2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Anytime computation algorithms for stochastically parametric approach-evasion differential games

Erich Mueller[1,2], Sze Zheng Yong[1,2], Minghui Zhu[1,2], Emilio Frazzoli[2]

Abstract— **We consider an approach-evasion differential game where the inputs of one of the players are upper bounded by a random variable. The game enjoys the order preserving property where a larger relaxation of the random variable induces a smaller value function. Two numerical computation algorithms are proposed to asymptotically recover the expected value function. The performance of the proposed algorithms is compared via a stochastically parametric homicidal chauffeur game. The algorithms are also applied to the scenario of merging lanes in urban transportation.**

I. Introduction

Most autonomous robots must operate in the presence of uncertainties in the decisions of non-communicating robots or humans. Thus, an autonomous robot can at best plan based on its "belief" of how likely different outcomes are. Assuming the worst case scenario can result in a very conservative solution, while stochastic approaches lack guarantees of success because there is some probability, however small, of failure. Hence we propose a class of stochastically parameterized differential games that balances conservativeness with system performance. In this approach, we take the belief into account by using a stochastic model to represent the probability of the upper bound on the disturbance inputs. Given the probability density function, we can evaluate the expected minimum time, which is naturally shorter than in the robust case.

Literature review. Differential games provide a quantitative framework to study strategic interconnections among competing decision makers whose actions are constrained by dynamic systems. There have been a limited number of deterministic differential games whose analytic solutions are known, including the homicidal chauffeur and the lady-in-the-lake games [6]. Numerical methods must be employed to determine solutions to more complicated games including those based on partial differential equations; e.g., in [2], viability theory; e.g., in [4] and level-set methods; e.g., in [13]. Stochastic differential games have been studied from both analytical and numerical points of view as well. The technical approaches in [8], [9] rely upon the celebrated Markov chain approximation method; e.g., in [10].

Another relevant set of references is that of robotic motion planning, since approach-evasion differential games can be viewed as robotic motion planning involving multiple agents. Thus, the same challenges of planning a feasible, collision-free path through a cluttered environment are present. But, this also means that sampling based algorithms, such as the Rapidly-exploring Random Tree (RRT) [11], and its asymptotically optimal variant (RRT*) [7] provide a means of solving such games efficiently. Note, however, that RRT-based algorithms only provide open loop strategies, while strategies for approach-evasion differential games are closed-loop. For stochastic motion planning, a sampling-based algorithm that employs chance constraints to guarantee probabilistic feasibility in the presence of motion and sensing noises is proposed in [12], and extended to be asymptotically optimal in [3]. Similarly, chance constraints are applied in [18] where the environment is stochastically parameterized.

Contributions. We consider an approach-evasion differential game where the inputs of one of the players are upper bounded by a random variable. The game enjoys the order preserving property where a larger relaxation of the random variable induces a smaller value function. We propose an extension of the algorithms in [14], namely the coupled-iGame algorithm, which exploits the order preserving game structure. Inspired by incremental sampling in [11], [7], the coupled-iGame algorithm incrementally samples the range of the random variable, constructs a family of discrete games parameterized by the samples where each new game is initialized by utilizing the latest estimates of previously generated games, and incrementally updates the value functions of the discrete games. We show that the weighted sums of the discrete value functions asymptotically recover the expected value function. In order to demonstrate the advantage of exploiting the order preserving property, we propose the decoupled-iGame algorithm where the discrete games are independently updated and use a stochastically parametric homicidal chauffeur game to numerically compare two proposed algorithms. We also apply the coupled-iGame algorithm to merging lanes.

II. Motivating example

To motivate the problem discussed in this paper, we consider the scenario of merging lanes (see Figure 1) with a white accelerating human-driven vehicle in the main traffic lane and a blue (semi-)autonomous vehicle in the merge lane. The dynamics of both vehicles are assumed to be decoupled:

$$\dot{x}_p = f_p(x_p, w), \quad \dot{x}_q = f_q(x_q, u),$$

where $x_p := \begin{bmatrix} p & \dot{p} \end{bmatrix}^\top$ and $x_q := \begin{bmatrix} q & \dot{q} \end{bmatrix}^\top$ are the states of the human-driven vehicle and of the (semi-)autonomous

[1] Equal contribution from these authors.

[2] The authors are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA (e-mail: {emueller, szyong, mhzhu, frazzoli}@mit.edu).

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

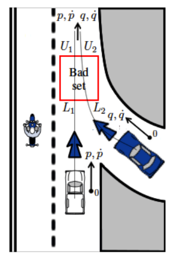


Fig. 1: Merging lanes scenario (modified from [15]).

vehicle, respectively. The control input of the autonomous vehicle has a known bound, $u \in [u_L, u_H]$, whereas the unknown input of the human-driven vehicle is bounded as $|w| \leq r$, where r is a parameter, representing, e.g., the allowable deviation from a nominal behavior. The value of this parameter is modeled as a random variable with a known probability density function, that represents the likelihood for a certain maximum deviation r, obtained either through experiments, or customized as a setting of risk preference, with a risk-averse driver skewing the density to a larger value, and a dare-devil to a smaller value.

