JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 32, No. 1, January–February 2009

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes should not exceed 2500 words (where a figure or table counts as 200 words). Following informal review by the Editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Trajectory Shaping of Projectiles Through Cross-Entropy-Minimization-Based Search

Nicolas Léchevin*
Numerica Technologies Inc., Québec, Québec G3J 1X5, Canada
and
Franklin Wong† and Camille Alain Rabbath‡
Defence Research and Development Canada—Valcartier, Québec, Québec G3J 1X5, Canada

DOI: 10.2514/1.39231

I. Introduction

PRECISION weapon-effect delivery aims at reducing collateral damage in terms of both infrastructure and loss of life and at providing timely and efficient debilitation of the intended target by the use of a small number of rounds that are quickly and accurately placed on target. To be effective in an operational theatre, synthesis of the optimal trajectory needed to guide the precision weapon must be promptly achieved. In the context of artillery precision-guided munitions, compliance with time-critical operating conditions and with time-varying, as well as partially known, disturbances such as atmospheric turbulence calls for a trajectory-shaping solution that integrates last-minute information just before firing the round. The challenges that arise in deriving a fast trajectory-shaping solution increase with the concurrence of meeting control constraints such as actuator saturation and with dealing with a dynamic system that is highly nonlinear.

The trajectory-shaping solution proposed in this Note consists of providing, before launch, a lateral-acceleration time or waypoint trajectory that the controlled projectile robustly tracks by means of an appropriate guidance law and autopilot. The trajectory-shaping functionality can be interpreted as a constrained feedforward control law that drives, under nominal operating conditions, the projectile to a prescribed target set. Such a set is defined by a number of constraints that involve the projectile's range, terminal speed, and terminal flight-path angle. In such context, the guidance law is assumed to provide, during the course of the flight, some robustness when operating conditions differ from the nominal conditions.

Received 19 June 2008; revision received 31 July 2008; accepted for publication 7 August 2008.

*Research Engineer, 2459 Pie-XI North; Nicolas.Lechevin.Numerica@drdc-rddc.gc.ca.
†Defence Scientist, 2459 Pie-XI North; Franklin.Wong@drdc-rddc.gc.ca.
‡Defence Scientist, 2459 Pie-XI North; Camille-Alain.Rabbath@drdc-rddc.gc.ca.

Several approaches can tackle this constrained-type problem. Backward-reachable set theory [1] and the related invariant set theory are among the interesting approaches to deal with trajectory shaping. Indeed, by deriving an invariant set from the prescribed target set, every trajectory that starts in the invariant set, if it exists, is guaranteed to be captured by the target set [1]. Application of level-set theory [1–3] and viability theory [4,5] allows derivation of an invariant set, although at the expense of fast computation because it typically requires solving time-dependent Hamilton–Jacobi–Isaacs partial differential equations. The latter is therefore not well tailored to fast trajectory shaping. Differential flatness is another concept that can be used to derive commands that enable a dynamic system to follow a prescribed trajectory that aims to reach a target set [6]. To do so, a flat output, possibly a nonlinear function of the projectile's position, speed, and flight-path angle, is found so that the lateral-acceleration command can be expressed as a function of the flat output and its time derivatives. This approach may not be appropriate when the trajectory-shaping problem must account for control saturation, because it leads to a nonlinear nonconvex optimization problem [7]. Nongradient-based search, such as the genetic algorithm technique, has been used in trajectory shaping to compute lookup-table-based line-of-sight bias that is included in a proportional navigation guidance law to enable interception of a constant-velocity target with specific terminal constraints [8]. Such a search technique is proven efficient when an existing homing guidance law can be modified to comply with the prescribed crossing angle at intercept or with the maximized closing velocity.

In this Note, we propose a trajectory shaping over the entire course of the flight obtained by means of a cross-entropy-minimization-based search (CEMBS). To deal with complex dynamic systems and control constraints, an iterative simulation-based approach is adopted. The cross-entropy method was originally developed for rare-event simulations and for complex combinatorial optimization problem-solving such as the traveling salesman problem (TSP) [9]. The main contribution of this Note lies in the generation of a finite-in-time sequence of lateral-acceleration or waypoint commands obtained with a nonhomogeneous Markov chain building upon the concept of CEMBS. These commands are then used to yield a sample set from which a batch of simulations of the guided projectile is performed. Analysis of the resulting statistics drives new iterations by updating the Markov chain's transition matrices. The trajectory-shaping solution is found when the metric defined with respect to the target set is minimized. The proposed approach is compared with the results obtained through brute-force search and numerical simulations. It is shown that a near-optimal trajectory-shaping solution can be computed in a relatively short period of time for a given target set, which is the key practical benefit of the proposed trajectory shaping.

