I. Shapira and J. Z. Ben-Asher, “The Singular Perturbation Technique Model of Air to Air Interception in the Horizontal Plane,” Reston, VA, 1996.
 - [22] J. Z. Ben-Asher, *Optimal Control Theory with Aerospace Applications*. American Institute of Aeronautics and Astronautics, January, 2010.
 - [23] P. Zarchan, *Tactical and Strategic Missile Guidance, Sixth Edition*. American Institute of Aeronautics and Astronautics, March, 2012.
 - [24] Thewellman, “Lark airframe at Point Mugu Missile Park, California.” Wikimedia Commons, 2015. Accessed: 2026-07-02.
 - [25] U.S. Navy, “A U.S. Navy XSAM-N-2 Lark surface-to-air missile launched at the Naval Air Materiel Test Center, Point Mugu, California (circa 1949).” Wikimedia Commons / Naval History & Heritage Command (Accession: USN 400916), 1949. Published in: Evans, M. L. & Grossnick, R. A. (2015) "United States Naval Aviation 1910-2010", p. 242. Accessed: 2026-07-02.

-
- [26] D. Ćuk, *Lecture Notes on the Subject of Missile Guidance and Control*. University of Belgrade, Faculty of Mechanical Engineering, 2008.
 - [27] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*. Hemisphere, New York, 1975, pp. 153–155.
 - [28] J. Z. Ben-Asher and I. Yaesh, *Advances in Missile Guidance Theory*. American Institute of Aeronautics and Astronautics, 1998.
 - [29] G. Cherry, “A General, Explicit, Optimizing Guidance Law for Rocket-Propelled Spaceflight,” *Astrodynamics Guidance and Control Conference*, 1964.
 - [30] E. J. Ohlmeyer and C. A. Phillips, “Generalized Vector Explicit Guidance,” *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 2, pp. 261–268, 2006.
 - [31] C. F. Lin, *Modern Navigation Guidance and Control Processing*. Prentice–Hall, Upper Saddle River, NJ, 1991, Chap. 8.6.
 - [32] D. Serakos and C.-F. Lin, “Linearized Kappa Guidance,” *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 5, pp. 975–980, 1995.
 - [33] A. E. Bryson, “Linear Feedback Solutions for Minimum Effort Interception, Rendezvous, and Soft Landing,” *AIAA Journal*, vol. 3, no. 8, pp. 1542–1544, 1965.
 - [34] F. W. Nesline. and P. Zarchan, “A New Look at Classical vs Modern Homing Missile Guidance,” *Journal of Guidance and Control*, vol. 4, no. 1, pp. 78–85, 1981.
 - [35] M.-J. Tahk, C.-K. Ryoo, and H. Cho, “Recursive Time-to-Go Estimation for Homing Guidance Missiles,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 13–24, 2002.
 - [36] N. Dhananjay and D. Ghose, “Accurate Time-to-Go Estimation for Proportional Navigation Guidance,” *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1378–1383, 2014.
 - [37] J. Z. Ben-Asher and I. Yaesh, “Optimal Guidance with Reduced Sensitivity to Time-to-Go Estimation Errors,” *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 158–163, 1997.
 - [38] R. G. Cottrell, “Optimal Intercept Guidance for Short-Range Tactical Missiles,” *AIAA Journal*, vol. 9, no. 7, pp. 1414–1415, 1971.
 - [39] M. Kim and K. V. Grider, “Terminal Guidance for Impact Attitude Angle Constrained Flight Trajectories,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-9, no. 6, pp. 852–859, 1973.
 - [40] R. J. York and H. L. Pastrick, “Optimal Terminal Guidance with Constraints at Final Time,” *Journal of Spacecraft and Rockets*, vol. 14, no. 6, pp. 381–383, 1977.
 - [41] T. Song, S. Shin, and H. Cho, “Impact Angle Control for Planar Engagements,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 4, pp. 1439–1444, 1999.

-
- [42] C.-K. Ryoo, H. Cho, and M.-J. Tahk, "Optimal Guidance Laws with Terminal Impact Angle Constraint," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, pp. 724–732, 2005.
 - [43] C.-K. Ryoo, H. Cho, and M.-J. Tahk, "Time-to-Go Weighted Optimal Guidance with Impact Angle Constraints," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 483–492, 2006.
 - [44] V. Shaferman and T. Shima, "Linear Quadratic Guidance Laws for Imposing a Terminal Intercept Angle," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1400–1412, 2008.
 - [45] I. Rusnak, H. Weiss, R. Eliav, and T. Shima, "Missile Guidance with Constrained Intercept Body Angle," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 2, pp. 1445–1453, 2014.
 - [46] V. Glizer, "Optimal Planar Interception with Fixed End Conditions: Closed-form Solution," *Journal of Optimization Theory and Applications*, vol. 88, no. 3, pp. 503–539, 1996.
 - [47] M. Idan, O. M. Golan, and M. Guelman, "Optimal Planar Interception with Terminal Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 6, pp. 1273–1279, 1995.
 - [48] P. Lu and F. Chavez, "Nonlinear Optimal Guidance," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 0–11, 2006.
 - [49] P. Lu, "Intercept of Nonmoving Targets at Arbitrary Time-Varying Velocity," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 176–178, 1998.
 - [50] Y. Ulybyshev, "Terminal Guidance Law Based on Proportional Navigation," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, pp. 821–824, 2005.
 - [51] P. Lu, D. B. Doman, and J. D. Schierman, "Adaptive Terminal Guidance for Hypervelocity Impact in Specified Direction," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 2, pp. 269–278, 2006.
 - [52] A. Ratnoo and D. Ghose, "State-Dependent Riccati-Equation Based Guidance Law for Impact-Angle Constrained Trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 320–326, 2009.
 - [53] P. Lu, "Introducing Computational Guidance and Control," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 193–193, 2017.
 - [54] P. Lu, C. W. Brunner, S. J. Stachowiak, G. F. Mendeck, M. A. Tigges, and C. J. Cerimele, "Verification of a Fully Numerical Entry Guidance Algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 230–247, 2017.
 - [55] R. B. Padhi and M. Kothari, "Model Predictive Static Programming: A Computationally Efficient Technique For Suboptimal Control Design," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 2, pp. 399–411, 2009.
 - [56] H. B. Oza and R. Padhi, "Impact-Angle-Constrained Suboptimal Model Predictive Static Programming Guidance of Air-to-Ground Missiles," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, pp. 153–164, 2012.

-
- [57] A. Maity, H. B. Oza, and R. Padhi, "Generalized Model Predictive Static Programming and Angle-Constrained Guidance of Air-to-Ground Missiles," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 6, pp. 1897–1913, 2014.
 - [58] X. Liu, Z. Shen, and P. Lu, "Closed-Loop Optimization of Guidance Gain for Constrained Impact," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 453–460, 2017.
 - [59] J. T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, 2012.
 - [60] Z. Chen and T. Shima, "Nonlinear Optimal Guidance for Intercepting a Stationary Target," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 11, pp. 2418–2431, 2019.
 - [61] I.-S. Jeon, H. Cho, and J.-I. Lee, "Exact Guidance Solution for Maneuvering Target on Relative Virtual Frame Formulation," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 7, pp. 1330–1340, 2015.
 - [62] H. Li, J. Wang, S. He, and C.-H. Lee, "Nonlinear Optimal Impact-Angle-Constrained Guidance with Large Initial Heading Error," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 9, pp. 1663–1676, 2021.
 - [63] Y. Zhu, C. Zhou, S. Chen, X. Song, K. Li, and C. Li, "A Parameterized Solution to Optimal Guidance Law Against Stationary Target With Impact Angle Constraint," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 61, no. 2, pp. 3993–4003, 2025.
 - [64] B. S. Kim, J. G. Lee, and H. S. Han, "Biased PNG Law for Impact with Angular Constraint," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 277–288, 1998.
 - [65] A. Ratnoo and D. Ghose, "Impact Angle Constrained Interception of Stationary Targets," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1817–1822, 2008.
 - [66] A. Ratnoo and D. Ghose, "Impact Angle Constrained Guidance Against Nonstationary Nonmaneuvering Targets," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 1, pp. 269–275, 2010.
 - [67] K. S. Erer and O. Merttopçuoglu, "Indirect Impact-Angle-Control Against Stationary Targets Using Biased Pure Proportional Navigation," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 700–704, 2012.
 - [68] C.-H. Lee, T.-H. Kim, and M.-J. Tahk, "Interception Angle Control Guidance Using Proportional Navigation with Error Feedback," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 5, pp. 1556–1561, 2013.
 - [69] T.-H. Kim, B.-G. Park, and M.-J. Tahk, "Bias-Shaping Method for Biased Proportional Navigation with Terminal-Angle Constraint," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 6, pp. 1810–1816, 2013.
 - [70] B.-G. Park, T.-H. Kim, and M.-J. Tahk, "Biased PNG With Terminal-Angle Constraint for Intercepting Nonmaneuvering Targets Under Physical Constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 3, pp. 1562–1572, 2017.

