

Stochastic Model Predictive Control for Guided Projectiles Under Impact Area Constraints

Jonathan Rogers

Assistant Professor

Woodruff School of Mechanical Engineering,

Georgia Institute of Technology,

Atlanta, GA 30332

The dynamics of guided projectile systems are inherently stochastic in nature. While deterministic control algorithms such as impact point prediction (IPP) may prove effective in many scenarios, the probability of impacting obstacles and constrained areas within an impact zone cannot be accounted for without accurate uncertainty modeling. A stochastic model predictive guidance algorithm is developed, which incorporates nonlinear uncertainty propagation to predict the impact probability density in real-time. Once the impact distribution is characterized, the guidance system aim point is computed as the solution to an optimization problem. The result is a guidance law that can achieve minimum miss distance while avoiding impact area constraints. Furthermore, the acceptable risk of obstacle impact can be quantified and tuned online. Example trajectories and Monte Carlo simulations demonstrate the effectiveness of the proposed stochastic control formulation in comparison to deterministic guidance schemes.

[DOI: 10.1115/1.4028084]

1 Introduction

Precision-guided projectile systems are an active area of research for the U.S. military. Projectile guidance and control may be tasked with reducing dispersion, shaping the vehicle trajectory, extending range, or some combination. Gun-launched projectiles present a difficult environment in which to implement precise guidance and control for several reasons including extreme acceleration loads at launch, highly nonlinear dynamics, and low cost requirements. Many current projectile guidance schemes, such as impact point predictors [1–4], are based on some type of model propagation and often yield highly efficient control sequences. Propagation of a projectile model to impact is a computationally demanding task, which has led to development of projectile linear theory and modified linear theory [5,6]. Fresconi et al. [7] have compared performance of an IPP given propagation models of varying fidelity, demonstrating tradeoffs between accuracy and computational burden.

The deterministic nature of many projectile guidance laws limits their ability to incorporate impact area constraints. These constraints may take the form of obstacles, which impede ballistic flight to the target, or areas near the target such as buildings or civilian areas. Since any guidance system is prone to error due to estimation uncertainty, unmodeled dynamics, or wind perturbations, ignorance of such constraints near the target will inevitably lead to a portion of guided rounds impacting within these obstacles or constrained areas. Stochastic guidance schemes are required that account for the probability of obstacle impact given uncertainty at the current time. Such stochastic guidance laws may adjust the flight path accordingly, possibly accepting an increase in miss distance for a reduction in obstacle impact probability and collateral damage.

Stochastic missile control is not a new idea and has been explored by numerous authors throughout the past several decades. Speyer [8] derived an optimal, linear terminal guidance law for missiles subject to measurement and target acceleration uncertainty by minimizing the expected value of an exponential function. In Ref. [9], Zarchan outlined a technique to predict the performance of missiles subject to stochastic disturbances through a combination of covariance propagation of an equivalent linear system and analysis of the adjoint system. Hull et al. [10] considered inner-loop autopilot design for a missile subject to uncertainty in the nonlinear aerodynamic model. Yavin et al. developed a bang–bang guidance law to mitigate stochastic perturbations to a stabilized seeker platform attached to a missile airframe, with performance evaluated through point-mass [11] and rigid body models [12]. More recently, Hexner and Shima [13] derived a stochastic nonlinear missile autopilot for bounded acceleration by assuming a linear model for zero-effort-miss during terminal air-to-air engagement. Results showed that the stochastic guidance law outperformed deterministic guidance for missiles with bounded acceleration inputs. Although not applied to the missile or projectile problem specifically, recent work on reach-avoid stochastic control [14–16] is especially relevant to this paper in that the problem of obtaining a goal state while avoiding obstacles under uncertainty is considered. Solution methods for optimal stochastic reach-avoid controllers have to date typically employed a numerical optimization approach such as dynamic programming.

One aspect common to much of the stochastic missile control literature is that missile–target dynamics are considered linear and planar in many cases, significantly simplifying uncertainty propagation, which can then be performed with a Kalman filter. In the gun-launched projectile domain, lateral and longitudinal dynamics are highly coupled (and nonlinear) due to nontrivial spin rates, and vehicle dynamics may vary significantly over short time scales (for instance, in the transonic regime). Thus, uncertainty propagation over a time horizon beyond several seconds or so may not be possible using linear models. Furthermore, control authority is usually much more limited than in missile control—thus, uncertainty quantification must be performed well in advance of terminal guidance. This paper describes a novel stochastic control scheme, which solves a constrained optimization problem using impact distribution data from nonlinear uncertainty propagation. During each control cycle, Monte Carlo trajectory analysis is performed in real-time using a parallel computational architecture to generate a predicted ground impact probability distribution. Obstacle impact probabilities are calculated, and the aim point is subsequently shifted from the target to reduce the obstacle impact probability below a user-defined threshold. The result is a guidance system that optimally trades miss distance for a reduction in probability of impact within constrained areas. First, the novel stochastic control scheme is described, followed by a brief description of the nonlinear projectile dynamic model. Both example and Monte Carlo simulation results are presented, demonstrating controller performance in several challenging environments including defilade scenarios. Results demonstrate that the control scheme is effective in reducing the probability of obstacle impact by an arbitrary amount, while still guiding the projectile close enough to the target to remain largely effective.

2 Projectile Stochastic Model Predictive Control

2.1 Deterministic Model Predictive Guidance. Model predictive control may be instantiated for guided weapons systems in the form of IPP. IPP guidance is performed by propagating a model from the current state to ground impact and computing control inputs according to the difference between the predicted impact and target (or zero-effort miss, ZEM). Consider a projectile equipped with movable canards for aerodynamic control. Define a body frame B attached to the projectile, as shown in the rearward view in Fig. 1, and let δ be the canard variable deflection angle with respect to the body-fixed axes. For a canard mounted

Contributed by the Dynamic Systems Division of ASME for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received November 19, 2013; final manuscript received June 25, 2014; published online October 21, 2014. Assoc. Editor: Dejan Milutinovic.