II. Trajectory Shaping of Guided Projectile

A. Context and Problem Formulation

We consider a guided projectile for which the nonlinear model is given as

$$\dot{X} = f(X, u, w), \qquad Y = [\,x \quad y \quad v \quad \gamma\,]^T = g(X) \tag{1}$$

where state $X = [X_p^T, X_c^T]^T$ results from nonlinear flight dynamics in a closed loop with controllers such as a guidance law, an autopilot, and

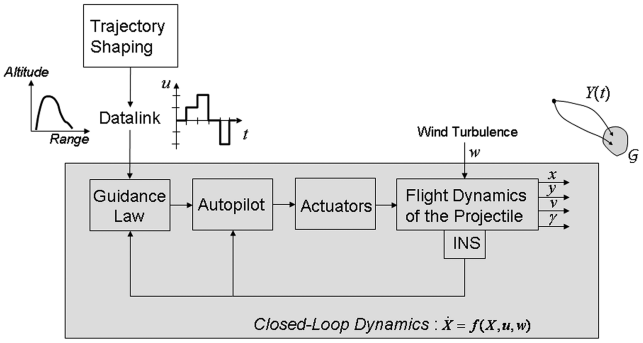


Fig. 1 Trajectory-shaping commands for the closed-loop dynamics (INS denotes the inertial navigation system).

servomechanisms (see Fig. 1); X_p denotes the state-space vector of the projectile's dynamics; X_c corresponds to the state-space vector of various control laws; w represents exogenous disturbances such as wind turbulence and the output signals x, y, v, and γ stand for the range, altitude, speed, and flight-path angle of the projectile, respectively. Equation (1) not only includes algebraic nonlinearities that are typically related to the flight dynamics, but also accounts for hard nonlinearities such as actuator saturation.

For the purpose of this work, which was focused on the development of the CEMBS method for trajectory shaping, a 3-degree-of-freedom model was used to represent the flight dynamics of the projectile. As such, the problem was formulated in terms of the lateral-acceleration command sets that are applied directly to the closed-loop system shown in Fig. 1. Extension of the method to a 6-degree-of-freedom model and waypoint command sets can be readily accomplished without loss of generality.

To obtain candidate trajectories, the following two-step operation is carried out as illustrated in Fig. 2. First, the autopilot aims at despinning the projectile behavior during the ascending phase of the ballistic motion. The trajectory-shaping output u remains zero. Then the finite-state trajectory shaping is applied from the apogee at t_a to the impact ($y = 0$) at t_f. In the sequel, t_f^* denotes the initial estimate of t_f and is computed with a simplified trajectory model with no commanded accelerations.

Definition 1 (target set). The set of constraints that the projectile must satisfy at the end of flight is defined by

$$\mathcal{G} = \{(x, y, v, \gamma) \in \mathbb{R}^3 \times [-\pi, 0] / x_{\min} \le x \le x_{\max}, v_{\min} \le v \le v_{\max}, \gamma_{\min} \le \gamma \le \gamma_{\max}, y = 0\} \tag{2}$$

Definition 2 (control constraints). The finite-state trajectory shaping provides a piecewise constant signal $u(t) = u_k \in \mathcal{U}$, for all $t \in [t_{\text{sw},k}, t_{\text{sw},k+1})$, where

$$\mathcal{U} = \{-U, \ -(n-1)U/n, \ldots, -U/n, \ 0, \times \ U/n, \ldots, (n-1)U/n, \ U\}$$

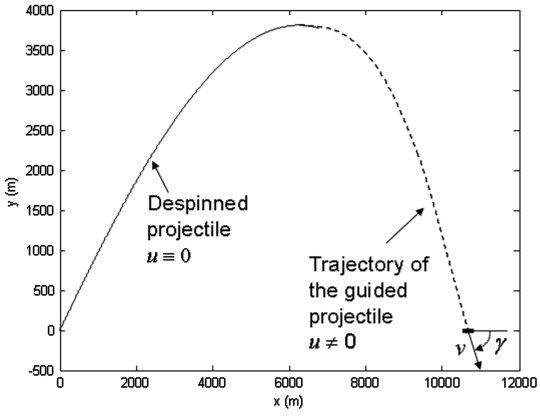


Fig. 2 Phases of the guided projectile.

where $U \in \mathbb{R}$, $n \in \mathbb{N}$, and $t_{\text{sw},k}$, defined for all $k = 1, \ldots, p$, are time instants at which switches may occur. Time instants $t_{\text{sw},k}$ are fixed before flight and are uniformly distributed over $[t_a, \eta t_f^*]$, where $t_{\text{sw},1} = t_a$, $t_{\text{sw},p} = \eta t_f^*$, and $\eta \ge 1$; that is,

$$t_{\text{sw},k} - t_{\text{sw},k-1} = \frac{t_{\text{sw},p} - t_a}{p - 1}$$

for all $k = 2, \ldots, p$. Let z denote the sequence $\{u_k, 1 \le k \le p\}$.

