Noname manuscript No.
(will be inserted by the editor)

Stochastic Optimal Coordination of Small UAVs for Target Tracking using Regression-based Dynamic Programming

Steven A. P. Quintero, Michael Ludkovski,
João P. Hespanha

the date of receipt and acceptance should be inserted later

Abstract We study the problem of optimally coordinating multiple fixed-wing UAVs to perform vision-based target tracking, which entails that the UAVs are tasked with gathering the best joint vision-based measurements of an unpredictable ground target. We utilize an analytic expression for the error covariance associated with the fused measurements of the target's position, and we employ stochastic fourth-order models for all vehicles, thereby incorporating a high degree of realism into the problem formulation. While dynamic programming can generate an optimal control policy that minimizes the expected value of the fused geolocation error covariance over time, it is accompanied by significant computational challenges due to the curse of dimensionality. In order to circumvent this challenge, we present a novel policy generation technique that combines simulation-based policy iteration with a robust regression scheme. The resulting control policy offers a significant advantage over alternative approaches and shows that the optimal control strategy involves coordinating the UAVs' distances to the target rather than their viewing angles, which had been a common practice in target tracking.

Keywords target tracking; unmanned aerial vehicle; autonomous vehicle; regression Monte Carlo; motion planning; probabilistic planning

1 Introduction

Small unmanned aerial vehicles (UAVs) are relatively inexpensive mobile sensing platforms capable of reliably and autonomously performing numerous tasks, such

S. Quintero
Department of Electrical and Computer Engineering,
University of California, Santa Barbara, CA 93106 USA,
E-mail: quintero@ece.ucsb.edu

M. Ludkovski
Department of Statistics and Applied Probability,
University of California, Santa Barbara, CA 93106, USA

J. Hespanha
Department of Electrical and Computer Engineering,
University of California, Santa Barbara, CA 93106 USA

as mapping, search and rescue, surveillance and tracking, and real-time monitoring. One problem of particular interest is that of using small, fixed-wing UAVs to perform *vision-based target tracking*, which entails that one or more camera-equipped UAVs is responsible for autonomously tracking a moving ground target.

In vision-based target tracking, image processing software determines the centroid pixel coordinates of a target moving in the image frame. Given these pixel coordinates, the intrinsic and extrinsic camera parameters, and the terrain data, one can estimate the three-dimensional location of the target in inertial coordinates and compute the associated error covariance. This is the process of *geolocation* for video cameras [1]. The geolocation error is highly sensitive to the relative position of a UAV with respect to the target. When a UAV is far from the target, relative to its height above the target, the associated error covariance is significantly elongated in the viewing direction. The smallest geolocation error comes when the UAV is directly above the target, in which case the associated covariance is circular. While a UAV would ideally hover directly above the target to minimize the error, the relative dynamics between a UAV and target typically preclude this viewing position from being held over a period of time.

To mitigate the effects of a single UAV's inability to maintain close proximity to the target, one can employ multiple UAVs to gather the best *joint* measurements. In this scenario, the objective is to minimize the *fused* geolocation error covariance of the target position estimate obtained by fusing the individual geolocation measurements. Thus, in this work, we seek optimally coordinated behavior between two UAVs aimed at improving the estimate of the target state.

The fused geolocation error is small when at least one UAV is close to the target and only slightly less when both aircraft are directly above the target. When both UAVs are far from the target relative to their altitudes, the fused geolocation error is greatly reduced when the UAVs have orthogonal viewing angles, though this error is still significantly greater than when at least one UAV is on top of the target. Of course, these configurations are static, yet in a realistic scenario, the target motion is unpredictable and the UAVs have limited control effort and experience stochasticity in their dynamics.

The purpose of this work is thus to present and study an effective solution to the problem of optimally coordinating two UAVs to track a moving ground target under fairly realistic conditions. More specifically, the objective for the camera-equipped UAVs is to gather the best *joint* vision-based measurements of a randomly moving ground target whilst themselves being subject to limited control effort and experiencing stochasticity in their dynamics. The class of UAVs under consideration are hand or catapult launched fixed-wing aircraft that fly at a constant altitude and have an autopilot that regulates roll angle, airspeed, and altitude to the desired setpoints via internal feedback loops. Furthermore, these underactuated aircraft are assumed to fly at a constant airspeed since the range of permissible airspeeds for such small aircraft may be very limited, as noted in [2] and §5.1 of [3]. In addition, frequent changes in airspeed may be either undesirable for fuel economy or simply unattainable. The roll angle setpoint is hence the sole control input that affects the horizontal plant dynamics. The target is modeled as a nonholonomic vehicle that randomly turns and accelerates.

As determining the optimal control policy (feedback law) is a challenging problem in the area of stochastic optimal control, we now take note of the numerous

solutions that have been proposed over the past decade for similar problems in the area of target tracking.

1.1 Related Work

Much research has proposed coordinated target tracking controllers in a deterministic setting without directly optimizing mission performance with respect to a desired objective function. For two UAVs, a generally accepted practice is to have the UAVs orbit the target at a nominal standoff distance (to remain outside a critical threat range) and maintain an angular separation of 90°. The 90° separation angle minimizes the joint / fused geolocation (target localization) measurement error for the given standoff distance, as the individual measurement error ellipses are orthogonal [4]. These principles give rise to what is henceforth referred to as cooperative (or coordinated) standoff tracking, which constitutes the majority of the work in the general area of coordinated target tracking. When more than two UAVs are considered, the goal generally becomes having the group achieve a uniform angular separation on a circle centered at the target.

Standoff tracking has been a longstanding goal in the general area of target tracking and has been addressed using numerous approaches that include "Good Helmsman" steering [5], Lyapunov guidance vector fields [6], nonlinear model predictive control [7], nonlinear feedback [8], and methods combining vector field guidance with adaptive control [9, 10]. Since multiple fixed-speed aircraft cannot maintain a uniform angular spread at a fixed distance from a constant-velocity target, works such as [3] and [11] have explored the notion of spreading agents uniformly in time along a periodic trajectory at a fixed distance from the moving target.

A number of approaches have employed stochastic optimal control to mitigate the effects of stochastic target motion while also respecting a maximum turn-rate / bank angle. In [12], Anderson and Milutinović studied the problem of optimal standoff tracking in the continuous time setting and model the target as a Brownian particle and the UAV as a deterministic Dubins vehicle. By minimizing the expected cost of the total squared distance error discounted over an infinite horizon, the authors generate an optimal bang-bang turn-rate controller that is highly robust to unpredictable target motion. In [13], the authors studied the problem of having a single UAV optimally perform vision-based target tracking with a limited sensing region, wherein the cost objective was a function of the desired viewing geometry. A comparison was made between a game theoretic approach (addressing evasive target motion) and a stochastic optimal control approach (addressing random target motion) and showed that the latter approach performed better in actual field tests. Hence, in the present work, we use a refined version of the stochastic kinematic UAV model from [13] and adopt a similar stochastic target model.

Others have employed optimal control to study optimal UAV coordination when the objective is to improve target state estimation. Miller et al. pose the problem of multiple UAVs tracking multiple targets as a partially observable Markov decision process (POMDP) in [14] and present a new approximate solution, as nontrivial POMDP problems are typically intractable to solve exactly [15]. In [16], Stachura et al. studied the problem of two variable-airspeed UAVs with bearing-

only sensors tracking a stochastic ground target in the presence of packet losses in the communication with the base station, where target state estimation takes place. The solution involved an online receding horizon controller that maximized the expected information (inverse covariance) of the target state estimate in an extended information filter over a short planning horizon, showing that one UAV will act as a relay to the base station when the target is far from the base. In [17], Ding et al. studied the problem of optimally coordinating two camera-equipped Dubins vehicles with bang-off-bang turn-rate control to maximize the geolocation information of a stochastic ground target over a short planning horizon. The results showed that a 90° separation in the viewing angle was essential in the case of terrestrial pursuit vehicles and less pronounced with airborne pursuit vehicles.

We emphasize the fact that the preceding optimal control approaches illustrate a trend among optimization-based coordination strategies. Namely, shorter planning horizons are often considered to reduce the computational complexity of the dynamic optimization. While this is justified from a pragmatic standpoint, short horizons are not adequate for the cost function considered here. In particular, since the main feature of the cost function is that it (in effect) penalizes the minimum UAV distance to the target, a short planning horizon inhibits the UAVs from realizing the long term benefits of distance coordination, i.e., keeping their peak distances from the target out of phase. Moreover, if the target is traveling considerably slower than the UAVs, the aircraft must perform loops to remain close to it and must realize the long term benefit of doing these loops in a coordinated fashion.

In [18], however, the authors optimized the coordination of two UAVs over long planning horizons of at least one minute by minimizing the fused geolocation error covariance, thereby gathering the best joint vision-based measurements of the target. The results showed that coordination of the distances to target is more effective for achieving the said goal than the traditional practice of solely coordinating viewing angles, thus motivating the use of optimization-based control strategies with longer planning horizons. These studies were conducted in a deterministic setting for UAVs that used bang-off-bang turn-rate control to track a constant-velocity target. Lastly, we note that a number of works including [19] and [20] have proposed using sinusoidal turn-rate control inputs that approximate the optimal behavior of [18] at higher speeds; however, our tests indicate that the proposed oscillatory control strategies appear to be non-robust to stochastic target motion while roll dynamics are not considered.

In all of the preceding works, at least one or more assumptions are made that impose severe practical limitations. Namely, the works mentioned thus far assume at least one of the following:

1. Coordinated circular trajectories are optimal, namely those trajectories resulting from standoff tracking.
2. Input dynamics are first order and roll dynamics have been ignored.
3. The UAV airspeed can be changed quickly and reliably over a significant range
4. Target motion is predictable
5. Short/greedy planning horizons are adequate for optimal tracking

The present paper removes all of these assumptions to promote a more practical solution that yields optimal coordination under more realistic conditions, namely higher order stochastic dynamics with explicit input constraints.

1.2 Contributions

To remedy the aforementioned simplifications / assumptions, we formulate a stochastic optimal control problem whose objective is for two fixed-wing UAVs to gather the best joint vision-based measurements of a randomly moving ground target over a sufficiently long planning horizon. The cost function utilizes an analytical expression for the geolocation error covariance while fourth order stochastic kinematic models are utilized for all vehicles to describe realistic vehicle dynamics. More specifically, the stochasticity in the ground vehicle model encompasses the unpredictable nature of the target motion while that of the UAV model addresses environmental disturbances, e.g., wind gusts, as well as unmodeled dynamics. Most importantly, this aircraft model has produced successful field test results in related target tracking applications [13,21]. Lastly, an upper limit is imposed on the maximum absolute roll-angle setpoint, which is the sole control input for each aircraft.

To determine the optimal control policy, one must solve a moderate dimensional stochastic optimal control problem for which grid-based approximations to the dynamic programming value function are infeasible. Hence, we present a regression-based dynamic programming technique that has been adapted from the simulation-based policy iteration technique known as regression Monte Carlo (RMC). More specifically, the original RMC algorithm has been modified to become a policy *generation* technique so as to remove the need for an initial policy that is close to the optimal. In addition, to address the high dimensionality of the system dynamics, we use a partitioned robust regression scheme (based on work in [22]) that is both fast and scalable with the number of Monte Carlo simulations. Since the overall method generates an approximately optimal control policy offline, this controller can be readily implemented in realtime. While the original RMC algorithm has been successfully applied to stochastic control problems in finance and epidemic management, here we demonstrate its utility for high-dimensional autonomous vehicle applications.

Lastly, we provide a thorough demonstration of the nature and performance of the resulting control policy. First, we show the benefits of the proposed approach over alternative methods, including an uncoordinated control strategy in which the multiple UAVs solve independent optimizations and (non-optimal) stand-off tracking. Second, we show that while viewing-angle coordination is certainly facilitated by the optimal policy, the more pronounced behavioral characteristic of the optimal strategy is the coordination of distances to the target. Overall, we show that optimization-based control techniques can produce results that differ from and significantly outperform traditional techniques relying on heuristics.

1.3 Paper Outline

The remainder of the paper is organized as follows. Section 2 describes the main components of the stochastic optimal control problem, namely the stochastic kinematic models for the vehicles, the fused geolocation error covariance, and the overall state space. Section 3 firstly provides an overview of the basic dynamic programming solution to the problem and secondly details the more sophisticated regression Monte Carlo algorithm. The remainder of the section is devoted to describing the partitioned robust regression tool. Section 4 discusses some of the pa-

rameters and modifications to the algorithm that are specific to the present target tracking application. Section 5 opens with a description of the overall simulation setup for a realistic scenario. The remainder of the section provides a comparison with alternative methods to establish the benefit of the proposed approach; the section concludes with an analysis of the coordination behavior. Section 6 provides conclusions of the overall work and discusses opportunities for future research.

2 Problem Formulation

Consider a group of two UAVs tasked with autonomously tracking an unpredictable moving target vehicle using gimbaled video sensors. The UAVs fly at a constant altitude and fixed nominal airspeed yet experience stochasticity in their dynamics. The target is a nonholonomic ground vehicle that moves on the ground and exhibits stochasticity in both its turning and acceleration. The main objective is to optimize the coordination of the UAVs to gather the best joint vision-based measurements of the target. Since all vehicles experience stochasticity in their dynamics, the dynamic optimization is inherently a stochastic optimal control problem, whose key components are a description of the stochastic evolution of the states and the cost associated with each state. Accordingly, we first describe the stochastic kinematic models for the UAVs and target and then discuss the video measurement model and the associated geolocation error covariance, which will constitute the cost.

2.1 Overview of Stochastic Dynamics

The UAV and target states are assumed to evolve stochastically according to discrete-time Markov Decision Processes. Accordingly, the probability of transitioning from UAV j's state $\boldsymbol{\xi}_j$ at the current time k to the next state $\boldsymbol{\xi}_j'$ at time $k+1$ under control action u_j is given by the controlled state transition probability function $p_a(\boldsymbol{\xi}_j' \,|\, \boldsymbol{\xi}_j, u_j)$. For simplicity, we assume that the UAVs have identical stochastic kinematics, though this can be easily generalized to a heterogenous team. Likewise, the probability of transitioning from the current target state $\boldsymbol{\eta}$ to the next target state $\boldsymbol{\eta}'$ is given by the state transition probability function $p_g(\boldsymbol{\eta}' \,|\, \boldsymbol{\eta})$.

Rather than deriving explicit formulas for these state transition probabilities, which are not needed for our approach, we use the agents' kinematics to draw Monte Carlo samples $\tilde{\boldsymbol{\xi}}_1^{(i,u_1)}$, $\tilde{\boldsymbol{\xi}}_2^{(i,u_2)}$, and $\tilde{\boldsymbol{\eta}}^{(i)}$, $i \in \{1, 2, \ldots, N_p\}$, from the conditional probability density functions of UAV 1, UAV 2, and the target, respectively. These Monte Carlo samples provide an empirical characterization of the stochastic dynamics of the overall system's state $\boldsymbol{z}$, which includes UAV states relative to those of the target and evolves according to a controlled state transition probability function $p(\boldsymbol{z}' \,|\, \boldsymbol{z}, \boldsymbol{u})$. The ability to sample this state transition probability will suffice to effectively approximate the dynamic programming solution.

2.2 UAV Dynamics

In practice, the UAVs are subject to environmental disturbances, such as wind gusts, that introduce stochasticity into the dynamics. Although a real UAV's kinematics are most accurately described by a 6 degree-of-freedom (DoF) aircraft model, we use a 4-state stochastic model of the kinematics, in which stochasticity accounts for the effects of both unmodeled dynamics (arising from the reduced 4^{th} order model) and environmental disturbances. The model was successfully employed in field tests both for a single UAV performing vision-based target tracking with sensing limitations in [13] and for flocking with multiple UAVs in [21].

While the majority of the work on target tracking uses continuous-time motion models, this work treats the optimization in discrete time. Thus, each UAV is initially modeled by fourth-order continuous-time dynamics, and then a T_s-second zero-order hold (ZOH) is applied to each UAV's control input to arrive at the discrete-time dynamics for the aircraft.

