2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

A Graph Theoretical Approach Toward a Switched Feedback Controller for Pursuit-Evasion Scenarios

Brian Goode, Andrew Kurdila and Mike Roan

***Abstract*— This research introduces a novel method for constructing a switched feedback control system to be used for an autonomous agent. The state space is partitioned into sets of states where a specific control is applied. Each partition is represented by nodes of a digraph where the success of the control in traversing the partitions is represented by a connecting edge. Using the concept of capture sets in the field of differential games, it is shown that a set of states included in a particular partition is capable of reaching the target set if the eigenvalues of the adjacency matrix representing the digraph are all zero and none of the partitions are invariant. The advantage of this method is that it is possible to assign finite horizon controls to each partition that are easier to calculate than infinite horizon methods, but still maintain the infinite horizon guarantee of reaching the target. An example is given to illustrate the implementation of the proposed controller.**

I. Introduction

One challenge in the autonomous control field is guiding an agent through a state space with physical obstacles. The agent must both identify and provide guidance around these obstacles to reach a specified target in finite time. Pursuit-evasion scenarios without explicit physical obstacles are similar in that the obstacles now become an opposing agent's dynamic capabilities and strategy. However, they differ in that an opposing agent's strategy may not be measured with a sensor, and has to be predicted based on what is known about the dynamic capabilities from the output states. Fortunately, using parameter estimation and filtering techniques, the dynamic capabilities can be measured, but all possible actions of the opposing agent need to be taken into consideration. The control presented here takes an obstacle avoidance approach to pursuit-evasion by forming a switched control over state partitions. Here, "obstacles" are the inability to transition from one partition to the next.

Classic solutions to pursuit-evasion implement Bellman optimality by solving the infinite horizon Hamilton-Jacobi-Bellman (HJB) [1] and Hamilton-Jacobi-Isaacs (HJI) [2] equations to obtain the agent controls using a Value function approach. Calculating the Value function [2]–[5] effectively exhausts all possible open-loop controls, because the solution iteration begins at the set of target states and expands outward. If a descending gradient in the Value exists along trajectories to the target, then the agent will succeed in target capture. However, there are several disadvantages to these methods such as the curse of dimensionality, and states that are farthest from the target take the longest to calculate. Solutions such as the Fast Marching Semi-Lagrangian (FMSL) methods [3] seek to alleviate this burden by reducing the set of states involved in the calculations to those where the value has not yet converged. Methods using extremal aiming techniques have a front-based propagation making them faster for short distances, but must identify and compensate for discontinuities that develop as the front propagates as seen in [6]. Other difficulties arise, because the opposing agents' strategies must be known in order to calculate the infinite horizon solution. This is not an entirely unreasonable requirement for many scenarios, but it is still not practical to re-calculate the controls for games when the opposing agent's strategy is learned.

Authors 1 and 3 sponsored by ONR Contract No. N00014-06-1-0356
Department of Mechanical Engineering, Virginia Polytechnic Institute and State University, 136 Durham Hall, Blacksburg, VA 24060 bjgoode@vt.edu, kurdila@vt.edu, mroan@vt.edu

Methods using a finite prediction horizon tend to offer fewer computations as a solution technique. These types of solutions appear in work such as [7]. A good method for finding solutions to probabilistic differential games with observational uncertainty using a one-step Nash approach was proposed in [8]. Finite horizon approaches are very attractive for their quick computational aspects and incorporating observations. However, [9] shows that even with perfect observations assigned to both agents, a finite horizon is not enough to successfully guide the agent to the target set. It is possible for the controller to produce trajectories that may either enter a stable equilibrium or form a limit cycle outside of the target set. In systems where the finite step objective is evaluated repeatedly with learning, methods do exist to evaluate their temporal performance [10]. However, these strategies may still converge to an equilibrium that is not in the target set.

This work seeks to increase the structural complexity of a finite controller in order to reduce the computational load to process the control. First, an intuitive method of assigning finite horizon objectives over partitions of the state space is given. Using a graph theoretical approach, the trajectory may then be evaluated over the infinite horizon for target reachability. The opposing agent dynamics are then estimated using the FMSL method in select partitions, reducing computational time. This gives the controller more flexibility in adapting to various scenarios and opponents.

