# Ergodic Exploration with Stochastic Sensor Dynamics

Gerardo De La Torre, Kathrin Flaßkamp, Ahalya Prabhakar, and Todd D. Murphey

***Abstract*— Ergodic exploration has been shown to be an effective framework for autonomous sensing and exploration. The objective of ergodic control is to minimize the difference between the distribution of the time-averaged sensor trajectory and a spatial probability distribution function representing information density. Therefore, the time a sensor spends sampling a particular region is manipulated to correspond to the anticipated information density of that region. This paper introduces a trajectory optimization approach for ergodic exploration in the presence of stochastic sensor dynamics. The stochastic differential dynamic programming algorithm is formulated in the context of ergodic exploration. Numerical studies demonstrate the proposed framework's ability to mitigate stochastic effects.**

## I. INTRODUCTION

Autonomous sensing and exploration is essential in systems that operate in unknown environments. In this context, the high-level goal is to use resources (e.g. time and energy) judiciously to maximize the amount of acquired information. Specifically, effective active sensing or sensor path planning requires the optimization of sensor parameters, such as position and orientation. For example, in UAV surveillance the airframe's position/attitude and the orientation of a gimbaled camera should be controlled to maximize the visual information captured [1]. Many algorithms and frameworks have been proposed that allow for autonomous sensing and exploration. Developed archetypes include random walk [2], lawnmower coverage, and information maximization [3].

Recently, autonomous exploration based on ergodic principles has been studied [4]. In this research thrust, sensor paths are computed to minimize the difference between the distribution of the time-averaged sensor trajectory and a spatial probability distribution function representing information density [5]. That is, in this framework the time a sensor spends sampling a particular region should correspond to the perceived information density of that region. The ergodic exploration of distributed information (EEDI) algorithm, proposed by Miller and Murphey [6], [7], generates feasible trajectories for general sensors with nonlinear dynamics. Experimental validation of the algorithm was conducted on the SensorPod robot, a platform inspired by electric field sensing fish [2]. When compared to previously proposed algorithms, such as the Information Gradient Ascent Controller [8] and the Information Maximization Controller [3], the SensorPod robot had a larger probability of successfully locating an object when the EEDI algorithm was used [7]. Furthermore, unlike exploration algorithms that rely on way-point selection, EEDI explicitly considers sensor dynamics and, therefore, remains effective for systems exhibiting complex behavior. However, the effects of stochastic dynamics on the ability of a sensor to perform ergodic exploration have not been explored.

The differential dynamic programming (DDP) algorithm generates optimal open and closed-loop control policies by computing a quadratic approximation of the cost-to-go function and utilizing quadratically approximated state space dynamics around a trajectory [9]. The same basic principles were used to develop iterative linear quadratic regulators (iLQR) [10], [11]. Extensions of the DDP algorithm have been developed in order to address state and control constraints [12], [13]. Furthermore, the algorithm has been successfully implemented in simulation to enable robust bipedal robotic walking [14] and has been successfully flight tested in suspended load operations [15]. Finally, the stochastic differential dynamic programming (S-DDP) algorithm considers stochastic system dynamics with additive control- and state-dependent noise and finds optimal open and closed-loop control policies to minimize the expectation of a given cost [16]. In this paper, the S-DDP algorithm is used to enable a system to mitigate the effects of stochasticity while performing ergodic exploration.

The main contribution of this paper is the development of an algorithm for ergodic exploration in the presence of stochastic and nonlinear dynamics. A new ergodic metric that is compatible with the S-DDP algorithm is introduced. Numerical studies compare the proposed algorithm to one where deterministic dynamics are considered. It is demonstrated that the proposed algorithm results in trajectories with greater and more predictable ergodicity. Furthermore, the total cost of trajectories, including the control cost induced by the computed closed-loop controller, is reduced when the proposed algorithm is used.

The organization of this paper is as follows. Section II gives an overview of the S-DDP algorithm. The concept of ergodicity and ergodic metrics are introduced in Section III. In addition, Section III formulates the trajectory optimization problem that is considered. Section IV presents results from numerical experiments. Conclusions are discussed in Section V.

G. De La Torre, A. Prabhakar, and T. D. Murphey are with the Department of Mechanical Engineering, McCormick School of Engineering and Applied Science, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208, USA. E-mail: gerardo.delatorre@northwestern.edu, ahalyaprabhakar2013@u.northwestern.edu, t-murphey@northwestern.edu.

K. Flaßkamp is with Optimization and Optimal Control, Center for Industrial Mathematics (ZeTeM), Bibliothekstr. 1, 28359 Bremen, Germany. E-mail: kathrin.flasskamp@uni-bremen.de.

This work was supported by Army Research Office grant W911NF-14-1-0461.

## II. Overview of S-DDP

The stochastic differential dynamic programming (S-DDP) algorithm numerically solves nonlinear stochastic optimal control problems using first and second order expansions of stochastic dynamics and cost along nominal trajectories. The algorithm is iterative in nature such that it computes optimal control deviation given a nominal input signal. The nominal input is then updated using the computed optimal deviation and the process can be repeated. In this section, we give an overview of the approach and refer to Reference [16] for a complete and detailed treatment of the topic.