Assumption 1. The guidance law and the autopilots are able to robustly track the command provided by the trajectory shaping.

Control Objective. From Definition 1, Definition 2, and Assumption 1, find the sequence of control signals z that minimizes $d_{\mathcal{G}}(x_f, v_f, \gamma_f; z)$, where

$$d_{\mathcal{G}}(x_f, v_f, \gamma_f; z) = d_{\mathcal{G}}(x_f; z) + d_{\mathcal{G}}(v_f; z) + d_{\mathcal{G}}(\gamma_f; z)$$

and

$$d_{\mathcal{G}}(\chi_f; z) = \begin{cases} 0 & \text{if } \chi_f \in [\chi_{\min} \ \ \chi_{\max}] \\ \min(|\chi_{\min} - \chi_f| / \chi_{\min}, |\chi_{\max} - \chi_f| / \chi_{\max}) & \text{otherwise} \end{cases} \tag{3}$$

where χ_f denotes either x_f, v_f, or γ_f; subscript f indicates that variables are valued at t_f defined by $y(t_f) = 0$; and argument z in $d_{\mathcal{G}}$ specifies that the terminal state is obtained by applying the sequence of control signals z to the system in Eq. (1).

Remark. Parameter η in Definition 2 is selected such that $t_{\text{sw},p} = \eta t_f^*$ bounds the time of flight entailed by every possible sequence z; η has to be selected carefully because switching time instants are uniformly distributed over $[t_a, \eta t_f^*]$. Too large a parameter η would result in $t_{\text{sw},k'} > t_f$ and thus unused entries $\{u_{k'}, \ldots, u_p\}$ of z, as illustrated in Fig. 3.

Fixing $t_{\text{sw},k}$ and forcing z to evolve in the finite set $\mathcal{U}^p$ enables satisfying the control objective by means of a combinatorial optimization, where one seeks, as explained in the next section, to minimize a metric defined with respect to $\mathcal{G}$ by deriving an appropriate sequence z.

B. Iterative Simulation-Based Approach of Trajectory Shaping

The rationale that lies behind the proposed iterative simulation-based approach is depicted in Fig. 4. First, a sample of command sequences Z_1^N is drawn from an independent, identically distributed, random sequence Z_i associated with $z \in \mathcal{U}^p$. Second, a batch of simulations is carried out for every sequence Z_i. Third, the results of the simulations are analyzed to drive the selection, at the next iteration, of a new sample set Z_1^N that tends to decrease $d_{\mathcal{G}}(x_f, v_f, \gamma_f; z)$.

In the next section, we derive the mechanism CEMBS from which Z_1^N is obtained. One naïve approach would consist of implementing a brute-force search in lieu of CEMBS. Such a technique becomes prohibitive as p or n increase, because each $z \in \mathcal{U}^p$ corresponds to one of $(2n+1)^p$ permutations with repetition. Yet, brute-force search has been implemented in this study as a means to evaluate the performance of CEMBS. CEMBS, a stochastic search method, is proposed as a means to quickly find a near-optimal solution to the trajectory-shaping problem. It is to be noted, however, that the time required to find the solution depends on the computational load

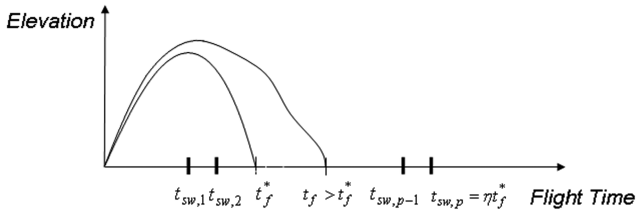


Fig. 3 Selection of η and $t_{\text{sw},k}$.

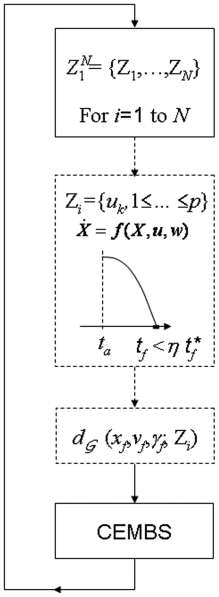


Fig. 4 Simulation-based trajectory-shaping design.

associated with solving Eq. (1) and on the size of the sample set Z_1^N. As our ultimate aim is to use a detailed model of the guided projectile, the efficiency of CEMBS is thus critical to achieve fast computation of near-optimal z.

