Automatic Generation of Communication Requirements for Enforcing Multi-Agent Safety*

Eric S. Kim Murat Arcak Sanjit A Seshia
Department of Electrical Engineering and Computer Sciences
UC Berkeley
Berkeley, CA
{eskim, arcak, sseshia}@eecs.berkeley.edu

BaekGyu Kim Shinichi Shiraishi
Toyota InfoTechnology Center, U.S.A.
Mountain View, CA
{bkim, sshiraishi}@us.toyota-itc.com

Distributed controllers are often necessary for a multi-agent system to satisfy safety properties such as collision avoidance. Communication and coordination are key requirements in the implementation of a distributed control protocol, but maintaining an all-to-all communication topology is unreasonable and not always necessary. Given a safety objective and a controller implementation, we consider the problem of identifying *when* agents need to communicate with one another and coordinate their actions to satisfy the safety constraint. We define a coordination-free controllable predecessor operator that is used to derive a subset of the state space that allows agents to act independently, without consulting other agents to double check that the action is safe. Applications are shown for identifying an upper bound on connection delays and a self-triggered coordination scheme. Examples are provided which showcase the potential for designers to visually interpret a system's ability to tolerate delays when initializing a network connection.

1 Introduction

Interaction amongst agents can come in various forms such as coupled dynamics, coupling constraints, or a joint optimization objective. A common facet of multi-agent systems is the use of a distributed control architecture, where each agent has authority over different sets of actuators, and an accompanying communication network for agents to coordinate their actions. Communication and collective decision making facilitate complex interactions amongst agents and enable them to reliably achieve collective behaviors that would otherwise be difficult to accomplish without some coordination protocol.

In this paper, we consider the problem of satisfying a safety objective with a controller that is distributed over multiple agents. We say that these agents are coordinating within a given time step if they communicate and collectively agree upon actions to execute. As a motivating example, consider two fully autonomous vehicles equipped with vehicle-to-vehicle (V2V) communication and tasked with avoiding a collision. At one extreme are scenarios where no communication is necessary due to a sufficiently large distance between the vehicles, while at the other extreme are near miss scenarios where collisions are only avoided through precise timing, actuation, or luck. Preemptive cooperation enabled by V2V communication is designed to help the vehicles avoid these danger scenarios and for vehicles to negotiate collision-free trajectories.

How can one distinguish between these extremes and determine when multi-system coordination is and is not necessary to maintain a safety objective? We present a method that takes a closed loop control system and a safety requirement, then identifies a subset of the state space that is robustly safe against temporary communication losses. This subset naturally shrinks with time as the duration of the communication loss increases. At its core, our method iterates an appropriate operator which propagates

*This work was supported in part by NSF grant CNS-1545116, co-funded by the DOT.

M. Gleirscher, S. Kugele, S. Linker (Eds.): 2nd International Workshop on Safe Control of Autonomous Vehicles (SCAV 2018)
EPTCS 269, 2018, pp. 3–16, doi:10.4204/EPTCS.269.2

a coordination-free region and resembles fixed point algorithms in the literature on symbolic system verification. This operator is defined such that it incorporates information about the system dynamics and the controller architecture. These results are first used to consider a scenario when multiple agents want to cooperate, but can only do so after some delay. We then develop a self-triggered coordination scheme where agents can preemptively schedule when they would like to communicate, while still maintaining safety guarantees.

This paper tackles a new problem that has not, to the best of our knowledge, been addressed within the control theory literature and is motivated by applications to autonomous vehicle safety. Compared to other work, we do not assume a decomposition of the state space as in [5][4] nor is the objective assumed to be decomposable [4]. Instead we only consider a decomposition of the input space and can thus accommodate instances when there are complex coupling dynamics that are best handled monolithically. This work leverages compositional tools and techniques developed for formal controller synthesis. These may involve constructing abstractions compositionally [13], decomposing the controller synthesis procedure [9][10], or decomposing the controller itself [15]. Assume-guarantee reasoning has also been used for compositional synthesis with multiple agents by abstracting out internal information that is irrelevant to reason about system interactions [11]. Our self-triggering communication scheme may be compared to similar schemes in the self-triggered control literature [8], where often the objective is to minimize the energy expended by sensors and actuators subjected to a stability constraint [2][6]. Our work instead seeks to minimize the communication overhead incurred as multiple agents negotiate safe actions.