This lane merging scenario is an instance of the stochastically parametric differential game which will be defined in Section III-B, where the human-driven vehicle can be seen as the demon and the (semi-)autonomous vehicle the angel. The driver desires to steer the system from the initial state $x(t_0) = (p_0, \dot{p}_0, q_0, \dot{q}_0)$ to a goal set $\mathcal{X}_{\text{goal}} = \{(p, \dot{p}, q, \dot{q}) \in \mathbb{R}^4 \mid q > q_{\text{goal}}\}$ in minimum time while maintaining the system states outside the "bad" set $\mathcal{X}_{\text{bad}} := \{(p, \dot{p}, q, \dot{q}) \in \mathbb{R}^4 \mid (p, q) \in [L_1, U_1] \times [L_2, U_2]\}$, where collision is inevitable. Both vehicles must also adhere to speed limits, i.e. $\mathcal{X} = \{(p, \dot{p}, q, \dot{q}) \in \mathbb{R}^4 \mid \dot{p}, \dot{q} \in [v_{min}, v_{max}]\}$, hence the constraint set is given by $\mathcal{X}_{\text{free}} = cl(\mathcal{X} \setminus \mathcal{X}_{\text{bad}})$, where $cl(.)$ denotes the closure of a set. Conversely, the demon aims to steer the system towards $\mathcal{X}_{\text{bad}}$ quickly and away from $\mathcal{X}_{\text{goal}}$.

III. Problem Formulation

A. *Ordered differential game*

Consider a dynamic system, in which two players, say the angel and the demon, simultaneously control the system:

$$\dot{x}(t) = f(x(t), u(t), w(t)), \quad x(0) = x_0, \tag{1}$$

where the vector $x(t) \in \mathcal{X} \subset \mathbb{R}^N$ represents the system state and the vector $u(t) \in \mathcal{U}$ (resp. $w(t) \in \mathcal{W}(r)$) the control of the angel (resp. the demon). For system (1), the set of admissible control strategies of the angel is given by:

$$\mathcal{U} \triangleq \{u(\cdot) \,:\, [0, +\infty) \to U, \text{ measurable}\},$$

where $U \subset \mathbb{R}^{m_a}$; and the admissible set of strategies for the demon is parameterized by the scalar $r \geq 0$ as follows:

$$\mathcal{W}(r) \triangleq \{w(\cdot) \,:\, [0, +\infty) \to W(r), \text{ measurable}\},$$

with the parametric set $W(r) \triangleq \{w \in \mathbb{R}^{m_d} \mid \|w\| \leq r\}$. Given the initial state x_0 and controls of $u \in \mathcal{U}$ and $w \in \mathcal{W}(r)$, we denote the solution to system (1) as $\phi_r(\cdot; x, u, w) \triangleq \{\phi_r(t; x, u, w)\}_{t \geq 0}$.

The interests of the angel and the demon are exact opposites of each other's. The angel desires to drive the system state $x(t)$ to the open goal set $\mathcal{X}_{\text{goal}} \subset \mathbb{R}^N$ in minimum time, while always maintaining $x(t)$ inside the closed constraint set $\mathcal{X}_{\text{free}} \subset \mathbb{R}^N$, whereas the demon desires to steer $x(t)$ to leave $\mathcal{X}_{\text{free}}$ quickly and keep $x(t)$ away from $\mathcal{X}_{\text{goal}}$. We will refer to the game as the time-optimal approach-evasion (TO-AE, for short) differential game parameterized by r. To formalize the above conflicting objectives, we define as $t_r(x, u, w)$ the first time that the trajectory $\phi_r(\cdot; x, u, w)$ enters $\mathcal{X}_{\text{goal}}$ while staying in $\mathcal{X}_{\text{free}}$ before $t_r(x, u, w)$. More precisely, given the trajectory $\phi_r(\cdot; x, u, w)$, the first hitting time is defined as:

$$t_r(x, u, w) \triangleq \inf\{t \geq 0 \mid \phi_r(t; x, u, w) \in \mathcal{X}_{\text{goal}}, \; \phi_r(s; x, u, w) \in \mathcal{X}_{\text{free}}, \; \forall\, s \in [0, t]\}.$$

If $\phi_r(\cdot; x, u, w)$ leaves $\mathcal{X}_{\text{free}}$ before reaching $\mathcal{X}_{\text{goal}}$ or never reaches $\mathcal{X}_{\text{goal}}$, then $t_r(x, u, w) = +\infty$. The angel strives to minimize the cost $t_r(x, u, w)$, while the demon desires to maximize the same cost functional.