II. Problem Formulation

The system consists of two agents. One agent has a known set of dynamic capabilities whose strategy will be designed before the onset of the game. The other agent is considered a disturbance where their strategy is unknown and the dynamic capabilities are not known until the game begins at $t = 0^+$. A solution of the system is a trajectory, $w = (x, u, d)$, where

$w \in \mathcal{W}$ is the set of valid trajectories. The state, x, is a member of the compact state space $X \subset \mathbb{R}^n$. The control of the known agent, u, is a finite set $U \subset \mathbb{N}$, and the disturbance input, d, caused by the disturbing agent is a finite set $D \subset \mathbb{N}$. It is assumed that both u and d are mappings to compact subsets in $\mathbb{R}$.

Each trajectory in $\mathcal{W}$ must satisfy the dynamic equations

$$\dot{x} = \mathbf{f}(x,u,d) = \mathbf{f}_A(x,u) + \mathbf{f}_D(x,d)$$

for all time, $t \in \mathbb{R}^+$, from some initial state, $x^{t_0} = x_0$. In this formulation, the dynamics of the agents are exogenous and separable. State evolutions for a particular control strategy, $u(\cdot)$, and disturbance, $d(\cdot)$, are found using

$$x^{t_f} = \mathcal{O}[\mathbf{f}(x,u,d), x_0, t_f] \tag{1}$$

where $\mathcal{O}$ is an operator that integrates $\mathbf{f}(x,u,d)$ over the interval, $[t_0, t_f]$.

The switched control should form a trajectory in $\mathcal{W}$ such that $x^{t>t^*} \in \mathcal{T}$ for some $t_0 < t^* < \infty$ where $\mathcal{T} \subset X$. $\mathcal{T}$ is termed the target set for the known agent and is positively invariant such that a trajectory never leaves upon entry. The game terminates when the state has entered the target set, a stable equilibrium point, or limit cycle.

III. Control Law

The control law used in this formulation has the canonical feedback form

$$u = \mathcal{M}(x) \tag{2}$$

It is desired that the function, $\mathcal{M}(x)$, form a valid trajectory from a maximal number of initial states to reach target $\mathcal{T}$ in a finite time.

A. Switched Feedback Structure

The control law considered is of the form

$$\mathcal{M}(x) = \mathbf{m}^T \mathbf{I} \tag{3}$$

where $\mathbf{m}$ is a vector of q feedback controls

$$\mathbf{m} = \begin{bmatrix} m_1(x) & m_2(x) & \cdots & m_q(x) \end{bmatrix}^T \tag{4}$$

and membership of a control to a particular set of states is governed by

$$\mathbf{I} = \begin{bmatrix} \mathbf{1}(x \in \sigma_1) & \mathbf{1}(x \in \sigma_2) & \cdots & \mathbf{1}(x \in \sigma_q) \end{bmatrix}^T \tag{5}$$

where $\mathbf{1}()$ is the indicator function and $\sigma_1, \ldots, \sigma_q$ are partitions of states.

B. State Partitions

The state partitions are a cover of the state space, X, that are identified by a set of symbols

$$\Sigma := \{\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_q\}$$

where q corresponds to the number of partitions in the state space, each of which has an associated feedback control $m_q(x)$. The subscript of the partition label is referred to as the degree. It is required that the cover satisfy

$$X = \bigcup_{k=0}^{q} \sigma_k$$

where the partitions are open along borders of higher degree partitions and closed along boundaries of lower degree partitions. The only stipulation placed on the assignment of states to partitions is that $\sigma_0 = \mathcal{T}$.

IV. Capture Set

The main objective of this method is to identify the states using the switched control of Eqn. 3 that are capable of traversing to the target set in finite time.

