Optimal control and numerical optimization for missile interception guidance

Conference Paper · June 2014

DOI: 10.1109/ECC.2014.6862503

Spilios Theodoulis
Deutsch-Französisches Forschungsinstitut Sain…

Philippe Wernert
French-German Research Institute of Saint-Lou…

Florian Holzapfel
Technische Universität München

2014 European Control Conference (ECC)
June 24-27, 2014. Strasbourg, France

Leif Walter, Gunther Schlöffel, Spilios Theodoulis, Philippe Wernert,

Ekaterina Kostina, and Florian Holzapfel

Optimal Control and Numerical Optimization for Missile Interception Guidance

Abstract— **Conventional guidance algorithms for exo-atmospheric missile interception, such as Proportional Navigation, provide guidance laws that are derived from conceptual heuristics where certain simplifications have to be made. Typically, these guidance laws have the disadvantage of not being able to fully take into account system and/or vehicle limitations such as maximum thrust. In this paper, optimal control and numerical optimization methods are used to provide a guidance scheme that may incorporate these critical constraints, while still ensuring a successful hit-to-kill interception of the target. Therein, a direct optimal control approach is deployed, based on multiple shooting and a sequential quadratic programming algorithm for solving the resulting nonlinear optimization problem. Numerical results for three distinctive transcription methods for deriving the nonlinear program are presented that illustrate the overall efficiency of numerical optimization as a guidance scheme.**

INTRODUCTION

In exo-atmospheric missile interception, a kinetic kill vehicle aims for achieving a hit-to-kill interception with a hostile target. In typical concepts, the guidance routine for this kind of vehicle is provided by traditional guidance algorithms such as *Proportional Navigation*-based laws or *Zero-Effort-Miss* guidance, see [1], [2]. These algorithms are based on geometric heuristics to derive efficient laws for computing required acceleration commands. The disadvantage, however, is that they typically do not incorporate constraints on the system (such as the seeker's field of view) or limitations of the vehicle (such as maximum thrust). As a consequence, these guidance laws may not achieve successful interceptions when applied within real systems, elaborated simulations or difficult endgame situations. Additionally, offline simulations become more efficient when these constraints can be addressed within the guidance scheme. Optimal control and numerical optimization are alternative methods to approach the so-called endgame problem of missile interception guidance and may very well incorporate above mentioned constraints. Indirect optimal control methods in particular sliding mode methodology are used in [29], [30] to derive guidance-to-collision laws, which show superior performance compared to classical proportional navigation. Direct optimal control methods have been used to provide aerospace guidance algorithms for several decades now, with applications ranging from aircraft fighter evasion, [3], spacecraft reentry, [4], to anti-satellite missiles [5].

The emphasis of this paper is to present an efficient optimization framework that potentially enables real-time guidance for interceptor missiles. This framework is based on the direct multiple shooting approach, [6], [7], to discretize the system's state variables and a *sequential quadratic programming* (SQP) algorithm, [8], [9], [10], to solve the resulting nonlinear optimization problem. The real-time scheme is obtained by an efficient *nonlinear model predictitve control* (NMPC) formulation, [11], [12], that is based on key improvements, [13], to conventional *brute force* sequential optimization.

In this paper, three different so-called transcription schemes to formulate and model endgame guidance as an optimal control problem are introduced. This is done to provide an overview of possible applications of numerical optimization within guidance algorithms and to compare their efficiency for this application. In practical applications, the choice of transcription method is one of the key drivers to the algorithm's overall performance. Numerical results are presented that are compared to results by traditional *Proportional Navigation* (PN). These trials illustrate the overall efficiency of the here introduced method and its performance as endgame guidance scheme.

The structure of the paper is as follows. Section II will introduce the optimal control formulation used within this paper, and will outline the direct multiple shooting approach with the corresponding transcription methods. In Section III, the optimization algorithm for solving the optimal control problem is outlined. Section IV will cover the real-time algorithm while numerical results will finally be presented in Section V.

OPTIMAL CONTROL AND DIRECT MULTIPLE SHOOTING

In optimal control, the task is to identify an optimal control history $u(t)$ that will drive a dynamic system on a given time horizon $[t_0,t_f]$, from its initial state $x(t_0)$ to its final state $x(t_f)$, in an optimal way. Optimality is measured by a performance index J, defined as

$$J(x,u) = \int L(x(t),u(t),t)dt + \Psi(x_f,u_f,t_f), \qquad (1)$$

where L is a *Lebesque* integrable function and Ψ is the so-called endpoint term that defines conditions for the final system state and/or controls. At the same time, the process is subject to equality and inquality constraints. Most importantly, the process' state dynamics must be satisfied, expressed by the condition

L. Walter, G. Schlöffel, S. Theodoulis, and P. Wernert are with the department of Guidance, Navigation, and Control, at the French-German Research Institute of Saint Louis, Saint Louis, France (phone: +33-389-69-5041; e-mail: leif.walter@isl.eu).