To define the value of each differential game, we need the notion of nonanticipating or causal strategy in the sense of [16]. The set Γ_r^a of such strategies for the angel is such that $\gamma_r^a \,:\, \mathcal{W}(r) \to \mathcal{U}$ satisfies for any $T \geq 0$, $\gamma_r^a(w(t)) = \gamma_r^a(w'(t))$ for $t \in [0, T]$ if $w(t) = w'(t)$ for $t \in [0, T]$. The lower value of the TO-AE differential game is given by:

$$T_r^*(x) = \inf_{\gamma_r^a(\cdot) \in \Gamma_r^a} \sup_{w(\cdot) \in \mathcal{W}(r)} t_r(x, \gamma_r^a(w(\cdot)), w(\cdot)),$$

which we henceforth refer to as the minimum time function. We further define the discounted cost functional $J_r(x, u, w) = \Psi \circ t_r(x, u, w)$, where we considered the Kružkov transform of $\Psi(s) = 1 - e^{-s}$ to normalize the hitting time, and the discounted lower value v_r^* as follows:

$$v_r^*(x) = \inf_{\gamma_r^a(\cdot) \in \Gamma_r^a} \sup_{w(\cdot) \in \mathcal{W}(r)} J_r(x, \gamma_r^a(w(\cdot)), w(\cdot)),$$

which we refer to as the optimal value function for the TO-AE differential game parameterized by r. It can be observed that $v_r^*(x) = \Psi \circ T_r^*(x)$ for $\forall x \in \mathcal{X}$.

Each parametric TO-AE differential game is identical to those in [4], [14] except that the input set of the demon is restricted to be a ball. This family of parametric differential games consists of an *ordered* differential game in the sense that $v_{r'}^*(x) \leq v_r^*(x)$ for any pair of $0 \leq r' \leq r \leq R$ and all $x \in \mathcal{X}$. Note that our ordering is different from that of monotone control systems; e.g., in [1], [5] which is defined on the state space with respect to state and input.

B. *Stochastically parametric differential game*

In this paper, we are interested in the stochastically parametric differential game where r in $\mathcal{W}(r)$ is a random variable defined on the probability space $(\Omega, \sigma(\mathcal{C}([0, R])), P)$ with $\sigma(\mathcal{C}([0, R]))$ being the σ-field generated by the intervals within $[0, R]$. The distribution function of r is denoted by $G(r)$ which is assumed to be absolutely continuous with the density function $g(r)$. With this, the expected optimal value function can be defined as follows:

$$\mathbb{E}[v_r^*(x)] = \int_{[0,R]} v_r^*(x) G(dr) = \int_0^R v_r^*(x) g(r) dr.$$

The objective of the paper is to design and analyze anytime algorithms to asymptotically recover $\mathbb{E}[v_r^*(x)]$ and further synthesize feedback control policies for the angel.

C. Notations and assumptions

Throughout this paper, we make the following assumptions on the dynamic system:

Assumption 3.1: The following properties hold:

(A1) The sets $\mathcal{X}$ and U are compact.

(A2) The function f is continuous in (x, u, w) and Lipschitz continuous in x for any $(u, w) \in U \times W(r)$ and $r \in [0, R]$.

(A3) Given any $r \in [0, R]$, for any pair of $x \in \mathcal{X}$ and $u \in U$, $F_r(x, u)$ is convex where the set-valued map $F_r(x, u) \triangleq \cup_{w \in W(r)} f(x, u, w)$.

Remark 3.1: Since the set $W(r)$ is convex, a sufficient condition for (A3) in Assumption 3.1 is that the function of f is affine with respect to w. •

Because of (A1) and (A2) in Assumption 3.1, $M(r) \triangleq \sup_{x \in \mathcal{X},\, u \in U,\, w \in W(r)} f(x, u, w)$ is well-defined. Let $\ell(r)$ be the Lipschitz constant of f with respect to x for any $(u, w) \in U \times W(r)$. We denote $M \triangleq \sup_{r \in [0,R]} M(r)$ and $\ell \triangleq \sup_{r \in [0,R]} \ell(r)$. Let $\mathcal{B}(x, r)$ be the closed ball centered at x with radius r. Denote by $|\Omega|$ the cardinality of set Ω. Some primitive procedures are defined as follows:

Sample(S, n): return n states which are uniformly and independently sampled from the set S.

Order(Θ): return the set where the elements of set Θ are ordered in a non-deceasing way.

Pre(a, Θ): return $\Theta(i-1)$ if $a = \Theta(i)$ for some $i \geq 2$; otherwise, return a.

IV. COUPLED-IGAME ALGORITHM

In this section, we will present an anytime computation algorithm, namely coupled-iGame, to solve the stochastically parametric differential game and study its asymptotic convergence properties.

A. Algorithm statement

The coupled-iGame is an extension of our iGame algorithm in [14] and informally stated as follows. The variable $\text{flag}(n) \in \{0, 1\}$ is used to determine whether a new game is added at iteration n. If $\text{flag}(n) = 1$, a point r_n is uniformly sampled from the interval $[0, R]$. The distinct samples are stored in the set Θ_n in an increasing way. Then we construct a new TO-AE game for the sample r_n, namely **Game**(r_n), and its value function $v_n(\cdot; r_n)$ is initialized by inheriting the current estimates of **Game**(**Pre**(r_n, Θ_n)). After that, the existing **Game**$(\Theta_{n-1}(i))$ for $i = 1, \cdots, |\Theta_{n-1}|$ are updated in the same way as the iGame algorithm in [14]. More specifically, a sequence of discrete set-valued dynamic systems are incrementally built to approximate the continuous dynamic system (1) in Algorithm 2. A discrete value function is constructed for each discrete set-valued dynamic system, and its update rules are summarized in Algorithm 3. The readers are referred to [14] for detailed discussion on the iGame algorithm.