III. Search Based on Cross-Entropy Minimization

A. Review of the Cross-Entropy Method

The algorithm requires finding a probability distribution that aims at seeking a near-optimal solution z^+ in the sense that

$$\{z^+ \in \mathcal{U}^p, d_{\mathcal{G}}(x_f, v_f, \gamma_f; z^+) < \lambda^+\}$$

is a rare event for some level λ^+ typically close to zero; that is,

$$l = Pr_h(d_{\mathcal{G}}(x_f, y_f, \gamma_f; z^+) < \lambda^+) = E_h[\mathcal{I}_{\{d_{\mathcal{G}}(x_f, y_f, \gamma_f; z^+) < \lambda^+\}}] \quad (4)$$

is very small. In Eq. (4), $h(Z, P)$ denotes a probability distribution of random variable Z that is parametrized by some stochastic matrix P, which is derived in the next section.

Searching (z^+, λ^+) constitutes the associated stochastic problem (ASP) to that defined in the control objective. This problem is related to rare-event simulation because ASP requires estimating l, which is obtained by means of an iterative algorithm that depends on λ and P [9]. The guideline that leads to this algorithm is briefly recalled. Instrumental to simulation of rare events is the use of the importance sampling, which provides an unbiased estimator of l:

$$\hat{l} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{I}_{\{d_{\mathcal{G}}(x_f, y_f, \gamma_f; Z_i) < \lambda\}} W(Z_i) \quad (5)$$

where $W(Z_i) = h(Z, P)/h'(Z)$ is the likelihood ratio and $(Z_1, \ldots, Z_n)$ is a random sample drawn from h'. By using Eq. (5), one seeks to compute $\hat{l}$ with a sample set that would be smaller than that used by direct application of Monte Carlo simulation to Eq. (4). The distribution $h'(Z)$ that allows finding the best estimator of l is [9]

$$h^{*\prime}(Z) = \mathcal{I}_{\{d_{\mathcal{G}}(x_f, y_f, \gamma_f; Z_i) < \lambda\}} h(Z, P)/l$$

Because l is unknown, computation of $\hat{l}$ can be achieved if one is able to find a family of probability distributions $h'(Z, P)$, along with the reference parameter P, such that the cross entropy between $h'(Z, P)$ and $h^{*\prime}(Z)$ is minimized, yielding the following CEMBS algorithm [9]:

1) Initialize P to P_0. Select $\rho \in (0, 1)$, $\alpha \in (0, 1)$, and $\kappa \in \mathbb{N}$.

2) For the rth iteration, proceed as follows as long as the stopping condition in step 3 is not met:

a) Draw N samples $\{Z_1, \ldots, Z_N\} = Z_1^N$ according to $h'(Z, P_{r-1})$.

b) Select the $\lceil \rho N \rceil$ that best draws $\{Z'_1, \ldots, Z'_{\lceil \rho N \rceil}\}$ as defined by $d_{\mathcal{G}}(x_f, v_f, \gamma_f; z)$, which is obtained from Fig. 4 by simulation of Eq. (1) over $[t_a, \eta t_f^*]$. Let λ_r be the $\lceil \rho N \rceil$th-order statistics of the sequence

$$\{d_{\mathcal{G}}(x_f, v_f, \gamma_f; Z'_i), i = 1, \ldots, \lceil \rho N \rceil\}$$

c) Update, from $\{Z'_1, \ldots, Z'_{\lceil \rho N \rceil}\}$, the reference parameter Q that minimizes the cross entropy, which is obtained by maximizing

$$\sum_{i=1}^{\lceil \rho N \rceil} \ell_n\, h'(Z'_i, Q)$$

d) Update P_r as follows: $P_r = \alpha Q + (1 - \alpha) P_{r-1}$

3) For the stopping condition, $\lambda_r = \cdots = \lambda_{r-\kappa+1}$ or there exists $i = 1, \ldots, N$ such that $d_{\mathcal{G}}(x_f, v_f, \gamma_f; Z_i) = 0$.

B. Proposed Reference Parameter

The cross-entropy method has been successfully applied to various combinatorial optimization problems, among which the TSP is solved by selecting $h'(Z, P)$ as a function of a homogeneous Markov chain that expresses the probability of being at a given site at t_k conditioned to the location of the salesman at t_{k-1} [9]. We consider here a nonhomogeneous Markov chain [10], defined for all $k \in \{2, \ldots, p\}$ by

$$P_k = [p_{k,ij}]_{1 \le i,j \le 2n+1}, \qquad p_{k,ij} = Pr(U_k = u_k | U_{k-1} = u_{k-1}) \quad (6)$$

which models the transition from $u_{k-1} \in \mathcal{U}$ at $t_{\mathrm{sw},k-1}$ to $u_k \in \mathcal{U}$ at $t_{\mathrm{sw},k}$, as illustrated in Fig. 5. Notice that the transition matrix in Eq. (6) depends on k. A homogeneous Markov chain may be too restrictive when applied to a dynamic system with terminal state constraints, because a k-independent transition matrix would force every transition to be governed by the same probabilities.