E. Kostina is a professor at the Faculty of Mathematics and Computer Science, Philipps-University Marburg, Marburg, Germany (e-mail: kostina@mathematik.uni-marburg.de).

F. Holzapfel is a professor at the Institute for Flight System Dynamics, Technical University of Munich, Garching, Germany (e-mail: florian.holzapfel@tum.de).

$$\dot{x} = f(t,x(t),u(t)), \quad (2)$$

where f is a set of ordinary differential equations, or differential-algebraic equations. So-called path constraints may be imposed to the system as well, i.e.

$$h(t,x(t),u(t)) \geq 0. \quad (3)$$

These constraints typically consist of the above mentioned system or control constraints and are key to the efficiency of the here proposed approach. Finally, the process may be subject to terminal constraints, like preassigned initial and final system states (such as position and/or velocity). These conditions are typically formalized as

$$C(t_f,x_f,u_f) \geq 0. \quad (4)$$

Therefore, the optimal control problem (**OCP**) in standard form may be formulated as:

$$(\mathbf{OCP}) \quad \min \quad \int L(x(t),u(t),t)dt + \Psi(x_f,u_f,t_f)$$

$$\text{subject to} \quad \begin{aligned} \dot{x} &= f(t,x(t),u(t)) \\ 0 &\leq h(t,x(t),u(t)) \\ 0 &\leq C_f(t_f,x_f,u_f) \\ 0 &\leq C_0(t_0,x_0,u_0) \end{aligned}$$

Note that in this formulation, the task is to identify an optimal control history $u(t)$ and corresponding optimal state trajectories $x(t)$. Therefore, this formulation is called an infinite-dimensional problem. Even on a finite time horizon $[t_0,t_f]$, the number of actual variables is infinite. In order to resolve this, the formulation must be transformed to a finite-dimensional problem. This can either be done by direct or indirect methods. Within this approach, the direct method was chosen as they typically show better performance in practical applications and may be extended to closed-loop schemes as done in this paper. For complementary information on the indirect method, please see [7], [14], [15].

In the following subsections, the direct transcription via multiple shooting will be outlined. The process of direct transcription aims for finding a finite-dimensional representation of both control and state trajectories, i.e. $u(t)$ and $x(t)$, respectively.

A. State Discretization

Within the multiple shooting approach, state trajectories are discretized on a so-called multiple shooting grid $T=\{\tau_i\}_{i=0,\ldots,N-1}$ with

$$t_0 = \tau_0 < \tau_1 < \ldots < \tau_{N-2} < \tau_{N-1} = t_f,$$

representing the complete trajectory by a finite number of state values on grid nodes s_i, $i = 0, \ldots, N-1$, with N being the number of grid nodes. All values between these grid nodes only depend on the initial value of the respective segment, i.e. the corresponding node value. Therefore, the state trajectory on each segment is obtained by numerically integrating the initial value problem.

This is illustrated in Figure 1, where the state trajectory segmentation is shown. Multiple shooting grid nodes (circles) s_i are included as variables of optimization within the optimization problem. On each individual segment, an initial value problem is solved to obtain the segment's trajectory and, in particular, the segments final integration value (bold dots) $x_i(s_i,u_i;\, \tau_{i+1})$ at the next grid node. To ensure realistic,

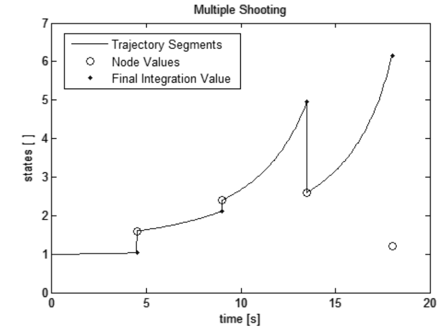

Figure 1. Multiple shooting illustration. The y-axis shows arbitrary, hence dimensionless, state values.

hence continuous state trajectories, the so-called *continuity condition,*

$$x_i(s_i,u_i;\, \tau_{i+1}) - s_{i+1} = 0, \quad (5)$$

is imposed. Note that $u_i(t)$ means the control input on the i-th segment, i.e. $u(t)$ for $t \in [\tau_i,\tau_{i+1}]$. Using this approach, the system's states can be represented by a finite number of variables within the optimization problem, and their adherence to the system's dynamics is assured by imposing (5). Also note that in figure 1, it is apparent that (5) is violated on all grid nodes.

B. Control Transcription

The control history $u(t)$ needs to be represented by a finite number of variables within the optimization problem as well. In fact, there is a multitude of so-called control transcriptions that all have their advantages and disadvantages. Within this paper, three transcription possibilities are introduced and investigated as part of numerical simulations.