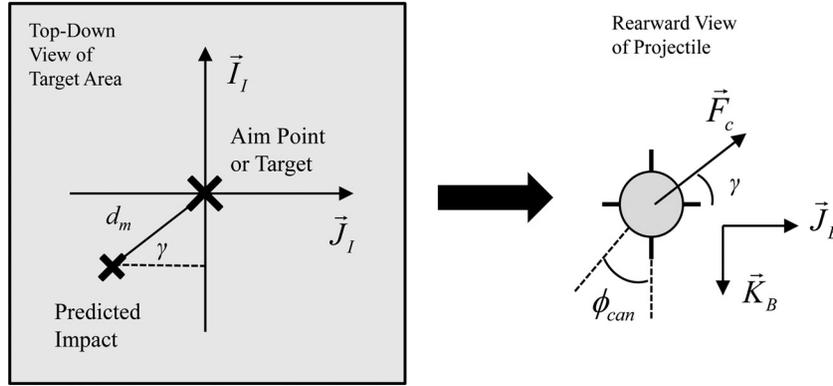


Fig. 1 Diagram of impact point prediction guidance parameters

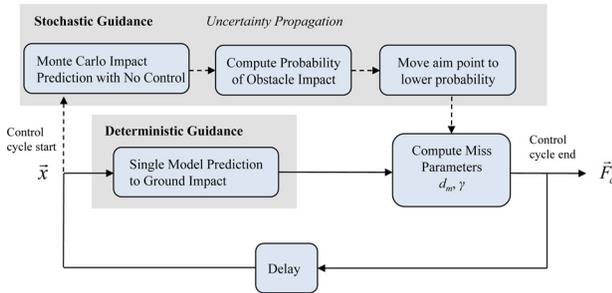


Fig. 2 Deterministic and stochastic impact point prediction guidance. Deterministic guidance follows the solid path; stochastic guidance follows the dashed path.

on the body at roll angle ϕ_{can} with respect to the \mathbf{K}_B axis (as shown in Fig. 2), canard deflection is determined from ZEM according to

$$\delta = \delta_{mag} (\cos \phi \cos \gamma + \sin \phi \sin \gamma) \cos(\phi_{can}) \quad (1)$$

where ϕ is the projectile roll angle and δ_{mag} is given by the saturation function

$$\delta_{mag} = \begin{cases} \delta_{max} & \text{if } d_m > d_0 \\ 0 & \text{if } d_m \leq d_0 \end{cases} \quad (2)$$

In Eqs. (1) and (2), δ_{max} is the maximum canard deflection, d_0 is the user-defined controller deadband, and d_m and γ are defined in Fig. 1. Deterministic model predictive guidance is outlined in the block diagram shown in Fig. 2.

2.2 Stochastic Model Predictive Guidance. Deterministic model predictive guidance is suitable for use when stochastic disturbances and/or model uncertainties are small enough that they induce trajectory errors below an acceptable threshold. In projectile guidance, various aspects of model uncertainty and stochastic forcing are encountered, including uncertainty in aerodynamics, winds, target location, and initial conditions from which model predictions are propagated. Taken together, these sources of uncertainty may lead to problematic miss distances with deterministic guidance systems even if reasonable control authority is available. In many urban environments where targets are embedded in narrow streets or close to buildings, such miss distances may present an unacceptable risk in terms of collateral damage and potentially inhibit target engagement.

While greater impact accuracy may be obtained through development of more accurate aerodynamic models and real-time estimators, cost considerations often impose limits on sensor

precision and model identification. As an alternative, a stochastic model predictive control system is derived, which acknowledges uncertainty within the guidance system and adjusts the guidance solution accordingly. To formulate the stochastic controller, consider the time-invariant discrete nonlinear system with process noise

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \Theta) + \mathbf{w}_t \quad t = 0, \dots, T-1 \quad (3)$$

where $\mathbf{x}_t \in \mathbb{R}^n$ is the state, $\mathbf{u}_t \in \mathbb{R}^m$ is the control input at timestep t , $\Theta \in \mathbb{R}^c$ are the system parameters, and \mathbf{w}_t is the process noise or stochastic forcing term. In general, \mathbf{w}_t is not restricted to be Gaussian white noise and may exhibit an arbitrary correlation with system parameters, time, and initial conditions. Because the dynamics in Eq. (3) are stochastic rather than deterministic, the uncertainty associated with the state vector \mathbf{x} may be characterized by a time-varying probability density function (PDF) of the state. Let $p_0(\mathbf{x}_0, \Theta)$ represent the initial state PDF at $t=0$, and $p_t(\mathbf{x}_t, \Theta)$ represent the state PDF at timestep t . An optimal controller may be derived that minimizes a cost function given by

$$J = E \left(\sum_{t=0}^{T-1} g_t(p_t(\mathbf{x}_t, \Theta), \mathbf{u}_t) + g_T(p_T(\mathbf{x}_T, \Theta)) \right) \quad (4)$$

where $g_t(p_t(\mathbf{x}_t, \Theta), \mathbf{u}_t)$ represents a function of the control and state probability density at time t , and $g_T(p_T(\mathbf{x}_T, \Theta))$ is a function of the final PDF of the state at time of impact T . Thus, the cost function is obtained by applying the expectation operator to some function of the stage and final state probability densities as well as control inputs. Note that this is a generalization of deterministic model predictive control, in which the state PDF is condensed to a single value and functions g_t usually take on a quadratic form. The stochastic control problem outlined in Eq. (4) is called a Markov decision process (MDP) and is typically solved through either value iteration or policy iteration depending on the specific application (as long as certain constraints on the process noise are satisfied). In the case of partial state observability, the problem becomes a partially observable MDP, and more advanced solution techniques are required. A survey of these techniques is provided in Ref. [17]. In the present application, both the state and action spaces are continuous with dimensions 12 and 2, respectively (along with two time-varying wind parameters). Typical MDP solution methods for continuous state spaces require discretization of the state space [18,19], but such methods are infeasible here due to the curse of dimensionality. Furthermore, runtime considerations for fire control computers, which must compute the optimal policy offline before flight in a short amount of time, mean that iterative MDP solution methods are of limited practical use since the number of iterations for convergence depends on many factors, including the initial guess, and solutions may not be achievable in the time required.

However, the general stochastic control problem is simplified significantly in this case since terminal accuracy is the driving factor in control calculations. Here, control inputs are based directly on a terminal state PDF generated through probabilistic predictions of ZEM, and thus the stage cost functions in Eq. (4) may be omitted leaving only the terminal cost component $g_T(p_T(\mathbf{x}_T, \Theta))$. The optimal control can therefore be derived by computing the state probability density at ground impact through uncertainty propagation at each control update. While PDF propagation is straightforward in the case of a linear plant with Gaussian uncertainty, the projectile problem of interest is nonlinear and subject to non-Gaussian uncertainty. Efficient methods of nonlinear uncertainty propagation must be used within the guidance loop to predict the state PDF at impact. Numerous methods of varying complexity have been developed for nonlinear uncertainty propagation including Monte Carlo [20], polynomial chaos [21], local linearization [22], and direct methods using the Fokker–Planck–Kolmogorov equation (FPKE) [23]. Methods such as FPKE and polynomial chaos are typically difficult to apply in high-dimensional state spaces. Monte Carlo is chosen as the PDF propagation method of choice for the stochastic MPC algorithm derived here due to its flexibility in handling large state spaces and arbitrary input distributions.