-
- [71] R. Tekin and K. S. Erer, "Switched-Gain Guidance for Impact Angle Control Under Physical Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 2, pp. 205–216, 2015.
 - [72] A. Ratnoo, "Analysis of Two-Stage Proportional Navigation with Heading Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 1, pp. 156–164, 2016.
 - [73] W.-j. Gu, R.-c. Zhang, and J.-y. Yu, "A Three-Dimensional Missile Guidance Law with Angle Constraint Based on Sliding Mode Control," in *2007 IEEE International Conference on Control and Automation*, pp. 299–302, 2007.
 - [74] T. Shima, "Intercept-Angle Guidance," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 2, pp. 484–492, 2011.
 - [75] S. R. Kumar, S. Rao, and D. Ghose, "Sliding-Mode Guidance and Control for All-Aspect Interceptors with Terminal Angle Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 4, pp. 1230–1246, 2012.
 - [76] S. R. Kumar, S. Rao, and D. Ghose, "Nonsingular Terminal Sliding Mode Guidance with Impact Angle Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1114–1130, 2014.
 - [77] S. He, T. Song, and D. Lin, "Impact Angle Constrained Integrated Guidance and Control for Maneuvering Target Interception," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 10, pp. 2653–2661, 2017.
 - [78] I. Manchester and A. Savkin, "Circular Navigation Guidance Law for Precision Missile/Target Engagements," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 2, pp. 1287–1292 vol.2, 2002.
 - [79] I. R. Manchester and A. V. Savkin, "Circular Navigation Missile Guidance with Incomplete Information and Uncertain Autopilot Model," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 6, pp. 1078–1083, 2004.
 - [80] O. Ariff, R. Zbikowski, A. Tsourdos, and B. A. White, "Differential Geometric Guidance Based on the Involute of the Target's Trajectory," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 5, pp. 990–996, 2005.
 - [81] M. Sun, Q. Xu, S. Du, Z. Chen, and D. Zhang, "Practical Solution to Impact Angle Control in Vertical Plane," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 1022–1027, 2014.
 - [82] F. P. Adler, "Missile Guidance by Three-Dimensional Proportional Navigation," *Journal of Applied Physics*, vol. 27, pp. 500–507, 1956.
 - [83] H. Li, J. Wang, H. Tao, and D. Lin, "Near-Optimal Impact-Vector-Control Guidance," *Journal of Guidance, Control, and Dynamics*, vol. 47, no. 9, pp. 1965–1972, 2024.
 - [84] N. Indig, J. Z. Ben-Asher, and N. Farber, "Near-Optimal Spatial Midcourse Guidance Law with an Angular Constraint," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 214–223, 2014.

-
- [85] J.-H. Kim, S.-S. Park, K.-K. Park, and C.-K. Ryoo, "Quaternion Based Three-Dimensional Impact Angle Control Guidance Law," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2311–2323, 2021.
 - [86] B. Biswas, A. Maity, and S. R. Kumar, "Finite-Time Convergent Three-Dimensional Nonlinear Intercept Angle Guidance," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 1, pp. 146–153, 2020.
 - [87] K. S. Erer and R. Tekin, "Impact Vector Guidance," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 10, pp. 1892–1901, 2021.
 - [88] K. S. Erer, R. Tekin, and M. K. Özgören, "Composite Impact Vector Control Based on Apollo Descent Guidance," *Chinese Journal of Aeronautics*, vol. 35, no. 7, pp. 67–74, 2022.
 - [89] L. Miličić, A. Obradović, I. Todić, and A. Pejčev, "Real-Time Algorithm for Nonlinear Optimal Impact Angle Guidance," *Aerospace*, vol. 13, no. 5, 2026.
 - [90] M. Guelman and J. Shinar, "Optimal Guidance Law in the Plane," *Journal of Guidance, Control, and Dynamics*, vol. 7, no. 4, pp. 471–476, 1984.
 - [91] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2 ed., 1992. Reprinted with corrections 1996.
 - [92] H. B. Keller, *Numerical Solution of Two Point Boundary Value Problems*. Society for Industrial and Applied Mathematics, 1976.
 - [93] R. Pepy and B. Hérissey, "An Indirect Method for Optimal Guidance of a Glider," *IFAC Proceedings Volumes*, vol. 47, pp. 5097–5102, 2014.
 - [94] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, p. 308–313, 1965.
 - [95] A. Cvetković and M. Spalević, *Numeričke metode*. Univerzitet u Beogradu, Mašinski fakultet, 2013.
 - [96] C. Ridders, "A New Algorithm for Computing a Single Root of a Real Continuous Function," *IEEE Transactions on Circuits and Systems*, vol. 26, no. 11, pp. 979–980, 1979.
 - [97] R. Brent, *Algorithms for Minimization without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall, 1973. Chapters 3, 4.
 - [98] P. J. Davis and P. Rabinowitz, "Chapter 2 - Approximate Integration over a Finite Interval," in *Methods of Numerical Integration (Second Edition)* (P. J. Davis and P. Rabinowitz, eds.), pp. 51–198, Academic Press, second edition ed., 1984.
 - [99] I. Todić, *Optimalno vođenje u uslovima velikih poremećaja i ograničenih performansi leta rakete*. PhD thesis, Univerzitet u Beogradu, Mašinski fakultet, Beograd, 2013.
 - [100] I. Todić and V. Kuzmanović, "Code optimization for strapdown inertial navigation system algorithm," in *Space Flight* (G. Dekoulis, ed.), InTech, 2018.

-
- [101] S. Park, J. Deyst, and J. How, “A New Nonlinear Guidance Logic for Trajectory Tracking,” *AIAA Guidance, Navigation, and Control Conference and Exhibit, Session: Tracking and Guidance*, Providence, Rhode Island, 2004.
 - [102] EDePro, “ALAS-XX.” <https://www.edepro.com/products-and-services/missiles/alas/alas-xx>. Accessed: 2026-06-22.
 - [103] EDePro, “5th Generation Cruise Missile,” 2026. Accessed: 2026-06-22.
 - [104] Yugoimport SDPR, “Self-Propelled Anti-Tank Land-Attack Missile Systems,” 2026. Accessed: 2026-06-22.
 - [105] L. Pontryagin, V. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*. (Russian), English translation by K. N. Trirogoff, ed. by L. W. Neustadt, 1962.
 - [106] I. Todić, D. Ćuk, and S. Pajić, *Flight Dynamics of Projectiles*. University of Belgrade, Faculty of Mechanical Engineering, 2021.
 - [107] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*. The Institution of Engineering and Technology, 2nd ed., 2004.
 - [108] D. Ćuk and S. Mandić, “Comparison of Different Lateral Acceleration Autopilots for a Surface-to-Surface Missile,” *Vojnotehnički glasnik (Military Technical Courier)*, vol. 59, no. 3, pp. 7–28, 2011.
 - [109] D. Ćuk and S. Mandić, “Synthesis of a Pitch Autopilot for a Missile with Highly Non-Stationary Dynamic Parameters,” *Naučnotehnički pregled (Scientific-Technical Review)*, vol. 50, no. 3, pp. 19–28, 2000. Original title in Serbian: Sinteza autopilota propinjanja rakete sa izrazito nestacionarnim dinamičkim parametrima.

Appendices

Appendix A

Minimum Principle

The optimal guidance algorithm developed in this dissertation is based on the minimum principle of optimal control theory. Only the results required for the derivation of the guidance laws are summarized here. The presentation follows the formulation given in [22], while the underlying theoretical foundation is Pontryagin's maximum principle [105], often referred to in the aerospace literature as the *minimum principle*.

Consider the nonlinear dynamic system:

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (\text{A.1})$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector and $\mathbf{u}(t) \in \mathbb{R}^m$ is the control vector. It is assumed that the control is piecewise continuous and that the function:

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) = \begin{bmatrix} g_1(\mathbf{x}, \mathbf{u}, t) \\ g_2(\mathbf{x}, \mathbf{u}, t) \\ \vdots \\ g_n(\mathbf{x}, \mathbf{u}, t) \end{bmatrix} \quad (\text{A.2})$$

possesses continuous first and second partial derivatives with respect to all of its arguments.

Let $\mathbf{x}_f = \mathbf{x}(t_f)$ denote the terminal state. In the Mayer formulation of the optimal control problem, the objective is to minimize the terminal cost:

$$P(\mathbf{x}_f) \quad (\text{A.3})$$

subject to the state equation (A.1) and the prescribed initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$. Note that $\mathbf{x}_f$ is free, whereas t_f is prescribed. These assumptions will be relaxed later when terminal constraints and free-final-time problems are considered.