Each UAV is assumed to have an autopilot that regulates roll angle, airspeed, and altitude to the desired setpoints via internal feedback loops. In our model, UAV j flies at a fixed airspeed s_j and at a constant altitude h_j above the ground. UAV j's planar position $(x_j, y_j) \in \mathbb{R}^2$ and heading $\psi_j \in \mathbb{S}^1$ are measured in a local East-North-Up (ENU) earth coordinate frame while its roll angle $\phi_j \in \mathbb{S}^1$ is measured in a local North-East-Down (NED) body frame. In the latter coordinate frame, the x-axis points out of the nose, the y-axis points out of the right wing, and the z-axis completes the right-handed coordinate frame. As in [13] and [21], the roll/bank angle of the aircraft is the only controllable state that affects the horizontal plant dynamics. The roll angle is controlled through setpoint control, which entails that a given control policy determines the *desired* roll angle u_j that is provided to the autopilot's low-level control loops.

The development of the stochastic discrete-time kinematic model for the UAV begins with the deterministic continuous-time model:

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ \psi \\ \phi \end{pmatrix} = \begin{pmatrix} s\cos\psi \\ s\sin\psi \\ -(\alpha_g/s)\tan\phi \\ f(\phi,u) \end{pmatrix}, \tag{1}$$

where α_g denotes the acceleration due to gravity and the subscript j denoting the UAV index has been omitted, as the same dynamical model is used for both UAVs. The quantity $f(\phi, u)$ denotes the roll dynamics, and could be, for example, $f(\phi, u) = -\alpha_\phi(\phi - u)$ with $\alpha_\phi > 0$. In this case, $1/\alpha_\phi > 0$ can be regarded as the time constant corresponding to the autopilot control loop that regulates the actual roll angle ϕ to the *current* roll-angle setpoint u. However, we actually use a much more detailed model for the roll dynamics wherein we apply a T_s second zero-order hold (ZOH) on the roll-angle setpoint u and sample roll trajectories from a high-fidelity flight simulator that utilizes an aircraft model with 6 degrees of freedom. In this sampling process, we assume $u \in C$, where

$$C := \{0, \pm\Delta, \pm 2\Delta\},$$

and that the changes in u from one ZOH period to the next belong to the set $\{0, \pm\Delta\}$. Thus, $\Delta > 0$ is in essence the maximum allowable change in the roll-angle

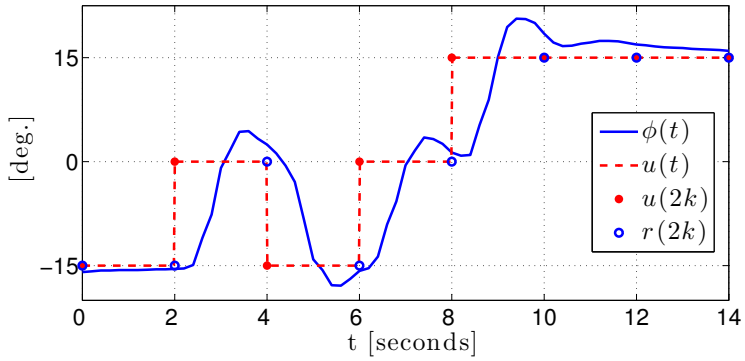


Fig. 1 Monte Carlo simulations to sample roll trajectories. Once every $T_s = 2$ seconds the roll-angle setpoint is randomly changed to $u(kT_s) \in U(r(kT_s))$, where each element of $U(r(kT_s))$ occurs with equal probability and $\Delta = 15^\circ$ is the maximum allowable change in roll-angle setpoints.

setpoint from one ZOH period to the next. In order to regulate the maximum allowable change in the setpoint, we must keep track of the previous setpoint $u_{k-1} = u(kT_s - T_s)$, where $k \in \mathbb{Z}_{\geqslant 0}$. To this end, we note that an autopilot with a properly tuned controller for roll will approximately achieve the setpoint at the end of the ZOH period, i.e., $\forall k \in \mathbb{Z}_{\geqslant 0}$, $\phi(kT_s + T_s) \approx u(kT_s)$. Moreover, we assume this is the case and define the discretized roll angle at time k as

$$r_k := \arg\min_{c \in C} |c - \phi_k|,$$

and assume $r_{k+1} = u_k$, which means $r_k = u_{k-1}$. Thus, our requirements that $u_k \in C$ and $(u_k - u_{k-1}) = (u_k - r_k) \in \{0, \pm\Delta\}$ are summarized by requiring $u_k \in U(r_k)$, where for $c \in C$

$$U(c) := \{c, c \pm \Delta\} \cap C.$$

We sample roll trajectories from a high fidelity flight simulator per the description of Figure 1 and generate a collection $\Phi(r, u)$ of N_p roll-angle trajectories $\phi_i(\tau, r, u)$, where $i \in \{1, 2, \ldots, N_p\}$ and $\tau \in [0, T_s]$, for each combination of $r \in C$ and $u \in U(r)$. Figure 2 illustrates a typical collection of roll-angle trajectories for particular values of r and u. One should observe that all of the roll-angle trajectories in the example approximately achieve the setpoint in accordance with the assumption that $r_{k+1} = u_k$. In like manner, all of the roll trajectories used in the model have this property. Since r well approximates the roll angle ϕ at discrete time instances $t = kT_s$ seconds for all $k \in \mathbb{Z}_{\geqslant 0}$, we define the state of a UAV as $\boldsymbol{\xi} := (x, y, \psi, r) \in \mathbb{R}^4$.

To make the aircraft model more realistic we also introduce stochasticity into the airspeed s, which is drawn from a symmetric triangle distribution (wherein the mode is equidistant from the support bounds) that is centered at a nominal value of μ_s and has support over the interval $[\mu_s - \sigma_s\sqrt{6}, \mu_s + \sigma_s\sqrt{6}]$, where σ_s denotes the standard deviation of the distribution. Also, the airspeed s in (1) is assumed to be constant over each ZOH period and independent across different ZOH periods.

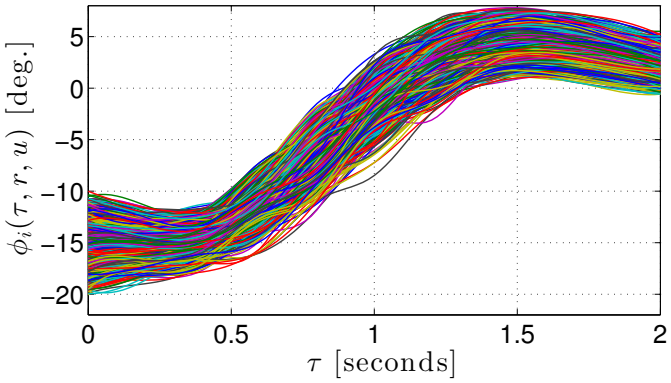


Fig. 2 One hundred roll-angle trajectories over a $T_s = 2$ second ZOH period resulting from an increase of $\Delta = 15^\circ$ from the previous roll-angle setpoint of $u_{k-1} = r_k = -15^\circ$. Moreover, the current setpoint is $u_k = 0^\circ$.

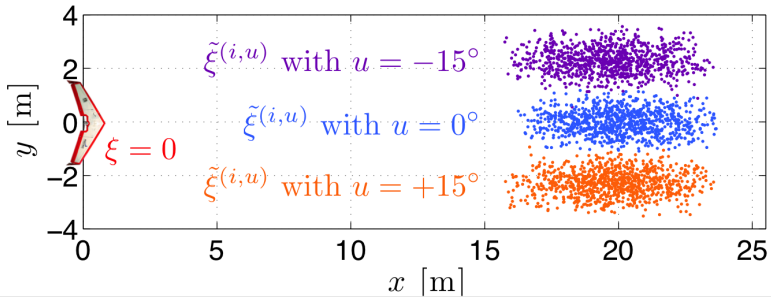


Fig. 3 Sample trajectories generated from the stochastic kinematic model for the UAV with s distributed according to a triangle distribution with mean $\mu_s = 10$ [m/s] and standard deviation $\sigma_s = 4/5$ [m/s]. Also, $T_s = 2$ seconds, and $\Delta = 15^\circ$. The initial UAV state is identically zero. For each $u \in U(0^\circ) = \{0^\circ, \pm 15^\circ\}$, 1,000 sample trajectories were generated. For each command, the vertical spread in final UAV positions is due to sampling different roll trajectories while the horizontal spread results from stochastic airspeed.

This modeling technique allows us to generate samples $\tilde{\boldsymbol{\xi}}^{(i,u)}$ for the next state $\boldsymbol{\xi}'$, given the current state $\boldsymbol{\xi}$ and the roll setpoint u. Specifically, the first three components of a sample $\tilde{\boldsymbol{\xi}}^{(i,u)}$ are the implicit solution to

$$\frac{d}{d\tau}\begin{pmatrix} x \\ y \\ \psi \end{pmatrix} = \begin{pmatrix} s_i \cos\psi \\ s_i \sin\psi \\ -\dfrac{\alpha_g}{s_i} \tan\left(\phi_i(\tau, r, u)\right) \end{pmatrix}$$

at the end of the T_s-second ZOH period with s_i drawn from the symmetric triangle distribution with mean μ_s and standard deviation σ_s and $\phi_i(\tau, r, u)$ randomly selected from the set $\Phi(r, u)$, where each element occurs with equal probability. The fourth component of $\tilde{\boldsymbol{\xi}}^{(i,u)}$ is deterministic and is simply $\tilde{r}^{(i,u)} = u$. The samples of the UAV's position and heading thus have two sources of randomness: stochasticity in the roll-angle dynamics captured by the collection of roll-angle trajectories $\Phi(r, u)$ and stochasticity in the airspeed. Figure 3 illustrates the stochastic UAV model.

2.3 Target Dynamics

As with the UAV state, the target state $\boldsymbol{\eta}$ is assumed to evolve stochastically according to a Markov Decision Process, where the state transition probability function $p_g(\boldsymbol{\eta}' \,|\, \boldsymbol{\eta})$ is implicitly defined by the following construction for the target motion.

The target is assumed to be a nonholonomic vehicle that travels in the ground plane and has the ability to turn and accelerate. Its state comprises its planar position $(x_g, y_g) \in \mathbb{R}^2$, heading $\psi_g \in \mathbb{S}^1$, and speed $v \in \mathbb{R}_{\geqslant 0}$ and is hence defined as $\boldsymbol{\eta} := (x_g, y_g, \psi_g, v)$. The target's dynamics are those of a planar kinematic unicycle, i.e.,

$$\dot{\boldsymbol{\eta}} = \frac{d}{dt}\begin{pmatrix} x_g \\ y_g \\ \psi_g \\ v \end{pmatrix} = \begin{pmatrix} v\cos\psi_g \\ v\sin\psi_g \\ \omega \\ a \end{pmatrix}, \tag{2}$$

where ω and a are the turn-rate and acceleration control inputs, respectively.

To model the behavior of an operator driving the ground vehicle safely and casually, yet unpredictably, the target's control inputs ω and a are drawn from continuous probability density functions. These inputs are assumed to be held constant over a T_s-second ZOH period synchronized with that of the UAVs and independent across different sampling intervals.

The target's acceleration a is drawn from a symmetric triangle distribution with support over the interval $[\max\{(\underline{v}-v)/T_s, -\alpha\}, \min\{(\bar{v}-v)/T_s, \alpha\}]$, for given positive scalars $\alpha, \bar{v}$, and $\underline{v}$. The support for the distribution guarantees that the absolute value of the acceleration does not exceed α and that the velocity $v' = v + aT_s$ at the end of the sampling period remains in the interval $[\underline{v}, \bar{v}]$.

The distribution for the target's turn rate ω is symmetric triangular with support in the interval $[-\bar{\omega}, \bar{\omega}]$, where $\bar{\omega} > 0$ is given by $\bar{\omega} := \min\{\bar{\omega}_\varrho, \bar{\omega}_a\}$. The support for the distribution guarantees that the target respects both the upper turn rate limit $\bar{\omega}_\varrho = \min\{v/\varrho, (v+aT_s)/\varrho\}$ set by the target vehicle's minimum turning radius $\varrho > 0$ as well as a given maximum allowable turn rate $\bar{\omega}_a > 0$. The quantity $\bar{\omega}_a$ is typically less than $\bar{\omega}_\varrho$ at moderate to high speeds and is used to further govern the target's turning behavior beyond the inherent minimum turning-radius limitation.

The discrete-time stochastic kinematic model is the solution of (2) at the end of the T_s-second ZOH period with the acceleration and turn rate having been drawn from their respective triangle distributions at the start of the ZOH period. This kinematic model is illustrated in Figure 4 with the parameters in Table 1.

Table 1 Stochastic target motion parameters

Parameter:	α	$\underline{v}$	$\bar{v}$	ϱ	$\bar{\omega}_a$	T_s
Value:	0.5	4.5	12.5	7	0.2	2
Units:	$\mathrm{m/s^2}$	m/s	m/s	m	rad./s	s

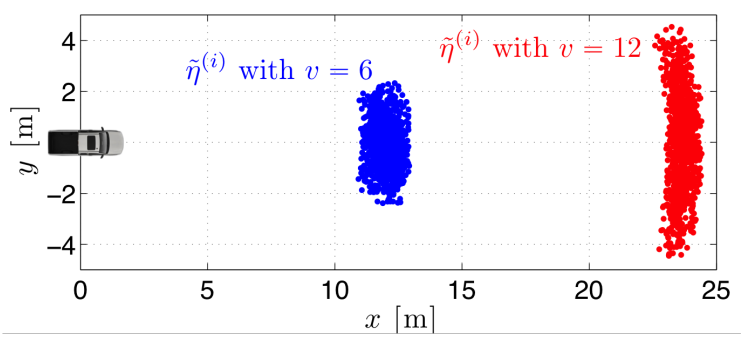


Fig. 4 Sample positions generated from the stochastic target motion model. The two initial target states depicted with different colors correspond to identical initial positions at the origin, but two distinct initial speeds of 6 and 12 [m/s]. For each initial condition, 1,000 samples are generated.

2.4 Target-Centric State Space

We consider a target-centric state space $\mathcal{Z}$ that has dimension $n = 9$. For $j \in \{1, 2\}$, we denote by $\mathfrak{r}_j$ the relative position of UAV j, which is given by

$$\mathfrak{r}_j \coloneqq \begin{bmatrix} \cos\psi_g & \sin\psi_g \\ -\sin\psi_g & \cos\psi_g \end{bmatrix} \begin{bmatrix} x_j - x_g \\ y_j - y_g \end{bmatrix}. \tag{3}$$

Also, we define the UAV j's pose (position and heading) relative to the target as $\mathfrak{p}_j \coloneqq (\mathfrak{r}_j, \psi_{r,j}) \in \mathbb{R}^2 \times [-\pi, \pi)$, where $\psi_{r,j} = \text{atan2}(\sin(\psi_j - \psi_g), \cos(\psi_j - \psi_g))$ and atan2 is the four-quadrant inverse tangent function. The state vector $\boldsymbol{z} \in \mathcal{Z} \subset \mathbb{R}^9$ is thus given by

$$\boldsymbol{z} \coloneqq (\mathfrak{p}_1, r_1, \mathfrak{p}_2, r_2, v),$$

where r_j and v denote UAV j's discretized roll-angle and the target's speed, respectively. The overall state transition probability $p(\boldsymbol{z}' \mid \boldsymbol{z}, \boldsymbol{u})$, where $\boldsymbol{u} \in U(r_1) \times U(r_2)$, is given implicitly by combining the stochastic kinematic models for the vehicles with the preceding description of the components of the states in $\mathcal{Z}$.