Consider a class of stochastic dynamical systems that evolve according to

$$\mathrm{d}x = f(x,u)\ \mathrm{d}t + \sum_{i=1}^{m} F_i(x,u)\ \mathrm{d}\omega_i, \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^p$ is the control input, and $\omega_i \in \mathbb{R}$ are independent Brownian noises. Furthermore, the considered cost is of the form

$$v(x,u,t) = \mathrm{E}\Big[h(x(t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} l(x(\tau),u(\tau),\tau)\ \mathrm{d}\tau\Big], \tag{2}$$

where $h(x(t_\mathrm{f}))$ is the terminal cost and $l(x(t),u(t),t)$ is the running cost. The S-DDP algorithm attempts to find the optimal sequence of discrete inputs to minimize the given cost such that the continuous input is then defined as $u(T_k) = u_k,\ T_k \in [t_0 + k\Delta t,\ t_0 + (k+1)\Delta t)$ where $\Delta t$ is the discretization time step. The algorithm also approximates continuous trajectories with a sequence of state vectors using a numerical integrator such that $x_1 = x(t_0)$, $x_2 = x(t_0 + \Delta t)$, $\ldots$, $x_N = x(t_\mathrm{f})$. It should be noted that the S-DDP algorithm is iterative. Specifically, given the $i^{\text{th}}$ iteration of the sequence of discrete inputs $U_i = \{u_1, \ldots, u_{N-1}\}$ the S-DDP algorithm finds the optimal control deviation, $\delta U_i^\star$, such that the control input is updated as

$$U_{i+1} = U_i + \gamma \delta U_i^\star, \tag{3}$$

where $\gamma$ is a user defined constant or is selected from an automated process (e.g. Armijo line search [17]). However, several iterations may be needed in order to arrive at a control input that is sufficiently close to the optimal solution.

To begin an overview of the derivation of the S-DDP algorithm it is assumed that a sequence of *nominal* discrete inputs $\bar{U} = \{\bar{u}_1, \ldots, \bar{u}_{N-1}\}$ and the associated state trajectory $\bar{X} = \{\bar{x}_1, \ldots, \bar{x}_N\}$ are given. The first-order linearization around the nominal trajectory is given as

$$\delta x_{k+1} = A_k \delta x_k + B_k \delta u_k + \sum_{i=1}^{m} \Gamma_{ik}\omega_{ik}. \tag{4}$$

where[1]

$$\begin{aligned}
A_k &= \mathrm{I} + f_x(\bar{x}_k,\bar{u}_k)\Delta t,\\
B_k &= f_u(\bar{x}_k,\bar{u}_k)\Delta t,\\
\Gamma_{ik} &= F_{i_x}(\bar{x}_k,\bar{u}_k)\delta x_k + F_{i_u}(\bar{x}_k,\bar{u}_k)\delta u_k + F_i(\bar{x}_k,\bar{u}_k),
\end{aligned}$$

and $\omega_{ik} \sim \mathcal{N}(0,\Delta t)$. Note that the discretization time step, $\Delta t$, needs to be sufficient small in order to accurately represent the considered dynamics. However, the linearization scheme given in equation (4) is not unique. Replacing the Euler linearization with a linearization derived from a variational integrator allows the use of relatively large discretization time steps with minimal degradation to the performance of the S-DDP algorithm [18]. In order to focus on the main contributions of this paper variational integrators are not discussed here.

Continuing with our derivation, in discrete time Bellman's Principle of Optimality is stated as:

$$V(x_k,t_k) = \min_{u_k}\Big[\mathrm{E}\big[L(x_k,u_k,t_k) + V(x_{k+1},t_{k+1})\big]\Big] \tag{5}$$

where $V(x_k)^2$ is the optimal cost-to-go function and $L(x_k,u_k,t_k) = l(x_k,u_k,t_k)\Delta t$. Therefore, the optimal control deviation, $\delta u_k^\star$, can be found by considering an appropriate expansion of equation (5). Utilizing the derived first-order linearization of the system dynamics a second-order expansion of the expectation cost-to-go function around the nominal trajectory is obtained:

$$\begin{aligned}
&\mathrm{E}\big[V(\bar{x}_{k+1} + \delta x_{k+1})\big] \approx \mathrm{E}\big[V(\bar{x}_{k+1})\\
&\quad + V_x(\bar{x}_{k+1})^\mathsf{T}\delta x_{k+1} + \frac{1}{2}\delta x_{k+1}^\mathsf{T} V_{xx}(\bar{x}_{k+1})\delta x_{k+1}\big]\\
&\quad = V(\bar{x}_{k+1}) + V_x(\bar{x}_{k+1})^\mathsf{T}(A_k\delta x_k + B_k\delta u_k)\\
&\quad + \delta x_k^\mathsf{T} A_k^\mathsf{T} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k + \tfrac{1}{2}\delta x_k^\mathsf{T} A_k^\mathsf{T} V_{xx}(\bar{x}_{k+1}) A_k \delta x_k\\
&\quad + \tfrac{1}{2}\delta u_k^\mathsf{T} B_k^\mathsf{T} V_{xx}(\bar{x}_{k+1}) B_k\delta u_k + \tfrac{1}{2}\delta x_k^\mathsf{T}\mathcal{F}\delta x_k + \tfrac{1}{2}\delta u_k^\mathsf{T}\mathcal{Z}\delta u_k\\
&\quad + \delta x_k^\mathsf{T}\mathcal{L}\delta u_k + \delta x_k^\mathsf{T}\mathcal{S} + \delta u_k^\mathsf{T}\mathcal{U} + \tfrac{1}{2}\mathcal{T},
\end{aligned}$$