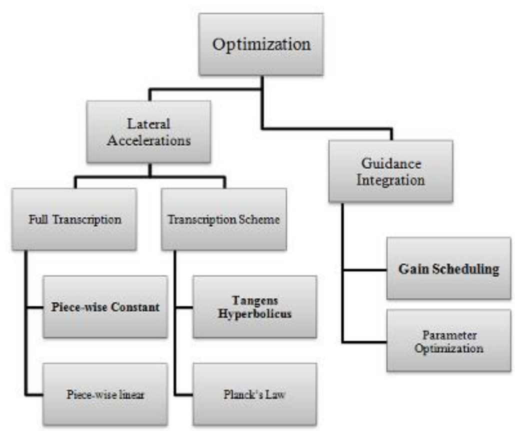

Figure 2. Control transcription methods for optimal control problems in interceptor guidance. Transcription methods for which numerical results are included in this paper are highlighted bold.

Figure 2 illustrates a number of transcription methods available for direct solution methods for optimal control problems in interceptor guidance. The left branch refers to optimizing the lateral acceleration profile itself. Such an optimized acceleration profile would ensure a successful hit of interceptor and its target. In contrary to conventional guidance laws, where an acceleration command is provided for each instant of time t, here, an optimized acceleration profile is computed in advance. Therein, lateral accelerations could either be represented by optimization variables directly, i.e. as piece-wise *Ansatz* functions (e.g. piece-wise polynomials of any degree) on a given control grid that is typically chosen to be finer than the multiple shooting grid. Additionally, a transcription scheme may be imposed where the general structure of the lateral accelerations is prescribed by some function, which may then be altered by optimizing certain variables or parameters within that function. Figure 3 illustrates two of such possibilities by either optimizing parameters within a *Tangens Hyperbolicus* function, (6), (dashed line), [16], [17], or within a function equivalent to *Planck's Law*, (7), that originally describes black body radiation (straight line), [18]. The shown thrust profiles are given by

$$u_1(t) = ½ \cdot (1 - \tanh(p_1 \cdot (t - p_2))), \quad (6)$$

$$u_2(t) = 100\, t^{-7} \cdot (\exp(p_3/t) - p_4)^{-1}, \quad (7)$$

where p_i would have been parameters to be optimized within the optimization algorithm.

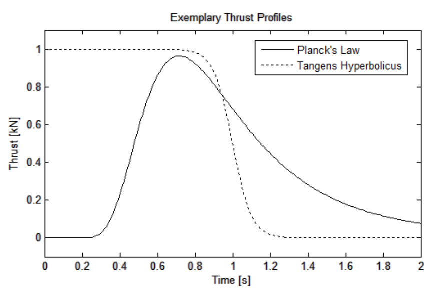

Figure 3. Exemplary thrust profiles when using a transcription scheme for parameterizing the control profile within the optimization problem.

The advantage of these transcription methods is twofold. First, the size of the resulting optimization problem is typically reduced significantly. In the above shown transcriptions as in (6) and (7), the whole control profile is represented by only two optimization variables. When choosing a full transcription via piece-wise polynomial functions, the number of optimization variables with respect to the controls is

$$n_u \cdot ((N_u - 1) \cdot (r + 1) + 1),$$

where n_u is the number of control variables, N_u the number of control grid segments, and r is the polynomial degree. Therefore, full control transcription typically yields much bigger optimization problems that will take significantly more time to be solved. Another advantage of these transcription schemes is that system characteristics can be expressed by the transcription scheme itself, rather than adding them as additional constraints to the optimization problem.

As an example, consider the interception endgame addressed here. It was shown, see [16], that due to geometrical heuristics, it may be assumed that the interceptor's thrusters should be fired as soon as possible. The instant of time when thrusters are cut off is to be optimized. In this case, the *Tangens Hyperbolicus* approach as in Figure (3) is a suitable representation as not only thrust is already implicitly constrained by its maximum value, but also the overall structure (*ON* to *OFF* transition) is already imposed. An approach like in equation (3) may be useful when thrusters have a special thrust profile with a known structure, and when it is mandatory that this profile is accurately represented within the optimization problem.

The right-hand branch in Figure 2 regards the integration of conventional guidance algorithms within the optimization and about using optimization variables to tweak the inherent guidance law. An example for this is to use Proportional Navigation for computing required lateral accelerations, while the optimization algorithm provides an optimal gain schedule for the PN gain. In this case, lateral accelerations would be given as

$$a_{cmd}(t) = N_{PN}(t) \cdot P'_{rel} \cdot d\lambda/dt, \quad (8)$$

where $N_{PN}(t)$ would be provided by a piece-wise constant gain schedule. Note that (4) is the so-called *true PN*, where P'_{rel} is the normalized augmented relative position, [19]. Similarly, any arbitrary guidance law may be used for this kind of integration. Due to the fact that Proportional Navigation is one of the best-known guidance laws, it was investigated in the numerical simulations performed for this paper.

For the remainder of this section, it is assumed that the control trajectory is parameterized as a finite-dimensional vector

$$u = u_1, u_2, \ldots, u_m, \quad (9)$$

which may apply for all control transcription presented above. Consequently, both infinite-dimensional unknowns, control and state trajectories, are represented by finite-dimensional parameterizations. Now, the finite-dimensional optimization problem itself is presented in the next subsection.