Definition 1: The capture set includes all the states that satisfy

$$\mathcal{C}_F(d(\cdot), \mathcal{M}(x)) := \Big\{x_0 \in X | (x,u,d) \in \mathcal{W}, \\ u = \mathcal{M}(x), x^{t>t^*} \in \mathcal{T}, t_0 < t^* < \infty\Big\}$$

where x^t is found using the operator, $\mathcal{O}[\mathbf{f}(x,u,d), x_0, t]$.

The capture set consists of all the initial states where there is a valid trajectory that terminates in the target set. This condition is usually a rather cumbersome task to evaluate using infinite horizon solutions. However, the rest of this paper is dedicated to showing that adding more complexity to the structure of the control through intuitive methods reduces the calculation time of the capture set.

A. Properties of the Adjacency Matrix

From a graph theoretical perspective, the partitions, Σ, form the nodes of a digraph where edges occur if

$$\exists (w \in \mathcal{W}, \bar{t}) \; s.t. \; x^t \in \sigma_k, x^{\bar{t}} \in \sigma_{j \neq k}, t < \bar{t} < \infty.$$

This defines a single transition, and there may be multiple such transitions of a given trajectory that form a path. In this formulation, a loop edge occurs when the agent tries to transition, fails, and remains in the same partition.

The digraph is represented as a square adjacency matrix, A, that is populated according to the rule,

$$a_{ij} = \begin{cases} 0, & \text{if there is no edge from node } i \text{ to } j \\ 1, & \text{if there is an edge from node } i \text{ to } j \end{cases}$$

In terms of state partitions, the diagonal of the matrix refers to failed attempts at transitioning. Everywhere else corresponds to a successful transition. Furthermore, it is well known that the matrix, A^q, contains a count of how many paths of length q exist from node i to j [11].

B. Inclusion of a Partition in the Capture Set

From Def. 1, an initial state that is to be included in the capture set must satisfy:

1) The trajectory, w, must be valid in the sense that $(x,u,d) \in W$ with $u = \mathcal{M}(x)$.
2) The trajectory must reach the target from the initial state in finite time.

By definition, if an edge exists between two node partitions, then the trajectory is valid. However, the question of reaching the target in finite time is still open. By partitioning similar states in the state space, a whole subset of states may be evaluated. Therefore, the capture set for a switched feedback controller can be formulated in terms of the partitions.

Theorem 1: Assume a switched feedback law, $\mathcal{M}(x)$, is given to an agent and the opposing agent uses a control, $d(\cdot)$. Set $\bar{A} = \sum_{k=1}^{q} A^k$. If

1) A has all 0 eigenvalues,
2) The entry, $\bar{A}_{i1} > 0$ for $i > 1$, and
3) for all entries, $\bar{A}_{j1} = 0, j > 1$, then, $\bar{A}_{ij} = 0, \forall i$.

then partition, $\sigma_i \in \mathcal{C}_F(d(\cdot), \mathcal{M}(x))$.

Proof: A matrix is nilpotent if and only if it has all 0 eigenvalues (for proof, see [12]). This means there is an integer,

$$\exists k \in \{1, \ldots, q\} \; s.t. \; A^k = \mathbf{0}$$

For higher powers,

$$A^j = A^k A^{j-k}, j > k$$

such that

$$A^j = \mathbf{0}, \forall j > k$$

Because, A^k determines how many paths of length k exist between two partitions, the zero eigenvalue guarantees that all paths from any node to another node are finite in the number of segments composing the path. There are no cycles or loops that the system can be caught in. Since the number of segments of every path is finite, then this guarantees that all trajectories across the partitions will stay in a final partition in a finite amount of time.