The sample Z_i in Fig. 4 is thus selected at each iteration of the cross-entropy algorithm from the set of $p - 1$ matrices $\{P_2, \ldots, P_p\}$. The first entry of Z_i is randomly selected from $\mathcal{U}$ with a uniform distribution.

The probability of generation of $z \in \mathcal{U}^p$ is given by

$$h'(z, P) = \prod_{k=2}^{p} \prod_{i=1}^{2n+1} \prod_{j=1}^{2n+1} (p_{k,ij})^{I_{\{z \in \mathcal{Z}_{ij}(k)\}}} \quad (7)$$

where $\mathcal{Z}_{ij}(k)$ denotes the set of z such that $u_{k-1} = i$ and $u_k = j$.

Step 2c of CEMBS involves minimizing the cross entropy between $h'(Z, P)$ and $h^{*\prime}(Z)$. This minimization, along with the use of Lagrangian multipliers $u_{i,k}$, which ensure that the ith row of $p_{k,ij}$ sums to one, is equivalent to the following maximization over P_k [9]:

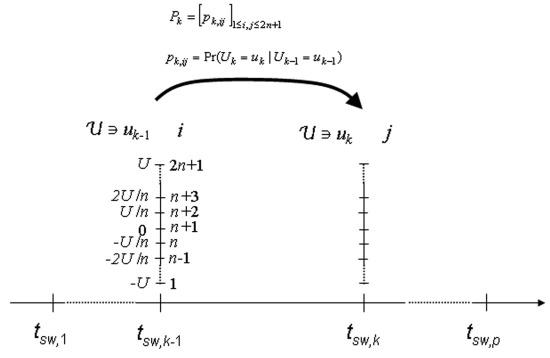


Fig. 5 Computation of z by means of probabilistic transition from u_{k-1} to u_k, $k = 1, \ldots, p$.

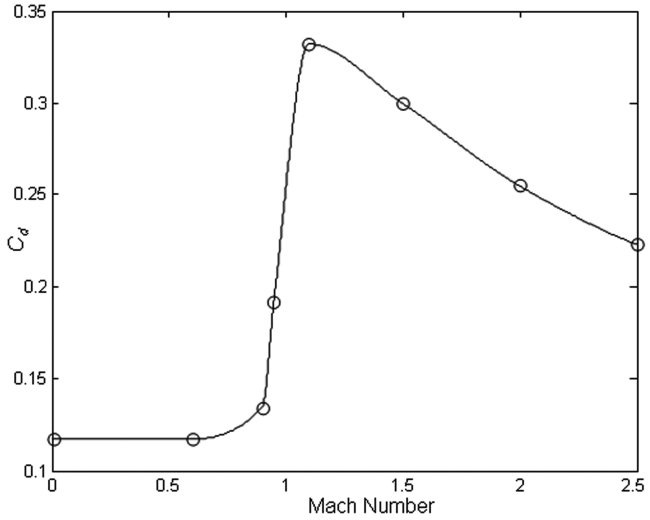


Fig. 6 Drag coefficient model.

$$\max_{P_k}\min_{u_{i,k}}\left[E_{P_k}I_{\{d_{\mathcal{G}}(x_f,y_f,\gamma_f;Z)<\lambda\}}\,\ell n\, h'(Z,P) + \sum_{k=2}^{p}\sum_{i=1}^{2n+1}u_{i,k}\left(\sum_{j=1}^{2n+1}p_{k,ij}-1\right)\right] \tag{8}$$

Expressing $\ell n\, h'(Z,P)$ as

$$\ell n\, h'(z,P)=\sum_{k=2}^{p}\sum_{i=1}^{2n+1}\sum_{j=1}^{2n+1}I_{\{z\in\mathcal{Z}_{ij}(k)\}}\,\ell n\, p_{k,ij} \tag{9}$$

and differentiating Eq. (8) with respect to $p_{k,ij}$ yields

$$\frac{E_{P_k}I_{\{d_{\mathcal{G}}(x_f,y_f,\gamma_f;Z)<\lambda\}}I_{\{Z\in\mathcal{Z}_{ij}(k)\}}}{p_{k,ij}}+u_{i,k}=0 \tag{10}$$

Summing Eq. (10) over $j=1,\ldots,2n+1$, where

$$\sum_{j=1}^{2n+1}p_{k,ij}=1$$

and letting $\mathcal{Z}_i(k)$ denote the set of z such that $u_{k-1}=i$, one obtains

$$p_{k,ij}=\frac{E_{P_k}I_{\{d_{\mathcal{G}}(x_f,y_f,\gamma_f;Z)<\lambda\}}I_{\{Z\in\mathcal{Z}_{ij}(k)\}}}{E_{P_k}I_{\{d_{\mathcal{G}}(x_f,y_f,\gamma_f;Z)<\lambda\}}I_{\{Z\in\mathcal{Z}_i(k)\}}} \tag{11}$$