C. Problem Formulation

The infinite-dimensional optimal control problem (**OCP**) can now be transcribed to a finite-dimensional formulation. Finite-dimensional nonlinear optimization problems (**NLP**) are expressed in standard form as:

$$(\textbf{NLP}) \qquad \min \quad F(z)$$
$$\text{subject to} \quad G(z) = 0$$
$$H(z) \geq 0$$

To match this formulation, the procedure outlined in the previous subsection need to be applied. The variable of optimization z now consists of both state and control representation, i.e.

$$z = (t_f, s_0, s_1, \ldots, s_{N-1}, u_1, \ldots, u_m)^T. \quad (10)$$

Typically, the final time t_f is also included within the optimization variables. The objective function in (**OCP**) is typically discretized as a finite sum approximation of the integral term, where the endpoint term is simply added, hence

$$F(z) = \textstyle\sum_j L_j(t_j,x_j,u_j) + \Psi(t_f,x_f,u_f). \quad (11)$$

Note that usually, the objective function F is evaluated on the multiple shooting grid. It is possible, however, to choose a custom grid to obtain the respective terms. The adherence to the system's dynamics is ensured by imposing the continuity condition as equality constraints within the optimization, i.e.

$$x_i(s_i,u_i;\tau_{i+1}) - s_{i+1} = 0, i = 0,\ldots,N\text{-}1. \qquad (12)$$

Path constraints h in (**OCP**) are normally evaluated on a separate path constraint grid $P = [\rho_k,\rho_{k+1}]$, $k=0,\ldots,N_h$. These function evaluations are formulated as equality and inequality constraints, respectively. Hence, it is imposed

$$h_k(t_k,x_k,u_k) \geq 0,\ k = 0,\ldots,N_h. \qquad (13)$$

It is important to note that adherence to these path constraints can only be guaranteed on the corresponding path constraint grid P. Between two grid nodes, path constraints may potentially be violated. Therefore, this grid must be chosen carefully and path constraints should always be checked outside the optimization routine as well. Additionally, h often includes upper and lower bounds to both system states and control variables.

To summarize, and for optimization variables (10), the optimal control problem (**OCP**) may now be formulated as a finite-dimensional optimal control problem (**FOC**):

$$\begin{array}{lll} (\textbf{FOC}) & \min & \sum_j L_j(t_j,x_j,u_j) + \Psi(t_f,s_{N\text{-}1},u). \\ & \text{subject to} & x_i(s_i,u_i;\tau_{i+1}) - s_{i+1} = 0,\ i = 0,\ldots,N\text{-}1, \\ & & h_k(t_k,x_k,u_k) \geq 0,\ k = 0,\ldots,N_h, \\ & & C_0(t_0,s_0,u_0) \geq 0, \\ & & C_f(t_f,s_{N\text{-}1},u_m) \geq 0, \end{array}$$

where terminal constraints are naturally added. Some remarks have to be made regarding the variables within (**FOC**). All state values x_j and x_k need to be extracted from the integrator at appropriate points, and their values depend on a corresponding preceding node value s_i. Since x_f is a function of $s_{N\text{-}1}$ and a subset of all components of u, the arguments in Ψ have been replaced accordingly. Note that u_i and u_k are vector arguments with, potentially, multiple components, depending on the control parameterization.

By formulating functions G and H appropriately, problem (**FOC**) can easily be written as a nonlinear program in standard form as in (**NLP**). In the next section, solution methods for these problems will be introduced and the algorithm used to obtain the numerical results of section V will be outlined.

NONLINEAR OPTIMIZATION

Problems as (**NLP**) are typically solved by deploying descent algorithms that generate a sequence of iterates

$$z_{k+1} = z_k + \alpha_k d_k, \qquad (14)$$

where α_k is the step length and d_k is the step direction. Starting from an initial guess z_0, the sequence is continued until the criteria for optimality, [20], [21], are satisfied and an optimal iterate z^* is identified. Typical examples for such descent algorithms are Newton-type methods, [9], or Gauss-Newton algorithms, [21], [22]. One of the most prominent techniques for providing d_k in (14) is *sequential quadratic programming* (SQP), where the step direction is obtained by sequentially solving an auxiliary quadratic problem, [9]. For the framework proposed here, such an SQP algorithm is deployed, see subsection A.

The determination of step length α_k is often neglected in so-called full step algorithms ($\alpha_k = 1$) that still deliver very practical results. Here, an algorithm based on the *Armijo* condition, [23], was implemented. The primary idea is to provide a step length such that the objective value decreases proportionally to the step length itself, and the steepness of the current iterates' gradient, [23], [24].

In the remainder of this section, some aspects of the framework for solving the (**NLP**) will be further outlined.