where

$$\begin{aligned}
\mathcal{F} &= \Delta t \textstyle\sum_{i=1}^m F_{i_x}(\bar{x}_k,\bar{u}_k)^\mathsf{T} V_{xx}(\bar{x}_{k+1}) F_{i_x}(\bar{x}_k,\bar{u}_k),\\
\mathcal{Z} &= \Delta t \textstyle\sum_{i=1}^m F_{i_u}(\bar{x}_k,\bar{u}_k)^\mathsf{T} V_{xx}(\bar{x}_{k+1}) F_{i_u}(\bar{x}_k,\bar{u}_k),\\
\mathcal{T} &= \Delta t \textstyle\sum_{i=1}^m F_{i}(\bar{x}_k,\bar{u}_k)^\mathsf{T} V_{xx}(\bar{x}_{k+1}) F_{i}(\bar{x}_k,\bar{u}_k),\\
\mathcal{L} &= \Delta t \textstyle\sum_{i=1}^m F_{i_x}(\bar{x}_k,\bar{u}_k)^\mathsf{T} V_{xx}(\bar{x}_{k+1}) F_{i_u}(\bar{x}_k,\bar{u}_k),\\
\mathcal{S} &= \Delta t \textstyle\sum_{i=1}^m F_{i_x}(\bar{x}_k,\bar{u}_k)^\mathsf{T} V_{xx}(\bar{x}_{k+1}) F_{i}(\bar{x}_k,\bar{u}_k),\\
\mathcal{U} &= \Delta t \textstyle\sum_{i=1}^m F_{i_u}(\bar{x}_k,\bar{u}_k)^\mathsf{T} V_{xx}(\bar{x}_{k+1}) F_{i}(\bar{x}_k,\bar{u}_k).
\end{aligned}$$

Next, a second-order expansion of equation (5) is given as

$$\begin{aligned}
&\min_{u_k}\Big[\mathrm{E}\big[L(x_k,u_k,t_k) + V(x_{k+1},t_{k+1})\big]\Big] \approx\\
&\quad \min_{\delta u_k}[Q(\bar{x}_k,\bar{u}_k) + \delta u_k^\mathsf{T} Q_u(\bar{x}_k,\bar{u}_k) + \delta x_k^\mathsf{T} Q_x(\bar{x}_k,\bar{u}_k)\\
&\qquad + \frac{1}{2}\delta u_k^\mathsf{T} Q_{uu}(\bar{x}_k,\bar{u}_k)\delta u_k + \frac{1}{2}\delta x_k^\mathsf{T} Q_{xx}(\bar{x}_k,\bar{u}_k)\delta x_k\\
&\qquad + \delta u_k^\mathsf{T} Q_{ux}(\bar{x}_k,\bar{u}_k)\delta x_k]
\end{aligned} \tag{6}$$

[1]For ease of exposition, notation for derivatives is condensed to $\nabla_z g = g_z$ and $\nabla_{xz} g = g_{xz}$.

[2]The dependence on time is no longer explicitly stated for ease of exposition.

where

$$
\begin{aligned}
Q(\bar{x}_k, \bar{u}_k) &= V(\bar{x}_{k+1}) + L(\bar{x}_k, \bar{u}_k) + \tfrac{1}{2}\mathcal{T}, \\
Q_x(\bar{x}_k, \bar{u}_k) &= L_x(\bar{x}_k, \bar{u}_k) + A_k^{\mathrm{T}} V_x(\bar{x}_{k+1}) + \mathcal{S}, \\
Q_u(\bar{x}_k, \bar{u}_k) &= L_u(\bar{x}_k, \bar{u}_k) + B_k^{\mathrm{T}} V_x(\bar{x}_{k+1}) + \mathcal{U}, \\
Q_{xx}(\bar{x}_k, \bar{u}_k) &= L_{xx}(\bar{x}_k, \bar{u}_k) + A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) A_k + \mathcal{F}, \\
Q_{uu}(\bar{x}_k, \bar{u}_k) &= L_{uu}(\bar{x}_k, \bar{u}_k) + B_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k + \mathcal{Z}, \\
Q_{xu}(\bar{x}_k, \bar{u}_k) &= L_{xu}(\bar{x}_k, \bar{u}_k) + A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k + \mathcal{L}.
\end{aligned}
$$