2 Formulation

2.1 Notation

Given two sets $\mathscr{A}$ and $\mathscr{B}$, let $|\mathscr{A}|$, $2^{\mathscr{A}}$, and $\mathscr{A} \times \mathscr{B}$ respectively represent $\mathscr{A}$'s cardinality, $\mathscr{A}$'s power set (set of all subsets), and the Cartesian product between $\mathscr{A}$ and $\mathscr{B}$. Let $\mathbb{R}$, $\mathbb{Z}$ represent the real and integer numbers respectively, while $\mathbb{R}_{\geq 0}$ and $\mathbb{Z}_{\geq 0} = \mathbb{N}$ are their non-negative counterparts. With an appropriate universal set Ω, $\mathscr{A}$'s complement $\mathscr{A}^C$ is defined as $\Omega \setminus \mathscr{A}$. Given a Cartesian product of M sets $\prod_{i=1}^{M} \mathscr{A}_i$ and a subset $L \subseteq \prod_{i=1}^{M} \mathscr{A}_i$, the projection operation $\pi_{\mathscr{A}_j} : \prod_{i=1}^{M} \mathscr{A}_i \to \mathscr{A}_j$ retains the coordinates associated with $\mathscr{A}_j$ and is defined as:

$$\pi_{\mathscr{A}_j}(L) = \{a_j \in \mathscr{A}_j : \exists (a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_M) \text{ such that } (a_1, \ldots, a_M) \in L\}. \tag{1}$$

2.2 Signals and Systems

An interval $[a,b]$ where $a,b \in \mathbb{Z}$ includes both end points. Let $[a,b) = [a,b-1]$ and $[a] = [a,a]$. Given a space $\mathscr{P}$, the space of trajectories evolving in $\mathscr{P}$ is $\mathscr{P}[\cdot]$. A trajectory $p[\cdot]$ over time interval I is a map $p[\cdot] : I \to \mathscr{P}$. Let $\mathscr{X}$ and $\mathscr{U}$ represent a system's state and input spaces respectively. Sets $\mathscr{X}[\cdot]$ and $\mathscr{U}[\cdot]$ are referred to as state and input trajectory sets. This paper deals with systems where the input space $\mathscr{U}$ consists of N components so that $\mathscr{U} = \prod_{i=1}^{N} \mathscr{U}_i$ [1]. Each of these N components is thought of as an individual agent. The system's discrete-time dynamics are given by a relation $f \subseteq \mathscr{X} \times \mathscr{U} \times \mathscr{X}$, which can also be viewed as a set-valued function $f : \mathscr{X} \times \mathscr{U} \to 2^{\mathscr{X}}$. Let $\mathscr{U}(x) = \{u \in \mathscr{U} : f(x,u) \neq \emptyset\}$ denote the set of non-blocking control inputs at x.

A memoryless controller for system f is a relation $C \subseteq \mathscr{X} \times \mathscr{U}$. The set of states $\mathscr{B} = \{x \in \mathscr{X} : (x,u) \notin C \text{ for all } u \in \mathscr{U}\}$ is the set of blocking states under controller C. A controller may also be viewed

[1] Some $\mathscr{U}_i$ may be multi-dimensional so N is not necessarily the dimension of $\mathscr{U}$.

as a function $C : \mathscr{X} \to 2^{\mathscr{U}}$ that maps states to sets of admissible inputs (states with no corresponding control input map to an empty set). A controller C and system f can be interconnected into a closed loop system denoted as $f \circ C : \mathscr{X} \to 2^{\mathscr{X}}$ [2]. The next state $x[k+1]$ satisfies $x[k+1] \in f \circ C(x[k]))$ if and only if there exists a $u[k] \in C(x[k])$ such that $x[k+1] \in f(x[k], u[k])$. All sequences $x[\cdot]$ that satisfy the aforementioned condition and $x[0] \in \mathscr{L}$ are said to be generated by the closed loop system $f \circ C$ with initial state set $\mathscr{L} \subseteq \mathscr{X}$.

2.3 Control for Safety

Safety is a common requirement for cyber-physical systems. We encapsulate this notion of safety as a region of the state space $\mathscr{S} \subseteq \mathscr{X}$ that should never be exited. For a vehicle, set $\mathscr{S}$ could represent a collision-free zone and a speed limit, while for a medical device $\mathscr{S}$ could represent safe blood sugar levels.