The Lagrange formulation:

$$\int_{t_0}^{t_f} f(\mathbf{x}(t), \mathbf{u}(t), t) \, dt \quad (\text{A.4})$$

can be converted into the Mayer form by introducing an additional state variable:

$$x_{n+1}(t_0) = 0, \quad \dot{x}_{n+1}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t) \quad (\text{A.5})$$

Consider an optimal state-control pair $(\mathbf{x}, \mathbf{u})$ and a small control variation $\delta \mathbf{u}$ producing the corresponding state variation $\delta \mathbf{x}$. Linearization of the state equations about the optimal trajectory yields:

$$\delta \dot{\mathbf{x}} = \mathbf{g}_x^T \delta \mathbf{x} + \mathbf{g}_u^T \delta \mathbf{u} \quad (\text{A.6})$$

where $\mathbf{g}_x$ and $\mathbf{g}_u$ are the Jacobian matrices of $\mathbf{g}$ with respect to $\mathbf{x}$ and $\mathbf{u}$, respectively:

$$\mathbf{g}_x = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_2}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_1} \\ \frac{\partial g_1}{\partial x_2} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_n}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_1}{\partial x_n} & \frac{\partial g_2}{\partial x_n} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix}, \quad \mathbf{g}_u = \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_2}{\partial u_1} & \cdots & \frac{\partial g_n}{\partial u_1} \\ \frac{\partial g_1}{\partial u_2} & \frac{\partial g_2}{\partial u_2} & \cdots & \frac{\partial g_n}{\partial u_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_1}{\partial u_m} & \frac{\partial g_2}{\partial u_m} & \cdots & \frac{\partial g_n}{\partial u_m} \end{bmatrix} \quad (\text{A.7})$$

The adjoint vector $\boldsymbol{\lambda}(t)$ is defined through the differential equation:

$$\dot{\boldsymbol{\lambda}} = -\mathbf{g}_x \boldsymbol{\lambda} \quad (\text{A.8})$$

Combining (A.6) and (A.8) gives:

$$\frac{d}{dt} (\boldsymbol{\lambda}^T \delta \mathbf{x}) = \boldsymbol{\lambda}^T \mathbf{g}_u^T \delta \mathbf{u} \quad (\text{A.9})$$

Since $\delta \mathbf{x}(t_0) = 0$, integration over the interval $[t_0, t_f]$ yields:

$$\boldsymbol{\lambda}^T(t_f) \delta \mathbf{x}_f = \int_{t_0}^{t_f} (\mathbf{g}_u \boldsymbol{\lambda})^T \delta \mathbf{u} dt \quad (\text{A.10})$$

The variation of the Mayer cost function is:

$$\delta P = \frac{\partial P}{\partial x_{1f}} \delta x_{1f} + \frac{\partial P}{\partial x_{2f}} \delta x_{2f} + \cdots + \frac{\partial P}{\partial x_{nf}} \delta x_{nf} = P_{\mathbf{x}_f}^T \delta \mathbf{x}_f \quad (\text{A.11})$$

Choosing the terminal condition for the adjoint variables as:

$$\boldsymbol{\lambda}(t_f) = P_{\mathbf{x}_f} \quad (\text{A.12})$$

and substituting into (A.10) gives:

$$\delta P = \int_{t_0}^{t_f} (\mathbf{g}_u \boldsymbol{\lambda})^T \delta \mathbf{u} dt \quad (\text{A.13})$$

The Hamiltonian function is introduced as:

$$H(\mathbf{x}, \mathbf{u}, t, \boldsymbol{\lambda}) = \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \quad (\text{A.14})$$

It follows directly that:

$$H_{\mathbf{x}}(\mathbf{x}, \mathbf{u}, t, \boldsymbol{\lambda}) = \mathbf{g}_{\mathbf{x}}(\mathbf{x}, \mathbf{u}, t) \boldsymbol{\lambda}(t), \quad H_{\mathbf{u}}(\mathbf{x}, \mathbf{u}, t, \boldsymbol{\lambda}) = \mathbf{g}_{\mathbf{u}}(\mathbf{x}, \mathbf{u}, t) \boldsymbol{\lambda}(t) \quad (\text{A.15})$$

where:

$$H_{\mathbf{x}}(\mathbf{x}, \mathbf{u}, t, \boldsymbol{\lambda}) = \begin{bmatrix} \frac{\partial H}{\partial x_1} \\ \frac{\partial H}{\partial x_2} \\ \vdots \\ \frac{\partial H}{\partial x_n} \end{bmatrix}, \quad H_{\mathbf{u}}(\mathbf{x}, \mathbf{u}, t, \boldsymbol{\lambda}) = \begin{bmatrix} \frac{\partial H}{\partial u_1} \\ \frac{\partial H}{\partial u_2} \\ \vdots \\ \frac{\partial H}{\partial u_m} \end{bmatrix} \quad (\text{A.16})$$

Using this definition, the adjoint equation can be written in the compact form:

$$\dot{\boldsymbol{\lambda}} = -H_{\mathbf{x}} \quad (\text{A.17})$$

while

$$H_{\mathbf{u}} = \mathbf{g}_{\mathbf{u}} \boldsymbol{\lambda} \quad (\text{A.18})$$

The following result provides a necessary condition for optimality.

Theorem A.0.1. *If $\mathbf{x}(t)$ and $\mathbf{u}(t)$ constitute an optimal trajectory and control history, then there exists a nontrivial adjoint vector satisfying:*

$$\dot{\boldsymbol{\lambda}} = -H_{\mathbf{x}}, \quad \boldsymbol{\lambda}(t_f) = P_{\mathbf{x}_f} \quad (\text{A.19})$$

and the stationarity condition:

$$H_{\mathbf{u}} = \mathbf{0} \quad (\text{A.20})$$

must hold almost everywhere on the interval $[t_0, t_f]$.

Proof. Since $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are optimal, the first variation of the cost function must vanish for every admissible variation $\delta \mathbf{u}(t)$. Therefore, Eq. (A.13) can be satisfied only if $\mathbf{g}_{\mathbf{u}} \boldsymbol{\lambda} = \mathbf{0}$ almost everywhere on $[t_0, t_f]$. Using the definition of $H_{\mathbf{u}}$, condition (A.20) follows immediately. $\square$

For constrained control problems, the above stationarity condition is replaced by Pontryagin's minimum principle.

Theorem A.0.2 (Minimum Principle). *Consider the system:*

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (\text{A.21})$$

with admissible controls $\mathbf{u}(t) \in \Omega$, where Ω is a compact set. If $\mathbf{x}(t)$ and $\mathbf{u}(t)$ provide a strong local minimum of the cost function $P(\mathbf{x}_f)$, then:

$$\mathbf{u}(t) = \arg \min_{\mathbf{q} \in \Omega} H(\mathbf{x}(t), \mathbf{q}, t, \boldsymbol{\lambda}(t)) \quad (\text{A.22})$$

where $\boldsymbol{\lambda}(t)$ satisfies Eqs. (A.8) and (A.12).

In many practical guidance problems the terminal state is not completely specified. Instead, the terminal conditions are represented by a set of algebraic constraints:

$$\boldsymbol{\Psi}(\mathbf{x}_f, t_f) = \mathbf{0} \quad (\text{A.23})$$

which define a terminal manifold, where $\boldsymbol{\Psi}(\mathbf{x}_f, t_f) \in \mathbb{R}^l$ and $l \leq n$.

The optimal control problem is then formulated as the minimization of:

$$P(\mathbf{x}_f, t_f) \quad (\text{A.24})$$

subject to the state equations and the terminal constraints above.

To account for the terminal constraints, a vector of Lagrange multipliers:

$$\mathbf{L}^T = [\eta_1 \quad \eta_2 \quad \cdots \quad \eta_l] \quad (\text{A.25})$$

is introduced together with the multiplier λ_0 , whose value is conventionally taken as $\lambda_0 = 1$ for normal optimal control problems. The resulting transversality conditions become:

$$\boldsymbol{\lambda}(t_f) = \lambda_0 P_{\mathbf{x}_f} + \mathbf{L}^T \boldsymbol{\Psi}_{\mathbf{x}_f} \quad (\text{A.26})$$

$$H(t_f) = -(\lambda_0 P_{t_f} + \mathbf{L}^T \boldsymbol{\Psi}_{t_f}) \quad (\text{A.27})$$

These conditions play a central role in the derivation of the optimal guidance law presented throughout this dissertation. In particular, the multipliers associated with the terminal constraints correspond directly to the parameters that appear in the guidance law expression derived in Chapter 2.