To show that the final partition can only be the target set (σ_0), consider a finite path of the digraph that terminates in any partition $\sigma_{j\neq 0}$. Termination in a partition implies positive invariance because it never leaves that set of states upon entry. There are two modes by which this will occur: (a) the trajectory started in σ_j or (b) the trajectory traversed into σ_j from another partition. If (a) is true, then $\bar{A}_{i1} = 0, 1 < j = i$ which contradicts condition 2. Likewise, if (b) is true, then $\bar{A}_{j1} = 0, 1 < j \neq i$ and $\bar{A}_{ij} \geq 1$, which contradicts condition 3. Therefore, if the conditions 1-3 hold, then the trajectory is guaranteed to terminate in σ_0. ■

This theorem presents the sufficient conditions to show that a given partition is in the capture set. The conditions presented here are restrictive in that they require all trajectories to traverse a finite number of partitions. It is quite trivial to contrive a case where some partitions will result in a cyclic path, but the trajectory remains in the part of the graph where paths are finite and terminating in σ_0. However, for pursuit-evasion scenarios, this is not desirable and the restriction is easy to impose. Furthermore, requiring that the trajectories are finite allows for a quick calculation of the capture set.

V. Implementation

The major result of the previous section is that the problem of ensuring that a trajectory reaches the target set may be decomposed into simpler subproblems. Each localized subproblem involves finding a control law to apply to a partition that guides the trajectory to another partition such that Thm. 1 holds. This section presents the methodology used to construct such a controller. The procedure begins by establishing a control law for the case of no disturbance ($\mathbf{f}_D(x,d) = \mathbf{0}$). Then, a technique is presented to find all the states capable of being captured by the opposing agent when $\mathbf{f}_D(x,d) > \mathbf{0}$. Essentially, obstacles are identified as conditions that prevent Thm. 1 from being satisfied.

A. Control with no Disturbance

The first step in constructing the switched feedback controller is to assume that the opposing agent's dynamics are zero, $\mathbf{f}_D(x,d) = \mathbf{0}$. For pursuit-evasion, this implies that the opposing agent is immobile. Then, the goal is to find both a set of partitions, Σ, and a corresponding control vector, $\mathbf{m}$, to apply to Eqn. 3 that guide the state to the target set. In the context of the game, this would be calculated off-line. The algorithm presented below is one method that may be used to find the switched controller, and it is beneficial because it utilizes simple finite horizon controls.

In this procedure, it is assumed that the control assigned to any of the partitions is of the form

$$m_q(x) = \arg\min_{u \in U} v_q(x, u, d).$$

The function being minimized, $v_q(x,u,d)$, is a member of a finite library of controls $\mathcal{V}$ available for the agent. In practice, such a library would consist of functions that maximize the capabilities of the agent. For example choices could include: "close in on a target fast as possible" or "operate with little sound output". The last assumption is that the state space may be reasonably approximated as a discrete grid, $\bar{X}$, so that the following algorithm has a finite number of states to calculate.

Algorithm 1: Constructing the switched objective system proceeds as follows:

1) INITIALIZE $\sigma_0 = \mathcal{T}$, $\hat{X} = \bar{X} \backslash \sigma_0$, $\mathbf{f}_D(x,d) = \mathbf{0}$, and $q = 0$.
2) SET $q = q + 1$.
3) SELECT a function $v_q(x,u,d) \in \mathcal{V}$ to minimize with the control $m_q(x)$ over the domain, $\hat{X}$. If $q = 1$, $v_q(x,u,d) = 0, \forall x \in \sigma_0$ and must be positive definite and increasing with $\|x\|_2$ for $x \notin \sigma_0$.
4) CALCULATE the reachability of the states by integrating the trajectories from the initial states $\hat{X}$ using the operator in Eqn. 1[1].
5) SET $\sigma_q := \left\{ x_0 \in \hat{X} | \exists x^{t^*} \in \bigcup_{p=0}^{q-1} \sigma_p, t_0 < t^* < \infty \right\}$
6) SET $\hat{X} = \bar{X} \backslash \bigcup_{p=0}^{q} \sigma_p$.

[1]There are several methods of integrating the vector field to determine find the state trajectories. For simple planar cases, the authors use a visual inspection.

7) REPEAT from step 2 until $\hat{X} = \emptyset$ or $\sigma_q = \{\emptyset | \; \forall v_q(x,u,d) \in \mathcal{V}\}$.
8) SET $\sigma_q = \hat{X}$ if $\sigma_q = \emptyset$ in step 7.