Definition 1. *Let $\mathscr{S} \subseteq \mathscr{X}$ be a set of safe states. A control policy $C : \mathscr{X} \to 2^{\mathscr{U}}$ and initial set $\mathscr{L} \subseteq \mathscr{S}$ is said to satisfy safety constraint $\mathscr{S}$ if all trajectories generated by a closed loop system $f \circ C$ with any initial state $x[0] \in \mathscr{L}$ never exit $\mathscr{S}$.*

At each state x, there is a set of admissible control inputs $C(x) \subseteq \mathscr{U}$. A controller is deterministic if $|C(x)| = 1$ only permits one action for all $x \in \mathscr{X}$. Although determinism simplifies analysis of a closed loop system, deterministic controllers may be too restrictive if the system needs to satisfy additional requirements on top of safety. For instance if two vehicles want to avoid a collision, then a safe controller can simply enforce that both vehicles have zero velocity but this prevents vehicles from reaching a desired location.

2.4 Loss of Safety Guarantees with a Distributed Controller

More permissive controllers can act as supervisors that restrict control actions only enough to ensure safety. They are useful because they can be combined with other controllers that seek to achieve other objectives such as reaching a region. When a distributed controller is deployed on multiple systems without an underlying communication scheme, the non-determinism contained in permissive controllers can lead to safety violations.

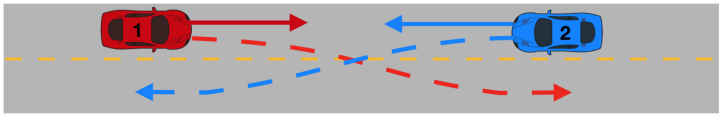
If $\mathscr{U} = \prod_{i=1}^{N} \mathscr{U}_i$ is decomposed into N inputs that are each under control from a different agent, then each must concurrently select a single input u_i such that

$$(u_1, \ldots, u_N) \in C(x). \tag{2}$$

It is this step where multiple agents concurrently select an input that leads to coordination hazards. Whenever $|C(x)| > 1$ then assuring that (2) holds is not always possible without explicit coordination and communication with other agents.

Example 1 (Illustrative Example)**.** *Consider a scenario depicted in Figure 1 where two vehicles are facing one another and a collision is imminent. Both vehicles can choose between staying in their lane or switching to the other lane and a collision is avoided only when one vehicle switches. Clearly it is possible for a collision to be avoided as long as the two vehicles are able to communicate and negotiate which one changes lanes. On the other hand suppose that these vehicles are not equipped with V2V*

[2] This notation was inspired by $\circ$'s usage as a function composition operator. However, it is not a composition in the strictest sense where $f(g(x)) = (f \circ g)(x)$.



	Right Vehicle Change	Right Vehicle Stay
Left Vehicle Change	Collision	No Collision
Left Vehicle Stay	No Collision	Collision

Figure 1: Motivating Example

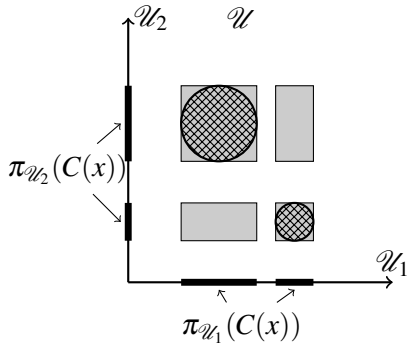


Figure 2: For some fixed $x \in \mathscr{X}$, the original safe control set $C(x)$ (patterned region) is projected onto the axes and yields $\pi_{\mathscr{U}_1}(C(x))$ and $\pi_{\mathscr{U}_2}(C(x))$ (thick lines). Combining the projections gives the coordination-free counterpart $\mathsf{IND}_C(x)$ (darker regions) defined in Section 3.

communications. If a collision does occur it is not possible to assign fault to solely one vehicle because from both vehicles' points of view its action was safe as long as the other vehicle responded with the appropriate action. Instead one can only attribute the fault to both agents' failure to negotiate.