Appendix B

Numerical Methods

B.1 Solving the Nonlinear Equation

B.1.1 Bisection Method

This section briefly summarizes the bisection method used to solve the nonlinear equation $\mathcal{N}(\alpha) = 0$. According to Algorithm 2, the root is sought only after its existence has been established by verifying that the function changes sign over the search interval, i.e.,

$$\mathcal{N}(a, s)\mathcal{N}(b, s) < 0, \quad s \in \{A, B, C\} \quad (\text{B.1})$$

Consequently, the initial sign test, which is normally the first step of the bisection method, may be omitted. Let the initial search interval be denoted by $[a, b]$, and let ϵ_α represent the desired accuracy of the solution (rather than the function value at the solution). The procedure is then given by Algorithm 7.

Algorithm 7 Bisection method for solving $\mathcal{N}(\alpha) = 0$.

```
1: while  $|b - a| \geq \epsilon_\alpha$  do
2:    $c \leftarrow (a + b)/2$ 
3:   if  $\mathcal{N}(c) = 0$  then
4:      $\alpha \leftarrow c$ 
5:     return
6:   end if
7:   if  $\mathcal{N}(a)\mathcal{N}(c) < 0$  then
8:      $b \leftarrow c$ 
9:   else
10:     $a \leftarrow c$ 
11:   end if
12: end while
13:  $\alpha \leftarrow (a + b)/2$ 
```

In practice, the product $\mathcal{N}(a)\mathcal{N}(c)$ need not be evaluated explicitly when determining the subinterval containing the root. During the real-time implementation on embedded hardware, the logical expression

$$(\mathcal{N}(a) < 0) \neq (\mathcal{N}(c) < 0)$$

was used instead, since it was found to provide the fastest sign comparison.

B.1.2 Derivatives Required for Root-Finding via Newton's Method

In case Newton's method is to be used for finding the root of the function $\mathcal{N}(\alpha)$, this section provides the expressions required for evaluating its derivative. The derivation assumes that all variables depend solely on α , corresponding to an instantaneous point along the flight trajectory, where the physical parameters $(\Delta x, \Delta y, \gamma)$ are treated as constants. Consequently, total derivatives are used throughout this section rather than the partial derivatives employed in the real-time analysis presented in Section 3.2.

For all three convexity cases, the function is given by Eq. (2.53):

$$\mathcal{N}(\alpha) = \Delta x I_D(\alpha) - \Delta y I_N(\alpha) \quad (\text{B.2})$$

The general form of the derivative of $\mathcal{N}(\alpha)$ with respect to α is given by:

$$\frac{d\mathcal{N}}{d\alpha} = \Delta x \frac{dI_D}{d\alpha} - \Delta y \frac{dI_N}{d\alpha} \quad (\text{B.3})$$

The difference between Cases A, B, and C lies in the expressions for the integrals I_N and I_D , which are given in the general form by Eq. (2.59):

$$I_N = \cos \alpha \mathcal{I}_1 - \sin \alpha \mathcal{I}_2 \quad (\text{B.4a})$$

$$I_D = \cos \alpha \mathcal{I}_2 + \sin \alpha \mathcal{I}_1 \quad (\text{B.4b})$$

Their derivatives can therefore be written in the general form given by:

$$\frac{dI_N}{d\alpha} = -\sin \alpha \mathcal{I}_1 + \cos \alpha \frac{d\mathcal{I}_1}{d\alpha} - \cos \alpha \mathcal{I}_2 - \sin \alpha \frac{d\mathcal{I}_2}{d\alpha} \quad (\text{B.5a})$$

$$\frac{dI_D}{d\alpha} = -\sin \alpha \mathcal{I}_2 + \cos \alpha \frac{d\mathcal{I}_2}{d\alpha} + \cos \alpha \mathcal{I}_1 + \sin \alpha \frac{d\mathcal{I}_1}{d\alpha} \quad (\text{B.5b})$$

According to Eq. (2.60):

$$\mathcal{I}_1 = \begin{cases} I_1(\gamma_0 - \alpha, \gamma_1 - \alpha), & \text{for case A} \\ I_1(\gamma_0 - \alpha, \pi/2) + I_1(\gamma_1 - \alpha, \pi/2), & \text{for case B} \\ I_1(\gamma_0 - \alpha, -\pi/2) + I_1(\gamma_1 - \alpha, -\pi/2), & \text{for case C} \end{cases} \quad (\text{B.6a})$$

$$\mathcal{I}_2 = \begin{cases} I_2(\gamma_0 - \alpha, \gamma_1 - \alpha), & \text{for case A} \\ I_2(\gamma_0 - \alpha, \pi/2) + I_2(\gamma_1 - \alpha, \pi/2), & \text{for case B} \\ I_2(\gamma_0 - \alpha, -\pi/2) + I_2(\gamma_1 - \alpha, -\pi/2), & \text{for case C} \end{cases} \quad (\text{B.6b})$$

determining the derivatives $d\mathcal{I}_1/d\alpha$ and $d\mathcal{I}_2/d\alpha$ requires differentiating the integrals I_1 and I_2 , defined in Equations (2.42) and (2.44). Because the variable α appears in the integration limits, the Leibniz integral rule is applied:

$$\frac{d}{d\alpha} \int_{a(\alpha)}^{b(\alpha)} f(\xi, \alpha) d\xi = f(b(\alpha), \alpha) \frac{db}{d\alpha} - f(a(\alpha), \alpha) \frac{da}{d\alpha} + \int_{a(\alpha)}^{b(\alpha)} \frac{\partial f}{\partial \alpha}(\xi, \alpha) d\xi \quad (\text{B.7})$$

For the integrals defined by Eqs. (2.42) and (2.44), the last term in the above expression vanishes because the integrands do not depend explicitly on α . Furthermore, the integration limits in Eq. (B.6) are such that, whenever α appears in a limit, it always has a negative sign. Consequently, the derivatives $da/d\alpha$ and $db/d\alpha$ are equal either to 0 or to -1 .

For Case A, the derivatives of the integrals I_1 and I_2 are therefore given by:

$$\frac{dI_1(\gamma_0 - \alpha, \gamma_1 - \alpha)}{d\alpha} = \sqrt{\cos(\gamma_0 - \alpha)} - \sqrt{\cos(\gamma_1 - \alpha)} \quad (\text{B.8a})$$

$$\frac{dI_2(\gamma_0 - \alpha, \gamma_1 - \alpha)}{d\alpha} = \frac{\sin(\gamma_0 - \alpha)}{\sqrt{\cos(\gamma_0 - \alpha)}} - \frac{\sin(\gamma_1 - \alpha)}{\sqrt{\cos(\gamma_1 - \alpha)}} \quad (\text{B.8b})$$

For Cases B and C, the expressions are identical because the upper integration limit is fixed at $\pm\pi/2$:

$$\frac{dI_1(\gamma_i - \alpha, \pm\pi/2)}{d\alpha} = \sqrt{\cos(\gamma_i - \alpha)}, \quad i \in \{0, 1\} \quad (\text{B.9a})$$

$$\frac{dI_2(\gamma_i - \alpha, \pm\pi/2)}{d\alpha} = \frac{\sin(\gamma_i - \alpha)}{\sqrt{\cos(\gamma_i - \alpha)}}, \quad i \in \{0, 1\} \quad (\text{B.9b})$$

By comparing Eqs. (B.8a) and (B.9a) with Eq. (2.44), it can be observed that the following relation holds for all three cases:

$$\frac{dI_1(a, b)}{d\alpha} = \frac{I_2(a, b)}{2} \quad (\text{B.10})$$

Therefore, based on (B.6), the following also holds:

$$\frac{d\mathcal{I}_1}{d\alpha} = \frac{\mathcal{I}_2}{2} \quad (\text{B.11})$$

As a result, Eqs. (B.5) can be rewritten in the more compact form:

$$\frac{dI_N}{d\alpha} = -\sin \alpha \left(\mathcal{I}_1 + \frac{d\mathcal{I}_2}{d\alpha} \right) - \cos \alpha \frac{\mathcal{I}_2}{2} \quad (\text{B.12a})$$

$$\frac{dI_D}{d\alpha} = \cos \alpha \left(\mathcal{I}_1 + \frac{d\mathcal{I}_2}{d\alpha} \right) - \sin \alpha \frac{\mathcal{I}_2}{2} \quad (\text{B.12b})$$

Consequently, the derivative $d\mathcal{N}/d\alpha$ can be evaluated by reusing the previously computed values of $\mathcal{I}_1$ and $\mathcal{I}_2$, which are already required for the evaluation of $\mathcal{N}(\alpha)$. The only additional quantity that must be computed is the derivative:

$$\frac{d\mathcal{I}_2}{d\alpha} = \begin{cases} \frac{\sin(\gamma_0 - \alpha)}{\sqrt{\cos(\gamma_0 - \alpha)}} - \frac{\sin(\gamma_1 - \alpha)}{\sqrt{\cos(\gamma_1 - \alpha)}}, & \text{for case A} \\ \frac{\sin(\gamma_0 - \alpha)}{\sqrt{\cos(\gamma_0 - \alpha)}} + \frac{\sin(\gamma_1 - \alpha)}{\sqrt{\cos(\gamma_1 - \alpha)}}, & \text{for cases B and C} \end{cases} \quad (\text{B.13})$$

In the general case, the derivative $d\mathcal{N}/d\alpha$ is evaluated by successively applying Eqs. (B.6), (B.13), (B.12), and (B.3).