The optimal control deviation is now solved for by minimizing equation (6) and is given as

$$\delta u_k^\star = -Q_{uu}^{-1}(Q_u + Q_{xu}^{\mathrm{T}} \delta x_k^\star), \tag{7}$$

where the optimal state deviation is propagated as

$$\delta x_k^\star = A_k \delta x_{k-1}^\star + B_k \delta u_{k-1}^\star, \quad \delta x_0^\star = 0. \tag{8}$$

Note that the optimal control deviation given by (7) contains a feed-forward and a feedback component. Therefore, the term $Q_{uu}^{-1} Q_{xu}^{\mathrm{T}}$ gives the optimal state feedback gain for a particular iteration. The feedback component attempts to keep the system's trajectory near the optimized trajectory, $\bar{x}$. Explicitly, supposing that $x(t_k)$ is the state of the system at time $t_k$ then the input signal is given as

$$\bar{v}(t_k) = \bar{u}(t_k) - Q_{uu}^{-1}(\bar{x}_k, \bar{u}_k) Q_{xu}^{\mathrm{T}}(\bar{x}_k, \bar{u}_k)(x(t_k) - \bar{x}_k) \tag{9}$$

where $\bar{u}$ is the optimized feed-forward input. Plugging $\delta u^\star$ back into equation (6) yields a backward propagating second-order approximation of the value function:

$$
\begin{aligned}
V(\bar{x}_k) &= V(\bar{x}_{k+1}) + L - \tfrac{1}{2} Q_u^{\mathrm{T}} Q_{uu}^{-1} Q_u + \tfrac{1}{2}\mathcal{T}, && (10) \\
V_x(\bar{x}_k) &= Q_x - Q_u^{\mathrm{T}} Q_{uu}^{-1} Q_{xu}^{\mathrm{T}}, && (11) \\
V_{xx}(\bar{x}_k) &= Q_{xx} - Q_{xu} Q_{uu}^{-1} Q_{xu}^{\mathrm{T}}, && (12)
\end{aligned}
$$

where initial conditions are given as $V(\bar{x}_N) = h(\bar{x}_N), V_x(\bar{x}_N) = h_x(\bar{x}_N)$, and $V_{xx}(\bar{x}_N) = h_{xx}(\bar{x}_N)$.

This completes the description of the S-DDP algorithm. The derived optimal control deviation, $\delta u^\star$, is used to update the nominal input as in equation (3). The process can then be repeated using the updated input as the new nominal input.

Recall that the control input is updated using a step size, $\gamma$ (see equation (3)). In the S-DDP (and DDP) implementations presented here an Armijo line search is used to automatically select an appropriate step size [17]. Note that deterministic dynamics can be considered by removing the diffusion vector fields from equation (1), $F_i(x, u) = 0$. The S-DDP algorithm is outlined in Algorithm 1.

## III. Ergodic Exploration

As discussed in the introduction, in an ergodic exploration framework sensor paths are computed to minimized the difference between the distribution of the time-averaged sensor trajectory and a spatial probability distribution function representing information density. That is, in this framework the time a sensor spends sampling a particular region should correspond to the anticipated information density of that region. As a result, coverage or centralized sampling can be expected, without modification, when the information density is uniformly distributed or highly concentrated at a single point, respectively.

**Algorithm 1** S-DDP with an Armijo Line Search

**Require:**
  Initial discrete control input $u(t)$, parameters $\alpha, \beta, \epsilon$ stochastic dynamics (1), and cost function (2)
  **while** Cost updates results in more than $\epsilon$ in difference **do**
    Propagate the discretized (deterministic) trajectory
    Linearize the value function and system dynamics
    Back-propagate equations (10)-(12)
    Compute $\delta U^\star$ and $\delta X^\star$
    **while** $\mathrm{Cost_p} > \mathrm{Cost} + \alpha\beta(v_x \delta X^\star + v_u \delta U^\star)$ **do**
      Find the proposed input $u_{\mathrm{p}} \leftarrow u + \beta^j \delta U^\star$
      Propagate deterministic trajectory, $x_{\mathrm{p}}$
      Find proposed cost $\mathrm{Cost_p} \leftarrow v(x_{\mathrm{p}}, u_{\mathrm{p}}, t)$
      Update $j \leftarrow j + 1$
    **end while**
    Update: $u \leftarrow u_{\mathrm{p}}, x \leftarrow x_{\mathrm{p}}, \mathrm{Cost} \leftarrow \mathrm{Cost_p}$
  **end while**

To formally introduce the concept of ergodicity define a spatial probability distribution function (PDF) $\phi(\chi)$ over an $N$-dimensional explorable domain $\mathcal{M} \subset \mathbb{R}^N$ defined as $[0, L_1] \times [0, L_2] \cdots \times [0, L_N]$. Furthermore, suppose the state of the dynamical sensor described by equation (1) is partitioned as $x = [\chi, \xi]^{\mathrm{T}}$. Then, the partial-state trajectory $\chi(t) \in \mathbb{R}^N,\ t \in [0, T]$ is ergodic if and only if,

$$\lim_{T \to \infty} \frac{1}{T} \int_0^T g(\chi(t))\ \mathrm{d}t = \int_{\mathcal{M}} \phi(\chi) g(\chi)\ \mathrm{d}\chi. \tag{13}$$

for all Lebesgue integrable functions, $g \in \mathcal{L}^1$ [4]. The system state is partitioned in order to distinguish between states that are contained, $\chi$, and not contained, $\xi$, in the explorable domain. For example, velocities are not typically contained in the explorable domain and, therefore, are not included in the definition of ergodicity. In general, any system state can be contained in the explorable domain and, therefore, included in equation (13).