2.5 Geolocation Error Covariance

Each UAV has a video sensor that makes image-plane measurements of the target. The dominant source of geolocation error stems from the error in the sensor attitude matrix that relates the line-of-sight vector in the sensor frame (centered at the UAV position) to that in the topographic coordinate frame. This error is amplified on the ground by UAV j's three dimensional distance d_j to the target. Hence, UAV j's individual error covariance, denoted by $\boldsymbol{P}_j \in \mathbb{R}^{2\times 2}$, is proportional to the product of d_j^2 and the covariance $\boldsymbol{R}_{\tilde{\boldsymbol{\theta}}} \in \mathbb{R}^{3\times 3}$ of the 3-2-1 Euler-angle sequence $\boldsymbol{\theta}_j \in \mathbb{R}^3$ describing UAV j's sensor attitude. For simplicity of notation, we take the covariance $\boldsymbol{R}_{\tilde{\boldsymbol{\theta}}}$ of each UAV's sensor attitude angle to be constant and

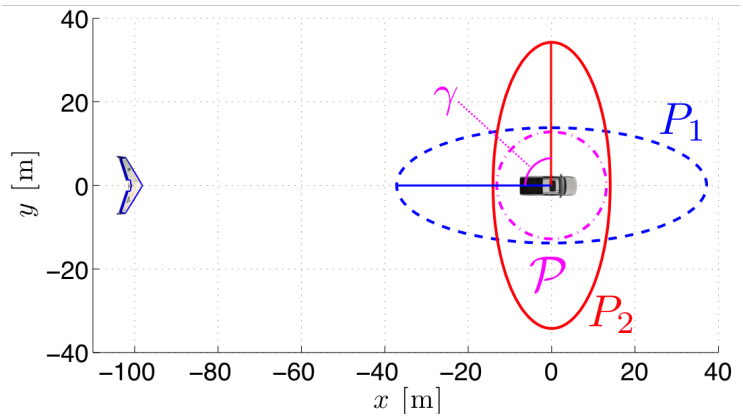


Fig. 5 Individual error ellipses $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ corresponding to the vision measurements from the blue and red UAVs having (x, y, z) coordinates (in meters) of $(-100, 0, 40)$ and $(0, 100, 45)$ respectively, where the latter UAV is not shown. Also depicted by the magenta, dash-dot line is the error ellipse $\mathcal{P}$ corresponding to the combination (fusion) of the measurements obtained from both UAVs, and the separation angle γ.

equal for all UAVs, which would be the case if the UAVs had similar sensors. The exact analytic expression for $\boldsymbol{P}_j$ is derived in [18].

With the UAVs collecting independent measurements of the target, the fused geolocation error covariance (GEC) $\mathcal{P}$ can be computed according to the following relationship

$$\mathcal{P}^{-1} = \sum_j \boldsymbol{P}_j^{-1}. \tag{4}$$

The nature of the error covariances, both individual and fused, is illustrated in Figure 5. Note that the fused covariance is determined by three degrees of freedom, namely the planar distances from the target and the UAVs' separation angle γ, which is given implicitly as

$$\mathfrak{r}_1^\top \mathfrak{r}_2 = \|\mathfrak{r}_1\|_2 \|\mathfrak{r}_2\|_2 \cos\gamma,$$

where the relative planar positions $\mathfrak{r}_j \in \mathbb{R}^2$ are given by (3).

To minimize the estimation errors associated with the fused GEC $\mathcal{P}$, we take the objective function of the stochastic optimal control problem to be

$$g(\boldsymbol{z}) \coloneqq \operatorname{trace}(\mathcal{P}), \tag{5}$$

which has units of meters squared and essentially minimizes the sum of the variances corresponding to the major and minor axes of the fused error ellipse.

The nature of this cost function is illustrated in Figure 6 for two UAVs. Note that if the second UAV's position is on the x-axis, then the UAVs are collinear, which entails that the major axes of their error ellipses are perfectly aligned. If however, its position is on the y-axis, then the UAVs have orthogonal viewing angles. Thus, one can see that the UAVs certainly benefit from having orthogonal viewing angles. However, being close to the target is even more beneficial. For example, if the second UAV's (x, y)-position is $(0, 100)$, such that the UAVs have orthogonal viewing angles, then $g(\boldsymbol{z}) \approx 56$ [m^2]; but if the second UAV is on

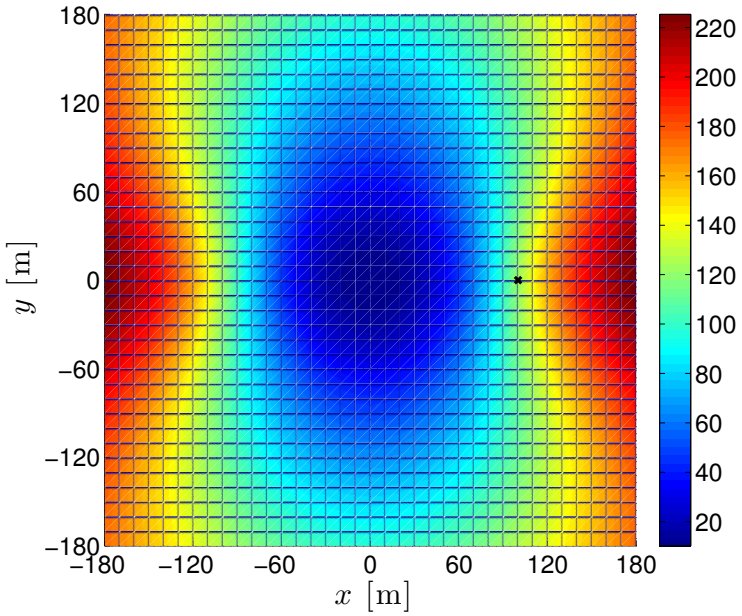


Fig. 6 Cost function $g(\boldsymbol{z}) = \text{trace}(\mathcal{P})$ with the target located at the origin and the first UAV located at three dimensional position $(x, y, z) = (100, 0, 40)$, where this UAV's (x, y) position is indicated by a black "×." The second UAV has an altitude of 45 [m], and the (x, y) coordinates in the plot represent its planar position.

top of the target, then $g(\boldsymbol{z}) \approx 10$ [m^2]. Thus, an effective coordination strategy would be to have at least one UAV close to the target (if possible), as a UAV's individual GEC is smallest in this setting and will dominate the cost through (4). The solution to the stochastic optimal control problem will determine if such a strategy is indeed possible and, in fact, optimal.

2.6 Stochastic Optimal Control Objective

The stochastic optimal control problem is to determine the optimal feedback control policy $\boldsymbol{\mu}_k^* : \mathcal{Z} \to C^2$, $k \in \{0, 1, \ldots, K-1\}$, that minimizes

$$J(\boldsymbol{z}) = \text{E}\left[\sum_{k=0}^{K} g(\boldsymbol{z}_k) \middle| \boldsymbol{z}_0 = \boldsymbol{z}\right], \ \forall \boldsymbol{z} \in \mathcal{Z}, \tag{6}$$

where $\boldsymbol{z}_k = \boldsymbol{z}(kT_s)$, $K \in \mathbb{N}$, $\text{E}[\cdot]$ denotes expectation, $g(\cdot)$ is given by (5), and $\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_K$ is a Markov Decision Process that evolves according to the transition probability $p(\boldsymbol{z}' \,|\, \boldsymbol{z}, \boldsymbol{u})$ determined by the models in Sections 2.2 and 2.3 and the state definitions in Section 2.4, under the feedback law $\boldsymbol{u}_k = \boldsymbol{\mu}_k^*(\boldsymbol{z}_k)$. Note that the state transition probability $p(\boldsymbol{z}' \,|\, \boldsymbol{z}, \boldsymbol{u})$ can also be written as $p(\boldsymbol{z}_{k+1} \,|\, \boldsymbol{z}_k, \boldsymbol{u}_k)$. To solve this problem, we present a novel optimal policy generation algorithm based upon the policy iteration technique known as regression Monte Carlo. To describe the method, we first introduce a few basic definitions and principles underlying *dynamic programming*.

3 Dynamic Programming

Dynamic programming exploits the Markovian nature of the dynamics and hinges on the notion of the *value function*, or *cost-to-go* from state $\boldsymbol{z} \in \mathcal{Z}$ at time $k \in \{0, 1, \ldots, K-1\}$, which is defined as

$$V_k(\boldsymbol{z}) := g(\boldsymbol{z}) + \min_{\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \ldots, \boldsymbol{u}_{K-1}} \mathrm{E}\left[\sum_{\ell=k+1}^{K} g(\boldsymbol{z}_\ell) \middle| \boldsymbol{z}_k = \boldsymbol{z} \right],$$

where $\boldsymbol{u}_k \in U(\boldsymbol{z}_k)$ and $U(\boldsymbol{z})$ denotes the state-dependent action space, which we assume is finite for all $\boldsymbol{z} \in \mathcal{Z}$ throughout this work. For $k = K$, one has that $V_K(\boldsymbol{z}) = g(\boldsymbol{z})$, and for $k \in \{0, 1, \ldots, K-1\}$, the cost-to-go is computed (offline) in reverse chronological order according to the following recursion

$$\begin{aligned} V_k(\boldsymbol{z}) &= g(\boldsymbol{z}) + \min_{\boldsymbol{u} \in U(\boldsymbol{z})} \mathrm{E}\left[V_{k+1}(\boldsymbol{z}') \middle| \boldsymbol{z}, \boldsymbol{u} \right] \\ &= g(\boldsymbol{z}) + \min_{\boldsymbol{u} \in U(\boldsymbol{z})} \int V_{k+1}(\boldsymbol{z}') p(\boldsymbol{z}' \,|\, \boldsymbol{z}, \boldsymbol{u}) d\boldsymbol{z}', \end{aligned} \tag{7}$$

which holds due to Bellman's principle of optimality (see [23], Chapter 6). As the minimization is performed, the optimal control policy can be formed as

$$\boldsymbol{\mu}_k^*(\boldsymbol{z}) = \arg\min_{\boldsymbol{u} \in U(\boldsymbol{z})} \Big(g(\boldsymbol{z}) + \mathrm{E}\left[V_{k+1}(\boldsymbol{z}') \middle| \boldsymbol{z}, \boldsymbol{u} \right] \Big). \tag{8}$$

Performing the sequence of computations in (7) for $k \in \{K-1, K-2, \ldots, 0\}$ ultimately yields $J^*(\boldsymbol{z}) = V_0(\boldsymbol{z}), \forall \boldsymbol{z} \in \mathcal{Z}$, where $J^*(\boldsymbol{z})$ is the minimum value of (6) under the feedback law (8).

3.1 Basic Monte Carlo Solution

A significant hurdle in computing (7) is the expectation, i.e., the integral over the implicitly specified state transition probability $p(\boldsymbol{z}' \,|\, \boldsymbol{z}, \boldsymbol{u})$. In standard Monte Carlo methods, this computation is approximated through empirical averaging. In particular, for a given $\boldsymbol{z} \in \mathcal{Z}$, one can take

$$V_k(\boldsymbol{z}) \approx g(\boldsymbol{z}) + \min_{\boldsymbol{u} \in U(\boldsymbol{z})} \frac{1}{N_s} \sum_{i=1}^{N_s} V_{k+1}\big(\tilde{\boldsymbol{z}}^{(i)}\big),$$

where the $\tilde{\boldsymbol{z}}^{(i)}$ are the N_s Monte Carlo random samples, extracted from the distribution $p(\boldsymbol{z}' \,|\, \boldsymbol{z}, \boldsymbol{u})$. Furthermore, to limit the computation of the value function to a finite number of points, one may restrict the computation of the value function $V_k(\boldsymbol{z})$ to a finite set $Z \subset \mathcal{Z}$ having M distinct elements. This leads to the following approximation of the value function and optimal control policy in (7) and (8), respectively:

$$V_k(\boldsymbol{z}) \approx g(\boldsymbol{z}) + \min_{\boldsymbol{u} \in U(\boldsymbol{z})} \frac{1}{N_s} \sum_{i=1}^{N_s} V_{k+1}\big(\boldsymbol{q}(\tilde{\boldsymbol{z}}^{(i)}, Z)\big) \tag{9}$$

$$\boldsymbol{\mu}_k^*(\boldsymbol{z}) = \arg\min_{\boldsymbol{u} \in U(\boldsymbol{z})} \Big[g(\boldsymbol{z}) + \frac{1}{N_s} \sum_{i=1}^{N_s} V_{k+1}\big(\boldsymbol{q}(\tilde{\boldsymbol{z}}^{(i)}, Z)\big) \Big],$$

where the computation is carried out only for $\boldsymbol{z} \in Z$ and $\boldsymbol{q}$ denotes the quantization function given by

$$\boldsymbol{q}(\boldsymbol{s}, X) \coloneqq \arg\min_{\boldsymbol{x} \in X} \|\boldsymbol{s} - \boldsymbol{x}\|_1$$

for $\boldsymbol{s}$ in $\mathbb{R}^n$ and a finite set $X \subset \mathbb{R}^n$. To lookup the optimal command $\boldsymbol{u}_k$ for an arbitrary state $\boldsymbol{z} \in \mathcal{Z} \backslash Z$, one takes $\boldsymbol{u}_k = \boldsymbol{\mu}_k^*(\boldsymbol{q}(\boldsymbol{z}, Z))$.

This method is suitable for smaller stochastic optimal control problems, such as the single-UAV target tracking scenario for which the state dimension n is 5, as demonstrated in [13]. However, such a state space quantization method is simply infeasible for larger state spaces, such as that corresponding to the two-UAV scenario wherein $n = 9$. The multi-agent stochastic optimal control literature has addressed even larger problems using tools such as factored MDPs [24] and path integral control [25], which would allow one to address problems involving larger UAV teams and even multiple targets. However, such approaches typically have restrictive requirements that involve such needs as an explicit state transition probability function, the solution of complex integrals, continuous state spaces with small hypervolume, and additive noise dynamics. To avoid such limitations that hinder a realistic problem formulation, we employ the sophisticated Regression Monte Carlo technique to determine an approximately optimal policy in the present setting of two agents tracking a single target. This will provide insight into the nature of the optimal solution required for larger problems as well as a policy suitable for real-world implementation. The topic of more UAVs and multiple targets is discussed in the concluding remarks of Section 6.

3.2 Regression Monte Carlo

Regression Monte Carlo (RMC) is a simulation-based policy iteration algorithm that was introduced to stochastic control in the context of optimal stopping by Longstaff and Schwartz in [26] and further formalized by Egloff in [27] with additional convergence analysis. It is suitable for moderate dimensional stochastic optimal control problems, e.g., those having state dimension in the $1 - 10$ range, wherein one may not have an analytic expression for the state transition probability but can easily generate samples. The power and versatility of RMC is underscored by its use in determining optimal policies for managing influenza outbreaks in [28], as well as optimal policies for autonomous vehicle coordination in the current setting. Here we present the method in the general setting following the description of [28]; however, we provide a novel perspective of the algorithm. In particular, we present RMC as a policy *generation* technique rather than as a policy iteration technique and discuss its relationship to the state space quantization method of Section 3.1.

3.2.1 Policy Generation

This work utilizes the Q-value (referred to as the *continuation cost* in [28], or perhaps more commonly as the Q-factor [29]), which is defined as

$$Q_k(\boldsymbol{z}, \boldsymbol{u}) \coloneqq \min_{\boldsymbol{u}_{k+1:K-1}} \mathrm{E}\left[\sum_{\ell=k}^{K} g(\boldsymbol{z}_\ell) \middle| \boldsymbol{z}_k = \boldsymbol{z}, \boldsymbol{u}_k = \boldsymbol{u} \right],$$

where $\boldsymbol{u}_{k+1:K-1}$ is shorthand notation for the sequence $\boldsymbol{u}_{k+1}, \boldsymbol{u}_{k+2}, \ldots, \boldsymbol{u}_{K-1}$. The Q-value is the expected cumulative (or pathwise) cost of being at a state $\boldsymbol{z}$ at time k, applying control action $\boldsymbol{u} \in U(\boldsymbol{z})$ at that time, and then applying an optimal policy from time $k+1$ onward. Since, for $t \in \{k+1, k+2, \ldots, K-1\}$, $\boldsymbol{u}_t$ is a feedback policy, i.e., $\boldsymbol{u}_t = \boldsymbol{\mu}_t(\boldsymbol{z}_t)$, the optimization is not over a fixed sequence but over the sequence of mappings $\{\boldsymbol{\mu}_t(\boldsymbol{z}_t)\}_{t=k+1}^{K-1}$.