The stochastic model predictive guidance scheme developed here is depicted in Fig. 2, where the upper branch (dashed path) is a series of uncertainty quantification steps added to the original deterministic IPP scheme. At each control cycle, the aim point is adjusted through Monte Carlo IPP incorporating uncertainty in aerodynamics, mass parameters, initial conditions, and winds. These N trajectory predictions are performed using a six degree of freedom dynamic model assuming ballistic flight (zero control effort). Once Monte Carlo impact predictions have been obtained, they are transformed into a probability density and centered such that the mean of the distribution lies at the target. This PDF is shifted around the target area until the probability of obstacle impact drops below a user-defined threshold. The optimal shift is determined as that which results in a minimum distance to the original target, yet produces an acceptable probability of obstacle impact as outlined in Sec. 2.3. Once the optimal shift is established, the mean of this shifted distribution is taken as the new aim point. Finally, to compute control inputs γ and δ_{mag} , a deterministic IPP (as used in the deterministic IPP) is compared to the new aim point, as shown in Fig. 1. Note that the entire purpose of stochastic IPP is to characterize the shape and size of the impact PDF given uncertainty at the current time (this is similar to a particle filter, except there is no recursion step as impact densities are completely recomputed at each guidance update). The impact density shape determines how far the aim point should be shifted from obstacles to reduce impact probabilities below a certain threshold. Standard IPP guidance can then proceed given the new aim point.

2.3 Aim Point Optimization. At each controller update, a new aim point is generated through solution of an optimization problem. First, Monte Carlo IPPs assuming no control (i.e., $\delta = 0 \forall t \in [0, T]$) produce N impact points given by $\{x_1, x_2, \dots, x_N\}$ and $\{y_1, y_2, \dots, y_N\}$. The dispersion pattern can be translated so that the mean impact point of the dispersion pattern overlays the target. This is accomplished by creating translated impact points

$$\{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N\} = \{x_1, x_2, \dots, x_N\} - (\mu_x - x_t) \quad (5)$$

$$\{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N\} = \{y_1, y_2, \dots, y_N\} - (\mu_y - y_t) \quad (6)$$

where (μ_x, μ_y) are the mean impact point coordinates of the ballistic dispersion pattern and (x_t, y_t) is the target location. Impact locations $\{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N\}$ and $\{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N\}$ are referred to as mean-shifted impact points. Once the impacts have been shifted, a

probability density $\hat{f}(x, y)$ is constructed from $\{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N\}$ and $\{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N\}$ through kernel density estimation. Note that in practice, impact distributions for ZEM trajectories tend to be unimodal, and hence, this mean shift procedure is reasonable.

Aim point optimization is performed by shifting the impact PDF around the target area to reduce the likelihood of obstacle impact. This optimization problem may be formulated as follows:

Given target location (x_t, y_t) , an estimate of the impact point PDF $\hat{f}_k(x, y)$ at time t_k , and a set of r two-dimensional obstacles $\{C_1, C_2, \dots, C_r\}$, find an aim point (x_a, y_a) that minimizes

$$J = \sqrt{(x_t - x_a)^2 + (y_t - y_a)^2} \quad (7)$$

subject to

$$\sum_{i=1}^r \int_{C_i} \hat{f}(x + (x_t - x_a), y + (y_t - y_a)) dx dy < \varepsilon_p \quad (8)$$

where $0 < \varepsilon_p \leq 1$ is a user-defined level of acceptable risk and (x_a, y_a) is the shifted location of the mean of $\hat{f}_k(x, y)$.

Solution of this optimization problem requires a critical, but not unreasonable, assumption that the shape of the impact point PDF remains static as it is shifted throughout the target area. If the aim point shift $(\Delta x, \Delta y) = (x_t - x_a, y_t - y_a)$ is small compared to the projectile's current slant range to the target, this represents a reasonable assumption since small control adjustments that "nudge" the projectile toward the new aim point will not introduce significant new uncertainty. However, if $(\Delta x, \Delta y)$ is large compared to the slant range to target, the shape of $\hat{f}_k(x, y)$ may change substantially as the aim point is moved around the target area. This would require that different PDFs be generated for each aim point shift $(\Delta x, \Delta y)$ under consideration, imposing prohibitive computational burden. For reasonable obstacle scenarios, as long as ε_p is kept above a small threshold on the order of 10^{-3} the assumption that the PDF's shape remains static as it is shifted around the target area remains valid, since the PDF naturally shrinks as the projectile nears the target and thus smaller and smaller values of $(\Delta x, \Delta y)$ solve the optimization problem.

Several implementation details are important in enabling real-time execution of the proposed stochastic IPP guidance. First, given the impact point dispersion pattern at each controller update, PDF estimation is carried out through multivariate kernel density estimation [24], in which a scaled sum of kernel density functions are used to approximate the true probability density. Since estimation errors are assumed to be Gaussian distributed, Gaussian kernels are used and a symmetrical kernel bandwidth $h = \sigma N^{-1/6}$ is selected, where σ is the average standard deviation of the impacts in the x and y directions [24]. Therefore, an estimate of the impact point PDF is given as

$$\hat{f}(x, y) = \frac{1}{nh} \sum_{i=1}^n \frac{1}{2\pi} \exp\left(-\frac{1}{2h} \left((x - \tilde{x}_i)^2 + (y - \tilde{y}_i)^2\right)\right) \quad (9)$$