Upon termination of the algorithm, the set of partitions, Σ, and control vector $\mathbf{m}$ will be realized. By the nature of the algorithm, the partition and control combination will produce the largest capture set possible given the library, $\mathcal{V}$. All of the states that are not grouped into a partition capable of reaching the target set are grouped into σ_q, which is an invariant set.

Proposition 1: The output partition set of Alg. 1, Σ, and control vector, $\mathbf{m}$ result in a digraph with a strictly lower triangular adjacency matrix.

Proof: Suppose a trajectory starting from σ_k, travels to a partition $\sigma_{j \geq k}$. Then from step 5, the state would not be included in σ_k, and is a contradiction. ■

Strictly lower triangular matrices are all nilpotent, and since no trajectory travels into the invariant set, σ_q, the states in all the other partitions are in the capture set for the agent. This initial mapping is a good starting point from which the final control mapping may be refined and calculated on-line.

B. Inclusion of the Opposing Agent Dynamics

Assume a pre-existing partition and control mapping, Σ and $\mathbf{m}$. When the opposing agent dynamics are included such that $\mathbf{f}_D(x,d) > 0$, then one of the following scenarios may occur:

1) *Stagnation*, the opposing agent is able to turn a partition σ_k into an invariant set.
2) *Cyclic Partition Response*, the opposing agent generates a cyclic path on the digraph.
3) *Traversal to an Invariant Partition*, the opposing agent steers the trajectory away from the target.
4) *Aid in Desired Path*, the opposing agent helps reach the target.

Response 4 is trivial in the sense that nothing has to be done on the part of the agent to counteract the opposing control disturbance. Such a scenario would occur if both agents have a similar target and were required to cooperate. Responses 1-3 are a result of the opposing agent seeking to disrupt the state from reaching the target. Response 1 is a result of the opposing agent having just enough strength to keep the trajectory still (though there may be movement within the partition). Two equally matched players at tug-of-war would fall into this category. This is visualized with a digraph by placing a 1 in the diagonal of the adjacency matrix. Response 2 is a result of the opposing agent being able to steer the state such that the strategies of the agents result in a cyclic strategy response. Such a scenario occurs when an evasive maneuver is made and a chase has to renew again. An example of this may be found in [9]. There are a variety of adjacency matrices that could result in this form. Response 3 is when the opposing agent is able to form a new path on the digraph that leads away from the target. This type of situation occurs when the adjacency matrix is not strictly lower triangular, and the paths in the upper triangular region lead to an invariant set. In pursuit-evasion, an evader that is faster than the pursuer would result in this type of response.

In this work, only Response 3 is considered. One possible solution to Responses 1 and 2 may be found in [13]. To guard against the opposing agent traversing another path on the digraph away from the target, all the states from which it is possible to deviate to an undesired partition must be identified. This is rather troublesome, particularly if the strategy of the opposing agent is unknown. The authors, therefore, assume that the disturbance caused by the opposing agent is a worst-case scenario. By enforcing this assumption, any deviation on the part of the opposing agent is only beneficial for the agent receiving the control.

A very useful tool for calculating this set is the Fast Marching Semi-Lagrangian (FMSL) method discussed in [3]. The FMSL solution generates a value function where, under certain assumptions on the running cost, it is possible to identify the states where the opposing agent can deviate from a path.

Proposition 2: Assume an invariant set $\mathcal{D}$, a control law $u(x) = \mathcal{M}(x)$ and the value, $V(x)$, is the solution to the partial differential equation (Dynamic Programming Principle)

$$0 = \min_{d \in D} \left[g(x,u,d) + \langle V_x(x), \dot{x} \rangle \right] \tag{6}$$

subject to the boundary condition

$$V(x) = 0, \; \forall x \in \mathcal{D} \tag{7}$$

and the constraints

$$g(x,u,d) = \begin{cases} 0, \; \forall x \in \mathcal{D} \\ (0,\infty), \; otherwise \end{cases} \tag{8}$$

then, the opposing agent is able to guide the state to the set $\mathcal{D}$ from an initial state $x_0 \in X$ when $V(x) < \infty$ in finite time.