The Q-value and the value function are related as follows:

$$\begin{aligned} Q_k(\boldsymbol{z}, \boldsymbol{u}) &= g(\boldsymbol{z}) + \mathrm{E}\left[V_{k+1}(\boldsymbol{z}') \middle| \boldsymbol{z}, \boldsymbol{u} \right] \\ &= g(\boldsymbol{z}) + \int V_{k+1}(\boldsymbol{z}') p(\boldsymbol{z}' \mid \boldsymbol{z}, \boldsymbol{u}) d\boldsymbol{z}', \end{aligned} \tag{10}$$

and

$$V_k(\boldsymbol{z}) = \min_{\boldsymbol{u} \in U(\boldsymbol{z})} Q_k(\boldsymbol{z}, \boldsymbol{u}).$$

Thus, the optimal control policy is also formed as

$$\boldsymbol{\mu}_k^*(\boldsymbol{z}) = \arg\min_{\boldsymbol{u} \in U(\boldsymbol{z})} Q_k(\boldsymbol{z}, \boldsymbol{u}). \tag{11}$$

The main idea of RMC methods is to determine $\boldsymbol{\mu}_k^*(z)$ from (11) for $k \in \{K-1, K-2, \ldots, 0\}$, by approximating $Q_k(\boldsymbol{z}, \boldsymbol{u})$ for each $\boldsymbol{u} \in U(z)$ and for all $\boldsymbol{z} \in \mathcal{Z}$ through Monte Carlo simulations of the right-hand-side of (10). To simplify the introductory discussion of RMC, we assume for now that the control action space $U(\boldsymbol{z})$ is the same for all $\boldsymbol{z} \in \mathcal{Z}$ and hence refer to it simply as U. In RMC, the continuation costs are estimated in reverse chronological order by regressing sample continuation costs onto statistics derived from the starting points in a stochastic mesh $Z \subset \mathcal{Z}$ that is generated at the start of the algorithm and is fixed over time. Moreover, in RMC, one generates a *single* realization of the continuation cost for each of the M points in the stochastic mesh $Z = \{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots, \boldsymbol{z}^{(M)}\}$ and for each control action in U and then carries out cross-sectional regression to fit the *entire* map $(\boldsymbol{z}, \boldsymbol{u}) \mapsto Q_{K-1}(\boldsymbol{z}, \boldsymbol{u})$. For now we take Z as a given quantity and discuss the selection of this quantity in the section that follows.

The use of a single Monte Carlo simulation for each point in Z differs from the state space quantization method of Section 3.1, where one estimates the expectation in (10) by generating N_s scenarios for each state $\boldsymbol{z} \in Z$ and each control action $\boldsymbol{u} \in U$ at time k and then taking the empirical average. While this is reasonable for a single point, it is impractical to do so for each control action in U and point $\boldsymbol{z} \in Z$, as this would require $N_u M N_s$ Monte Carlo simulations, where $N_u = |U|$ and M is typically large for sizable state spaces, e.g., those having dimension $n \geqslant 5$. Thus, whereas the state space quantization method yields a pointwise estimate for the value function using multiple Monte Carlo simulations at each point in Z, RMC utilizes regression to produce a parametric form of the Q-value based on a *single* Monte Carlo simulation for each point in Z.

In RMC, the algorithm begins at time $k = K-1$ by generating a "noisy" instance of the continuation cost $Q_{K-1}(\boldsymbol{z}, \boldsymbol{u})$ for each of the points in Z and for a particular control action $\boldsymbol{u} \in U$. These samples of the continuation cost are denoted by $\tilde{q}_{K-1}^{(i,\boldsymbol{u})}$ and are given by

$$\tilde{q}_{K-1}^{(i,\boldsymbol{u})} = g\left(\boldsymbol{z}_{K-1}^{(i)}\right) + g\left(\tilde{\boldsymbol{z}}_K^{(i,\boldsymbol{u})}\right),$$

where each $\tilde{\boldsymbol{z}}_K^{(i,\boldsymbol{u})}$ is a Monte Carlo sample of the state at time K starting from each of the points $\boldsymbol{z}^{(i)} \in Z$ at time $K-1$ and applying control action $\boldsymbol{u}$ to each of these points. In the preceding equation we have appended the subscript $K-1$ to each of the points in Z to distinguish them from future states. Furthermore, in this context, "noisy" refers to the stochastic uncertainty in the state transition probability distribution $p(\boldsymbol{z}_{k+1} \,|\, \boldsymbol{z}_k, \boldsymbol{u}_k)$. At this point, one regresses $\{\tilde{q}_{K-1}^{(i,\boldsymbol{u})}\}$ onto statistics derived from $\{\boldsymbol{z}_{K-1}^{(i)}\}$ in order to generate an approximation $\hat{Q}_{K-1}(\boldsymbol{z}, \boldsymbol{u})$ to the corresponding continuation cost $Q_{K-1}(\boldsymbol{z}, \boldsymbol{u})$. As the regression step is crucial to the performance of RMC, this will be addressed in one of the following sections.

Once this is done for each $\boldsymbol{u} \in U$, the approximately optimal policy at time $K-1$ is then given by

$$\hat{\boldsymbol{\mu}}_{K-1}^*(z) = \arg\min_{\boldsymbol{u} \in U} \hat{Q}_{K-1}(\boldsymbol{z}, \boldsymbol{u}),$$

where in general the notation $\hat{Q}_k(\boldsymbol{z}, \boldsymbol{u})$ denotes the estimate of the continuation cost at time k obtained through regression. Similarly, $\hat{\boldsymbol{\mu}}_k^*(\boldsymbol{z})$ refers to the estimate of the optimal policy map at time k. In standard Monte Carlo value iteration one would note that

$$\hat{V}_{K-1}(\boldsymbol{z}) = \min_{\boldsymbol{u} \in U} \hat{Q}_{K-1}(\boldsymbol{z}, \boldsymbol{u})$$

and repeat the same procedure for $k = K-2$ by substituting $\hat{V}_{K-1}(\boldsymbol{z})$ for $V_{K-1}(z)$ in (10). However, this practice generally leads to rapid error accumulation. To minimize this, RMC focuses on approximating the optimal policy map $\boldsymbol{\mu}_k^*(\boldsymbol{z})$ rather than the continuation cost. More specifically, at each time k one simulates a single trajectory for each point $\boldsymbol{z}_k$ in the stochastic mesh Z using control action $\boldsymbol{u}$ at time k and implementing future controls based on the newly constructed policy map $\hat{\boldsymbol{\mu}}_t^*(z)$, where $t \in \{k+1, k+2, \ldots, K-1\}$. One then sums the associated stage costs to generate a "noisy" sample for $Q_k(\boldsymbol{z}, \boldsymbol{u})$, which is denoted by $\tilde{q}_k^{(i,\boldsymbol{u})}$ and is an exact realization of the pathwise cost based on the policy constructed so far. In general, for $k \in \{0, 1, \ldots, K-2\}$, $\tilde{q}_k^{(i,\boldsymbol{u})}$ is given by

$$\tilde{q}_k^{(i,\boldsymbol{u})} = g\left(\boldsymbol{z}_k^{(i)}\right) + g\left(\tilde{\boldsymbol{z}}_{k+1}^{(i,\boldsymbol{u})}\right) + \sum_{t=k+1}^{K-1} g\left(\tilde{\boldsymbol{z}}_{t+1}^{(i,\boldsymbol{u}_t)}\right),$$

where $\boldsymbol{u}_t = \hat{\boldsymbol{\mu}}_t^*(z)$. As in the case of $k = K-1$, one then regresses the values of these sample continuation costs onto statistics derived from the corresponding points in the stochastic mesh Z to generate an approximator $\hat{Q}_k(\boldsymbol{z}, \boldsymbol{u})$ for the continuation cost at the current time k. Once this is done for each $\boldsymbol{u} \in U$, the optimal policy is formed in the same manner as when $k = K-1$. The algorithm then marches backward in time, repeating the same procedure of Monte Carlo simulations and regression until reaching time $k = 0$.

The overall produce described above is given by Algorithms 1 and 2 and has an overall computational complexity of $\mathcal{O}(K^2 M N_u)$. In a typical implementation, the inner **for** loop of Algorithm 1 is computed in parallel while the outermost **for** loop of Algorithm 2 is eliminated through the use of vectorized operations, i.e., the procedures described within the loop are performed on all elements of the stochastic mesh at (practically) the same time. Two key components of the algorithm must be selected to obtain acceptable performance, namely the stochastic mesh Z and the regression type used.

Algorithm 1 Regression Monte Carlo

Require: Set Z containing M states in $\mathcal{Z}$

$N_u \leftarrow |U|$
for $k = K-1, K-2, \ldots, 0$ **do**
  for $\ell = 1, 2, \ldots, N_u$ **do**
    Using Algorithm 2, generate cumulative cost realization vector $\boldsymbol{q} \in \mathbb{R}^M$ corresponding to control action $\boldsymbol{u}^{(\ell)} \in U$
    Regress q_i's against statistics derived from each $\boldsymbol{z}^{(i)} \in Z$ to determine $\hat{Q}_k(\boldsymbol{z}, \boldsymbol{u}^{(\ell)})$
  end for
end for
return Q-value approximators $\hat{Q}_k(\boldsymbol{z}, \boldsymbol{u})$, where $k \in \{0, 1, \ldots, K-1\}$

Algorithm 2 Generate a sample of the pathwise cost for each point in Z

Require: Set of states $Z \subset \mathcal{Z}$; control action $\boldsymbol{u}^{(\ell)} \in U$; time index k; $\hat{Q}_{k+1}(\boldsymbol{z}, \boldsymbol{u}), \hat{Q}_{k+2}(\boldsymbol{z}, \boldsymbol{u}), \ldots, \hat{Q}_{K-1}(\boldsymbol{z}, \boldsymbol{u})$ if $k \leqslant K-2$.

$M \leftarrow |Z|$
for $i = 1, 2, \ldots, M$ **do**
  Sample $\tilde{\boldsymbol{z}}^{(i)} \sim p(\boldsymbol{z}' \,|\, \boldsymbol{z}^{(i)}, \boldsymbol{u}^{(\ell)})$, where $\boldsymbol{z}^{(i)} \in Z$
  $q_i \leftarrow g\left(\boldsymbol{z}^{(i)}\right) + g\left(\tilde{\boldsymbol{z}}^{(i)}\right)$
  if $k+1 < K$ **then**
    for $t = k+1, k+2, \ldots, K-1$ **do**
      $\boldsymbol{z}^{(i)} \leftarrow \tilde{\boldsymbol{z}}^{(i)}$
      $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u} \in U} \hat{Q}_t(\boldsymbol{z}^{(i)}, \boldsymbol{u})$
      Sample $\tilde{\boldsymbol{z}}^{(i)} \sim p(\boldsymbol{z}' \,|\, \boldsymbol{z}^{(i)}, \boldsymbol{u}^*)$
      $q_i \leftarrow q_i + g\left(\tilde{\boldsymbol{z}}^{(i)}\right)$
    end for
  end if
end for
return $q \in \mathbb{R}^M$

3.2.2 Forming the Stochastic Mesh

In traditional RMC, the stochastic mesh Z corresponds to a collection of simulated paths $\{\boldsymbol{z}_{0:K}^{(i)}\}$, where $i \in \{1, 2, \ldots, M\}$, that are generated with an initial policy $\boldsymbol{\mu}_k^{(0)}(\boldsymbol{z})$ starting from a collection of initial conditions $\{\boldsymbol{z}_0^{(i)}\}$. Here, $\boldsymbol{z}_{0:K}^{(i)}$ denotes the i^{th} realization of the Markov Decision Process $\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_K$ with initial condition $\boldsymbol{z}_0^{(i)}$ and feedback law $\boldsymbol{u}_k = \boldsymbol{\mu}_k^{(0)}(\boldsymbol{z})$. Thus, Z is in reality a time dependent quantity in traditional RMC and is equal to $\{\boldsymbol{z}_k^{(1)}, \boldsymbol{z}_k^{(2)}, \ldots, \boldsymbol{z}_k^{(M)}\}$ at time k. As with any regression, a higher concentration of samples in a given neighborhood improves the prediction accuracy therein. Hence, one major source of

influence on the performance of the resulting policy map $\hat{\boldsymbol{\mu}}_k^*(\boldsymbol{z})$ is the initial policy map $\boldsymbol{\mu}_k^{(0)}(\boldsymbol{z})$, since it steers the stochastic evolution of the states to generate the stochastic mesh $\{\boldsymbol{z}_{0:K}^{(i)}\}$. Therefore, an initial policy map close to the optimal will lead to re-simulation trajectories in Algorithm 2 that lie close to the original trajectory set where the prediction accuracy is highest; otherwise, one is forced to perform *extrapolation* with the Q-value approximators $\hat{Q}_k(\boldsymbol{z}, \boldsymbol{u})$, which may lead to large errors.

To circumvent the need and influence of an initial policy map, we propose choosing a set $Z \subset \mathcal{Z}$ for which the majority of trajectories corresponding to the optimal controller $\boldsymbol{\mu}^*(\boldsymbol{z})$ will always remain close to this set in some sense. This avoids extrapolation in the regression-based prediction of the continuation cost, and hence, in principle, the prediction accuracy should remain sufficient for choosing the correct control action. Moreover, with intuition and insight into the problem, one can construct Z to have a majority of the samples near the steady-state optimal trajectories. Thus, we take the stochastic mesh Z to be randomly generated at the start of RMC according to some distribution over the state space. One may also generate deterministic grids for Z, as is common for the state space quantization method of Section 3.1; however, the dimensionality of the problem may hinder such an approach.

3.2.3 Regression

The regression type used is crucial to the performance of RMC because inaccurate estimates of the Q-value lead to incorrect control decisions. One should note that in Algorithm 2, at time $k = 0$, running the forward simulations to generate samples of the pathwise cost requires K sequential samples of the state transition probability for each point in Z. Moreover, as k decreases in Algorithm 1, the variance of the pathwise costs increases, and accordingly, robust / regularized regression is required to mitigate these effects.

A number of solutions are available to deal with the said challenge, and include such techniques as radial basis functions, smoothing splines, neural networks, multivariate adaptive regression splines (MARS), ℓ_1-regularized regression, random forests, and others. Each approach has its own tradeoffs in regard to tuning, computational requirements, scalability (both in the number of observations and dimensions), and predictive power, and we refer the reader to [30] for a detailed overview of each approach.

We adopt here a particularly effective technique that is inspired by [22] and involves building a k-d tree for the initial condition set Z and applying least-squares regression at each leaf. In particular, one takes L_j partitions of the j^{th} component of the state vector, such that there are a total of $N_p = \prod_j L_j$ partitions of the state space, each having the *same number of samples*. One should also take care to bound the domain of the local functions at each leaf for the purposes of extrapolation, which we discuss later. This entire process is illustrated in Figure 7 with a 2-dimensional example, where the bounds of each domain, denoted by D_ℓ for $\ell \in \{1, 2, \ldots, N_p\}$, are determined by the outermost points along each direction. In the presence of outliers, one may also wish to either remove the outliers before setting the domain limits or limit the domain to a fixed number of standard deviations along each direction, where standard deviation is computed using only the particles

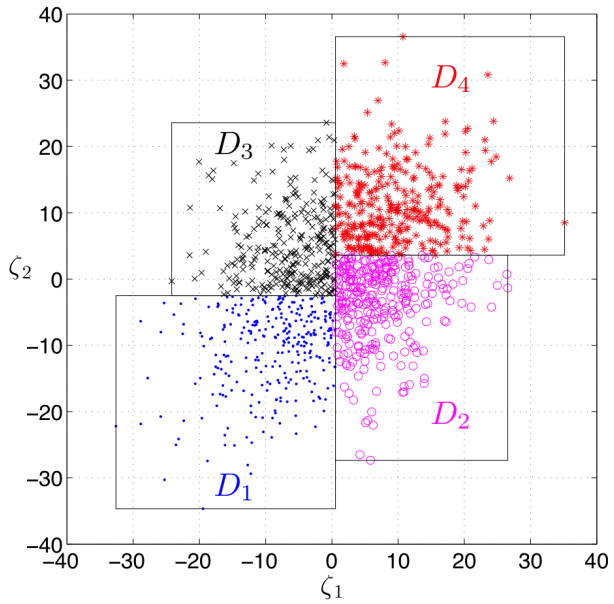


Fig. 7 Partitioning scheme for $\boldsymbol{L} = (2, 2)$ and $\boldsymbol{\zeta}^{(i)} \sim \mathcal{N}(0, C)$ with $C \in \mathbb{R}^{2\times 2}$. Here, $i \in \{1, 2, \ldots, M\}$ with $M = 1,200$, $c_{1,1} = c_{2,2} = 10$, and $c_{2,1} = c_{1,2} = 2$. The black rectangles indicate the individual domains. The split in the $\boldsymbol{\zeta}_1$ coordinate happens roughly at zero to divide the number of points in half, while the splits in the $\boldsymbol{\zeta}_2$ coordinate further subdivide the points such that each bin contains 300 samples.

at the given leaf. We employ the former practice, which is illustrated in Figure 7. The overall domain is $\mathcal{D} := \bigcup_\ell D_\ell$.