An advantage of this kernel density formulation is that a smooth estimate of the impact point PDF is available in closed form, enabling extremely efficient calculation of the integral sum in Eq. (8), which represents the probability of obstacle impact if the projectile continues to guide to the target. Assume that each obstacle is rectangular such that the i th obstacle is defined by a minimum and maximum location a_i and b_i , respectively, along \mathbf{I}_1 , as well as a minimum and maximum location c_i and d_i , respectively, along \mathbf{J}_1 . This is assumption is valid since any geometric obstacle can be decomposed into (potentially many) rectangular components. Then, using Eq. (9), the integral sum in Eq. (8) is given by

$$\begin{aligned}
& \sum_{i=1}^r \int_{C_i} \hat{f}(x + \Delta x, y + \Delta y) dx dy \\
&= \sum_{j=1}^r \sum_{i=1}^n \frac{1}{4n} \left(\operatorname{erf} \left(\frac{1}{\sqrt{2h}} (b_j - \tilde{x}_i - \Delta x) \right) \right. \\
&\quad \left. - \operatorname{erf} \left(\frac{1}{\sqrt{2h}} (a_j - \tilde{x}_i - \Delta x) \right) \right) \bullet \left(\operatorname{erf} \left(\frac{1}{\sqrt{2h}} (d_j - \tilde{y}_i - \Delta y) \right) \right. \\
&\quad \left. - \operatorname{erf} \left(\frac{1}{\sqrt{2h}} (c_j - \tilde{y}_i - \Delta y) \right) \right) \quad (10)
\end{aligned}$$

Given an impact point distribution and a set of r obstacles, the probability of obstacle impact for an arbitrary aim point shift may be calculated rapidly in closed form using Eq. (10).

Another detail pertinent to implementation is the feasibility of executing Monte Carlo simulation in real-time for this system. Monte Carlo simulations are inherently parallel requiring no synchronization between trajectories. In Ref. [25], the author showed that runtime requirements for Monte Carlo analysis could be reduced by 1–2 orders of magnitude when executed on a graphics processing unit (GPU), leading to the ability to run such simulations in real-time. In the stochastic MPC algorithm developed here, it is proposed that PDF prediction be accomplished through real-time Monte Carlo simulation on embedded GPU hardware, which is rapidly becoming available for low-power autonomous systems applications [26,27]. Preliminary runtime results are provided in Sec. 4.

Finally, the solution method to the optimization problem formulated in Eqs. (7) and (8) must be defined. While the cost function in Eq. (7) is clearly convex, the constraint (8) and (10) is not and thus gradient-based solvers are susceptible to convergence to local minima. There are few reliable solution techniques available for global, nonconvex optimization problems in continuous spaces such as that defined in Eqs. (7) and (8), especially given the complexity of the constraint function in Eq. (10). In this case, a search grid approach is reasonable since the optimization space is only two-dimensional, and discretization errors may be mitigated through repeated updating of the solution (feedback guidance). Thus, a grid of possible aim point locations (x_a, y_a) is created, and Eq. (10) evaluated for each one. If all possible aim points in this grid yield a probability of obstacle impact above the desired threshold ε_p , the search grid is doubled in size (resolution is halved to keep the number of points constant) and all probabilities are evaluated again using Eq. (10). This iterative solution process is continued until a valid solution is found. Note that solution of Eq. (10) is completely independent for each candidate aim point, and thus these evaluations may be performed in parallel on a GPU as well. Use of the GPU in combination with the probability evaluation function in Eq. (10) allows the optimization problem to be solved extremely efficiently.

3 Projectile Dynamic Model

The example projectile used to evaluate performance of the stochastic impact guidance algorithm is a roll-stabilized mortar projectile with four dithering canards. This design is similar to that proposed in Ref. [28] and uses two canards to control roll angle and two canards for aerodynamic maneuvering. A description of the dynamic equations of motion used in the simulation and impact point predictor are provided in Sec. 2.1, while a brief description of the inner-loop control system is given in Sec. 2.2.

3.1 Projectile Equations of Motion. The dynamic model and impact point predictor use standard rigid body equations of motion with a quaternion representation of orientation with respect to the inertial frame. This dynamic model is given in Eqs. (11)–(14).

$$\begin{aligned}
\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} &= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_0 q_2 + q_1 q_3) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_0 q_1 + q_2 q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \\
&\quad \times \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \quad (11)
\end{aligned}$$

$$\begin{Bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix} \quad (12)$$

$$\begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} = \frac{1}{m} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \quad (13)$$

$$\begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} = [I]^{-1} \begin{Bmatrix} L \\ M \\ N \end{Bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} [I] \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \quad (14)$$

In Eqs. (11)–(14), x , y , and z represent the inertial location of the projectile mass center, u , v , and w represent the body-frame velocity components of the mass center, and p , q , and r represent the body-frame components of the angular velocity. Normalized quaternion components are given by q_0 , q_1 , q_2 , and q_3 . Note that X , Y , and Z and L , M , and N in Eqs. (13) and (14) represent total forces and moments on the projectile in the body frame including steady aerodynamic, unsteady aerodynamic, weight, Magnus, and canard control contributions. Winds are included in the dynamic model during computation of the aerodynamic forces and moments. A complete description of the force and moment model is omitted here for brevity but can be found in Ref. [29]. In the dynamic simulation, the 13 equations given collectively by Eqs. (11)–(14) are integrated forward in time using a fourth-order, fixed-timestep Runge–Kutta algorithm to obtain a single trajectory. The impact point predictor uses an identical dynamic model with a variable timestep fifth-order Cash–Karp Runge–Kutta integrator [30] to reduce computation time.

The stochastic elements of the projectile model are hidden in Eqs. (11)–(14) as they appear implicitly in the force and moment description. Both model uncertainty (in the form of aerodynamic parameter mismatch) and stochastic wind disturbances are present in the aerodynamic force and moment computation, and thus Eqs. (11)–(14) represent a nonlinear stochastic dynamic system. Specific details on parameter uncertainty and wind disturbances used in this study are provided in Sec. 4.

3.2 Inner-Loop Control System. The example projectile is equipped with two sets of canards—one set for maneuver and the second for roll control. Roll control canards are located along the body $\pm \mathbf{J}_B$ axis, while the maneuver canards are located along the body $\pm \mathbf{K}_B$ axis. A proportional-derivative controller similar to that described in Ref. [28] is used to drive the roll angle of the projectile to the miss angle γ computed from the impact point predictor. This aligns the maneuver canards with the maneuver plane for optimum control authority. To that end, define the roll angle error signal as [28]

$$\Delta \phi = \tan^{-1} \left(\frac{\sin \phi \cos \gamma - \cos \phi \sin \gamma}{\cos \phi \cos \gamma + \sin \phi \sin \gamma} \right) \quad (15)$$

Then, the roll angle canard deflection is given by

$$\delta_{\text{roll}} = \operatorname{sat}(k_p p + k_\phi \Delta \phi, \delta_{\text{roll}}^{\max}) \quad (16)$$

where $\text{sat}(\bullet)$ denotes the saturation function and $\delta_{\text{roll}}^{\text{max}}$ is the maximum canard deflection angle for the roll canards. For all cases in this paper, $\delta_{\text{roll}}^{\text{max}} = 10$ deg and roll control gains are selected as $k_p = 0.1$ and $k_\phi = 2.0$.