The proof follows from the fact that the running cost, $g(x,u,d)$, is finite and accrues at every point not in $\mathcal{D}$. Since $\mathcal{D}$ is assumed invariant, if the trajectory never enters, then $V(x)$ will tend toward infinity. In practice, $\mathcal{D}$ will be set as all the bordering partitions of σ_k to which the opposing agent would seek to travel.

Once the set of states that lead into $\mathcal{D}$ for a partition has been calculated, then the digraph is updated and the adjacency matrix is analyzed to determine if it satisfies Thm. 1. If the theorem holds, then no further action has to be taken because there is an alternate path to the target set. In the case that the digraph fails to satisfy the theorem, then a new partition, σ_{q+1}, is created and a new control $m_{q+1}(x)$ is applied until $\mathcal{V}$ is exhausted. In general, the process is iterative in that it needs to be conducted for all of the partitions. However, partitions close to the state may be calculated first and those far away (in terms of paths on the digraph) may be ignored temporarily. In situations when the dynamics of the opposing agent are a small disturbance, then the controller is generally very efficient to calculate because the set of states for which the opposing agent may disrupt is small.

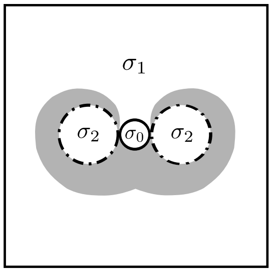

Fig. 1: Initial partitioning scheme of the Homicidal Chauffeur. The shaded region is the assumed region where the evader can traverse to an undesired partition.

VI. Illustrative Example

The classic pursuit-evasion Homicidal Chauffeur game is used to show applications when the switched control scheme is beneficial and when it is not. The premise of the game is that a faster pursuer is trying to capture a slower but more agile evader. In the original formulation, time-optimality, an infinite horizon solution to the HJI equation is used by both agents to determine the optimal actions. Furthermore, the agents were awarded perfect information of each other before the onset of the game.

In this example, the pursuer is to receive the switched control and a restriction will be placed that no information is known about the evader until the onset of the game at time $t = 0^+$. Therefore, it is desired to quickly calculate the control mapping. The parameters of the game will be adjusted to see the impact on the calculation time.

A. Problem Formulation

The agents move on the plane with the following dynamics in reduced form [2]:

$$\dot{x}_1 = -\frac{v_p x_2}{r} u + v_e \sin(d) \tag{9}$$

$$\dot{x}_2 = \frac{v_p x_1}{r} u - v_p + v_e \cos(d) \tag{10}$$

where x_1 and x_2 are the horizontal and vertical relative position of the evader with respect to the pursuer's forward heading. The parameters v_p and v_e are the speeds of the pursuer and evader. The parameter, r, is the pursuer turn radius. The inputs $u \in [-1, 1]$ and $d \in [0, 2\pi)$ are the steering controls provided to the the pursuer and evader, respectively. The bounds on the pursuer control represent the sharpest possible turn to the right or left and the evader is able to choose an instantaneous direction. The game terminates when the evader enters into a ball around the pursuer of radius, l.

$$\mathcal{T} = \{x \in X, \sqrt{x_1^2 + x_2^2} < l\}$$

B. Construction of the Control Law

Following the procedure discussed previously, the evading (disturbance) agent is temporarily assumed to have no dynamic control while an initial control mapping is constructed. Using Algorithm 1, the partitions shown in Fig. 1 are

$$\begin{aligned}
\sigma_0 &= \mathcal{T} \\
\sigma_1 &= X \backslash \{\sigma_0 \cup \sigma_2\} \\
\sigma_2 &= \left\{ x \in X, \sqrt{(x_1 \pm r)^2 + x_2^2} < r - l \right\}
\end{aligned}$$

and assigned the following switched control vector:

$$\mathbf{m}(x) = \begin{bmatrix} \min_{u \in U} \|x^{t+\Delta t}\| \\ \max_{u \in U} \|x^{t+\Delta t}\| \end{bmatrix} \tag{11}$$

The strategy here is to have the pursuer minimize the distance to the evader in σ_1. However, when the evader is located in the turning radius, σ_2, the pursuer must first increase the distance before reducing the separation distance. The adjacency matrix of this scenario is

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Because it is strictly lower triangular, Thm. 1 is satisfied, and in this case $\mathcal{C}(d, \mathcal{M}(x)) = X$.