The algorithm scales well with the number of sample points M, which we take to be a multiple of N_p. However, it is exponential in the dimension n and is hence suited to moderate dimensional problems. The original algorithm from [22] fits linear models at each leaf using standard least squares; consequently, the fits are not robust to the high variance samples and have a limited ability to capture nonlinearities in the continuation cost. To address these limitations, we propose using an ℓ_1-regularized quadratic fit in each partition.

Let $\mathcal{I}_\ell \subset \{1, 2, \ldots, M\}$ denote the subset of the indices of the particles that belong to partition ℓ, with $|\mathcal{I}_\ell| = m = M/N_p$. Furthermore, with $\mathcal{I}_\ell = \{i_1, i_2, \ldots, i_m\}$, we take $\boldsymbol{y}^{(\ell)} := (q_{i_1}, q_{i_2}, \ldots, q_{i_m}) \in \mathbb{R}^m$. Here, q_i is the pathwise cost sample that is generated from Algorithm 2 and associated with state $\boldsymbol{z}^{(i)} \in Z \subset \mathbb{R}^n$, where $i \in \mathcal{I}_\ell$. Additionally, we denote by $H^{(\ell)} \in \mathbb{R}^{m\times N_b}$ the predictor matrix, where $N_b = n + n(n + 1)$ is the number of basis functions, not including the constant term. Thus, the rows of the predictor matrix take the form $H^{(\ell)}_{i*} = \boldsymbol{h}^\top(\boldsymbol{z}^{(i)})$, where $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^{N_b}$ is used to evaluate the quadratic basis functions for each point $\boldsymbol{z}^{(i)}$ in partition ℓ. We assume that the regression equation is of the following form

$$\boldsymbol{y}^{(\ell)} = H^{(\ell)}\boldsymbol{\beta}^{(\ell)} + \beta_0^{(\ell)}\mathbf{1}_{m\times 1} + \boldsymbol{\epsilon}^{(\ell)}.$$

Here $\boldsymbol{\epsilon}^{(\ell)} \in \mathbb{R}^m$ is the vector of residuals in partition ℓ, $\mathbf{1}_{m\times 1}$ is an m-length vector of all ones, and $\beta_0^{(\ell)} \in \mathbb{R}$ and $\boldsymbol{\beta}^{(\ell)} = (\beta_1^{(\ell)}, \beta_2^{(\ell)}, \ldots, \beta_{N_b}^{(\ell)}) \in \mathbb{R}^{N_b}$ are the coefficients to be determined in the regression. To determine the regression coefficients in a robust fashion, we minimize

$$\|\boldsymbol{y}^{(\ell)} - \beta_0^{(\ell)}\mathbf{1}_{m\times 1} - H^{(\ell)}\boldsymbol{\beta}^{(\ell)}\|_2 + \lambda\|\boldsymbol{\beta}^{(\ell)}\|_1, \tag{12}$$

where $\lambda > 0$ is a tuning parameter. As noted in §3.4.4 of [30], this problem can be solved in the same time complexity as regular least squares, and hence it is suited for repeated use in the partitioned regression scheme, i.e., for each $\ell \in \{1, 2, \ldots, N_p\}$.

Once the regression coefficients have been determined, then the estimator for the Q-value in Algorithm 1, takes the form

$$\hat{Q}_k(\boldsymbol{z}, \boldsymbol{u}) = \langle \boldsymbol{\beta}_k^{(\ell,\boldsymbol{u})}, \boldsymbol{h}(\boldsymbol{z}_k)\rangle + \beta_{0,k}^{(\ell,\boldsymbol{u})}, \quad \boldsymbol{z}_k \in \mathcal{D}_\ell$$

where $\langle\ ,\ \rangle$ denotes inner product, $\boldsymbol{h}(\boldsymbol{z}_k) \in \mathbb{R}^{N_b}$ is the aforementioned mapping that forms the rows of the predictor matrix $H^{(\ell)}$, and ℓ is the index of the partition $\mathcal{D}_\ell$ to which $\boldsymbol{z}_k$ belongs. Also, we have indicated the dependency of the regression coefficients on both the time k and the control action u, which had been temporarily omitted for simplicity.

Ideally, the controller keeps the steady state trajectories in the domain $\mathcal{D}$; however, in the case that extrapolation must be performed, one should be wary of the behavior of the quadratic fit outside $\mathcal{D}$. Thus, if $\boldsymbol{z} \notin \mathcal{D}$, we evaluate $Q_k(\boldsymbol{z}, \boldsymbol{u})$ at the point in $\mathcal{D}$ closest to $\boldsymbol{z}$ using the 1-norm, as we have found this practice to produce satisfactory performance. Moreover, we expect the partitioned quadratic fits to interpolate well in the domain but avoid their use for extrapolation.

3.2.4 Convergence and Sample Size

Regarding the algorithm's convergence to the optimal solution, we note that this subject has been formally studied in [31], and the reader is referred to the said work for formal proofs of convergence. Nonetheless, we note that convergence to the true Q-values (and hence the optimal policy and true value function) are obtained in the joint limit as the total number of Monte Carlo samples M and the total number of partitions N_p (and in particular each L_j) tends to infinity. In [31], the obtained result is rather general, principally requiring the Q-values to be bounded. A limitation is that M must be exponential in N_p, i.e., the number of samples grows very quickly as the number of degrees of freedom for the regression is increased. We also note that [31] assumes unpenalized regression, i.e., $\lambda = 0$ in (12), though this is unlikely to affect the results.

Since computational tractability necessitates a finite sample size M, one particularly noteworthy result from [31] is that the approximation error is of order $\mathcal{O}((\log^c M)/\sqrt{M})$ for a certain $c > 0$, which is close to the classical Monte Carlo error of $\mathcal{O}(1/\sqrt{M})$. The precise estimates are rather complex and depend on a number of factors including functional smoothness of the Q-values and smoothness of the underlying stochastic dynamics, to name a few. More precise and detailed results about convergence rates have been given for the case of optimal stopping in [27].

To balance computational considerations with the accuracy of the solution using the partitioned robust regression scheme of the previous section, we recommend using a large number of samples per partition m that is linear in the number of basis functions and is on the order of a few hundred or more samples per basis function. As an example, for the problem setup that we consider in Section 5, we have found that just over 275 samples per degree of freedom in each of the partitioned quadratic fits produces satisfactory performance. Without symmetry arguments that eliminate the need for certain state space partitions to be considered, $M = mN_p = m\prod_j L_j$, and hence the total number of simulations M is exponential in the number of partitions. Thus, for moderate dimensional problems, the number of partitions L_j for state j is usually small (less than 3) unless the state enters into the dynamics in a more nonlinear fashion, thus requiring a greater number of partitions.

4 Regression Monte Carlo for Target Tracking

We now specialize the algorithms described in Section 3 to the problem of vision-based target tracking. In particular, we first present an adaptation to Algorithm 1 that addresses the fact that some of the components of the state space described in Section 2.4 were discrete. Next, we describe the stochastic mesh Z, and finally, we discuss a modification to the cost function that makes it radially unbounded, thereby ensuring the distances of the UAVs relative to the target remain bounded.

4.1 Modified Algorithm

Since the state space of the stochastic kinematic model of the UAV described in Section 2.2 had a few discrete components, the standard RMC algorithm must be slightly modified for the application of vision-based target tracking with two UAVs. In particular, we need to run Monte Carlos simulations for all roll-angle pairs $\boldsymbol{r}$ belonging to a finite set $\mathcal{C} \subset C^2$ combined with all (finitely-many) allowable roll action pairs $\boldsymbol{u} \in \mathcal{U}(\boldsymbol{r})$, where $\mathcal{C} = \{\boldsymbol{r}^{(1)}, \boldsymbol{r}^{(2)}, \ldots, \boldsymbol{r}^{(N_r)}\}$,

$$\mathcal{U}(\boldsymbol{r}) \coloneqq U(r_1) \times U(r_2),$$

and $N_r = |\mathcal{C}| = 15$ (versus 5×5) due to symmetry arguments discussed in the appendix. To accommodate these modifications, we remove the roll states from $\mathcal{Z}$ and denote the resulting continuous state space by $\mathcal{X} \subset \mathbb{R}^7$, where $\boldsymbol{\chi} \in \mathcal{X}$ is given by

$$\boldsymbol{\chi} \coloneqq (\mathfrak{p}_1, \mathfrak{p}_2, v)$$

and $\mathfrak{p}_j$ is described in Section 2.4. The resulting stochastic mesh described in Section 3.2.2 is denoted by X.

The modified RMC algorithm for vision-based target tracking with two-UAVs is presented in Algorithm 3. The primary differences with respect to Algorithm 1 is the addition of a **for** loop over all discrete-valued states, as well as the formation of the full initial condition set Z from the set X of continuous initial condition states and the given roll-angle pair $\boldsymbol{r} \in \mathcal{C}$. Furthermore, the regression is performed

using only the continuous states in X, and thus the dimensionality of the regression problem is reduced to $n_r = 7$. In practice, the two innermost loops are often combined and run in parallel for increased computational performance. On a final note, when generating the cumulative cost samples for each roll action with Algorithm 2, one should replace U with $\mathcal{U}(\boldsymbol{r})$ in the requirements section and U with $\mathcal{U}(\boldsymbol{r}^{(i)})$ in Line 8.

Algorithm 3 Regression Monte Carlo for Target Tracking

Require: Initial condition set $X \subset \mathbb{R}^7$, where $|X| = M$; set of roll-angle pairs $\mathcal{C} \subseteq C^2$; action space $\mathcal{U}(\boldsymbol{r})$

1: $N_r \leftarrow |\mathcal{C}|$
2: **for** $k = K-1, K-2, \ldots, 0$ **do**
3: **for** $s = 1, 2, \ldots, N_r$ **do**
4: Form initial condition set Z from X, such that for each $\boldsymbol{\chi}^{(i)} = (\mathfrak{p}_1^{(i)}, \mathfrak{p}_2^{(i)}, v^{(i)}) \in X$, the following relationship holds: $\boldsymbol{z}^{(i)} = (\mathfrak{p}_1^{(i)}, r_1^{(s)}, \mathfrak{p}_2^{(i)}, r_2^{(s)}, v^{(i)})$, where $i \in \{1, 2, \ldots, M\}$
5: $N_u \leftarrow |\mathcal{U}(\boldsymbol{r}^{(s)})|$
6: **for** $\ell = 1, 2, \ldots, N_u$ **do**
7: Using Algorithm 2, generate continuation cost realization vector $\boldsymbol{q} \in \mathbb{R}^M$ by applying control action $\boldsymbol{u}^{(\ell)} \in \mathcal{U}(\boldsymbol{r}^{(s)})$ to each point $\boldsymbol{z}^{(i)} \in Z$
8: Regress q_i's against statistics derived from corresponding $\boldsymbol{\chi}^{(i)}$'s to determine $\hat{Q}_k(\boldsymbol{z}, \boldsymbol{u}^{(\ell)})$
9: **end for**
10: **end for**
11: **end for**
12: **return** Q-value approximators $\hat{Q}_k(\boldsymbol{z}, \boldsymbol{u})$, where $k \in \{0, 1, \ldots, K-1\}$

4.2 Stochastic Mesh

While modified RMC approach offers significant computational savings over a basic Monte Carlo method for value iteration, it generally requires adjusting the initial condition set and the regression to obtain satisfactory performance. Thus, we now describe the initial condition set X of Algorithm 3 that comprises the continuous states. For UAV j, if the relative position states of $(\chi_{3j-2}, \chi_{3j-1})$ are represented in polar coordinates as $(\rho_j \cos\vartheta_j, \rho_j \sin\vartheta_j)$, then we take ρ_j to be normally distributed with mean $\mu_\rho > 0$ and variance σ_ρ^2 and ϑ_j to be uniformly distributed on $[\underline{\vartheta}, \overline{\vartheta}]$. Typically, $(\underline{\vartheta}, \overline{\vartheta}) = (-\pi, \pi)$. However, if one exploits symmetry per the discussion of the appendix, this need not be the case. Also, in the process of generating M samples of ρ_j, we only retain those samples that have strictly positive values and

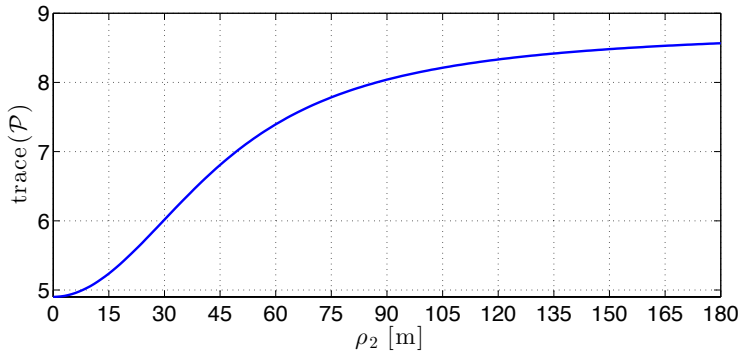


Fig. 8 Cost function $g(\boldsymbol{z}) = \text{trace}(\mathcal{P})$ with the target located at the origin and the first UAV located on top of the target at an altitude of 40 [m]. Note that the separation angle γ is 0 since the first UAV's planar distance is $\rho_1 = 0$; consequently, the fused GEC is completely characterized by the second UAV's planar distance ρ_2 from the target. The second UAV has an altitude of 45 [m].

those that are within 3σ of the mean, as the outer boundaries of the partitioning domain are set in the manner discussed and illustrated in Section 3.2.3. Next, we take χ_{3j}, the relative heading angle of UAV j, to be uniformly distributed on $[-\pi, \pi]$ and the target speed $\chi_7 = v$ to be uniformly distributed on $[\underline{v}, \bar{v}]$, where $\underline{v}$ and $\bar{v}$ are discussed in Section 2.3. Moreover, at the start of Algorithm 3, we generate M samples of the continuous states in the manner just described to form $X = \{\boldsymbol{\chi}^{(1)}, \boldsymbol{\chi}^{(2)}, \ldots, \boldsymbol{\chi}^{(M)}\}$, where the mean μ_ρ and variance σ_ρ^2 of the radial distribution of the relative planar position states are tuning parameters set beforehand.

4.3 Barrier Function

While in the single UAV case, the stage cost given by (5) directly penalizes distance from the target, this is not the case for both agents in the two-UAV scenario. In particular, having UAV 1 directly above the target and UAV 2 far away is only slightly worse than having both UAVs directly above the target, since the smallest planar UAV distance from the target is the dominant factor in the fused GEC. This is illustrated in Figure 8, where the range of trace values is drastically smaller than that of Figure 6. This disparity in the range of trace values arises from UAV 1 having zero planar distance from the target in Figure 8 and a planar distance of 100 [m] from the target in Figure 6. Moreover, UAV 2's position has an almost negligible effect on the fused GEC in the former scenario while in the latter scenario its position has a considerable impact on the fused GEC. Overall, this suggests that the cost function is not radially unbounded with respect to the second UAV's planar distance from the target.