Ergodicity describes the equality of the time-average trajectory distribution to a particular spatial distribution. Conceptually, this requires that the time spent sampling a particular region corresponds to the information density, described by $\phi(\chi)$, of that region. Figure 1 shows the qualitative difference between an ergodic trajectory and an information maximizing trajectory.

Though not discussed here, acquired information can be used to update $\phi(\chi)$. As demonstrated in Reference [7], Bayesian inference and Fisher information tools can be used to update and create information maps. Therefore, as information is gathered new optimized trajectories can be computed based on an updated $\phi(\chi)$.

Note that equation (13) does not lend itself to optimization since the expressed condition must hold for all Lebesgue integrable functions. However, ergodic metrics can be defined by considering Fourier decompositions of the spatial

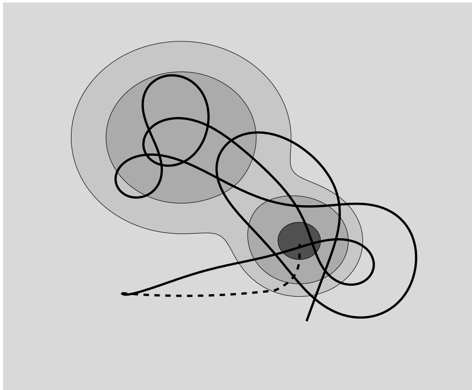

Fig. 1: Conceptual representations of an ergodic exploration trajectory (solid line) and an information maximizing trajectory (dotted line). In ergodic exploration a correspondence between the time average of the sensor trajectory and the information density is desired. Conversely, in an information maximizing trajectory only the region with the highest information density is considered.

and time-averaged trajectory distributions and a finite time horizon, $T$. As a result, optimization becomes feasible and adequate coverage of the $\mathcal{L}^1$ function space can be ensured with sufficient number of Fourier basis functions.

### A. Ergodic Metrics

A metric to quantify the ergodicity of a trajectory $\chi(t)$ with respect to a PDF $\phi(\chi)$ is needed to perform optimization. The ergodic metric used in [5] and [6] measures the differences in the Fourier decompositions of the spatial and time-averaged trajectory distributions:

$$\mathcal{E} = \sum_{k \in \mathcal{K}} \Lambda_k (c_k - \phi_k)^2 \tag{14}$$

where $k = (k_1, k_2, \ldots, k_N) \in \mathbb{Z}^N$ is a multi-index,

$$\mathcal{K} = \{k \in \mathbb{Z}^N : 0 \leq k_j \leq K\}, \tag{15}$$

for some selected $K > 0$, $\Lambda_k = \frac{1}{1+\|k\|^s}$, and $s = \frac{N+1}{2}$. Furthermore, $\phi_k$ is the Fourier coefficient associated with the basis function $G_k(\chi)$ of the distribution $\phi(\chi)$ calculated as

$$\phi_k = \int_{\mathcal{M}} \phi(\chi) G_k(\chi) \, \mathrm{d}\chi \tag{16}$$

and $c_k$ is the time-averaged value of the basis function $G_k(\chi)$ evaluated over the trajectory $\chi(t)$,

$$c_k = \frac{1}{T} \int_0^T G_k(\chi(t)) \, \mathrm{d}t. \tag{17}$$

The Fourier basis functions used are defined as $G_k(\chi) = \frac{1}{h_k} \prod_{i=1}^N \cos(\frac{k_i \pi}{L_i} \chi_i), \quad k \in \mathcal{K}$ where $h_k$ is a normalizing factor. Note that the ergodic metric places a higher importance on low frequency modes.

Due to the structure of the ergodic metric defined in (14), trajectory optimization formulations that incorporate the metric into the cost function are not in the form of a Bolza problem. It has been shown that a Bolza problem can be recovered if projection-based optimization techniques are used [6], [7]. However, a Bolza problem cannot be recovered when the S-DDP algorithm is used since a quadratic approximation of the cost-to-go function is needed.

In order to formulate a new ergodic metric a set of auxiliary system states are now introduced

$$y_k(t) = \frac{1}{t} \int_0^t G_k(\chi(\tau)) \, \mathrm{d}\tau - \int_{\mathcal{M}} \phi(\chi) G_k(\chi) \, \mathrm{d}\chi. \tag{18}$$

The auxiliary system state, $y_k(t)$, provides a distance from ergodicity at time $t$. Note that $y_k(t)$ is not defined at $t = 0$ and, therefore, a new auxiliary system state is introduced

$$z_k(t) = \int_0^t G_k(\chi(\tau)) \, \mathrm{d}\tau - t \int_{\mathcal{M}} \phi(\chi) G_k(\chi) \, \mathrm{d}\chi \tag{19}$$

where $z_k(t) = t y_k(t)$. Taking the time derivative of (19) results in the following equations of motion

$$\dot{z}_k(t) = G_k(\chi(t)) - \phi_k, \quad z_k(t_0) = 0, \quad k \in \mathcal{K}. \tag{20}$$