B.2 Numerical Integration

B.2.1 Gauss–Legendre Quadrature Formula

The n -point Gauss–Legendre quadrature approximates the integral of a function over the interval $[-1, 1]$ according to:

$$\int_{-1}^1 f(x) dx \approx \sum_{k=1}^n w_k f(x_k) \quad (\text{B.14})$$

where x_k are the quadrature nodes and w_k are the corresponding weights. The nodes are the roots of the Legendre polynomial of degree n , while the weights are determined so that the quadrature formula has an algebraic degree of precision equal to $2n - 1$, i.e., every polynomial of degree not exceeding $2n - 1$ is integrated exactly.

Since the quadrature rule is defined on the interval $[-1, 1]$, an integral over an arbitrary interval $[a, b]$ is transformed using:

$$x = \frac{a+b}{2} + \frac{b-a}{2}\xi, \quad \xi \in [-1, 1] \quad (\text{B.15})$$

which yields:

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{a+b}{2} + \frac{b-a}{2}\xi\right) d\xi \quad (\text{B.16})$$

Consequently,

$$\int_a^b f(x) dx \approx \frac{b-a}{2} \sum_{k=1}^n w_k f\left(\frac{a+b}{2} + \frac{b-a}{2}x_k\right) \quad (\text{B.17})$$

In this dissertation, a 10-point Gauss–Legendre quadrature rule was employed. Owing to the symmetry of the nodes and weights:

$$x_{11-k} = -x_k, \quad w_{11-k} = w_k \quad (\text{B.18})$$

only the positive nodes and their corresponding weights need to be stored. These values are listed in Table B.1.

Using the symmetry of the quadrature rule, the approximation can be written in the computationally efficient form:

$$\int_a^b f(x) \, dx \approx \frac{b-a}{2} \sum_{k=1}^5 w_k [f(x_m + x_r x_k) + f(x_m - x_r x_k)] \quad (\text{B.19})$$

where:

$$x_m = \frac{a+b}{2}, \quad x_r = \frac{b-a}{2} \quad (\text{B.20})$$

This expression was used throughout the numerical implementation presented in this dissertation.

Table B.1: Positive nodes and corresponding weights for the 10-point Gauss–Legendre quadrature rule [91].

Node x_k	Weight w_k
0.1488743389	0.2955242247
0.4333953941	0.2692667193
0.6794095682	0.2190863625
0.8650633667	0.1494513492
0.9739065285	0.0666713443

B.3 Minimax Polynomial Approximation

Many elementary functions cannot be evaluated exactly using a finite number of arithmetic operations. Consequently, they are often approximated by polynomials, whose evaluation requires only additions and multiplications. In this dissertation, minimax polynomial approximations were considered for the cosine function in order to reduce the computational cost of repeated function evaluations.

Unlike a Taylor polynomial, which minimizes the approximation error only in the vicinity of the expansion point, a minimax polynomial minimizes the maximum absolute approximation error over a prescribed interval. That is, the coefficients are chosen so that:

$$P_n(x) = \arg \min_{Q_n} \left(\max_{x \in [a,b]} |f(x) - Q_n(x)| \right) \quad (\text{B.21})$$

where $Q_n(x)$ denotes any polynomial of degree n . As a result, the approximation error is distributed nearly uniformly over the entire interval, making minimax polynomials particularly suitable for numerical implementations.

Since the cosine function is even, only even powers of the argument are required, so the approximation takes the form:

$$\cos x \approx P_{2m}(x) = c_0 + c_2x^2 + c_4x^4 + \cdots + c_{2m}x^{2m} \quad (\text{B.22})$$

In this dissertation, sixth-, eighth-, and tenth-order minimax approximations were analyzed. Their corresponding polynomial coefficients are listed in Table B.2. Based on the achieved trade-off between computational efficiency and approximation accuracy, the eighth-order approximation was adopted in the final implementation.

Table B.2: Coefficients of the minimax polynomial approximations of the cosine function.

Coefficient	6th order	8th order	10th order
c_0	1.0	1.0	1.0
c_2	−0.4999999975	−0.5000000000	−0.5000000000
c_4	0.0416666418	0.0416666667	0.0416666667
c_6	−0.0013888378	−0.0013888889	−0.0013888889
c_8	—	2.480155×10^{-5}	2.480159×10^{-5}
c_{10}	—	—	$−2.755732 \times 10^{-7}$

The resulting polynomial can be evaluated efficiently by first computing the even powers of the argument:

$$x^2, x^4, x^6, \dots \quad (\text{B.23})$$

and then forming the weighted sum of the corresponding polynomial terms. This approach avoids the relatively expensive evaluation of the standard library cosine function while maintaining sufficient approximation accuracy over the interval of interest.

Appendix C

Six-Degree-of-Freedom Simulation Model

The mathematical model presented in this appendix is intended to provide the reader with sufficient detail to reproduce the six-degree-of-freedom flight simulations used throughout this dissertation and to facilitate further development or verification of the methods presented herein. For a more comprehensive treatment of flight dynamics and inertial navigation, the reader is referred to Refs. [106, 107].

C.1 Coordinate Frames

This section briefly introduces the coordinate frames required to understand the mathematical model presented in the following section. These reference frames are standard in aerospace engineering and are therefore described only to the extent necessary for the subsequent derivations.

The first reference frame is the Earth-centered inertial (*i*-frame), whose origin is located at the center of the Earth and whose axes are non-rotating with respect to the fixed stars. Its Oz_i axis coincides with the Earth's axis of rotation.

For terrestrial navigation, the Earth-fixed (*e*-frame) is of greater practical importance. This frame also has its origin at the Earth's center, but its axes rotate together with the Earth. The frame is defined by the axes Ox_e , Oy_e , and Oz_e , where Oz_e is aligned with the Earth's rotation axis, while Ox_e lies along the intersection of the Greenwich meridian plane and the Earth's equatorial plane, as illustrated in Fig. C.1. The Earth-fixed frame rotates relative to the inertial frame with the angular velocity Ω about the Oz_i axis.

The vehicle position is represented by the geographic coordinates latitude (*lat*), longitude (*lon*), and altitude (*h*), which correspond to the spherical-coordinate representation in the Earth-fixed frame.

The navigation frame (*n*-frame), commonly referred to as the North-East-Down (NED) frame, is a local geographic frame whose origin coincides with the vehicle position P . Its axes are aligned with the local north (*N*), east (*E*), and downward vertical (*D*) directions, as shown in Fig. C.1.

Throughout this dissertation, the vehicle velocity is expressed as the components of the ground velocity (velocity relative to the Earth) resolved in the navigation frame.

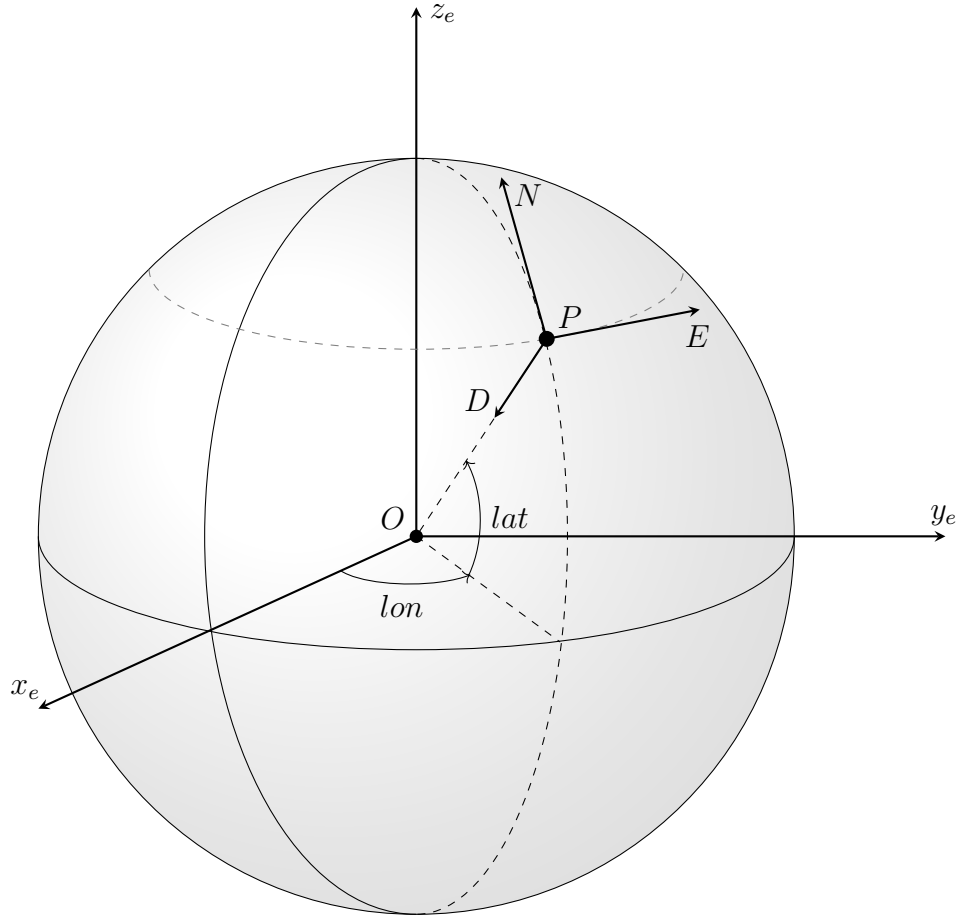


Figure C.1: Coordinate frames for terrestrial navigation.