Algorithm 1 coupled-iGame()

Require: Initially, choose a (small) $\alpha > 0$, $r_0 = 0$, $\Theta_0 = \{r_0\}$ and initialize **Game**(r_0) as follows:

$$S_0 \leftarrow \textbf{Sample}(\mathcal{X}_{\text{free}}, 1), \quad U_0 \leftarrow \textbf{Sample}(U, 1),$$
$$W_0(r_0) \leftarrow \textbf{Sample}(W(r_0), 1), \quad v_0(x; r_0) = 0, \quad x \in S_0.$$

Ensure: At each iterate $n \geq 1$, the following steps are executed:

1: If $\text{flag}(n) = 0$, let $\Theta_n = \Theta_{n-1}$ and execute Algorithm 2;

2: If $\text{flag}(n) = 1$, execute the following steps:
- $\Theta_n = \Theta_{n-1}$;
- Execute Algorithm 2;
- $r_n \leftarrow \textbf{Sample}([0, R], 1)$;
- If $r_n \notin \Theta_{n-1}$, $\Theta_n = \textbf{Order}(\Theta_{n-1} \cup \{r_n\})$ and initialize **Game**(r_n) in the following way:

$$W_n(r_n) \leftarrow \textbf{Sample}(W(r_n), 1),$$
$$v_n(x; r_n) = v_n(x; \textbf{Pre}(r_n, \Theta_n)), \quad x \in S_n;$$

3: Compute the estimate $\mathbb{E}[v_n(x); \Theta_n]$ of $\mathbb{E}[v_r^*(x)]$ as follows:

$$\begin{aligned}\mathbb{E}[v_n(x); \Theta_n] &= \sum_{i=1}^{|\Theta_n|} (\Theta_n(i+1) - \Theta_n(i)) \\ &\times g(\Theta_n(i)) \min_{y \in \mathcal{B}(x, d_n) \cap S_n(\Theta_n(i))} v_n(y; \Theta_n(i)) \\ &\times \frac{1}{\sum_{i=1}^{|\Theta_n|} (\Theta_n(i+1) - \Theta_n(i)) \times g(\Theta_n(i))},\end{aligned} \tag{2}$$

where we use $\Theta_n(|\Theta_n + 1|) = R$.

Algorithm 2

1: $y_n \leftarrow \textbf{Sample}(\mathcal{X}_{\text{free}}, 1)$;

2: $S_n = S_{n-1} \cup \{y_n\}$;

3: $U_n = U_{n-1} \cup \textbf{Sample}(U, 1)$;

4: Update the dispersion d_n, generate the time discretization size $h_n = d_n^{\frac{1}{1+\alpha}}$, and the dilation size $\alpha_n = 2d_n + \ell h_n d_n + M \ell h_n^2$;

Ensure: For each $i \in \{1, \cdots, |\Theta_n|\}$, the following steps are executed for **Game**$(\Theta_n(i))$:

1: Generate the interpolated function $\tilde{v}_{n-1}(\cdot; \Theta_n(i))$: $\tilde{v}_{n-1}(y_n; \Theta_n(i)) = 1$ and for $x \in S_{n-1}$,

$$\tilde{v}_{n-1}(x; \Theta_n(i)) = v_{n-1}(x; \Theta_n(i));$$

2: Update the estimate $v_n(\cdot; \Theta_n(i))$ as follows: for each $x \in S_n \setminus \mathcal{B}(\mathcal{X}_{\text{goal}}, M h_n + d_n)$, execute **VI'**; for each $x \in S_n \cap \mathcal{B}(\mathcal{X}_{\text{goal}}, M h_n + d_n)$,

$$v_n(x; (\Theta_n(i))) = \tilde{v}_{n-1}(x; (\Theta_n(i))).$$

The coupled-iGame algorithm is formally stated in Algorithm 1. Some notations are defined as follows. The state dispersion d_n is the quantity such that for any $x \in \mathcal{X}_{\text{free}}$, there exists $x' \in S_n$ such that $\|x - x'\| \leq d_n$. We let $h_n = d_n^{\frac{1}{1+\alpha}}$ and $\kappa_n = h_n - d_n$.

B. Algorithm analysis

In the coupled-iGame algorithm, the new games are generated infinitely often. This is formally stated as follows:

Assumption 4.1: The event $\text{flag}(n) = 1$ occurs infinitely often.

The following theorem summarizes the asymptotic convergence properties of the coupled-iGame algorithm.