Finally, the new ergodic metric is given as

$$\mathcal{E} = \tfrac{q_{\mathrm{f}}}{2} z(T)^{\mathrm{T}} \Lambda z(T) + \int_0^T \tfrac{q}{2} z(\tau)^{\mathrm{T}} \Lambda z(\tau) \, \mathrm{d}t \tag{21}$$

where $q, q_{\mathrm{f}} \geq 0$, $z = [z_1, z_2, \ldots, z_{|\mathcal{K}|}]^{\mathrm{T}}$ and $\Lambda \in \mathbb{R}^{|\mathcal{K}| \times |\mathcal{K}|}$ is a diagonal matrix whose elements are defined as in equation (14). If $q = 0$ and $q_{\mathrm{f}} = \frac{2}{T^2}$ then the ergodic metrics (14) and (21) are equivalent.

### B. The Trajectory Optimization Problem

The trajectory optimization problem can now be formally introduced: Minimize the cost function

$$v(x, z, u, t) = \mathrm{E}\Big[ \tfrac{q_{\mathrm{f}}}{2} z^{\mathrm{T}}(t_1) \Lambda z(t_1) + \int_{t_0}^{t_{\mathrm{f}}} \left( \tfrac{q}{2} z^{\mathrm{T}}(\tau) \Lambda z(\tau) + \tfrac{1}{2} u^{\mathrm{T}}(\tau) R u(\tau) \right) \mathrm{d}\tau \Big] \tag{22}$$

subject to the stochastic system dynamics augmented with auxiliary system states

$$\begin{bmatrix} \mathrm{d}x \\ \mathrm{d}z \end{bmatrix} = \begin{bmatrix} f(x, u) \\ G(x) - \phi \end{bmatrix} \mathrm{d}t + \sum_{i=1}^m \begin{bmatrix} F_i(x, u) \\ 0 \end{bmatrix} \mathrm{d}\omega_i \tag{23}$$

where $R \in \mathbb{R}^{p \times p}$ is a positive definite matrix, $G(x) = [G_1(\chi), G_2(\chi), \ldots, G_{|\mathcal{K}|}(\chi)]^{\mathrm{T}}$, and $\phi = [\phi_1, \phi_2, \ldots, \phi_{|\mathcal{K}|}]^{\mathrm{T}}$.

Note that the proposed set of auxiliary variables enables the optimization problem to be posed in Bolza form and, thereby, allows the S-DDP algorithm to be used.

## IV. Numerical Experiments

In this section, the trajectory optimization problem given in equations (22) and (23) is numerically solved using the S-DDP and DDP optimization algorithms. It is shown that the S-DDP algorithm can be used to mitigate stochastic effects and results in a smaller and a more predictable ergodic metric particularly when the additive noise is significantly large. That is, the mean (expectation) and variance of the resulting distribution of the ergodic metric are smaller. Furthermore, the same is true with the cost function (22)

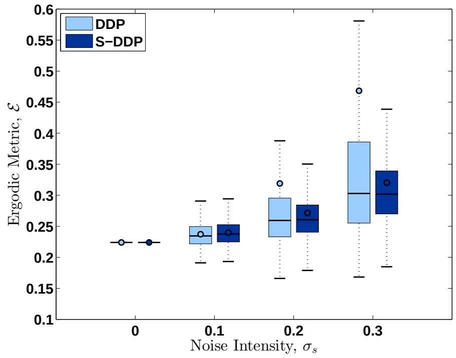


Fig. 2: Distributions of the ergodic metric given in equation (21) for noise intensities, $\sigma_s$, ranging from 0 to 0.3. Circle markers represent the mean of the resulting data set. Data outliers are not shown. While the performance of the algorithms are similar when the noise intensity is low, the S-DDP algorithm results in a smaller ergodic metric variance as the noise intensity increases.

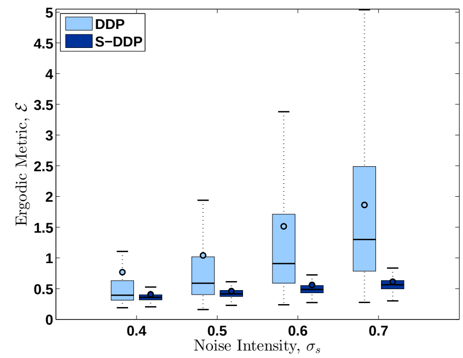


Fig. 3: Distributions of the ergodic metric given in equation (21) for noise intensities, $\sigma_s$, ranging from 0.4 to 0.7. Circle markers represent the mean of the resulting data set. Data outliers are not shown. The mean and the variance of the ergodic metric are significantly smaller, in particular for larger noise intensities, when the S-DDP algorithm is used. Furthermore, when the typical DDP algorithm is utilized the distribution of the metric exhibits a greater change when the noise intensity is increased.