A. SQP Strategy

The principle idea of SQP algorithms is to obtain the step direction d_k in (14) as the solution of a quadratic auxiliary problem that originates from a second-order Taylor extension of the original problem (**NLP**). This can be understood by investigating the Lagrangian for (**NLP**), as

$$L(x,\lambda,\mu) = F(x) - \lambda^T G(x) - \mu^T H(x), \qquad (15)$$

where λ and μ are so-called Lagrange multipliers. Performing a second-order expansion of (15) for obtaining the next iterate yields:

$$\begin{aligned} L(x_{k+1},\lambda_{k+1},\mu_{k+1}) &= L(x_k,\lambda_k,\mu_k) \\ &\quad + \nabla L(x_k,\lambda_k,\mu_k)^T d \\ &\quad + \tfrac{1}{2}\, d^T \nabla^2 L(x_k,\lambda_k,\mu_k) d \\ &= F(x) - \lambda^T G(x) - \mu^T H(x) \\ &\quad + (\nabla F(x)^T - \nabla G(x)^T \lambda - \nabla H(x)^T \mu) d \\ &\quad + \tfrac{1}{2}\, d^T \nabla^2 L(x_k,\lambda_k,\mu_k) d \\ &= F(x_k) + \nabla F(x_k)^T d \\ &\quad + \lambda(\nabla G(x_k)^T d + G(x_k)) \\ &\quad + \mu(\nabla H(x_k)^T d + \mathrm{H}(x_k)) \\ &\quad + \tfrac{1}{2}\, d^T \nabla^2 L(x_k,\lambda_k,\mu_k) d\ , \end{aligned}$$

where $d = \Delta(x,\lambda,\mu)$ is the increment between the current iterate and the next one. This quadratic expansion is minimized in d, such that $F(x_k)$ may be omitted. Furthermore, the linearizations of equality and inequality conditions are formulated as conditions for the auxiliary quadratic problem (**QP**):

$$\begin{array}{lll} (\textbf{QP}) & \min & \nabla F_k^T d + \tfrac{1}{2}\, d^T \nabla^2 L_k d \\ & \text{subject to} & \nabla G_k^T d + G_k = 0 \\ & & \nabla H_k^T d + \mathrm{H}_k \geq 0 \end{array}$$

Here, the following abbreviations have been used:

$$F_k = F(x_k),\ G_k = G(x_k),\ H_k = H(x_k), \text{ and } L_k = L(x_k,\lambda_k,\mu_k).$$

Problem (**QP**) is a typical quadratic optimization problem with a quadratic objective and linear constraints. There is a multitude of algorithms in quadratic programming that address this type of problem. Typical algorithms include, but are not limited to, *active set methods*, *Cholesky* solvers, or *trust-region methods*, [9], [21]. Stock implementations in *Matlab®* include `QuadProg` or `fmincon` subroutines. Within the framework proposed here, `QuadProg` with an active set setting has delivered the most robust and efficient results.

In each iteration in (14), a new (**QP**) is formulated and solved for step direction $d_k = d_{QP,k}^*$, where $d_{QP,k}^*$ is the solution for (**QP$_k$**). Further information on quadratic programming can be found in [9], [21], and [25]. Note that in most practical applications, the second derivative, i.e. the Hessian matrix of the Lagrangian in (**QP**), cannot be computed directly. Therefore, this matrix is approximated by certain update formulas that guarantee positive definiteness and appropriate convergence rates, [9], [21]. In this framework, a BFGS update formula was implemented, [26]. In the next section, some further remarks on the setup of (**QP**) will be made.

B. Gradient Generation via Sensitivity ODE

When setting up the (**QP**) for a given iteration k, derivatives of the problem's defining functions F, G, and H, are required. Typically, derivatives within numerical algorithms are obtained through finite differences and/or automatic differentiation. Due to the multiple shooting approach and the resulting special structure of the problem function's derivatives, a different method may be applied, [27]. When inspecting the continuity conditions in G for a particular multiple shooting node i, i.e.

$$\begin{aligned} \partial G_i/\partial z &= \partial/\partial z\, (x_i(s_i,u_i;\tau_{i+1}) - s_{i+1}) \\ &= \partial x_i(s_i,u_i;\tau_{i+1})/\partial z - \partial s_{i+1}/\partial z, \end{aligned}$$

and by considering optimization variables as in (10), it is apparent that it holds:

$$\partial G_i/\partial z = [\ 0 \ \dots\ 0\ \partial x_i(\tau_{i+1})/\partial s_i\ \text{-}1\ 0\ \dots\ 0\ ].$$

The term $\partial x_i(\tau_{i+1})/\partial s_i$ is often referred to as *sensitivity term* because it represents the changes of the final integration value with respect to changes in s_i. The question remains of how to compute this value. Since this value is obtained through numerical integration, it cannot be known prior to the integration process. This is also the reason why numerical differentiation of the sensitivity term via finite differences is particularly inefficient as multiple integration processes are required.