Another local reference frame of particular importance is the azimuth frame. It is obtained by rotating the NED frame about the local vertical through the azimuth angle (Az). Missile range and cross-range deviation are conveniently expressed in this frame, which is defined as follows. Consider the great circle (shown in red in Fig. C.2) connecting the launch point (L) and the target point (T). The arc of this great circle between the two points represents the desired range, and the x_{Az} axis is therefore defined as the curvilinear coordinate measured along this arc. The y_{Az} axis represents the perpendicular distance from the missile to the plane containing the great circle, while the z_{Az} axis is defined as the negative altitude. This definition of the azimuth frame is used consistently throughout this dissertation.

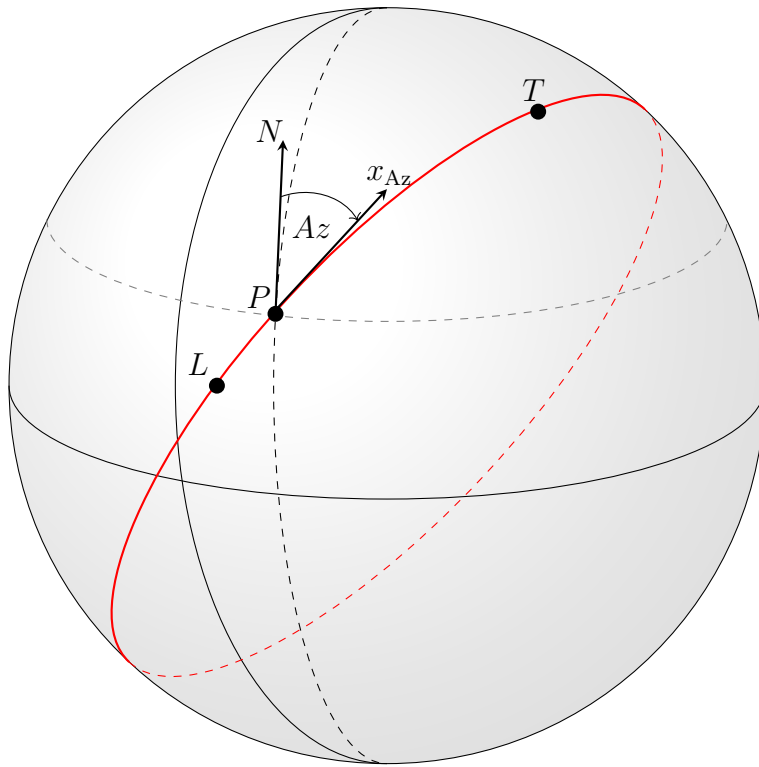


Figure C.2: Azimuth coordinate frame.

Finally, the body frame (b -frame), illustrated in Fig. C.3, is a Cartesian coordinate frame fixed to the vehicle, with its origin at the vehicle center of mass C . Its axes coincide with the roll, pitch, and yaw axes of the vehicle.

A closely related reference frame is the aeroballistic frame, which is also attached to the vehicle. Unlike the body frame, however, it does not rotate about the longitudinal (x) axis. This property makes it particularly convenient for defining the commanded specific force, which is subsequently transformed into the body frame, as described by Eq. (3.80).

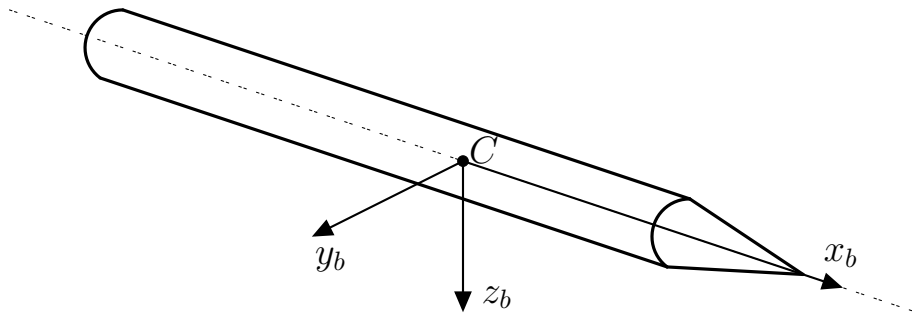


Figure C.3: Body coordinate frame.

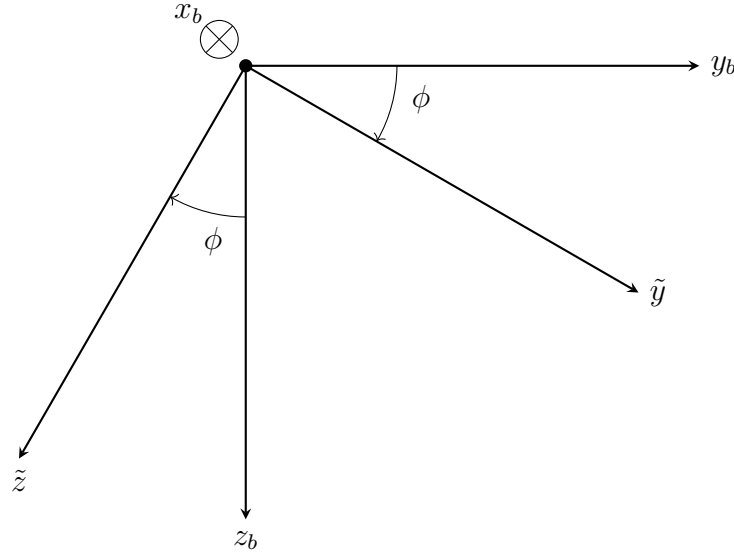


Figure C.4: Aeroballistic coordinate frame.

C.2 Mathematical Model

A few remarks regarding the notation used throughout this appendix:

- For transformation matrices between two coordinate frames, the subscript denotes the frame from which a vector is transformed, while the superscript denotes the frame in which it is expressed.
- For angular rate vectors (denoted by ω), the two subscripts indicate the frames between which the angular rate is defined (the latter relative to the former). The superscript denotes the frame in which the vector is expressed. For example, ω_{ib}^b represents the angular rate of the body (b) frame relative to the inertial (i) frame, expressed in the body frame.
- For other vectors (for example, force vectors), the subscript denotes the vector type or origin, while the superscript indicates the coordinate frame in which it is expressed. For example, $\mathbf{F}_g^n$ denotes the gravitational force, expressed in the local geographic (n) frame.

The mathematical model is based on the navigation algorithm presented by Titterton [107]. Since global navigation over the Earth's surface is not the focus of this dissertation, but rather the guidance problem itself, a spherical Earth model is assumed.

The primary objective of the navigation algorithm is to determine the vehicle velocity with respect to the Earth, expressed in the local geographic frame:

$$\mathbf{v}_e^n = \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} \quad (\text{C.1})$$

and the vehicle position in terms of latitude, longitude, and altitude:

$$\begin{bmatrix} lat \\ lon \\ h \end{bmatrix} \quad (C.2)$$

based on the specific force vector, which in practice would be measured by a triad of body-fixed accelerometers:

$$\mathbf{f}^b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \quad (C.3)$$

and the angular rate of the body frame with respect to the inertial frame, which in practice would be measured by a triad of body-fixed gyroscopes:

$$\boldsymbol{\omega}_{ib}^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (C.4)$$

The fundamental navigation equation is:

$$\dot{\mathbf{v}}_e^n = \mathbf{C}_b^n \mathbf{f}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}_e^n + \mathbf{g}_l^n \quad (C.5)$$

where $\mathbf{C}_b^n$ is the direction cosine matrix used to transform the measured specific force vector from the body frame to the navigation frame:

$$\mathbf{C}_b^n = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \quad (C.6)$$

This matrix evolves according to:

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \boldsymbol{\Omega}_{nb}^b \quad (C.7)$$

where:

$$\boldsymbol{\Omega}_{nb}^b = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (C.8)$$

is the skew-symmetric matrix associated with $\boldsymbol{\omega}_{nb}^b$, the angular rate of the body frame relative to the navigation frame:

$$\boldsymbol{\omega}_{nb}^b = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b (\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \quad (C.9)$$