evaluated with the resultant closed-loop input. The state vector of the considered stochastic dynamical model is $x(t) = [X(t), Y(t), V(t), \theta(t)]^{\mathrm{T}}$ where $X(t)$ and $Y(t)$ are the sensor's Cartesian coordinates, $\theta(t)$ is the heading angle, and $V(t)$ is the velocity of the sensor along the heading angle. The system has two inputs $u(t) = [u_1, u_2]^{\mathrm{T}}$ and is affected by two stochastic processes. The state of the system evolves as

$$\mathrm{d}X(t) = V(t)\cos(\theta(t))\ \mathrm{d}t, \tag{24}$$

$$\mathrm{d}Y(t) = V(t)\sin(\theta(t))\ \mathrm{d}t, \tag{25}$$

$$\mathrm{d}\omega(t) = 10u_1(t)\ \mathrm{d}t + V^2(t)\ \mathrm{d}\omega_1, \tag{26}$$

$$\mathrm{d}V(t) = u_2(t)\ \mathrm{d}t + V^2(t)\ \mathrm{d}\omega_2, \tag{27}$$

where $\omega_1, \omega_2 \sim \mathcal{N}(0, \sigma_s^2)$ are independent random variables. Initial conditions were set as $X(t_0) = 8$, $Y(t_0) = 8$, $\theta(t_0) = \pi/2$, $V(t_0) = 0$. Furthermore, $\phi(\chi)$ was selected to be a Gaussian PDF with expected values $\mu_x, \mu_y = 5$ and variances $\sigma_x, \sigma_y = 0.5$ defined on the domain $\mathcal{M} = [0, 10] \times [0, 10]$. Set (15) was constructed for $K = 8$ and a time horizon of $T = 10$ was used. Furthermore, $q = 100$, $q_{\mathrm{f}} = 1000$, and $R = \mathrm{diag}(0.1, 0.1)$.

The noise intensity $\sigma_s$ was varied to study its effect on the S-DDP and DDP algorithms. The DDP solution was only computed once since it does not account for stochastic effects while the S-DDP solution was recomputed for each value of $\sigma_s$. A series of 1000 Monte Carlo simulations (Brownian noise history varied) were conducted for each $\sigma_s$ selected. The optimal feedback gains produced by the S-DDP and DPP algorithms were utilized in these simulations. However, only the gains associated with sensor states where used and those associated with the auxiliary variables were excluded.

Figures 2 and 3 show a summary of the conducted numerical experiments. Notice that the mean and variance of the ergodic metric is lower when the S-DDP algorithm is used. Furthermore, the difference between the S-DDP and DDP algorithms becomes more apparent as the noise intensity is increased. Figure 4 shows the distributions of the cost function if the closed-loop input cost is considered. That is, the input signal used to evaluate the control is given by equation (9). As with the ergodic metric, the mean and variance of the closed-loop cost function is lower when the S-DDP algorithm is used.

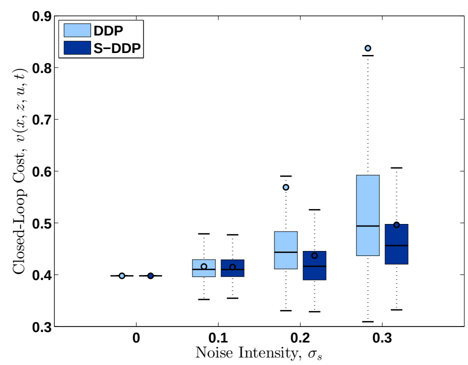


Fig. 4: Distributions of the cost function given in equation (22) evaluated with the resultant close-loop input for noise intensities, $\sigma_s$, ranging from 0 to 0.3. Circle markers represent the mean of the resulting data set. Data outliers are not shown. While the performance of the algorithms are similar when the noise intensity is low, the S-DDP algorithm results in a smaller closed-loop cost function variance as the noise intensity increases.

Figures 5 and 6 show the optimized trajectories at different noise intensities. Notice that when no stochastic effects are considered a trajectory similar to those reported in previous work is computed. However, the computed trajectory changes when stochasticity is considered. Figure 6 shows how the

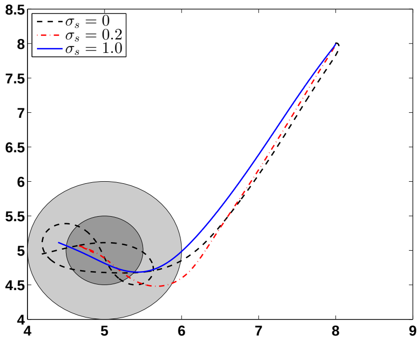


Fig. 5: Optimized trajectories for three noise intensities computed using the S-DDP algorithm. The gray discs indicate the $(\sigma_x, \sigma_y)$ and $(2\sigma_x, 2\sigma_y)$ level sets. The optimized trajectory is largely dependent on the noise intensity. As the noise intensity increases, the trajectory is reshaped in order to mitigate the amount of induced noise while still reducing the ergodic metric.