Theorem 4.1: Suppose Assumptions 3.1 and 4.1 hold. Then the sequence of $\mathbb{E}[v_n(\cdot); \Theta_n]$ generated by the coupled-iGame algorithm converges pointwise to $\mathbb{E}[v_r^*(\cdot)]$; i.e., for

Algorithm 3 VI'

1: Implement the following step:

$$W_n(\Theta_n(i)) = W_{n-1}(\Theta_n(i)) \cup \textbf{Sample}(W(\Theta_n(i)), 1);$$

2: Solve the following problem:

$$v_n(x; \Theta_n(i)) = 1 - e^{-\kappa_n} + e^{-\kappa_n} \times \max_{w \in W_n(\Theta_n(i))} \min_{u \in U_n, y \in \mathcal{B}(x + h_n f(x,u,w), \alpha_n) \cap S_n} \tilde{v}_{n-1}(y; \Theta_n(i)),$$

and set $u_n(x; \Theta_n(i))$ to be one of the solution to u.

Algorithm 4 decoupled-iGame()

Require: Initially, choose a (small) $\alpha > 0$, $r_0 = 0$, $\Theta_0 = \{r_0\}$ and initialize $\textbf{Game}(r_0)$ as follows:

$$S_0(r_0) \leftarrow \textbf{Sample}(\mathcal{X}_{\text{free}}, 1), \;\; U_0(r_0) \leftarrow \textbf{Sample}(U, 1),$$
$$W_0(r_0) \leftarrow \textbf{Sample}(W(r_0), 1), \;\; v_0(x; r_0) = 0, \;\; x \in S_0(r_0).$$

Ensure: At each iterate $n \geq 1$, the following steps are executed:

1: If flag$(n) = 0$, let $\Theta_n = \Theta_{n-1}$ and execute Algorithm 5;

2: If flag$(n) = 1$, execute the following steps:

- $\Theta_n = \Theta_{n-1}$;
- Execute Algorithm 5;
- $r_n \leftarrow \textbf{Sample}([0, R], 1)$;
- If $r_n \notin \Theta_{n-1}$, then $\Theta_n = \Theta_{n-1} \cup \{r_n\}$ and initialize $\textbf{Game}(r_n)$ in the following way:

$$S_n(r_n) \leftarrow \textbf{Sample}(\mathcal{X}_{\text{free}}, 1), \;\; U_n(r_n) \leftarrow \textbf{Sample}(U, 1),$$
$$W_n(r_n) \leftarrow \textbf{Sample}(W(r_n), 1),$$
$$v_n(x; r_n) = 0, \quad x \in S_n(r_n);$$

3: Compute the estimate as (2).

any $x \in \mathcal{X}_{\text{free}}$, it holds that

$$\lim_{n \to +\infty} \mathbb{E}[v_n(x); \Theta_n] = \mathbb{E}[v_r^*(x)]. \qquad (3)$$

V. Decoupled-iGame algorithm

In this section, we will present a variation, namely decoupled-iGame, of the coupled-iGame algorithm where the games for different samples are completely independent. In Section VI, we will compare two algorithms and demonstrate the advantage of exploiting the order preserving property.

Informally speaking, the decoupled-iGame algorithm samples the interval $[0, R]$ repeatedly and constructs a TO-AE game for each sample. Then the algorithm *independently* and incrementally refines the family of parametric TO-AE games and updates the associated value functions. The update rules for each parametric game are the same as those of the iGame algorithm in [14]. Here the *independency* means that the parametric TO-AE games do not share the sample sets for the state and controls as well as the estimates of their value functions. The algorithm then produces an estimate of the expected value function by weighting the current estimates with the probability densities. The decoupled-iGame algorithm is formally stated in Algorithm 4. Some notations are defined as follows. The state dispersion $d_n(\Theta_n(i))$ is the quantity such that for any $x \in \mathcal{X}_{\text{free}}$, there exists $x' \in S_n(\Theta_n(i))$ such that $\|x - x'\| \leq d_n(\Theta_n(i))$. We let $h_n(\Theta_n(i)) = d_n(\Theta_n(i))^{\frac{1}{1+\alpha}}$ and $\kappa_n(\Theta_n(i)) = h_n(\Theta_n(i)) - d_n(\Theta_n(i))$.

The following theorem summarizes the asymptotic convergence properties of the decoupled-iGame algorithm.

Algorithm 5

Ensure: For each $i \in \{1, \cdots, |\Theta_n|\}$, the following steps are executed for $\textbf{Game}(\Theta_n(i))$:

1: $y_n(\Theta_n(i)) \leftarrow \textbf{Sample}(\mathcal{X}_{\text{free}}, 1)$;

2: $S_n(\Theta_n(i)) = S_{n-1}(\Theta_n(i)) \cup \{y_n(\Theta_n(i))\}$;

3: Update the dispersion $d_n(\Theta_n(i))$, generate the time discretization size $h_n(\Theta_n(i)) = d_n(\Theta_n(i))^{\frac{1}{1+\alpha}}$, and the dilation size $\alpha_n(\Theta_n(i)) = 2d_n(\Theta_n(i)) + \ell h_n(\Theta_n(i)) d_n(\Theta_n(i)) + M\ell h_n(\Theta_n(i))^2$;