4 Results

4.1 Example Projectile and Control System Parameters.

The example projectile is a fin-stabilized mortar projectile approximately 81 mm in diameter with mass, axial inertia, and transverse inertia given by 5.11 kg, 0.00391 kg-m², and 0.0757 kg-m² respectively. The projectile mass center is located approximately 32.9 cm from the base of the round. Two maneuver canards, each of area 8.06 cm², are assumed to be located approximately 16.2 cm in front of the projectile center of mass with $\delta_{\text{max}} = 10$ deg. Roll canards have the same station line location and area of 7.43 cm². For all cases in this section, control is initiated approximately at apogee, and IPP cycles are executed at a rate of 2 Hz.

Real-time Monte Carlo simulations executed by the stochastic IPP use $N = 256$ simulations. At initiation of each control cycle, Monte Carlo initial conditions are generated first by perturbing the true state with a random bias generated from a Gaussian distribution (simulating estimation error). Then, each of the 256 simulations is perturbed by an additional Gaussian-distributed error to create variation in the Monte Carlo run (simulating stochastic disturbances). Error distribution parameters are listed in Table 1. At the start of each trajectory simulation, model and wind errors are captured by perturbing the mean wind and aerodynamic parameters used by the stochastic IPP. These distribution parameters are listed in Table 2. Random biases are added to the wind vector along the inertial x and y directions at the start of the simulation. Likewise, random biases are added to all projectile and canard aerodynamic parameters using a zero-mean Gaussian distribution. At the initiation of each GPU-based Monte Carlo simulation, further aerodynamic model errors are added on a per-trajectory basis using the standard deviations listed in Table 2 (where aerodynamic coefficients reference the aerodynamic expansion outlined in Ref. [29]). Finally, wind errors are also added to each trajectory of the GPU-based Monte Carlo simulations to capture effects from wind estimation error throughout flight. Runtime experiments showed that, for a 256-case Monte Carlo simulation starting from apogee for the first example case shown below, a Tesla K40 GPU requires approximately 0.99 s for execution, leading to the conclusion that the guidance system may be sampled at a reasonable rate (1.5 s or longer) and real-time performance may be obtained.

4.2 Example Obstacle Geometry. Consider the obstacle geometry shown in Fig. 3, in which a target lies approximately

Table 1 Standard deviations for state errors in Monte Carlo simulations performed by stochastic IPP. These error parameters are used at each control cycle to generate initial conditions for Monte Carlo trajectory prediction.

State	Bias standard deviation	Error standard deviation
x (m)	3.05	3.05
y (m)	3.05	3.05
z (m)	3.05	3.05
ϕ (deg)	3.44	3.44
θ (deg)	2.86	2.86
ψ (deg)	2.86	2.86
u (m/s)	1.52	1.52
v (m/s)	0.12	0.12
w (m/s)	0.12	0.12
p (rad/s)	3.00	3.00
q (rad/s)	0.30	0.30
r (rad/s)	0.30	0.30

Table 2 Wind and aerodynamic error parameters for stochastic IPP. These distributions are used to create model error at the beginning of each simulation (model mismatch), and to perturb aerodynamic parameters and winds within each Monte Carlo run.

Quantity	Standard deviation
Wind errors	
Standard deviation of initial bias error	3.05 m/s
Standard deviation of trajectory perturbations	3.05 m/s
Standard deviation of aerodynamic coefficient errors	
C_{X0}	5%
C_{X2}	3%
can	5%
C_{YPA}	6%
C_{Y0}	4%
C_{Z0}	4%
C_{LP}	5%
C_{MQ}	4%
C_{LDD}	4%
All canard aerodynamics	5%
Canard trim pitch angle random bias standard deviation.	2.29 deg

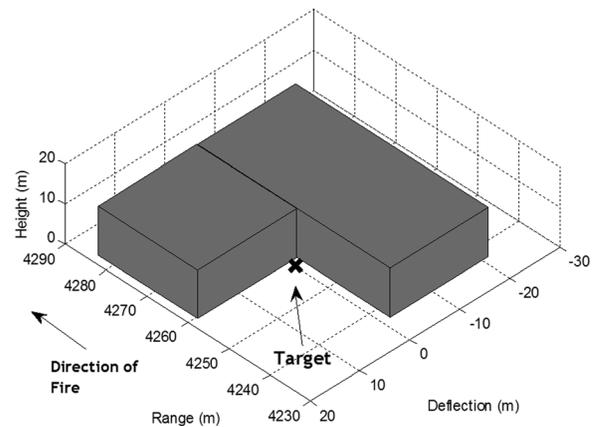


Fig. 3 Obstacle geometry, example scenario

2 m from the corner of an L-shaped building. In this case, the projectile's angle of fall is severely constrained, and there is significant risk of impacting the building and causing undesired collateral damage. An example simulation of the guided mortar projectile using stochastic IPP guidance is performed using muzzle velocity of 274 m/s, quadrant elevation of 1100 mrad, risk threshold $\epsilon_p = 0.15$, and zero winds. Figure 4 shows the aim point evolution for this example case. Note that initially, the optimal aim point determined by the stochastic IPP overlays the target. After apogee, the aim point moves toward the upper left quadrant away from the obstacles while uncertainty is still large (corresponding to a larger, more diffuse impact PDF). During terminal flight, the impact PDF shrinks and the aim point moves back toward the target. This evolution, in which the aim point moves toward open areas and collapses back to the target during terminal maneuvers, is representative of typical performance. The impact point shown in Fig. 4 is approximately 15 m from the target in a direction clear of the obstacles.