To formalize the notion of coordination, we first define a minimal independent controller IND_C associated with C. The set of possible controller actions at x is $\mathsf{IND}_C(x)$ and depicted in Figure 2.

$$\mathsf{IND}_C(x) := \prod_{i=1}^{N} \pi_{\mathscr{U}_i} C(x). \tag{3}$$

The projection $\pi_{\mathscr{U}_i} C(x)$ of this controller onto each agent i's individual component $\mathscr{U}_i$ yields the set of all control inputs permitted at state x without any information about how other agents behave. Any input $u_i \notin \pi_{\mathscr{U}_i} C(x)$ indicates that agent i is either reckless or malicious. If all agents pick a $u_i \in \pi_{\mathscr{U}_i} C(x)$ then they have all reasonably attempted to satisfy the safety condition by selecting a point $(u_1, \ldots, u_N) \in \mathsf{IND}_C(x)$, but the joint condition $(u_1, \ldots, u_N) \in C(x)$ is not necessarily satisfied because $C(x) \subseteq \mathsf{IND}_C(x)$. The independent controller IND_C may also be viewed as the set of possible control actions that are reasonable in the undesirable situation where each agent believes itself to be the leader and relies on the other agents to be followers that respond to the leader's choice. The set $\mathsf{IND}_C(x) \subseteq \mathscr{U}$ is the minimal independent set that contains $C(x)$.

Throughout the rest of this paper, we analyze properties of the new closed loop system $f \circ \mathsf{IND}_C$, which is derived from $f \circ C$ but exhibits additional behaviors due to the absence of coordination.

Note that the set of trajectories that are exhibited under $f \circ C$ is a subset of those exhibited under $f \circ \mathsf{IND}_C$. Thus, even though the original system $f \circ C$ may be safe, $f \circ \mathsf{IND}_C$ may exhibit unsafe trajectories.

Problem 1. *Given a set of dynamics f, a distributed controller IND_C, a safe region $\mathscr{S}$, and coordination-free interval $I = [a,b)$ identify a subset of the state space $\mathscr{L}$ such that all behaviors of $f \circ \mathsf{IND}_C$ with initial state $x[a] \in \mathscr{L}$ remain in $\mathscr{S}$ within the interval I.*

2.5 Remarks on Coordination with Mesh Networks

V2V technology also enables the creation of ad hoc vehicular mesh networks which enables applications in cooperative cruise control, vehicular platoons, and congestion mitigation. Suppose each agent is represented by a vertex in an undirected graph and two agents with a V2V have their corresponding vertices connected by an edge. Such a graph can be grouped into equivalence classes corresponding to its connected components. We assume that agents in the same class can communicate instantly even if they are separated by more than one edge.

Assumption 1. *Each agent in an equivalence class can coordinate with all other agents in that class within each time step k.*

In practice, Assumption 1 is a requirement that the time scale over which messages is passed in the network are effectively instantaneous relative to the time scale of the physical dynamics. The independence definition of Equation (3) was stated under the assumption that each $\mathscr{U}_i$ corresponded to one agent and that no agents cooperate. If agent cooperation occurs over a mesh network with P connected components, then the independence condition corresponds to the connected components of the graph. For each of $l = 1,\ldots,P$ equivalence classes, let $\hat{\mathscr{U}}_l$ be the Cartesian product of the coordinates $\mathscr{U}_i$ that belong to that class.

$$\mathsf{IND}_C(x) := \prod_{l=1}^{P} \pi_{\hat{\mathscr{U}}_l} C(x). \tag{4}$$

This formulation allows for a platoon to be treated as a single agent instead of a collection of vehicles. For notational simplicity, we simply assume that the decomposition into equivalence classes is given and use Equation (3) throughout the rest of this paper.

3 Coordination-Free Operator

Given some controller $C \subseteq \mathscr{X} \times \mathscr{U}$, we use the associated minimally restrictive independent controller from Equation (3) as a formal characterization of all the possible actions with a distributed implementation of C in the absence of coordination.

The set of predecessor states which enforce membership within a region $Z \subseteq \mathscr{X}$ without coordination is computed with the operator

$$\mathsf{IPRE}(Z) = \{x : x \in \pi_{\mathscr{X}}(\mathsf{IND}_C)\} \cap \{x : \emptyset \neq f(x,u) \subseteq Z \text{ for all } u \in \mathsf{IND}_C(x)\}. \tag{5}$$

The first set ensures that there is always a valid input because $\pi_{\mathscr{X}}(\mathsf{IND}_C)$ is a state domain over which the controller produces admissible inputs. The second set takes into account the system dynamics and ensures that all states are in Z. A state in $\mathsf{IPRE}(Z)$ is robust in the sense that all future possible next states $f(x,u)$ are contained in Z *despite* uncertainty about which $u \in \mathsf{IND}_C(x)$ is chosen.