4: Generate the interpolated function $\tilde{v}_{n-1}(\cdot; \Theta_n(i))$: $\tilde{v}_{n-1}(y_n; \Theta_n(i)) = 1$ and for $x \in S_{n-1}(\Theta_n(i))$,

$$\tilde{v}_{n-1}(x; \Theta_n(i)) = v_{n-1}(x; \Theta_n(i));$$

5: Update the estimate $v_n(\cdot; \Theta_n(i))$ as follows: for each $x \in S_n(\Theta_n(i)) \setminus \mathcal{B}(\mathcal{X}_{\text{goal}}, M h_n(\Theta_n(i)) + d_n(\Theta_n(i)))$, execute **VI**; for each $x \in S_n(\Theta_n(i)) \cap \mathcal{B}(\mathcal{X}_{\text{goal}}, M h_n(\Theta_n(i)) + d_n(\Theta_n(i)))$,

$$v_n(x; (\Theta_n(i))) = \tilde{v}_{n-1}(x; (\Theta_n(i))).$$

Algorithm 6 VI

1: Implement the following steps:

$$U_n(\Theta_n(i)) = U_{n-1}(\Theta_n(i)) \cup \textbf{Sample}(U, 1),$$
$$W_n(\Theta_n(i)) = W_{n-1}(\Theta_n(i)) \cup \textbf{Sample}(W(\Theta_n(i, 1)), 1);$$

2: Solve the following problem:

$$v_n(x; \Theta_n(i)) = 1 - e^{-\kappa_n(\Theta_n(i))} + e^{-\kappa_n(\Theta_n(i))} \times \max_{w \in W_n(\Theta_n(i))} \min_{u \in U_n(\Theta_n(i))} \min_{y \in \mathcal{B}(x + h_n(\Theta_n(i)) f(x,u,w), \alpha_n(\Theta_n(i))) \cap S_n(\Theta_n(i))} \tilde{v}_{n-1}(y; \Theta_n(i)),$$

and set $u_n(x; \Theta_n(i))$ to be one of the solution to u.

Theorem 5.1: Suppose Assumptions 3.1 and 4.1 hold. Then the sequence of $\mathbb{E}[v_n(\cdot); \Theta_n]$ produced by the decoupled-iGame algorithm converges pointwise to $\mathbb{E}[v_r^*(\cdot)]$; i.e., for any $x \in \mathcal{X}_{\text{free}}$, it holds that

$$\lim_{n \to +\infty} \mathbb{E}[v_n(x); \Theta_n] = \mathbb{E}[v_r^*(x)]. \qquad (4)$$

VI. Simulation results

A. Homicidal Chauffeur

First, we consider a variant of the classic homicidal chauffeur problem, in which a faster, but less agile pursuer attempts to capture a slower, more maneuverable evader [14]. This problem can be reduced to 2 dimensions, in which the dynamics are given by:

$$\dot{x} = u_p y + v_e \cos u_e - v_p \qquad \dot{y} = -u_p x - v_e \sin u_e$$

We take on the role of the pursuer, whose input is denoted u_p, and attempt to capture the evader, whose input is denoted u_e, in minimum time. In this case, the maximum speed at which the evader moves is uncertain. This problem was used to compare the performance of decoupled-iGame and coupled-iGame. Figure 2 shows the value function approximation error for both, using a 300x300 fixed grid dynamic programming solution for reference. This data represents the mean approximation error over the points of a 50x50 uniform grid in the region $x \in [-1, 1]$ and $y \in [-1, 1]$. The pursuer input was restricted to $u_p \in [-1, 1]$ and the evader input u_e

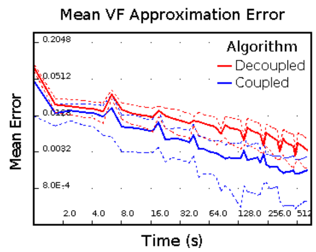


Fig. 2: Value function approximation error vs elapsed computation time for the decoupled-iGame and coupled-iGame algorithms. The error is averaged over 20 trials, with dashed lines indicating the standard deviation. Note the logarithmic scaling of both axes.

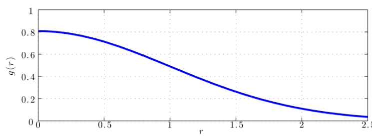


Fig. 3: Probability density function of random variable r (One-sided truncated normal distribution).

was unrestricted, however the evader speed was restricted as $|v_e| \in [0, r]$. The random input bound r was considered to have a uniform probability distribution between 0.1 and 0.9. A new game was added to the collection every 300 iterations.

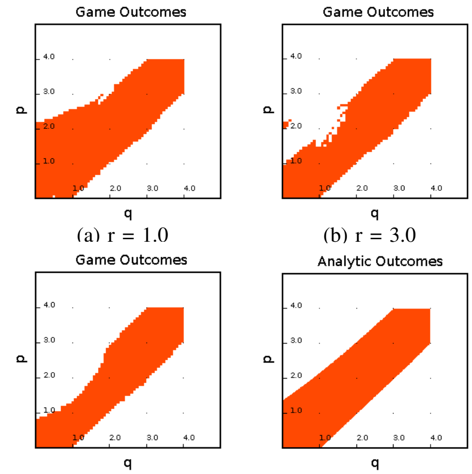
These results demonstrate the expected advantage of sharing value function information between the different games, as the coupled version converges more quickly to the high resolution fixed grid solution.