Effectiveness of the stochastic control system is more clearly demonstrated through Monte Carlo simulation rather than an example trajectory, since deterministic control may be more effective in one specific instance but will be less effective overall given stochastic disturbances. Monte Carlo simulations of 200 cases are performed by randomizing initial conditions at launch as well as winds according to normal distributions with standard deviations provided in Table 3. All Monte Carlo simulation results using the

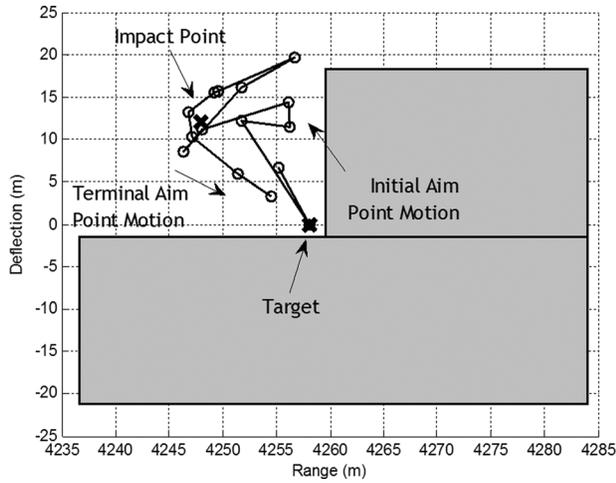


Fig. 4 Aim point evolution, for example scenario

Table 3 Initial condition and wind distributions for Monte Carlo simulations

State	Mean value	Standard deviation
x (m)	0	3.05
y (m)	0	3.05
z (m)	0	0
ϕ (deg)	Uniform between 0 and 360 deg	
θ (deg)	63.0 (Section 4.2) or 82.0 (Section 4.3)	1.00
ψ (deg)	0	1.00
u (m/s)	274	3.66
v (m/s)	0	1.07
w (m/s)	0	1.07
p (rad/s)	0	1.00
q (rad/s)	0	1.50
r (rad/s)	0	1.50
Wind along \mathbf{I}_I	0	3.05
Wind along \mathbf{J}_I	0	3.05

stochastic IPP are compared with a Monte Carlo simulation using the deterministic impact point predictor described in Sec. 2.1. This enables a side-by-side comparison of the performance of the two control schemes.

Eleven Monte Carlo simulations are performed using different values of ϵ_p between 0.05 and 0.9. Figure 5 shows several impact point dispersion patterns and 50% CEP rings for these simulations, while Fig. 6 shows the percentage of cases in each Monte Carlo simulation, which impacted the three-dimensional obstacle. Note that using deterministic guidance, the mean impact point is only a few meters from the target, but a substantial number of cases impact the building due to dispersion. As expected, the probability of obstacle impact decreases as the risk threshold is reduced, from 55% in the deterministic case to zero in the case of $\epsilon_p < 0.4$. As less risk is tolerated, the dispersion pattern moves farther from the target although mean impact-centered CEP increases only by 1–2 m. Finally, note that when ϵ_p is increased beyond 0.6, performance matches the deterministic control system. This verifies that the proposed guidance system reduces to deterministic IPP if the risk tolerance threshold is high enough.

It is evident from Fig. 6 that a mismatch exists between the risk threshold ϵ_p and the percentage of impacts that actually hit the obstacle area. The reason for this difference is that the stochastic IPP propagates zero-effort trajectories, which are not representative of the closed-loop trajectories actually flown. It is computationally infeasible to propagate hundreds of closed-loop trajectories with the stochastic predictor for several reasons, one of which is that

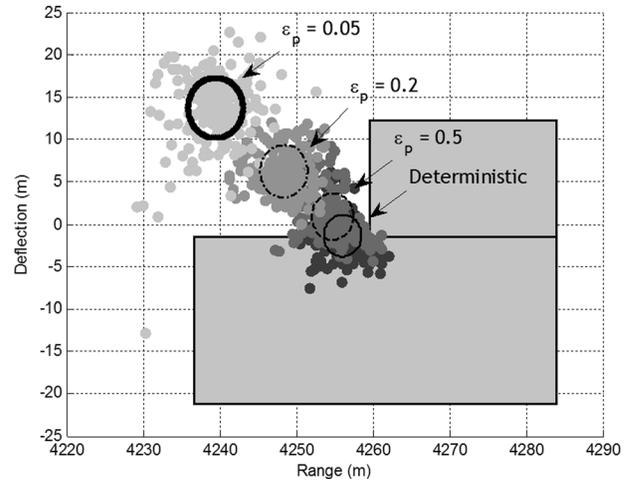


Fig. 5 Monte Carlo dispersion patterns and mean-centered CEP rings for varying ϵ_p , example geometry

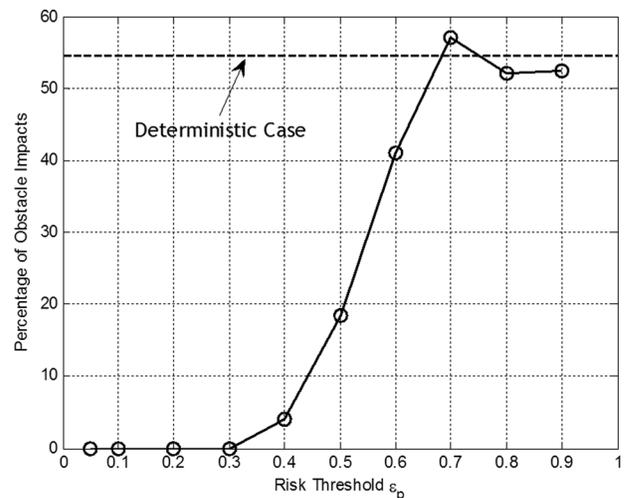


Fig. 6 Obstacle impact probability versus ϵ_p , for example scenario

each closed-loop trajectory prediction would itself have to implement massively parallel trajectory predictions to simulate the guidance system. This recursive loop leads to exponential scaling in the number of trajectory predictions required. As shown in this section, even with zero-effort trajectory predictions obstacle avoidance can still be accomplished quite effectively. In practice, since ϵ_p is not an accurate measure of the actual probability of obstacle impact for the closed-loop system, a fire control computer must be used to determine the mapping between ϵ_p and the true probability of impact for the specific obstacle scenario under consideration.