To avoid using the Q-value approximators far from the stochastic mesh, we added a barrier-type function to the stage cost that is non-negative, radially unbounded, and only nonzero for large distances. Hence, we present the following augmented cost function to be used in the dynamical optimization of Algorithm 3:

$$g_b(\boldsymbol{z}) := \text{trace}(\mathcal{P}) + b(\boldsymbol{z}), \tag{13}$$

where

$$b(\boldsymbol{z}) = \sum_{j=1}^{2} \max\{0, \rho_j - (\mu_\rho + 2\sigma_\rho)\},$$

ρ_j denotes UAV j's planar distance from the target, and μ_ρ and σ_ρ are the mean and standard deviation of the normally distributed planar distances from the target that form the initial condition set described in Section 4.2. While the barrier function $b(\boldsymbol{z})$ penalizes trajectories where the UAVs wander very far from the target, it has a negligible effect along optimal trajectories, which should remain close to the target.

5 Results

We now study the nature of the optimal coordination strategy and the effectiveness of the modified RMC approach in the optimal coordination of two UAVs to perform vision-based target tracking in a stochastic environment. To establish the benefit of the proposed control approach, we compare the performance of our (approximately) optimal controller against an effective baseline strategy, as well as the previously proposed approach of coordinated standoff tracking. Additionally, we seek to gain insight regarding the optimal control policy to understand the predominant behavior of the two fourth-order UAVs, as they cooperatively track the stochastic ground target.

5.1 Problem Setup and Solution Parameters

Throughout this section we extensively analyze the results of a fairly realistic tracking scenario that is summarized by the parameters pertaining to the target and UAVs provided in Tables 1 and 2, respectively. The scenario is similar to that considered by [2], wherein the authors presented field test results for a single UAV (capable of $15-20$ [m/s] airspeeds) tracking a target that traveled between $5-10$ [m/s].

Table 2 Parameters in Stochastic UAV dynamics

Parameter	Description	Value	Units
μ_s	nominal airspeed	18	m/s
σ_s^2	airspeed variance	16/25	$\mathrm{m}^2/\mathrm{s}^2$
α_g	gravitational accel.	9.81	m/s^2
C	roll command set	$\{0, \pm\Delta, \pm 2\Delta\}$	deg.
Δ	max roll change	15	deg.

The parameters pertaining to the general dynamical optimization of Section 2.6 are presented in Table 3. The planning horizon of $KT_s = 30$ seconds was chosen so that, within this time, the UAV could perform a loop at max bank, which for a maximum turn rate of $\omega_{\max} = \alpha_g \tan(2\Delta)/s$ [rad./s] is approximately 20 seconds. With the 30-second horizon, the optimization takes into account long-term impact

Table 3 General Parameters

Parameter	Description	Value	Units
$\boldsymbol{R}_{\tilde{\theta}}$	sensor attitude covariance	$9I_{3\times 3}$	deg^2
(h_1, h_2)	UAV altitudes	$(40, 45)$	m
T_s	zero-order hold period	2	s
K	planning horizon	15	-

Table 4 RMC Parameters

Parameter	Description	Value
m	samples per partition	$10,000$
$\boldsymbol{L}$	partitioning scheme	$(2, 2, 4, 2, 2, 4, 2)$
λ	regularization parameter	3
μ_ρ	radial distribution mean	70
σ_ρ^2	radial distribution variance	35^2

of committing to a loop, as the control policy $\hat{\boldsymbol{\mu}}_k^*(\boldsymbol{z})$ is applied in a receding horizon fashion, i.e., we always apply $\hat{\boldsymbol{\mu}}_0^*(\boldsymbol{z})$ at every time step.

The parameters pertaining to the RMC solution are presented in Table 4, where the augmented cost from (13) was minimized using Algorithm 3 along with the techniques for computational savings presented in the appendix. However, throughout this section the terms *cost* and *stage cost* refer to the original cost function given by (5), which is simply the trace of the fused GEC. Also, we henceforth refer to the resulting policy as the *optimal* policy with the understanding that this policy is in reality an approximation to the true optimal policy.

Regarding the regression, each partition had 36 degrees of freedom in the quadratic regression, as the dimension of the continuous state space $\mathcal{X}$ is 7. Hence, we chose m (the number of samples per partition) to avoid overfitting, while the regularization parameter λ was chosen to add robustness to process noise, where $\lambda \in [3, 10]$ constitutes a considerable degree of regularization and generally works well. Regarding the partitioning scheme, recall that $\boldsymbol{L} \in \mathbb{N}^7$ is the vector denoting the number of partitions for each component of the continuous state space $\mathcal{X}$, and hence the total number of partitions is $N_p = \prod_j L_j = 512$, though one does not need to estimate the Q-value in more than 320 partitions according to the symmetry considerations of the appendix. Thus, the size of the stochastic mesh X is $M = 320 \cdot 10^4$. Note that choosing 2 partitions for the relative x and y coordinates implies that the partitions of the planar positions correspond (approximately) to standard Cartesian quadrants, since the 2-dimensional distributions that generate the individual (planar) position samples of the initial condition set are radially symmetric per the discussion of Section 4.2. The mean and variance of the normally distributed planar distances are also given in this table. Also, we have found that the relative heading coordinates are the most sensitive to the number of regression partitions (due to the nonlinearity) and hence choosing L_3 or L_6 to be less than 3 typically yields poor performance. Through considerable testing, we found that this particular partition configuration is a good compromise between computational feasibility and mitigating the effects of the nonlinearity through additional partitions.

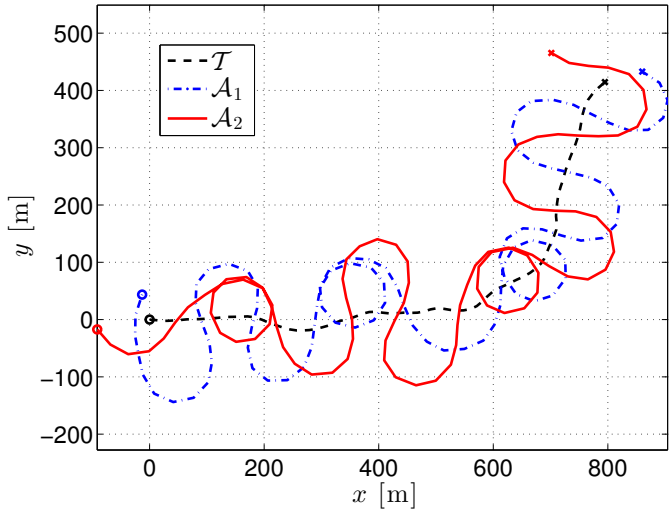


Fig. 9 Optimally coordinated trajectories over a three minute window. The starting positions of all vehicles are marked by an "○" while the ending positions are denoted by an "×". The target (denoted by $\mathcal{T}$) begins at the origin travelling at approximately 5.4 [m/s] and finishes its trajectory travelling at approximately 7.3 [m/s]. Both UAVs (denoted by $\mathcal{A}_1$ and $\mathcal{A}_2$) begin with zero roll.

To highlight key features of the optimal trajectories, we have provided a representative sample trajectory in Figure 9 and the corresponding performance parameters in Figure 10. From Figure 9, one should note how the optimal trajectories comprise both sinusoidal and orbital trajectories, where the latter is not necessarily centered around the target. At the beginning of the simulation, one UAV is performing an "S" turn (sinusoidal pattern) while the other is performing a loop. The UAVs switch roles and perform the same joint maneuver before both UAVs make out-of-phase loops and then out-of-phase "S" turns. From the top plot in Figure 10, distance coordination becomes apparent, as the peaks of the distance curves alternate. The second subplot of this figure indicates that the UAVs do not strive to maintain orthogonal viewing angles, as the curve does not cluster around $\gamma = 90^\circ$. However, the UAVs do benefit from orthogonal viewing angles when they are both moderately far, e.g., $t = 82$ [s], where the cost is kept from spiking by such a configuration. Overall, minimum distance is the dominant factor in the cost function, though viewing angle coordination does benefit the UAVs when they find themselves moderately far from the target.

While this particular instance of a controlled stochastic process does not establish distance coordination as the predominant coordination strategy, it does illustrate typical behaviors encountered with this policy. Namely, the optimal trajectories comprise a rich mixture of sinusoidal and orbital trajectories that occasionally pass over or near the target rather than just a single trajectory type, which is the primary goal in the vast majority of the target tracking literature.

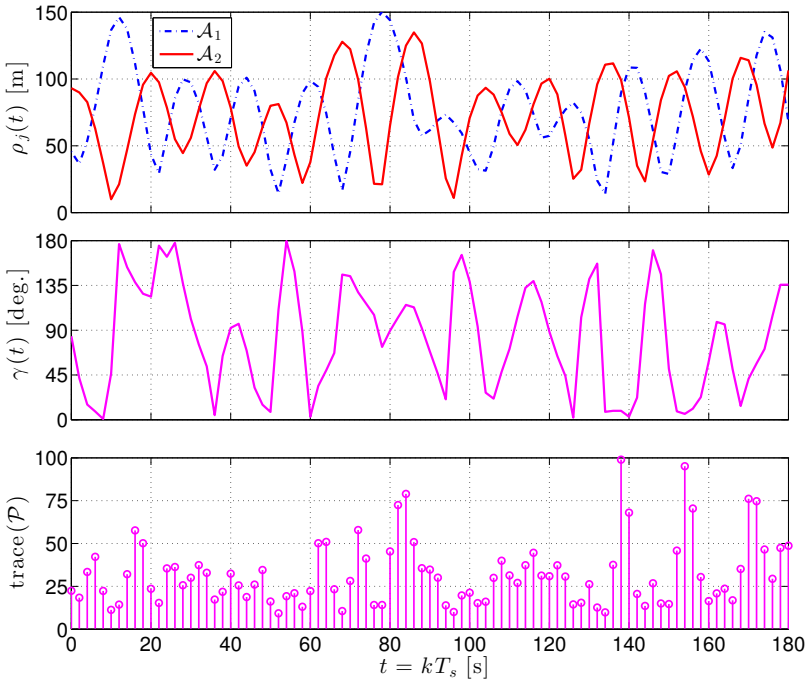


Fig. 10 Performance metrics of optimally coordinated UAVs: planar distances ρ_j, separation angle γ, and trace of the fused GEC $\mathcal{P}$.

5.2 RMC Performance

We now compare our RMC solution against alternative control strategies that roughly consider the same problem formulation. In particular, we first compare the strategy with two UAVs that are running uncoordinated optimal policies, and secondly we compare the strategy with the common approach of coordinated standoff tracking.

5.2.1 Comparison with Uncoordinated Optimal Controllers

To generate an appropriate uncoordinated baseline strategy, we solve the stochastic optimal control problem of Section 2.6 for a single UAV and then apply the same optimal control law for the two UAVs independently. Since the problem for a single UAV has modest dimension, one can solve it using the basic Monte Carlo solution of Section 3.1. As a result, we performed value iteration according to Section 3.1 to generate two individual control policies with the cost function (5) and the parameters of Table 3. We used $M = 1,000$ Monte Carlo samples in (9) with a finite state space Z described by Table 5. We denote the resulting policies as $\pi_k^{(1)}(\boldsymbol{\zeta}_k^{(1)})$ and $\pi_k^{(2)}(\boldsymbol{\zeta}_k^{(2)})$, where $k \in \{0, 1, \ldots, K-1\}$, $\boldsymbol{\zeta}^{(1)} = (\mathfrak{p}_1, r_1, v)$, $\boldsymbol{\zeta}^{(2)} = (\mathfrak{p}_2, r_2, v)$, and $\mathfrak{p}_j$ is defined in Section 2.4. As in the case of coordinated UAVs, we always apply these policies in a receding-horizon fashion, i.e., we always

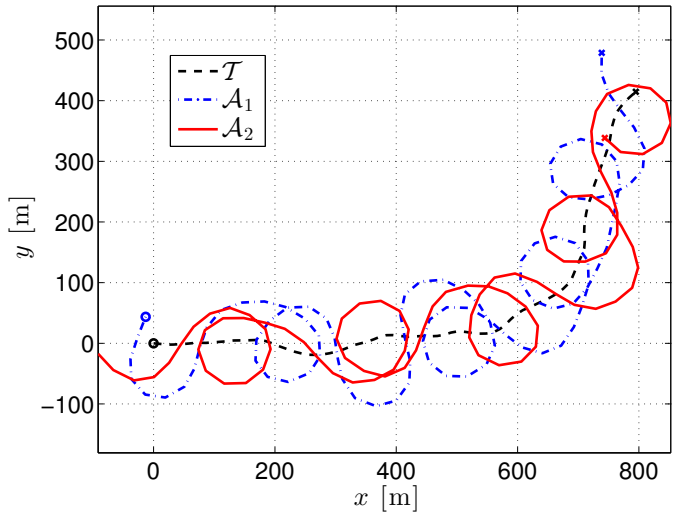


Fig. 11 Uncoordinated trajectories over a three minute window. The starting positions of all vehicles are marked by an "○" while the ending positions are denoted by an "×". The remaining notation, initial conditions, and target trajectory are the same as in Figure 9.

use the time-stationary policies $\pi_0^{(1)}(\zeta_k^{(1)})$ and $\pi_0^{(2)}(\zeta_k^{(2)})$ for the uncoordinated UAVs for all $k \in \mathbb{Z}_{\geqslant 0}$.

Table 5 State Space Discretization in One-UAV Scenario

Set	Description	Value	Units
Y	relative positions	$\{-225, -220, \ldots, 225\}$	m
Ψ	relative headings	$\{0, 15, \ldots, 345\}$	deg.
C	roll commands	$\{0, \pm 15, \pm 30\}$	deg.
W	target speeds	$\{4.5, 5.0, \ldots, 12.5\}$	m/s
Z	discrete state space	$Y^2 \times \Psi \times C \times W$	-

To illustrate the nature of this control strategy, we have provided plots in Figures 11 and Figures 12 illustrating the behavior and performance of uncoordinated controllers for the same initial conditions and target trajectory realization as in Figures 9 and 10. While each UAV minimizes its own individual GEC, we plot the fused covariance in the bottom chart of Figure 12. The most noticeable feature of Figure 11 is the fact that the UAVs primarily make orbital trajectories around the target, which enables them to keep their worst-case distance from the target smaller. This is confirmed by the top chart of Figure 12, where the peak planar distance from the target is approximately 114 [m], whereas that of Figure 10 is approximately 150 [m]. One can also see the lack of coordination for $t \in [126, 132]$, as the cost is above 100 during this time period when both UAVs are moderately far from the target and have viewing angles that are quite far from being orthogonal. On a final note, the time-averaged cost for this run was approximately 39.6 [m^2] while that of the coordinated control policy was 32.4 [m^2]. It is interesting to note that coordination allows the UAVs to deviate further from the target without

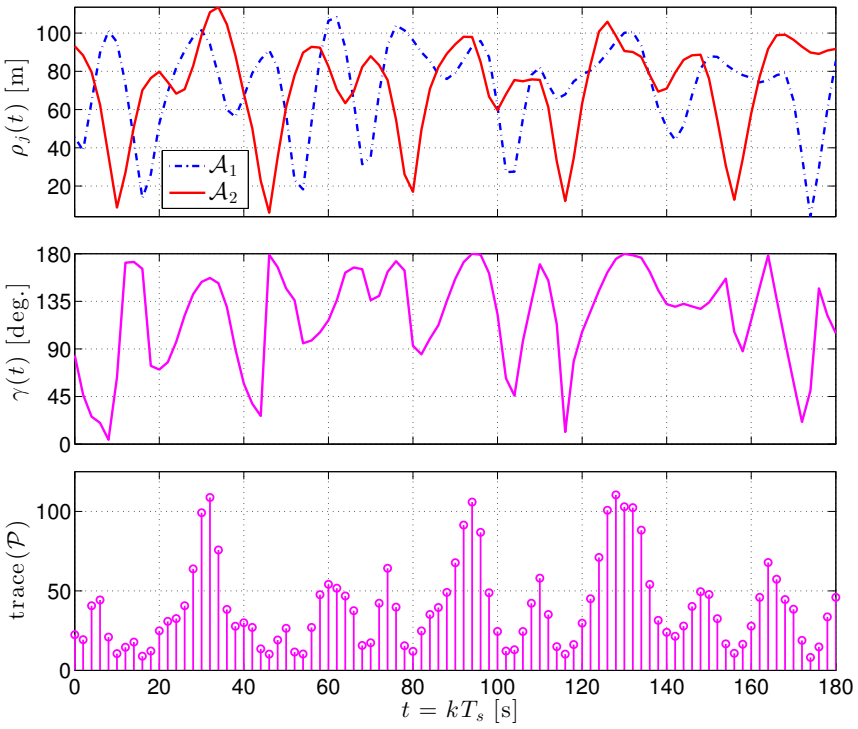
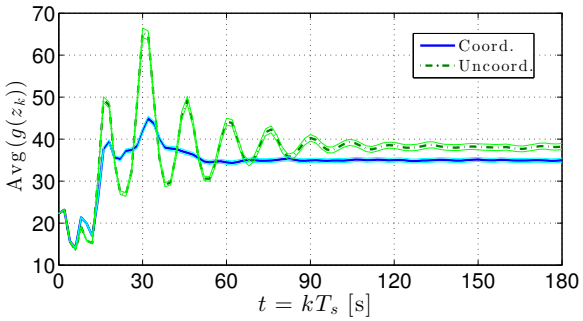


Fig. 12 Performance metrics of uncoordinated UAVs: planar distances ρ_j, separation angle γ, and trace of the fused GEC $\mathcal{P}$.