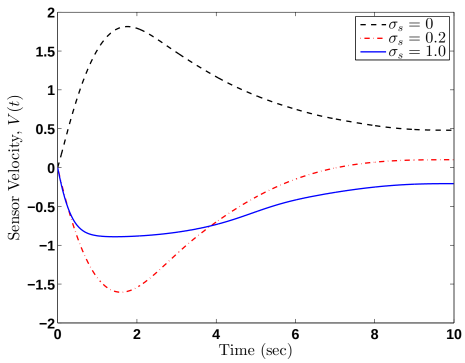


Fig. 6: Optimized velocity trajectories for three noise intensities computed using the S-DDP algorithm. The optimized velocity trajectory is largely dependent on the noise intensity. As the noise intensity increases, the S-DDP algorithm attempts to limit the amount of induced noise by reducing the magnitude of $V(t)$ (see equations (26) and (27)).

S-DDP algorithm attempts to reduce the noise intensity by reducing $V^2(t)$. Recall that the cost function considered is an expectation and a policy that induces an unnecessary level of noise (large $V^2(t)$) is detrimental. On the other hand, a trajectory in which $V(t) = 0$ is not optimal since no ergodic coverage is achieved. Fundamentally, there exists a trade-off between completing the task and inducing stochastic effects.

## V. Conclusion

This paper introduces a trajectory optimization approach for ergodic exploration in the presence of stochastic sensor dynamics. A set of auxiliary variables are proposed in order to pose the ergodic exploration optimization problem in a Bolza formulation. As a result, the problem becomes solvable via the DDP and S-DDP algorithms. Numerical studies demonstrated that the S-DDP algorithm is able to mitigate the effects of stochastic sensor dynamics better than standard DDP. Trajectories are optimized in order to limit the amount of induced noise while still reducing the ergodic metric.

The proposed approach is used to attenuate the effects of stochasticity when ergodic exploration is performed. However, an entirely different perspective can be adopted. Can designed stochasticity be imposed on a system such that its trajectories become ergodic? That is, can ergodic exploration occur if induced noise is considered a control signal? One can imagine, for highly noisy systems, that the available stochasticity can be utilized, instead of mitigated, to achieve ergodicity. This perspective will involve an entirely different solution approach and the feasibility of physical implementation will need to be investigated.

## References


[1] J. Cooper and M. Goodrich, "Towards combining UAV and sensor operator roles in UAV-enabled visual search," in *International Conference on Human-Robot Interaction*, 2008, pp. 351–358.

[2] J. R. Solberg, K. M. Lynch, and M. A. MacIver, "Active electrolocation for underwater target localization," *The International Journal of Robotics Research*, vol. 27, no. 5, pp. 529–548, 2008.

[3] J. Vander Hook, P. Tokekar, and V. Isler, "Cautious greedy strategy for bearing-based active localization: Experiments and theoretical analysis," in *International Conference on Robotics and Automation*, 2012, pp. 1787–1792.

[4] S. E. Scott, T. C. Redd, L. Kuznetsov, I. Mezìc, and C. K. Jones, "Capturing deviation from ergodicity at different scales," *Physica D: Nonlinear Phenomena*, vol. 238, no. 16, pp. 1668 – 1679, 2009.

[5] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 45, pp. 432 – 442, 2011.

[6] L. M. Miller and T. D. Murphey, "Trajectory optimization for continuous ergodic exploration," in *American Control Conference*, 2013, pp. 4196–4201.

[7] L. M. Miller, Y. Silverman, M. A. MacIvler, and T. D. Murphey, "Ergodic exploration of distributed information," *Transactions on Robotics*, conditionally accepted, available online.

[8] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte, "Information based adaptive robotic exploration," in *International Conference on Intelligent Robots and Systems*, 2002, pp. 540–545.

[9] D. H. Jacobson and D. Q. Mayne, *Differential dynamic programming*. Elsevier Publishing Company, 1970.

[10] E. Todorov and W. Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference*, 2005, pp. 300 –306.

[11] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *International Conference on Informatics in Control, Automation, and Robotics*, 2004, pp. 222–229.

[12] G. Lantoine and R. P. Russell, "A hybrid differential dynamic programming algorithm for robust low-thrust optimization," in *AAS/AIAA Astrodynamics Specialist Conference and Exhibit*, 2008.

[13] S. Yakowitz, "The stagewise Kuhn-Tucker condition and differential dynamic programming," *IEEE Transactions on Automatic Control*, vol. 31, no. 1, pp. 25–30, 1986.

[14] J. Morimoto and C. G. Atkeson, "Minimax differential dynamic programming: Application to robust biped walking," in *Intelligent Robots and Systems*, 2003, pp. 1927 –1932.

[15] G. De La Torre, "Autonomous suspended load operations via trajectory optimization and variational integrators," Ph.D. dissertation, Georgia Institute of Technology, 2015.

[16] E. Theodorou, Y. Tassa, and E. Todorov, "Stochastic differential dynamic programming," in *American Control Conference*, 2010, pp. 1125–1132.

[17] C. T. Kelley, *Iterative methods for optimization*. SIAM, 1999.

[18] G. De La Torre and E. A. Theodorou, "Stochastic variational integrators for system propagation and linearization," in *IMA Conference on Mathematics of Robotics*, 2015.