Having established an initial control before the scenario, the evader is no longer assumed to remain still at $t > 0^+$. Here, the interest is in finding the states where the evader can cause a transition from σ_1 to σ_2. In this case, the evader seeks to make σ_2 an invariant set by always remaining within the turning radius of the pursuer. The calculation proceeds by setting $\mathcal{D} = \sigma_2$ and invoking Proposition 2 using the FMSL method [3]. These states are shown in Fig. 1 by the shaded region around σ_2. When the evader is able to secure the transition, the adjacency matrix becomes,

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

which does not satisfy the criteria of Thm. 1. To remedy this, the shaded region is labeled as a new partition, σ_3, and for this case the control vector is assigned

$$\mathbf{m}(x) = \begin{bmatrix} \min_{u \in U} \|x^{t+\Delta t}\| \\ \max_{u \in U} \|x^{t+\Delta t}\| \\ \max_{u \in U} \|x^{t+\Delta t}\| \end{bmatrix} \tag{12}$$

to prevent entry into σ_2. It can be verified that the adjacency matrix of this new partition-control scheme satisfies Thm. 1.

C. Results

In the following set of trials an array of simulations was conducted to show the effect of the agent parameters on the efficiency of the control calculation. It is well known that the size of the shaded region of Fig. 1 is a function of the speed ratio, $\gamma = \frac{v_e}{v_p}$, and turning radius, r, of the pursuer [2]. In particular, when the inequality,

$$\frac{l}{r} > \sqrt{1 - \gamma^2} + \sin^{-1} \gamma - 1 \tag{13}$$

is satisfied, it is possible for the pursuer to catch the evader. As the parameters approach this line, the size of the set where the evader is able to transition to σ_2 increases. This directly

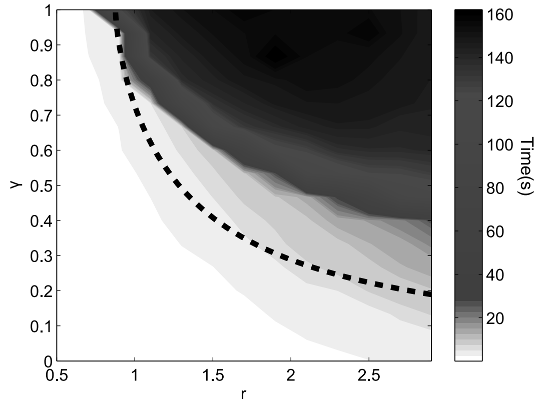

Fig. 2: The effect of calculation time on vehicle parameters for the Homicidal Chauffeur game.

impacts the number of calculations needed to find the control, affecting efficiency.

In the simulations, the speed ratio, γ, was evaluated on the interval $[0, 1]$, and the turning radius, r, was evaluated on the interval $[0.5, 3]$. The parameters of the speed ratio were chosen such that the analytical solution given by Eqn. 13 would remain valid. When $\gamma > 1$, there is no effect in the size of the calculation because the evader may secure an undesired transition from any initial state. The turning radius interval was chosen to give a good array of data from which to compare to the analytical solution. All calculations were performed on a FitPC2 mobile platform with Intel Atom 1.10 GHz processor and 1GB of RAM.