4.3 Targets in Defilade. Consider another example with geometry shown in Fig. 7, in which a target is concealed within an urban environment and in defilade with respect to the gun. This represents an extremely challenging scenario even for guided projectiles since substantial trajectory shaping is required to avoid impact with the building. Since the target is approximately 3 m from the building, angles of fall less than 86.2 deg will result in building impact. An example simulation of the stochastic IPP, guidance system is performed and compared to the deterministic IPP described in Sec. 2.1. Initial conditions for this case are identical to the above scenario, except with a quadrant elevation of 1431 mrad and $\epsilon_p = 0.05$. Figure 8 shows the aim point evolution

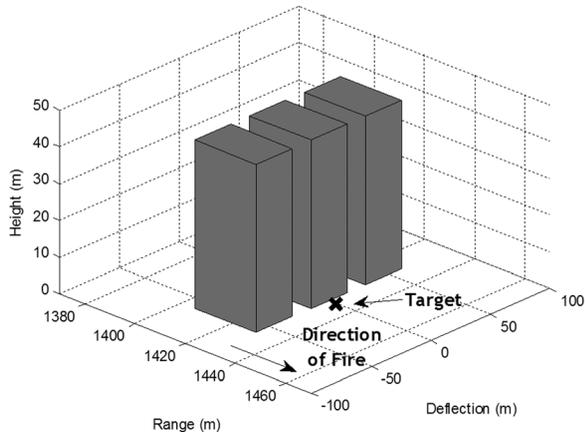


Fig. 7 Obstacle geometry, defilade case

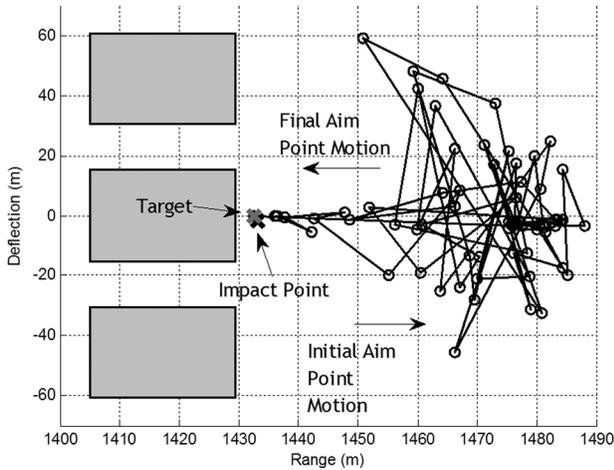


Fig. 8 Aim point evolution for stochastic IPP guidance, defilade case

for this case, in which once again the aim point begins at the target, moves away from the obstacles during midcourse, and collapses back toward the target during terminal maneuvers. The final impact point is only 2 m from the target. Figure 9 is particularly interesting in terms of the tradeoff between deterministic and stochastic IPP guidance. In the deterministic case, the projectile's terminal angle of fall of 83.3 deg is insufficient to clear the

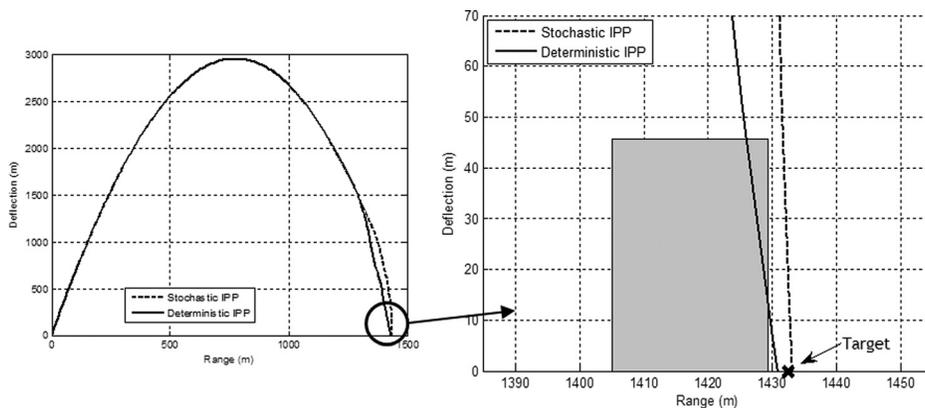


Fig. 9 Range versus altitude and zoom view of target area, defilade example case

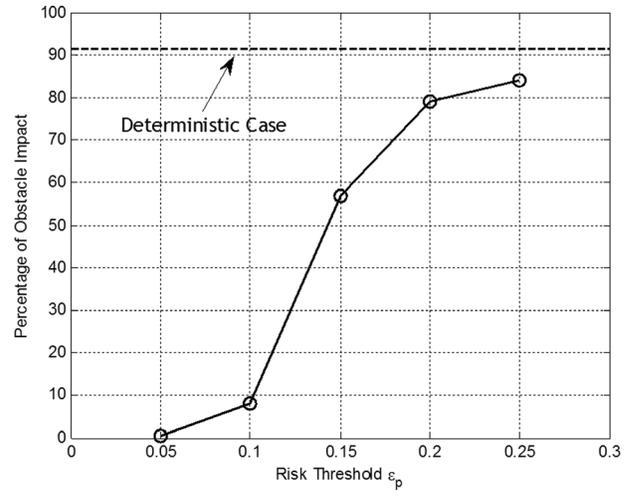


Fig. 10 Obstacle impact probability versus ϵ_p , defilade case

obstacle. In the stochastic case, the guidance system actually performs terminal trajectory shaping through evolution of the aim point as demonstrated in Fig. 8. During midcourse, the aim point moves downrange of the target, which initiates a slight range extension. Then, as the impact PDF shrinks and the aim point moves back toward the target, a subsequent pull-down maneuver is commanded, which increases the angle of fall to approximately 87.7 deg at impact. Thus, the stochastic IPP implements trajectory shaping during midcourse and terminal guidance through evolution of the aim point.

Monte Carlo simulations are performed for this example defilade scenario, using 200 simulations each drawn from random initial conditions with distributions provided in Table 3. Risk threshold ϵ_p was varied between 0.05 and 0.25. Figure 10 shows the percentage of cases that impacted the building obstacles as a function of ϵ_p . Note that at high values of ϵ_p , stochastic guidance performs similar to deterministic IPP guidance. As the risk threshold decreases, it was noted that the average angle of fall increased steadily to almost 90 deg at $\epsilon_p = 0.05$ while the percentage of obstacle impacts drops below 1%. This type of autonomous trajectory shaping is not possible using deterministic IPP guidance since impact density information is not readily incorporated into the algorithm. Overall, the success with which the stochastic guidance scheme successfully performs target engagement in this defilade scenario demonstrates its potential as an intelligent trajectory shaping mechanism that can augment standard IPP control.

5 Conclusion

A novel stochastic model predictive guidance scheme has been derived for guided projectiles under target area constraints by combining nonlinear uncertainty propagation with a constrained optimization problem. The stochastic controller takes the form of a massively parallel IPP scheme. Impact PDF's are processed to determine an optimal aim point shift that reduces obstacle impact probabilities below a certain threshold. Multivariate kernel density estimation is used to rapidly predict obstacle impact probabilities from impact point dispersion patterns. The entire guidance scheme may be parallelized and implemented on emerging embedded GPUs. Comparisons with a standard deterministic guidance algorithm show that the new stochastic guidance law not only reduces obstacle impact probabilities by an arbitrary amount, but also serves as an efficient mechanism for trajectory shaping in defilade scenarios and other challenging geometries.