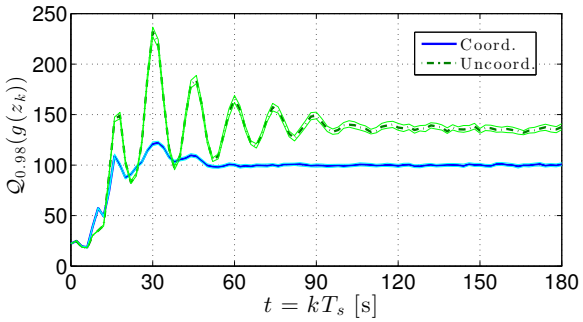
sacrificing performance. Of course, this deviation must be done in an alternating fashion, as illustrated by the distance curves of Figure 9.

To better demonstrate the temporal nature of both control strategies in an expected sense, we have selected an initial condition that is a good starting point for both strategies and run 50,000 Monte Carlo simulations from this initial condition with the same realizations of target trajectories to compute both the mean value and 98^{th}-percentile statistics of the cost, which are provided in Figure 13. By inspecting Figure 13a, one can see that the optimal control policy converges to the mean steady-state cost of (approximately) 35 [m^2] within one minute while the uncoordinated controllers take nearly 2 minutes to converge to the mean steady-state cost of (approximately) 38 [m^2]. In addition, the peak average value is significantly less in the case of the optimal coordinated control policy than in the case of uncoordinated policies. Note that the distribution of steady-state costs is independent of the initial conditions.

Another benefit of the coordinated control policy can be seen in Figure 13b, where the plot indicates that the tail of the steady-state cost distribution is often significantly wider in the case of uncoordinated policies. In fact, the 98^{th}-percentile of the steady-state costs for the uncoordinated policies is about 33% higher than that of the optimal policy. Moreover, although we have illustrated transient response performances for a specific initial condition, the plots in Figures 13a and 13b illustrate typical benefits of the optimal control policy. Namely, with coor-



(a) Mean Value (Ensemble Average)



(b) 98$^{\text{th}}$ Percentile

Fig. 13 Transient response for initial condition $\boldsymbol{z}_0 = (-60, 0, -\pi/2, -30^\circ, 0, -60, 0, -30^\circ, 8.5)$, where $\boldsymbol{z}_k = \boldsymbol{z}(kT_s)$. In this initial condition, the UAVs have orthogonal viewing angles and are banked max left at a distance roughly equal to their minimum turning radius of 57.2 [m]. The lighter, thinner lines indicate the 95% confidence intervals for the given statistics.

dination, the recovery from initial conditions is typically faster (in an expected sense), and the tail of the cost distribution is significantly smaller in steady state, which entails that high cost events are more rare than in the uncoordinated case.

To provide a more objective comparison, we have performed another test over a wide range of initial conditions. More specifically, we generated $M = 50,000$ initial conditions randomly according to Section 4.2 and then ran 12-minute Monte Carlo simulations with each control strategy from these initial conditions using the same realizations of target trajectories for each approach. To reduce the effects of initial conditions, we truncated the first two minutes of each run. Computing the sample mean (over time) of the stage costs associated with each run yields the histogram presented in Figure 14. Hence, whereas the previous test illustrated the first few minutes of a transient response and computed certain statistics across samples, here we are computing the mean *over time* with the first few minutes of each simulation removed. In this plot, the sample mean and sample standard deviation of the time-averaged costs associated with the optimal policy are 34.91 [m^2] and 2.33 [m^4], respectively; those associated with the uncoordinated control policies are 37.92 [m^2] and 4.09 [m^4], respectively. Furthermore, the standard error of the mean is less than 0.02 [m^2] in both cases. One can observe that, while the optimally coordinated control policy reduces the mean of the time-averaged costs

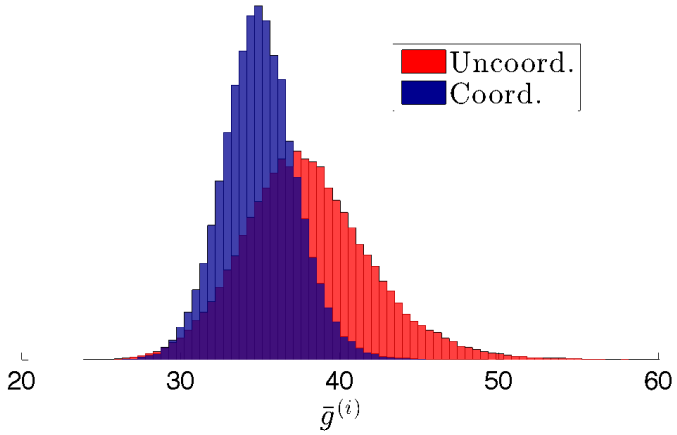


Fig. 14 Histogram of the stage-cost mean $\bar{g}^{(i)} = (1/301)\sum_{k=60}^{360} g(\boldsymbol{z}_k^{(i)})$ for 10-minutes of steady-state behavior with 50,000 Monte Carlo simulations. The outliers for the uncoordinated policy are not all shown, as they extend out to nearly 73 [m^2].

by only about 8%, it reduces their standard deviation by nearly 43%. This reduction in standard deviation is illustrated by the widths of the distributions, which is considerably less in the case of the optimally coordinated strategy. Thus, the optimally coordinated control policy improves the predictability of the tracking performance substantially.

One final comparison is provided by plotting the histogram of the steady-state costs with the effects of time-averaging removed. More specifically, in Figure 15 we provide a histogram of the stage costs given by (5) at each time step for each of the $M = 50,000$ ten-minute Monte Carlo simulations in steady state. The number of counts is presented with a logarithmic scale to focus on the tails of the distributions. One can see that the tail of the cost distribution corresponding to the uncoordinated policies decays slower than that corresponding to the optimally coordinated policy. This plot highlights that rare events are less frequent and less severe with the coordinated control policy than with uncoordinated control policies. In fact, for stage costs exceeding 400 [m^2], the frequency of such costs with the coordinated control policy are an order of magnitude lower than with the uncoordinated policies. Since we expect the controlled processes to be ergodic, these histograms are representative of a single very-long run for each of the cooperative tracking approaches, e.g., a run lasting hundreds of hours.

5.2.2 Comparison with Standoff Tracking

To establish a fair comparison with the standoff tracking approach, we note that the minimum allowable standoff distance, ϱ_s, imposed by the maximum bank angle $\phi_{\max}$, is given by Equation 5.37 in [3] as follows:

$$\varrho_s \geqslant \frac{(v+s)^2}{\alpha_g \tan(\phi_{\max})}, \tag{14}$$

where v, s, and α_g denote target speed, UAV airspeed, and gravitational acceleration, respectively. With the target traveling at the minimum allowable speed of

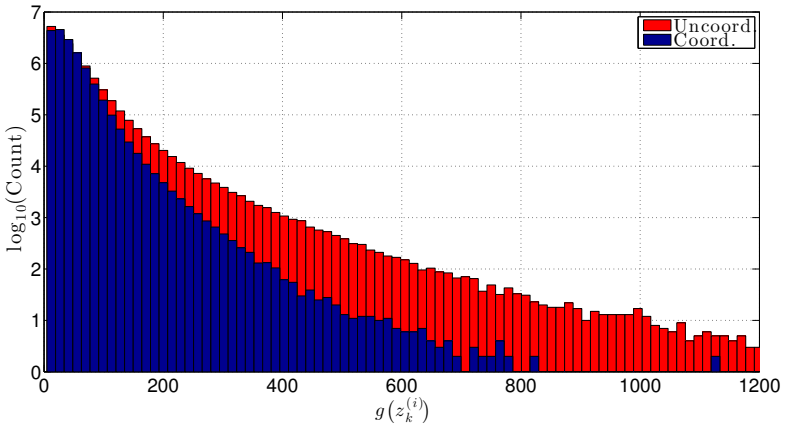


Fig. 15 Histogram of the stage-cost (not averaged over time) for 50,000, 10-minute Monte Carlo simulations. The outlying costs of the uncoordinated policy are not all shown, as they exceed 2,000 [m^2].

4.5 [m/s] and the remaining parameters given in Table 2, we have

$$\varrho_s \geqslant \frac{(4.5+18)^2}{9.81\tan(30\pi/180)} \approx 89.4 \text{ [m]}.$$

In an ideal setting for standoff tracking, the target is traveling at a constant velocity and the UAVs have orthogonal viewing angles at the nominal standoff distance of $\varrho_s = 90$ [m]. Hence, with the altitudes of Table 3, trace$(\mathcal{P}) \approx 46$ [m^2]. Recalling that both the time-averaged cost and ensemble-averaged cost of the optimal policy in steady state (over many target velocity realizations) were both approximately 35 [m^2], one can see that the optimally coordinated policy offers a significant advantage in terms of average cost, even in this slow target scenario. If the target were instead traveling at $v = 9$ [m/s], or half the UAV's airspeed, standoff tracking requires $\varrho_s \geqslant 128.7$ [m] according to (14). Thus, with $\rho_s = 129$ [m], trace$(\mathcal{P}) \approx 92.2$ [m^2], and we have that the average steady steady cost of the optimal policy is nearly 2.5 times less than that of ideal standoff tracking. Of course, constant speed aircraft cannot hold a 90° separation angle at a fixed nominal distance from a constant-velocity target, nor does a target travel at a fixed velocity in a real-world setting. Thus, the numbers presented here for standoff tracking are optimistic.

Overall, the stochastic optimal control approach presents substantial improvements in performance over standoff tracking when the cost is the fused GEC. Recall that the fused GEC is determined by three degrees of freedom, namely the UAV distances ρ_j and their separation angle γ. Accordingly, when one proposes a standoff tracking approach, one loses two of these three degrees of freedom, namely the UAV distances, which are the dominant factors in the cost function. Hence, the performance one can expect from standoff tracking is inherently limited. Thus, while certain applications might require a minimum standoff distance, the degradation in tracking performance with vision sensors is substantial and perhaps warrants the use of alternative sensors, e.g., radar, though such equipment may require larger UAVs.

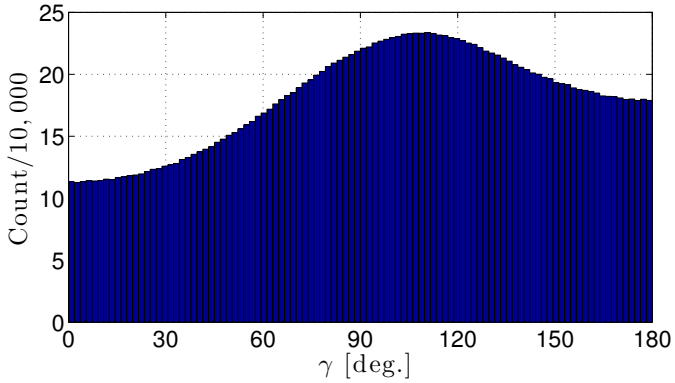


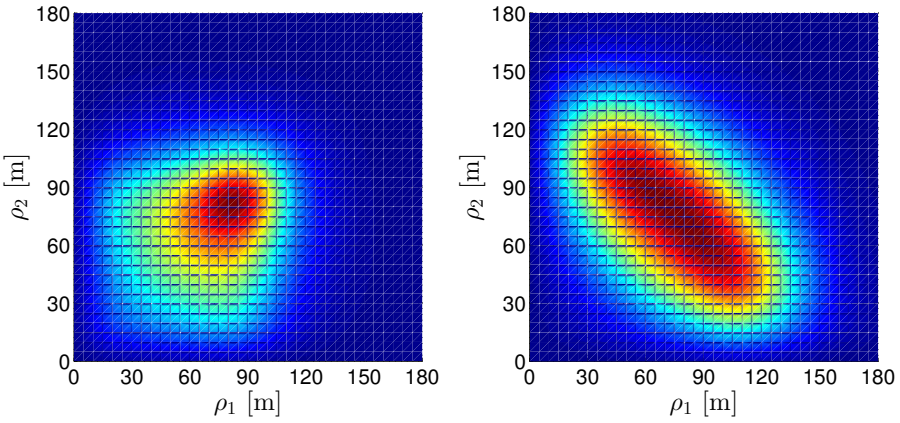
Fig. 16 Histogram of the separation angle γ incurred by the optimal policy during steady-state at each time step and for each of the 50,000 ten-minute Monte Carlo simulations.

5.3 Nature of Optimal Solution

Since we have established the benefits of the optimal policy, we now seek to understand its behavioral qualities. We again use the uncoordinated strategy to generate baseline statistics. We utilize the $M = 50,000$ Monte Carlo simulations that were described at the end of Section 5.2.1 to generate Figures 14 and 15. Recall that each simulation comprises a ten-minute trajectory in steady state.

To assess the level of viewing angle coordination, we have generated the histogram of Figure 16. From this figure, one can see that the optimal control strategy yields orthogonal viewing angles more often than collinear viewing angles, which occur either when $\gamma = 0^\circ$ or $\gamma = 180^\circ$. Even so, while orthogonal viewing angles occur nearly twice as often as $\gamma = 0^\circ$ with the optimal coordinated control policy, they are only 23% more frequent than $\gamma = 180^\circ$. Additionally, the distribution is not nearly an impulse function at $\gamma = 90^\circ$, as would be achieved in an ideal setting with coordinated standoff tracking. In fact, the mode of the distribution occurs near $\gamma = 111^\circ$. Moreover, we conclude that viewing angle coordination is certainly facilitated by the optimal policy but is not necessarily a dominant behavior.

We now assess the level of distance coordination achieved by the optimal policy in comparison with the uncoordinated strategy. To do this, we have smoothed the scatterplot data of the 15.05 million UAV distance pairs to estimate the joint probability density function of planar UAV distances for each control strategy. The results are provided in Figure 17. The joint density function corresponding to uncoordinated policies in Figure 17a is nearly circular around $(\rho_1, \rho_2) = (80, 80)$, which is not surprising since we expect the uncoordinated policies to be equivalent to statistical uncorrelation. However, the joint density function corresponding to the optimal policy in Figure 17b is significantly elongated and shows strong anti-correlation, which indicates that when one UAV is far from the target, the other is most often fairly close to the target. These plots also show that uncoordinated policies generally keep each UAV's distance below 115 meters while the optimal coordinated policy keeps each UAV's distance below 140 meters, as indicated by the maximal values of ρ_1 and ρ_2 associated with the turquoise regions of the probability density estimates. This demonstrates the desired effect of the



(a) Uncoordinated Policies (b) Optimal Policy

Fig. 17 Joint probability density of UAV distances ρ_1 and ρ_2, as determined through Gaussian kernel smoothing. The heat maps range from dark blue to dark red, corresponding to low and high density regions, respectively. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

barrier function of Section 4.3, which becomes active beyond a planar distance of 140 [m] in the computation of the optimal coordinated policy and thus deters individual UAVs from wandering unnecessarily far from the target. Overall, while intuition suggests that minimizing each UAV's individual worst-case distance from the target might be the best strategy based on the fused covariance's sensitivity to distance, it is the coordination of distances that yields optimal performance since Figure 14 is effectively the projection of the two dimensional plots in Figures 17a and 17b into one dimension based on the trace($\mathcal{P}$) functional. Hence, from this test, we conclude that the coordination of distances is the predominant behavior of the optimal control policy.