Let $S_i = \partial x_i(\tau_{i+1})/\partial s_i$ define the sensitivity term at the i-th multiple shooting node. Instead of computing S_i directly, an ordinary differential equation may be defined. Deriving S_i with respect to the time yields

$$\begin{aligned} \dot{S}_i &= \mathrm{d}/\mathrm{d}t\ \partial x_i(\tau_{i+1})/\partial s_i \\ &= \partial/\partial s_i\ \mathrm{d}x_i(\tau_{i+1})/\mathrm{d}t \\ &= \partial/\partial s_i f(\tau_{i+1}, x_i,u) \\ &= \partial xi/\partial s_i\ \partial f(\tau_{i+1}, x_i,u)/\partial x_i \\ &= \partial f(\tau_{i+1}, x_i,u)/\partial x_i \cdot S_i. \end{aligned}$$

Here, it was used that $\dot{x} = f(t,x,u)$, as well as the chain rule. Therefore, a differential equation in S_i is defined that may be integrated along with the dynamic system itself. Using numerical integration to obtain S_i, $i = 0, \dots, N\text{-}1$, all required Jacobian entries for the continuity equality constraints may be occupied. Due to the sparsity of the resulting derivative matrix, the quadratic problem (**QP**) has quite formidable properties. More information on the structure of the Jacobian derivative matrix can be found in [27], [28].

The qualities of the defining matrices in (**QP**) can further improved by applying condensing techniques that are tailored towards multiple shooting, [8], [10], [28], and which yield dense quadratic problems.

REAL-TIME ALGORITHM

Traditionally, optimal control and optimization schemes have only provided a means of open-loop simulation to obtain an optimal, yet offline solution. The principle idea of an online closed-loop algorithm is to consecutively solve optimization problems that would provide updates to the control input, depending on any disturbances or perturbations that may impact the system.

Let $P_j = P(t_j,x_0(t_j))$ denote the optimal control (**OCP**) at the j-th instant of time t_j. A straightforward brute force approach is to solve problems P_j consecutively by using z_{j-1}^*, i.e. the previous problem's optimal solution, as initial guess. Even though this often yields quite convincing results, the disadvantage is that there still is a full-scale optimization problem to solve. Due to computational cost, the update rate may not be sufficiently high.

An improved idea was elaborated in [11], [12], where a quadratic expansion of the next problem is suggested. This leads to a *nonlinear model predictive control* (NMPC) (sources) scheme, where quadratic problems need to be solved in each update step. Therein, let z_0^* denote the optimal solution of the full-scale open-loop optimization problem. A similar procedure as above enables the formulation of a quadratic update problem that may be solved efficiently. Now let $L(x_j,\lambda_j,\mu_j)$ denote the Lagrangian of problem P_j at the j-th NMPC update step. Note that this is now not the expansion around the current optimization iteration but rather the expansion around the current instant of time. In order to formulate the quadratic update problem at the j-th instant of time, a quadratic expansion is performed around $L(x_j,\lambda_j,\mu_j)$. Therefore, it follows

$$\begin{aligned} L(x_{j+1},\lambda_{j+1},\mu_{j+1}) = {} & F(x_j) + \nabla F(x_j)^T d \\ & + \lambda(\nabla G(x_j)^T d + G(x_j)) \\ & + \mu(\nabla H(x_j)^T d + \mathrm{H}(x_j)) \\ & + \tfrac{1}{2}\, d^T \nabla^2 L(x_j,\lambda_j,\mu_j) d\,, \end{aligned}$$

from which a quadratic auxiliary problem may be derived as it was done above. The resulting problem can be written similarly to above as

$$\begin{aligned} (\mathbf{QP_{NMPC}}) \qquad \min \quad & \nabla F_j^T d + \tfrac{1}{2}\, d^T \nabla^2 L_j d \\ \text{subject to} \quad & \nabla G_j^T d + G_j = 0 \\ & \nabla H_j^T d + \mathrm{H}_j \geq 0, \end{aligned}$$

where the constraint evaluations at the current step are inserted. So instead of solving the full problem at the j-th

NMPC step, rather a quadratic approximation is solved for an update Δz_j to obtain

$$z_{j+1} = z_j + \Delta z_j.$$

The idea of solving these quadratic problems in the online procedure yields two significant advantages. First, the required time to solve (**QP**$_{\mathbf{NMPC}}$) is much less than to solve the full-scale optimization problem (**NLP**). Second, problems (**QP**$_{\mathbf{NMPC}}$) can actually be set up before the j-th NMPC step is reached. Therein, gradients from previous steps may be used within (**QP**$_{\mathbf{NMPC}}$) and may be updated less frequently (i.e. on a wider grid than the NMPC grid). As a consequence, the system state at the j-th NMPC step may be directly inserted as vectors G_j, H_j, so that the problem may be solved right away. A further modification was introduced in [13], where it is suggested to only perform a single update step for solving (**QP**$_{\mathbf{NMPC}}$). So instead of solving the auxiliary update problem for an optimal point, i.e. until convergence is achieved, a single optimization step is performed. This has proven to be a very efficient, yet sufficiently accurate method for obtaining updates within a closed-loop optimization scheme.