The estimator of Eq. (11), in which expectations are replaced by averaging over the sample $Z_1^N=\{Z_1,\ldots,Z_N\}$, is expressed as

$$\hat{p}_{\,k,ij}=\frac{\sum_{q=1}^{N}I_{\{d_{\mathcal{G}}(x_f,y_f,\gamma_f;Z_q)<\lambda\}}I_{\{Z_q\in\mathcal{Z}_{ij}(k)\}}}{\sum_{q=1}^{N}I_{\{d_{\mathcal{G}}(x_f,y_f,\gamma_f;Z_q)<\lambda\}}I_{\{Z_q\in\mathcal{Z}_i(k)\}}} \tag{12}$$

The reference parameter Q in step 2c of CEMBS is thus equal to $[\hat{P}_2,\ldots,\hat{P}_p]$. The ijth entry of $\hat{P}_k$ is given by $\hat{p}_{k,ij}$ in Eq. (12), where λ is replaced by λ_r obtained in step 2b.

Determining the estimator in Eq. (12) involves the computation of $(p-1)(2n+1)^2$ parameters. Thus, the number of levels in $\mathcal{U}$ and of switching times $t_{\mathrm{sw},k}$, which can be interpreted as the quantization and the time discretization of u_k with respect to a continuous-time trajectory-shaping signal, respectively, influence the computation time required to find a near-optimal sequence z. Although one may be tempted to increase p and n so that z approaches an analog signal, computational limits constrain the resolution of z. Furthermore, the dynamics of the projectile acts as a filter, which actually bounds, from above, practical values of p.

IV. Simulations

Applying Newton's law of motion to a projectile and assuming that the drag force predominates over the lift force yields the following model of a despinned projectile:

$$\dot{v}_x=-C_d^*(v,y)\hat{v}(v_x-w_x)-\frac{uv_y}{mv},$$
$$\dot{v}_y=-C_d^*(v,y)\hat{v}(v_y-w_y)+\frac{uv_y}{mv}-g,\qquad \dot{x}=v_x,\qquad \dot{y}=v_y \tag{13}$$

where

$$v=\sqrt{v_x^2+v_y^2},\qquad \hat{v}=\sqrt{(v_x-w_x)^2+(v_y-w_y)^2},$$
$$C_d^*(v,y)=\frac{\rho SC_d(M)}{2m},\qquad M=\frac{v}{v_s(y)}$$

where $v_s(y)$, m, S, ρ, and $\tilde{u}$ stand for the sound speed, which is a nonlinear function of the altitude, the mass of the projectile, the reference area, the air density, and the control input signal, respectively. A typical drag coefficient $C_d(M)$ profile is shown in Fig. 6.

To represent neglected dynamics, the input signal $\tilde{u}$ in Eq. (13) is expressed as the filtered and delayed command u:

$$\tilde{u}(s)=\frac{\omega^2e^{\tau_d s}}{s^2+(2\xi/\omega)s+\omega^2}\,\mathrm{sat}_{10N}(u(s)) \tag{14}$$

where function $\mathrm{sat}_{10N}(u)$ saturates the lateral acceleration at $\pm 10N$.

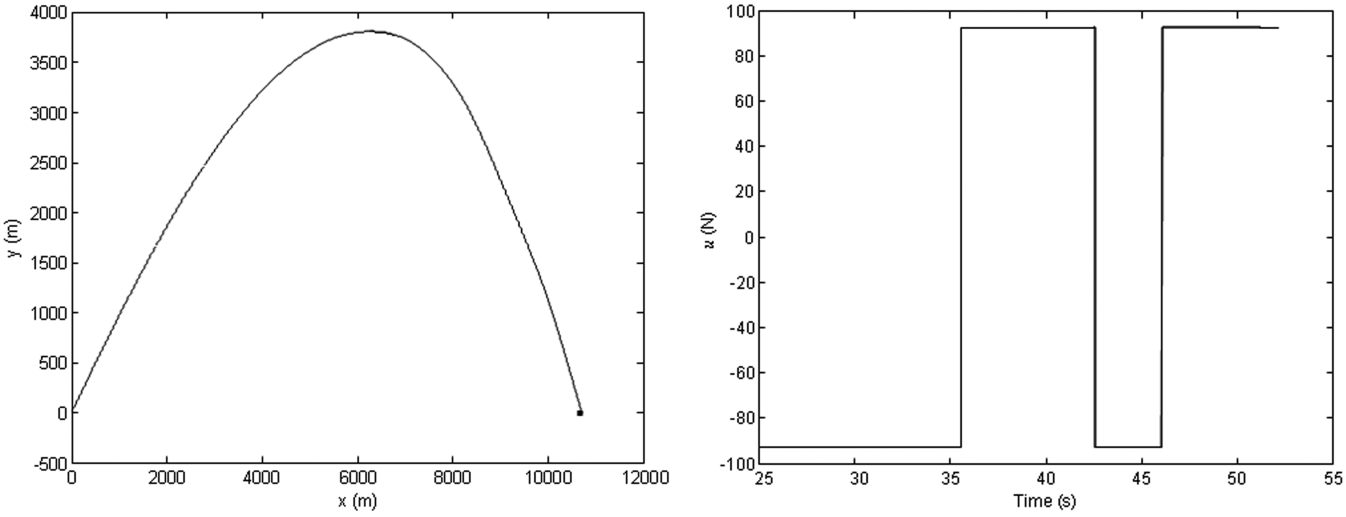
The following parameter values are used: $m=18.5$ kg, $S=18.8\times10^{-3}$ m^2, and $g=9.81$ m/s^2. A ballistic trajectory is obtained with the following initial conditions: $x(0)=0$ m, $y(0)=0$ m, $v(0)=625$ m/s, and $\gamma(0)=\pi/4$ rad. The apogee [6271, 3804] m is reached at 24.7 s. The range of the ballistic course is 11,276 m. The impact of the projectile on the ground occurs at 54.4 s with a terminal velocity of $v^*(t_f)=268$ m/s and terminal flight-path angle of -1.05 rad (-60.2 deg).