Figure 2 shows the time trial results for a 51×51 grid of nodes ($\bar{X}$) in an approximation of the state space[2]. The dashed line represents Eqn. 13 as an equality. The result shows two regions emerging in the time trials. The darker region corresponds to parameters where it is possible for the evader to escape, and it is possible for the evader to enter into the pursuer's turning radius from all states. All the grid nodes are used in the FMSL calculation, causing a major increase in the calculation time. The lower bound of this region follows the inequality boundary well. On the lower side of the dashed line, the calculation time is greatly reduced. This region corresponds to parameters where the evader is not capable of steering the trajectory into the pursuer turning radius from all initial states. The efficiency of the controller increases in this region because the pursuing agent can establish the correct control with fewer nodes in a small amount of time (usually < 1s). In between the light and dark region is a narrower gray transition region that takes the general trend of the dashed line. The gray color band spans a very large interval on the time color scale from 20% of the bottom increasing to 20% from the top which corresponds to a very sharp change in efficiency for a narrow change in the parameter space.

Several observations may be made regarding using the controller in a time efficient manner. Mainly, the controller performs better when the disturbance has a smaller impact on the trajectory. This is because much of the state space may be removed from the calculation. In comparison, traditional methods use the entire state state space by default. This means the calculation time would be the same for all parameters and equal the time seen in the top right corner of Fig. 2. For the case where the parameters of the agents are close to the narrow gray region, then there is very little warning of a sharp increase in calculation time. However, the time is bounded by the time needed to calculate the control for all states. Furthermore, because the control front propagates outward from the current position, the known agent has the advantage of having the control propagate in front of its desired motion.

[2]Qualitatively, alternate grid sizes will produce similar results relative to the parameter domain.

VII. Discussion and Conclusions

This work provides a methodology by which a control law is partitioned into regions of the state space. The requirement of designing the control law so that an agent may capture the target set from a given partition is readily evaluated using Thm. 1. As a result, several benefits emerge. The partitions and control laws may be constructed using a more intuitive approach. Because the opposing agent's strategy is not known, the assumed worst-case scenario may now be evaluated on the borders of the partitions rather than the entire state space. As illustrated by the Homicidal Chauffeur scenario, cases where the opposing agent disturbance deviates from the worst-case strategy results in faster calculations. This allows for controllers to be constructed during a game for scenarios where the agent is unaware of the opponent until the game starts.

References

[1] M. Caputo, *Foundations of Dynamic Economic Analysis*. Cambridge University Press, 2005.

[2] R. Isaacs, *Differential Games*. New York, NY: John Wiley and Sons, Inc., 1965.

[3] E. Cristiani and M. Falcone, "A fast marching method for pursuit-evasion games," *Communications to SIMAI Congress*, vol. 1, 2006.

[4] M. Falcone, "Numerical methods for differential games based on partial differential equations," *International Game Theory Review*, vol. 8, no. 2, pp. 231–272, 2006.

[5] N. N. Krasovskiĭ, A. I. Subbotin, and S. Kotz, *Game-theoretical control problems*. New York, NY, USA: Springer-Verlag New York, Inc., 1987.

[6] V. Patsko and V. L. Turova, "Numerical study of the homicidal chauffeur game."

[7] R. R. Brooks, J.-E. Pang, and C. Griffin, "Game and information theory analysis of electronic countermeasures in pursuit-evasion games," *IEEE Trans. Syst., Man, Cybern. A*, vol. 38, no. 6, pp. 1281–1293, Nov. 2008.

[8] J. Hespanha, M. Prandini, and S. Sastry, "Probabilistic pursuit-evasion games: A one-step nash approach," in *Proc. of 39th IEEE Conf. on Decision and Control*, Sep. 2000, pp. 2272–2277.

[9] B. Goode, A. Kurdila, and M. Roan, "Pursuit-evasion with acoustic sensing using one step nash equilibria," in *Proc. American Control Conference, 2010*.

[10] D. Fudenberg and D. K. Levine, *The Theory of Learning in Games*, ser. MIT Press Books. MIT Press, December 1998, vol. 1.

[11] B. Andrásfai, *Graph Theory: flows, matrices*. Akadémiai Kiadó, 1991.

[12] V. V. Prasolov and S. Ivanov, *Problems and theorems in linear algebra*. American Mathematical Society, 1994, vol. 134.

[13] B. Goode, A. Kurdila, and M. Roan, "Adaptive fuzzy control of switched objective functions in pursuit-evasion scenarios," in *IEEE Conference on Decision and Control, 2010*, to be published.