NUMERICAL RESULTS

Numerical simulations have been performed for a two-dimensional benchmark scenario as described in [16] and [28] and briefly sketched here. Since it is assumed that the interception take place out of the atmosphere there are no aerodynamic forces acting on either vehicle. Furthermore, it is assumed that the target is non-maneuverable and is on a strictly ballistic trajectory. The endgame process is initiated at interceptor handover, after the kill vehicle is separated from the last booster stage and has acquired to target. Both vehicles are following a collision course with a certain error so that a miss distance will occur if there is no endgame maneuvering performed. The kill vehicle is assumed to maneuver using four impulsive thrusters (two for each spatial dimension). Under these assumptions, the endgame interception problem is uniquely defined by both vehicles' position, and their relative velocity vector. Therefore, for each fixed instant of time, the endgame scenario may be transformed in a two-dimensional plane. The details of mathematical formulation can be found in [16] and [28].

The aim of numerical experiments was to investigate different guidance solutions for a severely-constrained maximum thrust of

$$T_{max} = 0.01 \text{ km/s}^2.$$

The initial conditions of coordinates and their velocities for both interceptor and target were defined as

$x_{ICP}(0) = 0$ km	$x_{TGT}(0) = 200$ km
$y_{ICP}(0) = 0$ km	$y_{TGT}(0) = 0$ km
$v_{x,ICP}(0) = 2$ km/s	$v_{x,TGT}(0) = -4$ km/s
$v_{y,ICP}(0) = 0$ km/s	$v_{y,TGT}(0) = 0.2$ km/s

During numerical experiments guidance solutions based on Proportional Navigation and optimization methods with different transcription strategies described in the paper were compared.

Numerical experiments have shown that the guidance methods based on PN has delivered solutions with successful hit only if there were no constraints imposed on a thrust maximum. As soon as the maximum thrust was severely constrained, PN no longer achieved a successful hit. The guidance algorithm was tested with a wide variety of gains and none have yield a successful interception.

Contrary to PN the optimization procedure described in this paper achieved a successful hit with all three transcription methods.

TABLE I. NUMERICAL COMPARISON OF THE SIMULATIONS PERFORMED FOR THIS PAPER.

Guidance	Miss Distance	Time	Magnitude
Pure ProNav	3418.0 m	-	.8199
True ProNav	4309.4 m	-	.2533
Full Transcription	3.1 m	32.55 s	.5575
Tangens Hyperbolicus	3.2 m	32.45 s	.6123
Gain Schedule	0.5 m	33.33 s	.2573

Table 1 shows some numerical results for the different guidance schemes investigated here. It is apparent that all three optimization solutions (full transcription, *Tangens Hyperbolicus* and Gain Scheduling) display a significantly better performance than both conventional PN solutions.

Especially the solution for the gain schedule is interesting, as the optimization procedure has computed the gain such that the thrust does not exceed its maximum value, while still ensuring a successful hit. Let us note that the numerical experiments also have shown that the proposed scheme is s real-time feasible, since the single iteration of solving (**QP**$_{\mathbf{NMPC}}$) is performed at each real-time instant, which is basically equivalent to matrix-vector product. Let us also note that thrust profiles computed by all procedures were different and the state constraint was satisfied.

CONCLUSION

In this paper, a closed-loop optimization scheme for exo-atmospheric missile interception was introduced. The emphasis here was on investigating different possibilities to transcribe the guidance problem as a nonlinear optimization problem. Therein, three different transcription methods have been introduced and their individual performance has been illustrated by numerical simulations. Also, comparison with conventional Proportional Navigation has shown that the approach via optimization has promising benefits. Especially the application of numerical optimization for gain scheduling transcription is a very interesting approach and should be elaborated further by investigating more elaborated scenarios and vehicle modeling.

Furthermore, a real-time optimization framework was proposed, based on sequential quadratic programming and a nonlinear model predictive control scheme, where auxiliary quadratic problems are solved to obtain updates on a given NMPC grid in order to account for disturbances and perturbations. More simulations are encouraged to investigate the performance of the proposed algorithm in difficult encounters and when applied to more complex models. The results in terms of real-time capability have been very promising but need further exploration with regard to stability and robustness.

REFERENCES

[1] Siouris, G. M., *Missile Guidance and Control Systems*, Springer, 2004.

[2] Zarchan, P., *Tactical and Strategic Missile Guidance*, Progress in Astronautics and Aeronautics, American Institute of Aeronautics and Astronautics, 2007.

[3] Karelahti, J., Virtanen, K., and Raivio, T., "Near-Optimal Missile Avoidance Trajectories via Receding Horizon Control," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, September-October 2007, pp. 1287-1298.

[4] Trélat, E., "Optimal Control of a Space Shuttle, and Numerical Simulations," *Proceedings of the Fourth International Conference on Dynamical Systems and Differential Equations*, May 2002.

[5] Pontani, M. and Conway, B. A., "Optimal Trajectories of Anti-Satellite Missiles," *Communications to SIMAI Congress*, Vol. 3, No. 265, 2009, pp. 12pp.

[6] Bock, H. G. and Plitt, K., "A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems," *Proceedings to the IFAC 9th World Congress*, 1984, pp. 242-247.