Two sets of constraints are considered. The first set, C1, is defined by $x_{\min}=10,599$ m, $x_{\max}=10,712$, $v_{\min}=241.6$ m/s, $v_{\max}=v^*(t_f)$, $\gamma_{\min}=-1.15$ rad, and $\gamma_{\max}=-0.94$; The second set, C2, corresponds to C1 with $v_{\max}=1.1v^*(t_f)$. The algorithm proposed in Sec. III is implemented with $n=1$, $p=10$, $U=92.5$ N, $\eta=1.1$, $\alpha=0.08$, $\rho=0.1$, $\kappa=5$, and $N=p^2$.

The stochasticity of the cross-entropy method may entail fluctuating performances from one simulation to another in terms of computation time and distance $d_{\mathcal{G}}(x_f,v_f,\gamma_f;z)$ at impact. One hundred simulations were run on a 1 GB, 2.4-GHz Xeon computer with MATLAB/Simulink-compiled code [11] to analyze the performance of the proposed approach. Table 1 shows the results obtained with CEMBS over $\mathcal{U}^p$. CEMBS is compared with minimum seeking of $d_{\mathcal{G}}(x_f,v_f,\gamma_f;z)$ obtained by means of brute-force search (BFS). The latter thus requires, at most, $3^{10}=59,049$ executions of the compiled model. Both algorithms will stop if $(x_f,v_f,\gamma_f;z)\in\mathcal{G}$ is satisfied. Otherwise, CEMBS stops when step 3 is satisfied and BFS stops when the 59,049 iterations have been performed. The symbol

Table 1 Performance of the cross-entropy-based trajectory shaping

Set of constraints	C1		C2	
Type of search	CEMBS	BFS	CEMBS	BFS
Average best relative error with respect to $\mathcal{G}$ at t_f	0.97% $[d_{\mathcal{G}}(v_f;z)]$	0.87% $[d_{\mathcal{G}}(v_f;z)]$	0%	0%
Average computation time (speedup ratio)	4 min 35 s (60)	4 h 30 min	27 s (29)	13 min 22 s
EF$[(x_f,v_f,\gamma_f)\in\mathcal{G}]$	0% (EF$[(x_f,\gamma_f)\in\mathcal{G}\backslash v]=100\%$)	N/A	100%	N/A



a) b)

Fig. 7 Trajectory shaping with set of constraints C2: a) trajectory of the guided projectile and b) sequence of command z.

EF in Table 1 denotes empirical frequency. $\mathcal{G}\backslash v$ denotes the set of (x, γ) that satisfy constraints $x_{\min} \leq x \leq x_{\max}$ and $\gamma_{\min} \leq \gamma \leq \gamma_{\max}$.