B. Merging Lanes

We shall now revisit the motivational example with the fully autonomous vehicle in Section II. We model the human-driven vehicle as in [17], as $\ddot{p} = \beta + \gamma w$, where β and γ are the mean acceleration and its standard deviation obtained from experiments with human subjects, and $|w| \leq r$ is an unknown input, where the input bound r is a random variable with probability density function $g(r)$ on a finite support $[0, R]$ shown in Figure 3. For the (semi-)autonomous vehicle, we also adopt the model in [17]: $\ddot{q} = au - b - c\dot{q}^2$, where a, b and c are positive constants corresponding to input gain, static friction and air drag coefficient terms, respectively, while the control input is bounded, $u \in [u_L, u_H]$.

For this example, we simulated the problem with the following parameters (similar to [17]): $p_0 = 0\,m$, $\dot{p}_0 = 1\,\frac{m}{s}$, $q_0 = 0\,m$, $\dot{q}_0 = 1\frac{m}{s}$. $q_{\text{goal}} = 4\,m$, $L_1 = L_2 = 3\,m$, $U_1 = U_2 = 4\,m$, $v_{min} = 0.35\frac{m}{s}$, $v_{max} = 1.1\frac{m}{s}$, $R = 2.5$, $\beta = 0.3505\frac{m}{s^2}$, $\gamma = 0.1396\frac{m}{s^2}$, $a = 0.7\frac{m}{s^2}$, $b = 0.1\frac{m}{s^2}$, $c = 0.01\frac{1}{m}$, $u_L = -0.2$ and $u_H = 1$. The state space for this problem is chosen as $\mathcal{X} = [0, 5] \times [0.35, 1.1] \times [0, 5] \times [0.35, 1.1] \subset \mathbb{R}^4$.

This problem was first used to demonstrate the effect of erroneous estimation of the uncertain parameter r. Several fixed grid solutions were generated using different values for r. Each solution was used by the merging car to play the game from a set of initial conditions, against a non-merging car who used a fixed grid solution based on the correct value of r. Figure 4 shows a slice in position space of the set of initial conditions from which collision results, for several



(a) r = 1.0 (b) r = 3.0 (c) r = 5.0 (d) analytic, r = 3.0

Fig. 4: Safe (white) and unsafe (red) sets resulting from the use of different values of r. The merging car uses the specified value, while the other car uses the correct value $r = 3$. The analytically computed safe and unsafe regions with the correct value of r are shown in 4d. $\dot{p} = 0.75$ and $\dot{q} = 0.75$ for all points.

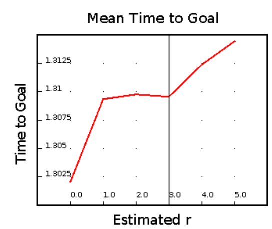
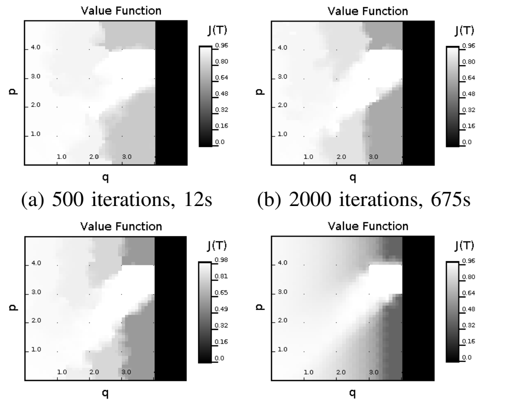


Fig. 5: Mean time to reach the goal across initial states safe for all estimates of r, as shown and described in Figure 4, versus the estimated value of r.

estimated values of r. These plots can be compared to the analytically computed sets, shown in Figure 4d. Figure 5 shows the mean time to reach the goal for the same set of tests, with the mean taken over all initial conditions from which the goal was safely reached for all tested estimates of r. These results show that it is indeed beneficial to use a more accurate estimate of the unknown parameter, as the size of the unsafe set is seen to shrink as the estimate becomes more conservative, while an overly conservative estimate can lead to an increase in the mean time to reach the goal. These results seem to show that it may be possible for an overly conservative estimate to be better than an accurate estimate. This should not be possible theoretically, and is most likely a result of the numerical discretization used for computation.

The merging lanes problem was also used to show convergence of the mean value function approximation generated by coupled-iGame to a high resolution fixed grid solution. Figure 6 shows the progression of the value function approximation and the fixed grid approximation. Here, 10 20x20x20x20 fixed grid solutions were used to compute the reference mean value function, seen in Figure 6d. A new game was added to the set used by coupled-iGame every 200 iterations. To compute the mean, the value functions for the different games were weighted according to their asso-



(a) 500 iterations, 12s (b) 2000 iterations, 675s

(c) 4000 iterations, 5629s (d) fixed grid

Fig. 6: Value function approximations for the lane merging problem, generated by the coupled-iGame algorithm, in comparison to a 160000 point fixed grid solution. The mean value function, taken across variation in the unknown parameter r, is shown. $\dot{p} = 0.75$ and $\dot{q} = 0.75$ for all points. Results are shown for different numbers of iterations.