Equation (C.7) may also be written in component form as:

$$\begin{aligned}\dot{c}_{11} &= c_{12}\omega_z - c_{13}\omega_y & \dot{c}_{12} &= c_{13}\omega_x - c_{11}\omega_z & \dot{c}_{13} &= c_{11}\omega_y - c_{12}\omega_x \\ \dot{c}_{21} &= c_{22}\omega_z - c_{23}\omega_y & \dot{c}_{22} &= c_{23}\omega_x - c_{21}\omega_z & \dot{c}_{23} &= c_{21}\omega_y - c_{22}\omega_x \\ \dot{c}_{31} &= c_{32}\omega_z - c_{33}\omega_y & \dot{c}_{32} &= c_{33}\omega_x - c_{31}\omega_z & \dot{c}_{33} &= c_{31}\omega_y - c_{32}\omega_x\end{aligned}\tag{C.10}$$

The Euler angles can be obtained directly from the direction cosine matrix. Provided that $\theta \neq 90^\circ$, they are calculated as:

$$\begin{aligned}\phi &= \text{atan}\left(\frac{c_{32}}{c_{33}}\right) \\ \theta &= \text{asin}(-c_{31}) \\ \psi &= \text{atan}\left(\frac{c_{21}}{c_{11}}\right)\end{aligned}\tag{C.11}$$

The remaining terms in Eq. (C.5) are defined as follows.

The vector $\boldsymbol{\omega}_{ie}^n$ is the Earth's rotation rate (Ω) expressed in the local geographic frame:

$$\boldsymbol{\omega}_{ie}^n = \begin{bmatrix} \Omega \cos(lat) \\ 0 \\ -\Omega \sin(lat) \end{bmatrix}\tag{C.12}$$

The vector $\boldsymbol{\omega}_{en}^n$ represents the angular rate of the local geographic frame with respect to the Earth-fixed frame (the transport rate). It can be expressed in terms of the latitude and longitude rates as:

$$\boldsymbol{\omega}_{en}^n = \begin{bmatrix} \dot{lon} \cos(lat) \\ -\dot{lat} \\ -\dot{lon} \sin(lat) \end{bmatrix}\tag{C.13}$$

where the latitude and longitude rates are given by:

$$\begin{aligned}\dot{lat} &= \frac{v_N}{(R_e + h)} \\ \dot{lon} &= \frac{v_E}{(R_e + h)}\end{aligned}\tag{C.14}$$

Here, R_e denotes the Earth's radius (taken as 6371 km in this work), while h is the altitude above the Earth's surface, which evolves according to:

$$\dot{h} = -v_D\tag{C.15}$$

The vector $\mathbf{g}_l^n$ is the local gravity vector, which accounts for both the Earth's gravitational attraction and the centripetal acceleration caused by the Earth's rotation:

$$\mathbf{g}_l^n = \mathbf{g}^n - \boldsymbol{\omega}_{ie}^n \times \boldsymbol{\omega}_{ie}^n \times \mathbf{R}^n \quad (\text{C.16})$$

In the last equation, $\mathbf{g}^n$ is the mass attraction gravitation vector, expressed in local frame:

$$\mathbf{g}^n = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \quad g = \frac{0.3986005 \times 10^{15}}{(R_e + h)^2} \quad (\text{C.17})$$

while $\mathbf{R}^n$ is the position vector with respect to the Earth's center, expressed in the local geographic frame:

$$\mathbf{R}^n = \mathbf{C}_e^n \mathbf{R}^e \quad (\text{C.18})$$

The vector $\mathbf{R}^e$ is obtained from the geographic coordinates (lat, lon, h) through the standard transformation from spherical to Cartesian coordinates, while $\mathbf{C}_e^n$ is the transformation matrix from the Earth-fixed frame to the local geographic frame.

It is also convenient to track the orientation of the azimuth frame relative to the local geographic frame. This orientation is described by the azimuth angle, whose rate of change is:

$$\dot{Az} = \frac{v_E}{R_e + h} \tan(lat) \quad (\text{C.19})$$

Once the azimuth angle has been updated, the ground velocity can be expressed in the azimuth frame as:

$$\mathbf{v}_e^{az} = \begin{bmatrix} \dot{x}_{Az} \\ \dot{y}_{Az} \\ -\dot{h} \end{bmatrix} = \mathbf{C}_n^{az} \mathbf{v}_e^n \quad (\text{C.20})$$

where:

$$\mathbf{C}_n^{az} = \begin{bmatrix} \cos(Az) & \sin(Az) & 0 \\ -\sin(Az) & \cos(Az) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{C.21})$$

The orientation of the velocity vector is of particular interest in this dissertation, especially the flight-path angle γ in the vertical plane, defined as:

$$\gamma = -\text{asin}\left(\frac{v_D}{V}\right) \quad (\text{C.22})$$

where $V = \sqrt{v_N^2 + v_E^2 + v_D^2}$ denotes the vehicle speed.

To provide the inputs required by the navigation algorithm described above, the specific force vector and angular rate vector must first be generated, as they would be measured by body-fixed accelerometers and gyroscopes.

The specific force vector is obtained as the sum of all non-gravitational forces divided by the vehicle mass:

$$\mathbf{f}^b = \frac{\sum \mathbf{F}^b}{m} - \mathbf{g}^b \quad (\text{C.23})$$

The angular rate vector is obtained by integrating Euler's rotational equation of motion:

$$\boldsymbol{\omega}_{ib}^b = \int \mathbf{J}^{-1} \left(\sum \mathbf{M}^b - \boldsymbol{\omega}_{ib}^b \times (\mathbf{J} \times \boldsymbol{\omega}_{ib}^b) \right) dt \quad (\text{C.24})$$

where $\mathbf{J}$ is the inertia tensor, which, for cruciform (axisymmetric) missiles, reduces to

$$\mathbf{J} = \begin{bmatrix} J_x \\ J_y \\ J_z \end{bmatrix} \quad (\text{C.25})$$

and $\sum \mathbf{M}^b$ denotes the sum of all external moments acting on the vehicle.

The resultant force acting on the vehicle consists of the gravitational, thrust, and aerodynamic forces. Expressed in the body frame, it is given by

$$\sum \mathbf{F}^b = \mathbf{C}_n^b \mathbf{F}_g^n + \mathbf{F}_t^b + \mathbf{F}_a^b \quad (\text{C.26})$$

where the gravitational force expressed in the local frame is:

$$\mathbf{F}_g^n = m \mathbf{g}^n \quad (\text{C.27})$$

and $\mathbf{g}^n$ is given by Eq. (C.17).

To evaluate the proposed guidance algorithm, no thrust misalignment was considered in the simulations. Consequently, the thrust force is modeled as:

$$\mathbf{F}_t^b = \begin{bmatrix} F_t(t) \\ 0 \\ 0 \end{bmatrix} \quad (\text{C.28})$$

where $F_t(t)$ is the thrust profile of the corresponding missile.

Under this assumption, no moments are generated by thrust misalignment, and the aerodynamic moment is therefore the only external moment acting on the vehicle. The aerodynamic force and moment are computed from the aerodynamic coefficients as:

$$\begin{aligned}\mathbf{F}_a^b &= \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} q_\infty S_{\text{ref}} \\ \mathbf{M}_a^b &= \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} q_\infty S_{\text{ref}} l_{\text{ref}}\end{aligned}\tag{C.29}$$

In the above expressions, l_{ref} and S_{ref} denote the reference length (typically the missile caliber d) and reference area, respectively:

$$S_{\text{ref}} = \frac{d^2 \pi}{4}, \quad l_{\text{ref}} = d\tag{C.30}$$

The dynamic pressure is given by:

$$q_\infty = \frac{1}{2} \rho V_a^2\tag{C.31}$$

where V_a is the airspeed (vehicle speed relative to the surrounding air) and ρ is the air density, which depends on altitude, i.e., $\rho = \rho(h)$.

The aerodynamic velocity is obtained by combining the ground velocity $\mathbf{v}_e$ and the wind velocity vector $\mathbf{w}$:

$$\mathbf{v}_a^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{C}_n^b \mathbf{v}_e^n + \mathbf{w}^b\tag{C.32}$$

from which the airspeed follows as $V_a = \sqrt{u^2 + v^2 + w^2}$.

The aerodynamic coefficients appearing in Eq. (C.29) depend on numerous flight parameters, and their expressions are specific to each missile considered in this dissertation. The principal flight parameters (in addition to those already introduced) are summarized below.

The Mach number is defined as:

$$M = \frac{V_a}{c}\tag{C.33}$$

where c is the speed of sound, which is also altitude-dependent, i.e., $c = c(h)$.