No sequence z, as obtained with BFS, leads to $(x_f, v_f, \gamma_f) \in \mathcal{G}$ when the set of constraints C1 is considered. In particular, the terminal velocity constraint is not satisfied, whereas $d_{\mathcal{G}}(x_f; z)$ and $d_{\mathcal{G}}(\gamma_f; z)$ are both equal to zero; that is,

$$\min_{z(\mathrm{BFS})} d_{\mathcal{G}}(x_f, v_f, \gamma_f; Z) = \min_{z(\mathrm{BFS})} d_{\mathcal{G}}(v_f; z)$$

where z(BFS) means that z is obtained with BFS. The fact that the best solution obtained with BFS does not reach $\mathcal{G}$, but nevertheless reaches $\mathcal{G}\backslash v$, allows a relative comparison between CEMBS and BFS. Indeed, the stochastic nature of the cross-entropy-based search incurs near-optimality rather than optimality, because the average relative error only increases to 11.5% when substituting CEMBS for BFS, whereas the terminal distance to $\mathcal{G}$ obtained with BFS is very small (0.87%). Note that the maximum deviations (not reported in Table 1) obtained with BFS are

$$\max_{z(\mathrm{BFS})} d\mathcal{G}(x_f, v_f, \gamma_f; z) = 49\%$$

and

$$\max_{z(\mathrm{BFS})} d\mathcal{G}(v_f; z) = 2.5\%$$

which, in the latter case, represents an increase of 287% with respect to $\min_{z(\mathrm{BFS})} d\mathcal{G}(x_f, v_f, \gamma_f; z)$.

Better performance could be obtained with CEMBS by enlarging the sample at the expense of fast computation. If some amount of optimality is sacrificed, the solution time can be sped up by a ratio of 60, as shown by the results obtained with C1. The set of constraints C2 leads to $\min_{z(\mathrm{BFS})} d\mathcal{G}(x_f, v_f, \gamma_f; z) = 0$, which is met with a frequency of 100% by BFS and CEMBS, meaning that the target set is always reached despite near-optimality of CEMBS. The speedup ratio is 29 in this case.

Figure 7 shows a typical sequence z, along with the trajectory of the projectile that reaches $\mathcal{G}$ in 52.2 s.

V. Conclusions

An iterative simulation-based trajectory shaping is proposed so that a sequence of commands enables a guided projectile to achieve a set of target constraints. The proposed approach, based on cross-entropy minimization, allows computing a near-optimal solution to the target-reaching problem in a relatively short period of time when compared with an equivalent solution found by the brute-force search method. A speedup ratio of several orders of magnitude can be obtained while preserving either small final errors with respect to a set of tight constraints or high empirical frequency of target-reaching occurrence.

Acknowledgments

This work was funded by the Defense Research and Development Canada (DRDC) Applied Research Program "Concept Development of Artillery Precision Guided Munitions." The authors would like to acknowledge Marc Lauzon from DRDC–Valcartier for his valuable comments and help for the simulations.

References

[1] Mitchell, I. M., Bayen, A. M., and Tomlin, C. J., "Time-Dependent Hamiltonian-Jacobi Formulation of Reachable Sets for Continuous Dynamics Games," *IEEE Transactions on Automatic Control*, Vol. 50, No. 7, 2005, pp. 947–957. doi:10.1109/TAC.2005.851439

[2] Bayen, A. M., Mitchell, I. M., Oishi, M. M. K., and Tomlin, C. J., "Aircraft Autolander Safety Analysis Through Optimal Control-Based Reach Set Computation," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 68–77. doi:10.2514/1.21562

[3] Oishi, M. M. K., Mitchell, I. M., Tomlin, C. J., and Saint-Pierre, P., "Computing Viable Sets and Reachable Sets to Design Feedback Linearizing Control Laws Under Saturation," *Proceedings of the 45th IEEE Conference on Decision and Control*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 2006, pp. 3801–3807.

[4] Aubin, J., *Viability Theory*, Birkhauser, Boston, 1991, Chap. 5.

[5] Cruck, E., and Saint-Pierre, P., "Nonlinear Impulse Target Problems Under State Constraints: A Numerical Analysis Based on Viability Theory," *Set-Valued Analysis*, Vol. 12, No. 4, 2004, pp. 383–416. doi:10.1007/s11228-004-6412-x

[6] Fliess, M., Lévine, J., Martin, P., and Rouchon, P., "Flatness and Defect of Nonlinear Systems: Introduction Theory and Application," *International Journal of Control*, Vol. 61, No. 6, 1995, pp. 1327–1361. doi:10.1080/00207179508921959

[7] Milam, M. B., Mushambi, K., and Murray, R. M., "A New Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems," *Proceedings of the 39th IEEE Conference on Decision and Control*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 2000, pp. 845–851.

[8] Brown Cribbs, H., "Genetics-Based Trajectory Discovery for Tactical Missiles," 1st AIAA Intelligent Systems Technical Conference, AIAA Paper 2004-6550, September 2004.

[9] Rubinstein, R. Y., and Kroese, D. P., "The Cross-Entropy Method—A Unified Approach to Combinatorial Optimization," *Monte-Carlo Simulation and Machine, Information Science and Statistics*, Springer, New York, 2004, pp. 41–57, 62–67.

[10] Brémaud, P., *Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues*, Springer–Verlag, New York, 2001, Chap. 8.

[11] Simulink, Software Package, Ver. 7, The MathWorks, Inc., Natick, MA, 2007.