TABLE I: False and missing warning rates

Iterations	Elapsed Time (s)	False Warnings	Missing Warnings
200	1.3	0.197	0.021
400	6.3	0.103	0.009
800	43	0.072	0.015
1600	354	0.053	0.003

ciated probability mass, as given in (2), with the probability density function shown in Figure 3. In a semi-autonomous application, the mean value function computed by coupled-iGame could be used by a warning system to trigger an alarm if an unsafe state is entered. One way to achieve this is to warn the driver if the mean value function at the current state exceeds some threshold. Figure 7 shows the result of imposing a warning threshold of 0.998 on the mean value function approximation computed by coupled-iGame and on a fixed grid. This threshold was selected to correspond to the latest possible time that the goal region can be reached by the merging car. Here we can see, as in Figure 6, that the solution generated by coupled-iGame converges to the much more accurate solution. Table I shows the mean over 20 trials of the fraction of safe states deemed unsafe by coupled-iGame (false warnings) and of unsafe states deemed safe (missing warnings) for this test, excluding states in the collision region, using the fixed grid solution shown in figure 6d as truth.

ACKNOWLEDGMENTS

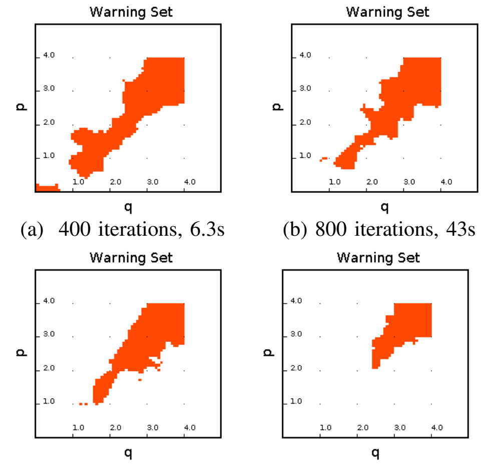
The work presented in this paper was supported in part by the National Science Foundation, grant #1239182, and by the Air Force Office for Scientific Research, grant #FA8650-07-2-3744.

REFERENCES

[1] D. Angeli and E. Sontag. Monotone control systems. *IEEE Transactions on Automatic Control*, 48(10):1684 – 1698, 2003.

[2] M. Bardi and I. Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Birkhäuser, 1997.

[3] A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 723–730, May 2011.



(a) 400 iterations, 6.3s (b) 800 iterations, 43s

(c) 1600 iterations, 354s (d) fixed grid

Fig. 7: Warning sets (darker, red) for semi-autonomous collision avoidance, generated by coupled-iGame. A fixed grid solution is also shown. $\dot{p} = 0.75$, $\dot{q} = 0.75$.

[4] P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre. Set-valued numerical analysis for optimal control and differential games. *Annals of the International Society of Dynamic Games*, 4(1):177 – 247, 1999.

[5] M. Hafner and D.D. Vecchio. Computational tools for the safety control of a class of piecewise continuous systems with imperfect information on a partial order. *SIAM Journal on Control and Optimization*, 49(6):2463 – 2493, 2011.

[6] R. Isaacs. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. Dover, 1999.

[7] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846 – 894, 2011.

[8] H.J. Kushner. Numerical approximations for stochastic differential games: the ergodic case. *SIAM Journal on Control and Optimization*, 42(6):1911 – 1933, 2004.

[9] H.J. Kushner. Numerical approximations for non-zero-sum stochastic differential games. *SIAM Journal on Control and Optimization*, 46(6):1942 – 1971, 2007.

[10] H.J. Kushner and P. Dupuis. *Numerical methods for stochastic control problems in continuous time*. Springer-Verlag, 2001.

[11] S.M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Dept., Iowa State University, 1998.

[12] B. Luders, M. Kothari, and J. P. How. Chance constrained RRT for probabilistic robustness to environmental uncertainty. In *AIAA Guidance, Navigation, and Control Conference (GNC)*, Toronto, Canada, August 2010. (AIAA-2010-8160).

[13] I. Mitchell, A. Bayen, and C. Tomlin. Validating a Hamilton Jacobi approximation to hybrid reachable sets. In *Hybrid Systems: Computation and Control*, pages 917 – 927, 2004.

[14] E. Mueller, M. Zhu, S. Karaman, and E. Frazzoli. Anytime computation algorithms for approach-evasion differential games. 2013. Preprint.

[15] California Department of Motor Vehicle. Motorcycle handbook.

[16] P. Varaiya, R.J. Elliott, E. Roxin, and N.J. Kalton. The existence of value in differential games. *American Mathematical Society*, (126), 1972.

[17] R. Verma and D. Del Vecchio. Semiautonomous multivehicle safety. *Robotics Automation Magazine, IEEE*, 18(3):44 –54, September 2011.

[18] M.P. Vitus and C.J. Tomlin. Belief space planning for linear, gaussian systems in uncertain environments. In *18th World Congress of the International Federation of Automatic Control, Milan, Italy*, pages 5902–5907, 2011.