6 Conclusion

We have presented and studied an effective solution to the problem of optimally coordinating two fixed-wing UAVs to gather the best joint vision-based measurements of a randomly moving ground target. An analytic expression was utilized for the fused geolocation error covariance (GEC) associated with the vision-based measurements, and stochastic fourth-order models were employed for all vehicles to capture realistic system dynamics. While this degree of realism is desirable from a practical point of view, it also renders a 9-dimensional stochastic optimal control problem for which grid-based solutions are impractical. Hence, we presented a simulation-based policy iteration technique known as regression Monte Carlo and adapted it into a policy generation algorithm to remove the need and influence of the initial policy map. To promote fast, reliable regression, we used a partitioned robust regression scheme that utilizes ℓ_1-regularized quadratic fits; as a result, the technique achieves spatial adaptivity and robustness to process noise while capturing nonlinearities in the Q-value.

We conducted a thorough study of the performance and nature of the optimal control policy. When compared with uncoordinated policies, the optimal coordinated policy was shown to achieve lower average costs with a significant reduction in the variance of these costs. Hence, the optimal control policy achieves performance that is not only improved, but also much more predictable. When compared with ideal standoff tracking costs determined for a constant-velocity target at various speeds, both the ensemble average of the optimal policy's costs in steady state and the mean value of its time-averaged costs were shown to be significantly lower. This can be explained by the fact that standoff tracking does not take advantage of the two most dominant of the three factors that determine fused GEC, namely each UAV's planar distance from the target. Moreover, while certain applications might require a minimum standoff distance, the degradation in tracking performance with vision sensors is substantial and perhaps warrants the use of larger UAVs that can carry heavier, active sensors, e.g., radar.

While the optimal policy was shown to facilitate angle coordination to a slight degree, the stronger, more pronounced behavior was shown to be the coordination of distances to the target. The associated optimal trajectories comprise a rich mixture of sinusoidal and orbital trajectories that occasionally pass over or near the target. These behaviors differ both from the standoff tracking approaches that aim to achieve coordinated orbital trajectories centered at the target and the heuristic approaches of [19] and [20] that aim to achieve out-of-phase sinusoids passing over the target. Furthermore, distance coordination is achieved in the presence of stochastic target motion, thereby offering a significant advantage. Nonetheless, should one design a heuristic controller for a multi-UAV target tracking application wherein a minimum standoff distance is not necessary and the cost is analogous to the fused GEC, one should focus on distance coordination rather than viewing angle coordination.

On a final note, we mention that in practice a target's motion may be deterministic over long time intervals, e.g., constant velocity, or it may have a fixed, deterministic control policy. So long as the target's motion respects the dynamical constraints of Section 2.3, such as maximum acceleration and maximum turn rate, it can be viewed as a realization of the stochastic process, albeit with very low probability. Moreover, the stochastic optimal controller is robust to any motion that can be explained by the stochastic model presented in Section 2.3. Of course, the controller is no longer necessarily optimal, since, for example, a constant-speed target that is turning at a constant rate deviates from the zero-mean assumption on the turn rate distribution. If one wished to play optimally against a given target policy, then one would have to either know and plan according to the policy a priori or learn the target's policy online. While the former option is rather impractical, the latter is certainly possible, but it is nontrivial and hence the subject of reinforcement learning [32]. Nonetheless, the present work provides robustness to a wide range of target motion encountered in practice.

One interesting topic for future work is that of using more than two UAVs to track multiple targets. Works that address multi-target tracking include [14] and [33], which typically rely on heuristics to form suboptimal, but computationally tractable, solutions. Using cost function simplifications and computational reductions from symmetry, the present approach could almost certainly be extended to the problem of tracking with three UAVs, which would prove useful for analyzing the return on investment for adding individual agents. Since the computational

demand of the regression scheme presented here grows exponentially in the state space dimension, one would likely need to consider another form of regression. One promising approach is an adaptive RMC approach presented in [34] that adds samples to the stochastic mesh in areas that yield the greatest expected improvement to the quality of the fit until some threshold is met. Moreover, the number and location of points in the stochastic mesh is selected automatically while Bayesian tree-based regression allows for the fits to be updated recursively and the resulting quality assessed. To track multiple targets, a clustering algorithm, such as that presented in [33] for tracking groups of targets traveling in close proximity to one another, could be used to track distinct groups of targets using teams of either one, two, or possibly even three UAVs.

As practical models that have been proven in the field were employed for the UAVs, a natural next step involves testing the optimal control policy in the field to validate its performance. One real-world condition not addressed in this work is wind, yet light to moderate steady winds can be merged with the target velocity to form an apparent target velocity which can then be used in the feedback policy. For more heavier, stochastic winds, one can incorporate wind velocity into the system dynamics, though this would increase the dimensionality of the problem. Nonetheless, since the problem is still tractable with RMC and because wind can play a significant role in the performance of small UAVs, this also remains an open area for future work. Lastly, since the aim of this work is to reduce the error of the vision-based position measurements and thereby facilitate more accurate reconstructions of the full target state with a filter, future work involves testing how the policy affects state estimates from filters such as a particle filter or the robust filter of [35].

7 Acknowledgments

This material is based upon work supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U. S. Army Research Office and by the National Science Foundation under Grant No. CNS-1329650.

8 Appendix: Exploiting Symmetry for Computational Savings

When performing modified RMC, one can exploit key symmetries in the problem to reduce the computational effort considerably. Firstly, the Q-value is symmetric about the relative x-axis in the target-centric state space $\mathcal{Z}$. To describe this, we introduce the reflection matrix $\mathcal{R} = \text{diag}(\boldsymbol{I}_{2\times 2} \otimes \boldsymbol{R}, 1) \in \mathbb{R}^{9\times 9}$, where $\boldsymbol{R} = \text{diag}(1, -\boldsymbol{I}_{3\times 3}) \in \mathbb{R}^{4\times 4}$. This matrix simply comprises 2 copies of the matrix R and unity in a block diagonal fashion. By multiplying the state vector $\boldsymbol{z} \in \mathcal{Z}$ by the reflection matrix, we reflect the relative poses of *both* UAVs simultaneously about the relative x-axis in the target-centric state space.

Taking note of dynamical symmetry, we have that $p(\boldsymbol{z}' \mid \boldsymbol{z}, \boldsymbol{u}) = p(\mathcal{R}\boldsymbol{z}' \mid \mathcal{R}\boldsymbol{z}, -\boldsymbol{u})$. This simply states that the dynamics of the UAV's pose relative to the target are symmetric about the relative x-axis. Furthermore, since simultaneously reflecting all UAV poses preserves both the UAV-target distances as well as the separation

viewing angle γ, $g(\boldsymbol{z}) = g(\mathcal{R}\boldsymbol{z})$. Combining these two properties in (10) yields $Q(\boldsymbol{z}, \boldsymbol{u}) = Q(\mathcal{R}\boldsymbol{z}, -\boldsymbol{u})$. Moreover, from (11), we have that

$$\boldsymbol{\mu}_k^*(\boldsymbol{z}) = -\boldsymbol{\mu}_k^*(\mathcal{R}\boldsymbol{z}),$$

which we henceforth refer to as the *reflection* property.

One can combine the reflection property with two-UAV symmetry for substantial computational savings. By two-UAV symmetry, we mean the property that one can simply relabel the UAVs to account for all possible state configurations when evaluating the cost-to-go. Note that this practice is in reality an approximation since the UAVs operate at different altitudes, although its effects are minor since the altitude difference is small in comparison to either of the altitudes. As an example of the two-UAV symmetry, the set of roll-angle pairs $\mathcal{C}$ can be defined as

$$\mathcal{C} \coloneqq \{\boldsymbol{r} \in C^2 : r_1 \geqslant r_2\}.$$

Thus, with $n_c = |C| = 5$ and $N_c = |\mathcal{C}|$, the total number of roll-angle pairs that needs to be considered has been reduced from $N_c = n_c^2 = 25$ to $N_c = n_c(n_c + 1)/2 = 15$, which is a significant reduction in the computational requirements of Algorithm 3.

Also, as mentioned in Section 5, the partitioning of the relative position states for regression is done approximately in quadrants. With two UAVs, we enforce the position states to be partitioned as quadrants a priori and ensure that m Monte Carlo samples from the initial condition set X exist in each quadrant, where m is the number of samples per regression partition. With this setup, there are initially $4^2 = 16$ possible combinations of quadrants (corresponding to the position states) wherein one needs to perform regression. However, by applying the reflection property, one can eliminate performing regression in the following pairs of quadrants: $(3,3)$, $(4,4)$, $(4,3)$, $(3,4)$, $(2,4)$, and $(4,2)$. Hence, one can eliminate at least $6L_3L_6L_7N_rm = 2880m$ Monte Carlo simulations, which is considerable since m is typically on the order of 10^4.

References

1. M. Mallick, "Geolocation using video sensor measurements," in *IEEE Int. Conf. Information Fusion*, (Quebec, Canada), July 2007.
2. G. E. Collins, C. R. Stankevitz, and J. Liese, "Implementation of a sensor guided flight algorithm for target tracking by small UAS," in *Ground/Air Multi-Sens. Interoperability, Integration, Netw. Persistent ISR II*, vol. 8047, SPIE, April 2011.
3. D. B. Kingston, *Decentralized Control of Multiple UAVs for perimeter and target surveillance*. PhD thesis, Brigham Young University, Dec. 2007.
4. G. Gu, P. R. Chandler, C. J. Schumacher, A. Sparks, and M. Pachter, "Optimum cooperative UAV sensing using a team of UAVs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, pp. 1446 – 1458, October 2006.
5. R. Rysdyk, "UAV path following for constant line-of-sight," in *Proc. 2nd AIAA Unmanned Unltd. Syst. Technol. Operations Aerosp. Land Sea Conf.*, 2003.
6. E. W. Frew, "Lyapunov guidance vector fields for unmanned aircraft applications," in *Am. Control Conf.*, 2007.
7. S. Kim, H. Oh, and A. Tsourdos, "Nonlinear model predictive coordinated standoff tracking of a moving ground vehicle," *J. Guid. Control Dyn.*, vol. 36, no. 2, pp. 557–566, 2013.

8. L. Ma and N. Hovakimyan, "Cooperative target tracking in balanced circular formation: Multiple UAVs tracking a ground vehicle," in *Am. Control Conf.*, pp. 5386–5391, IEEE, 2013.
9. T. H. Summers, *Cooperative Shape and Orientation Control of Autonomous Vehicle Formations*. PhD thesis, University of Texas at Austin, December 2010.
10. H. Oh, S. Kim, A. Tsourdos, and B. A. White, "Decentralised standoff tracking of moving targets using adaptive sliding mode control for UAVs," *J. Intell. Robot. Syst.*, pp. 1–15, 2013.
11. C. Peterson and D. A. Paley, "Multivehicle coordination in an estimated time-varying flowfield," *J. Guid. Control Dyn.*, vol. 34, no. 1, pp. 177–191, 2011.
12. R. Anderson and D. Milutinović, "A stochastic approach to Dubins feedback control for target tracking," in *IEEE / RSJ Conf. Intell. Robots Syst.*, pp. 3917–3922, 2011.
13. S. A. P. Quintero and J. P. Hespanha, "Vision-based target tracking with a small UAV: Optimization-based control strategies," *Control Eng. Pract.*, vol. 32, pp. 28 – 42, 2014.
14. S. A. Miller, Z. A. Harris, and E. K. P. Chong, "A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking," *EURASIP J. Adv. Signal Process.*, pp. 1–17, 2009.
15. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.
16. M. Stachura, A. Carfang, and E. W. Frew, "Cooperative target tracking with a communication limited active sensor network," in *Int. Workshop Robotic Wirel. Sens. Netw.*, (Marina Del Ray, CA), June 2009.
17. C. Ding, A. A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury, "Coordinated sensing and tracking for mobile camera platforms," in *Am. Control Conf.*, pp. 5114–5119, IEEE, 2012.
18. S. A. P. Quintero, F. Papi, D. J. Klein, L. Chisci, and J. P. Hespanha, "Optimal UAV coordination for target tracking using dynamic programming," in *IEEE Conf. Decis. Control*, (Atlanta, GA), Dec. 2010.
19. E. Lalish, K. Morgansen, and T. Tsukamaki, "Oscillatory control for constant-speed unicycle-type vehicles," in *IEEE Conf. Decis. Control*, 2007.
20. N. Regina and M. Zanzi, "UAV guidance law for ground-based target trajectory tracking and loitering," in *Aerosp. Conf.*, IEEE, Mar. 2011.
21. S. A. P. Quintero, G. E. Collins, and J. P. Hespanha, "Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach," in *Am. Control Conf.*, (Washington, D.C.), July 2013.
22. B. Bouchard and X. Warin, "Monte-carlo valorisation of American options: facts and new algorithms to improve existing methods," *Numer. Methods Finance, Springer Proc. Math., ed. R. Carmona, P. Del Moral, P. Hu and N. Oudjane*, 2011.
23. F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. Hoboken, New Jersey: John Wiley and Sons, 3rd ed., 2012.
24. C. Guestrin, M. Hauskrecht, and B. Kveton, "Solving factored mdps with continuous and discrete variables," in *Proc. 20th Conf. on Uncertain. in Artif. Intell.*, pp. 235–242, AUAI Press, 2004.
25. W. Wiegerinck, B. v. d. Broek, and H. Kappen, "Stochastic optimal control in continuous space-time multi-agent systems," in *Proc. 22nd Conf. on Uncertain. in Artif. Intell.*, 2006.
26. F. A. Longstaff and E. S. Schwartz, "Valuing American options by simulation: A simple least-squares approach," *Rev. Financial Stud.*, vol. 14, no. 1, pp. 113–147, 2001.
27. D. Egloff, "Monte carlo algorithms for optimal stopping and statistical learning," *Ann. Appl. Probab.*, vol. 15, no. 2, pp. 1396–1432, 2005.
28. M. Ludkovski and J. Niemi, "Optimal dynamic policies for influenza management," *Statistical Commun. Infect. Dis.*, 2010.
29. D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 2. Belmont, MA: Athena Scientific, 4th ed., 2012.
30. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2 ed., 2009.
31. D. Belomestny, A. Kolodko, and J. Schoenmakers, "Regression methods for stochastic control problems and their convergence analysis," *SIAM J. on Control and Optim.*, vol. 48, no. 5, pp. 3562–3588, 2010.
32. D. Vrabie, K. G. Vamvoudakis, and F. L. Lewis, *Optimal adaptive control and differential games by reinforcement learning principles*, vol. 81. IET, 2013.
33. H. Oh, S. Kim, H.-S. Shin, A. Tsourdos, and B. White, "Coordinated standoff tracking of groups of moving targets using multiple UAVs," in *Control & Automation (MED), 2013 21st Mediterranean Conf. on*, pp. 969–977, IEEE, 2013.

34. R. B. Gramacy and M. Ludkovski, "Sequential design for optimal stopping problems," *SIAM J. on Financial Math.*, 2015. Note: Accepted subject to minor revision.
35. L. Carrillo, W. Russell, J. Hespanha, and G. Collins, "State estimation of multiagent systems under impulsive noise and disturbances," *IEEE Trans. Control Syst. Technol.*, vol. 23, pp. 13–26, Jan 2015.