Acknowledgment

This material is based on work supported in part by the U.S. Army Research Laboratory under Cooperative Agreement W911NF-12-2-0027, and in part by the U.S. Army Research Office under Young Investigator Grant No. W911NF-12-1-0274 from the Complex Dynamics and Systems Program.

References

- [1] Ravindra, V. C., Bar-Shalom, Y., and Willett, P., 2010, "Projectile Identification and Impact Point Prediction," *IEEE Trans. Aerosp. Electron. Syst.*, **46**(4), pp. 2004–2021.
- [2] Hardiman, D. F., Kerce, J. C., and Brown, G. C., 2006, "Nonlinear Estimation Techniques for Impact Point Prediction of Ballistic Targets," Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets, Orlando, FL, May 19, Vol. 6236.
- [3] Kashigawi, Y., 1968, "Prediction of Ballistic Missile Trajectories," Stanford Research Institute, Menlo Park, CA, Technical Report No. SRI-H-8976.
- [4] Morgan, R. W., Tharp, H., and Vincent, T., 2011, "Minimum Energy Guidance for Aerodynamically Controlled Missiles," *IEEE Trans. Autom. Control*, **56**(9), pp. 2026–2037.
- [5] Hainz, L., and Costello, M., 2005, "Modified Projectile Linear Theory for Rapid Trajectory Prediction," *J. Guid., Control, Dyn.*, **28**(5), pp. 1006–1014.
- [6] Slegers, N., 2008, "Predictive Control of a Munition Using Low-Speed Linear Theory," *J. Guid., Control, Dyn.*, **31**(3), pp. 768–775.
- [7] Fresconi, F., Cooper, G., and Costello, M., 2011, "Practical Assessment of Real-Time Impact Point Estimators for Smart Weapons," *ASCE J. Aerosp. Eng.*, **24**(1), pp. 1–11.
- [8] Speyer, J., 1976, "An Adaptive Terminal Guidance Scheme Based on an Exponential Cost Criterion With Application to Homing Missile Guidance," *IEEE Trans. Autom. Control*, **21**(3), pp. 371–375.
- [9] Zarchan, P., 1979, "Complete Statistical Analysis of Nonlinear Missile Guidance Systems-SLAM," *J. Guid., Control, Dyn.*, **2**(1), pp. 71–78.
- [10] Hull, R. A., Schumacher, D., and Qu, Z., 1995, "Design and Evaluation of Robust Nonlinear Missile Autopilots From a Performance Perspective," Proceedings of the American Control Conference, Seattle, WA, Vol. 1, pp. 189–193.
- [11] Yavin, Y., Frangos, G., and Fourie, J., 1992, "The Performance of a Projectile Which Uses a Bang-Bang Type Guidance Law – Part 1," *Comput. Math. Appl.*, **23**(1), pp. 111–118.
- [12] Yavin, Y., Frangos, G., and Fourie, J., 1992, "The Performance of a Projectile Which Uses a Bang-Bang Type Guidance Law – Part 2 – A Semirigid Body Model," *Comput. Math. Appl.*, **24**(4), pp. 85–92.
- [13] Hexner, G., and Shima, T., 2007, "Stochastic Optimal Control Guidance Law With Bounded Acceleration," *IEEE Trans. Aerosp. Electron. Syst.*, **43**(1), pp. 71–78.
- [14] Abate, A., Prandini, M., Lygeros, J., and Sastry, S., 2008, "Probabilistic Reachability and Safety for Controlled Discrete Time Stochastic Hybrid Systems," *Automatica*, **44**(11), pp. 2724–2734.
- [15] Summers, S., and Lygeros, J., 2010, "Verification of Discrete Time Stochastic Hybrid Systems: A Reach-Avoid Decision Problem," *Automatica*, **46**(12), pp. 1951–1961.
- [16] Summers, S., Kamgarpour, M., Tomlin, C., and Lygeros, J., 2011, "A Stochastic Reach-Avoid Problem With Random Obstacles," 14th International Conference on Hybrid Systems: Computation and Control (HSCC'11), Chicago, IL, Apr. 12–14.
- [17] Thrun, S., Burgard, W., and Fox, D., 2006, *Probabilistic Robotics*, MIT, Cambridge, MA, p. 496.
- [18] Blondel, V., and Tsitsiklis, J., 2000, "A Survey of Computational Complexity Results in Systems and Control," *Automatica*, **36**(9), pp. 1249–1274.
- [19] Rust, J., 1997, "Using Randomization to Break the Curse of Dimensionality," Ph.D. dissertation, Yale University, Department of Economics, New London, CT.
- [20] Doucet, A., de Freitas, N., and Gordon, N., 2001, *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, pp. 6–14.
- [21] Wiener, N., 1938, "The Homogeneous Chaos," *Am. J. Math.*, **60**(4), pp. 897–936.
- [22] Roberts, J. B., and Spanos, P. D., 1990, *Random Vibration and Statistical Linearization*, Wiley, New York, pp. 122–176.
- [23] Risken, H., 1989, *The Fokker-Planck Equation: Methods of Solution and Applications*, Springer, New York, pp. 32–62.
- [24] Scott, D., 1992, *Multivariate Density Estimation: Theory, Visualization, and Practice*, Wiley, New York, p. 149.
- [25] Ilg, M., Rogers, J., and Costello, M., 2011, "Projectile Monte-Carlo Trajectory Analysis Using a Graphics Processing Unit," 2011 AIAA Atmospheric Flight Mechanics Conference, Portland, OR, Aug. 7–10.
- [26] SECO Corporation, 2012, "Carma Devkit – SECO," Accessed Dec. 20, <http://www.nvidia.com/object/carma-devkit.html>
- [27] SVTronics, Inc., 2013, "OMAP 5432 uEVM Development Board," Accessed Apr. 18, http://www.svtronics.com/index.php?route=product/product&product_id=33
- [28] Rogers, J., and Costello, M., 2010, "Design of a Roll-Stabilized Mortar Projectile With Reciprocating Canards," *J. Guid., Control, and Dyn.*, **33**(4), pp. 1026–1034.
- [29] Costello, M., and Rogers, J., 2011, "BOOM: A Computer-Aided Engineering Toolbox for Exterior Ballistics of Smart Projectiles," U.S. Army Research Laboratory Aberdeen Proving Ground, MD, Technical Report No. ARL-CR-670.
- [30] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B., 2007, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University, New York, pp. 714–722.