[7] Krämer-Eis, P., *Ein Mehrzielverfahren zur numerischen Berechnung Optimaler Feedback-Steuerung bei Beschränkten Nichtlinearen Steuerungsproblemen*, Ph.D. thesis, Universität Bonn, 1985.

[8] Leinweber, D. B., *Efficient Reduced SQP Methods for the Optimization of Chemical Processes Described by Large Sparse DAE Models*, Ph.D. thesis, Ruprecht-Karls-Universität Heidelberg, 1998.

[9] Nocedal, J. A. and Wright, S. J., *Numerical Optimization*, Springer Series in Operations Research, Springer, 2nd ed., 1999.

[10] Leinweber, D. B., Bauer, I., Bock, H. G., and Schlöder, J. P., „An Efficient Multiple Shooting Based SQP Strategy for Large-Scale Dynamic Process Optimization,“ 2002.

[11] Diehl, M., Real-Time Optimization for Large Scale Nonlinear Processes, Ph.D. thesis, Ruprecht-Karl-Universität Heidelberg, 2001.

[12] Kirches, C., Wirsching, L., Bock, H. G., and Schlöder, J. P., „Efficient Direct Multiple Shooting for Nonlinear Model Predictive Control on Long Horizons,“ *Journal of Process Control*, July 2011.

[13] Diehl, M., Bock, H. G., and Schlöder, J. P., „A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control,“ *SIAM Journal of Control and Optimization*, Vol. 43, No. 5, March 2005, pp. 1714-1736.

[14] Pontryagin, L., Boltyansky, V., and Gamkrelidze, R., *Mathematische Theorie Optimaler Prozesse*, Oldenburg, 1964.

[15] Gregory, J. and Lin, C., *Constrained Optimization in the Calculus of Variations and Optimal Control Theory*, Van Nostrand Reinhold, 1992.

[16] Walter, L., Schlöffel, G., and Holzapfel, F., „Optimal Control Framework for Impulsive Missile Interception Guidance,*“ AIAA Guidance, Navigation, and Control Conference*, Boston, August 2013.

[17] Sager, S., *Numerical Methods for Mixed-Integer Optimal Control Problems*, Ph.D. thesis, Interdisciplinary Center for Scientific Computing, Ruprecht-Karl-Universität Heidelberg, 2005.

[18] Hapke, B., Theory *of Reflectance and Emittance Spectroscop*y, Cambridge, 2nd ed., 2012.

[19] Shukla, U. and Mahapatra, P., "The Proportional Navigation Dilemma – Pure or True?" *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 26, No. 2, March 1990, pp. 382-392.

[20] Kuhn, H. W. and Tucker, A. W., "Nonlinear Programming," *Proceedings of the Second Berkeley Symposium on Matematical Statistics and Probability*, University of California Press, pp. 481-492, 1951.

[21] Fletcher, R., *Practical Methods of Optimization*, J. Wiley & Sons, 2nd ed., 1987.

[22] Binder, T. and Kostina, E., "Approximate Guass-Newton Method for Robust Parameter Estimation,“ *Meeting of the International Association of Applied Mathematics and Mechanics* (GAMM), Karlsruhe 2010.

[23] Armijo, L., "Minimization of Functions having Lipschitz Continuous First Partial Derivatives," *Pacific Journal of Mathematics*, Vol. 16, No. 1, pp. 1-3, 1966.

[24] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., *Some Theoretical Properties of an Augmented Lagrangian Merit Function*, Technical Report, September 1986.

[25] Gould, N. I., Hribar, M. E., and Nocedal, J., "On the Solution of Equality Constrained Quadratic Programming Problems Arising in Optimization," *SIAM Journal of Scientific Computing*, pp. 1376-1395, 2001.

[26] Broyden, C. G., "The Convergence of a Class of Double-Rank Minimization Algorithms," *Journal of the Institute of Mathematics and Its Applications*, Vol. 6, pp. 76-90, 1970.

[27] Albersmeyer, J., *Adjoint-based Algorithms and Numerical Methods for Sensitivity Generation and Optimization of Large Scale Dynamic Systems*, Ph.D. thesis, Karl-Ruprecht-Universität Heidelberg, 2010.

[28] Walter, L., Schlöffel, G., Holzapfel, F., and Kostina, E., „Multiple Shooting Condensing for Online Gain Scheduling in Interceptor Guidance,“ *SciTech/AIAA Guidance, Navigation, and Control Conference*, National Harbor, MD, 2014.

[29] Reisner, D, Shima, T., "Optima l Guidance-to-Collision for an Accelerating Exo-Atmospheric Interceptor Missile," *Journal Guidance, Control, and Dynamics*, Vol.36(6), pp. 1695-1708, 2013.

[30] Shima, T., and Golan, O.M. "Exo-Atmospheric Guidance of an Accelerating Interceptor Missile", *Journal of the Franklin Institute*, Vol. 349(2), pp. 622-637 2012.