The angle of attack and sideslip angle are defined as:

$$\alpha = \text{atan}\left(\frac{w}{u}\right), \quad \beta = \text{asin}\left(\frac{v}{V_a}\right)\tag{C.34}$$

The final component of the simulation model is the control system, whose purpose is to generate the control surface deflections for roll (ξ), pitch (η), and yaw (ζ).

The block diagram of the roll autopilot is shown in Fig. C.5. The input signal is the commanded roll angle ϕ_{cmd} , which is zero because both missile configurations considered in this dissertation are roll-stabilized. The feedback signals are the estimated roll angle ϕ and roll rate p , while the output is the commanded roll control surface deflection ξ . The controller gains K and G_R are

selected to achieve the desired dynamic response and command-tracking performance based on the aerodynamic data and mechanical properties for the corresponding missile, following the autopilot synthesis procedure described in Ref. [26].

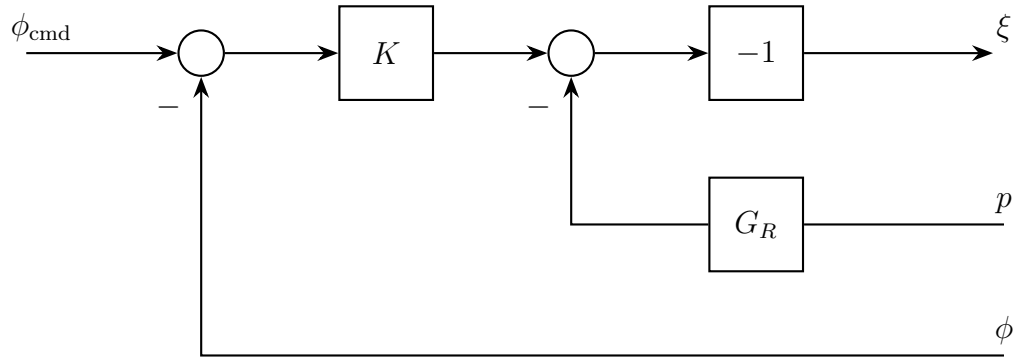


Figure C.5: Roll autopilot block diagram.

For the pitch channel, the conventional three-loop acceleration autopilot was adopted because of its well-established practical advantages, as discussed in Ref. [108]. The input is the commanded specific force along the body-frame z -axis, $f_{z,\text{cmd}}$, computed from Eq. (3.80a) using either Eq. (3.85) or Eq. (3.95), depending on the flight scenario. The feedback signals are the measured specific force f_z and the pitch rate q , while the output is the commanded pitch control surface deflection η . The controller gains K_i , k_{ac} , k_r , and k_a are determined using the aerodynamic data and mechanical properties according to the synthesis procedure presented in Ref. [109], so as to provide the required response characteristics and command-tracking performance.

The yaw autopilot has the same structure as the pitch autopilot. The commanded specific force $f_{y,\text{cmd}}$ replaces $f_{z,\text{cmd}}$, the yaw rate r replaces the pitch rate q , and the output is the commanded yaw control surface deflection ζ . In general, the controller gains differ from those of the pitch autopilot because they are determined independently from the corresponding aerodynamic characteristics using the same synthesis procedure.

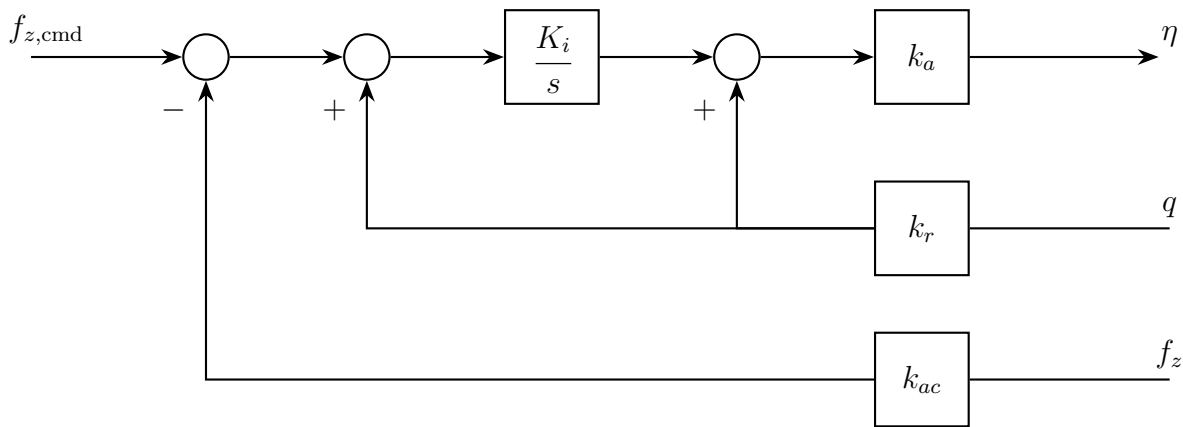


Figure C.6: Pitch autopilot block diagram.

Biography

Luka Miličić was born on August 26, 1996, in Belgrade. He completed his elementary education at “Filip Kljajić Fića” in 2011 with a perfect Grade Point Average (GPA) of 5.00/5.00. He subsequently enrolled in the “Aviation Academy” high school in Belgrade, specializing as an Air Traffic Technician, graduating in 2015 with a perfect GPA of 5.00/5.00.

In the same year, he enrolled in the Faculty of Mechanical Engineering at the University of Belgrade. During his undergraduate studies, he successfully competed in the academic competition “Mašinijada,” winning first place in Mathematics in both 2017 and 2018, as well as first place in Mechanics in 2018. He completed his Bachelor of Science (B.Sc.) studies in 2018 with a perfect GPA of 10.00/10.00. His final thesis, titled “*Optimization of Basic Parameters in the Conceptual Design Phase of an Unguided Artillery Rocket*”, was conducted under the supervision of Professor Momčilo Milinović, Ph.D., and defended with the highest grade (10).

Following his undergraduate graduation in 2018, he enrolled in Master of Science (M.Sc.) academic studies in the Weapon Systems module at the same faculty. During his first year of master’s studies, he was funded by the Ministry of Education, Science and Technological Development, and during his second year by the Ministry of Defence of the Republic of Serbia. He completed his Master’s degree in 2020 with a perfect GPA of 10.00/10.00. His master’s thesis, titled “*Synthesis of Guidance Algorithm for an Anti-Tank Guided Missile*”, was supervised by Assistant Professor Ivana Todić, Ph.D., and defended with a grade of 10. He received further commendations for outstanding academic success and was officially named the Valedictorian (Student of the Generation) for the student cohort enrolled in the 2015/2016 academic year.

In 2020, he enrolled in the Doctoral (Ph.D.) academic studies program at the Faculty of Mechanical Engineering, University of Belgrade. Since March 2021, he has been employed full-time at the Faculty of Mechanical Engineering as a Teaching and Research Assistant in the scientific field of Weapon Systems – Military Engineering. In this role, he conducts laboratory and tutorial classes for several courses, including *Missile Flight Mechanics*, *Flight Dynamics and Missile Aerodynamics*, and *Missile Guidance and Control*. Since June 2021, he has also been actively involved in research activities within the scientific project “*Integrated Research in the Field of Micro, Macro, and Nano Mechanical Engineering*.”

In addition to his academic career, he has been engaged since March 2020 as a Research Engineer in the Guided Missiles department at the company EDePro. In his professional work, he regularly utilizes the following programming languages and software tools: C, C++, C#, Python, MATLAB, Simulink, L^AT_EX, and Microsoft Office. Besides his native Serbian, he is fluent in English. He is married and the father of three children.

Изјава о ауторству

Име и презиме аутора: Лука Миличић

Број индекса: Д35/20

Изјављујем

да је докторска дисертација под насловом

New algorithm for optimal impact angle guidance
(Нови алгоритам оптималног вођења са ограничењем упадног угла)

- резултат сопственог истраживачког рада;
- да предложена дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, _____

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора:	Лука Миличић
Број индекса:	Д35/20
Студијски програм:	Докторске академске студије
Наслов рада:	New algorithm for optimal impact angle guidance (Нови алгоритам оптималног вођења са ограничењем упадног угла)
Ментор:	др Милош Марковић, ванредни професор

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду**.

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, _____

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

New algorithm for optimal impact angle guidance

(Нови алгоритам оптималног вођења са ограничењем упадног угла)

која је моје ауторско дело.

Дисертацију са свим прилозима предао сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио.

1. Ауторство (CC BY)
2. Ауторство – некомерцијално (CC BY-NC)
3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)
4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)
5. Ауторство – без прерада (CC BY-ND)
6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.
Кратак опис лиценци је саставни део ове изјаве).

Потпис докторанда

У Београду, _____

1. Ауторство. Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. Ауторство – некомерцијално. Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. Ауторство – некомерцијално – без прерада. Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. Ауторство – некомерцијално – делити под истим условима. Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. Ауторство – без прерада. Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. Ауторство – делити под истим условима. Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.