Operator $\mathsf{SIPRE}_{\mathscr{S}}$ below identifies states that can stay in Z and remain safely in $\mathscr{S}$ without coordination

$$\mathsf{SIPRE}_{\mathscr{S}}(Z) = Z \cap \mathsf{IPRE}(Z) \cap \mathscr{S}. \tag{6}$$

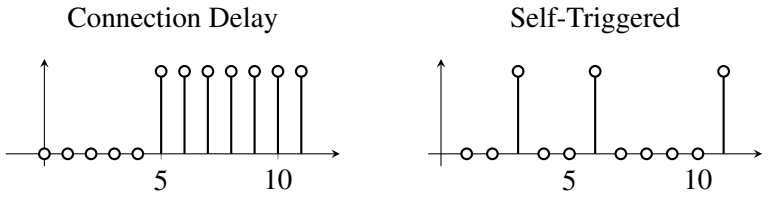


Figure 3: Two scenarios with intermittent connections. A high value signifies an established connection.

By iterating this operator k times, we can identify a region of the state space that remains in $\mathscr{S}$ for k time steps despite communication losses. Both operators are simple modifications on standard controllable predecessor operators [16].

3.1 Remarks about Algorithmic Implementation

Set intersection, union, negation, and projection are the main operations that are required to compute Equation (5) and Equation (6) exactly. In a continuous domain, support for these algebraic operations may only be possible to encode for a specific set of system dynamics and constraints (consider for instance linear system dynamics and constraints given as unions of polyhedra). However in the scenario where state and inputs spaces are finite, binary decision diagrams (BDDs)[3] are an efficient data structure that supports all of the aforementioned operations. Instead of imposing constraints on the system dynamics and safety region, we opt for the finite case by using a grid to approximate a continuous domain. Moreover, there exists a rich theoretical literature of abstraction methods [16] [12] and accompanying software tools such as [14] which construct approximately similar finite systems such that Assumption 2 is satisfied, even if the state and input spaces of system f are dense, continuous subsets of Euclidean space.

Assumption 2. *Both $\mathscr{X}$ and $\mathscr{U}$ are finite sets.*

4 Applications

We consider two applications. One is to characterize latency requirements for a wireless communication system and the other is a design for a self-triggered coordination scheme.

4.1 Maximum Allowed Connection Delay

Our first application involves N agents that seek to establish a wireless communication channel subject to a maximum connection delay $D \in \mathbb{N}$. Once a connection is established, it is assumed to be maintained as in the left of Figure 3 where $D = 5$. If all agents attempt to initiate a connection starting at time k, then they are able to jointly choose a control input starting at time $k+D$.

Definition 2. *A system in state $x[k]$ at time k is robustly safe to connection initialization delays of length D if $x[k,\infty) \in \mathscr{S}$ for all trajectories $x[k,\infty)$ generated by the time varying closed loop system*

$$x[k+1] \in f \circ \mathsf{IND}_C(x[k]) \text{ if } k \in [k, k+D) \tag{7}$$
$$x[k+1] \in f \circ C(x[k]) \text{ if } k \in [k+D, \infty) \tag{8}$$

where we adopt the convention $[k,k+D) = \emptyset$ if $D = 0$.

The approach to generating the set of states that are robust to connection initialization delays of length D is as follows. We first identify an invariance set $\mathscr{K}$ where the system $f \circ C$ remains in $\mathscr{S}$ along an infinite horizon $[k+D,\infty)$ once $x[k+D] \in \mathscr{K}$. Invariance set $\mathscr{K}$ is distinct from safe set $\mathscr{S}$ because a state $x[k] \in \mathscr{S} \setminus \mathscr{K}$ satisfies the safety condition at time k but is not guaranteed to do so along an infinite horizon. With set $\mathscr{K}$, we then iterate $\mathsf{SIPRE}_{\mathscr{S}}(\mathscr{K})$ D times to identify the states that are guaranteed to reach $\mathscr{K}$ at time $k+D$ without exiting $\mathscr{S}$ within $[k,k+D)$.

To identify $\mathscr{K}$, we define operators that are analogous to IPRE and SIPRE, except that IND_C is replaced with C

$$\mathsf{PRE}(Z) = \{x : x \in \pi_{\mathscr{X}}(C)\} \cap \{x : \emptyset \neq f(x,u) \subseteq Z \text{ for all } u \in C(x)\} \tag{9}$$

$$\mathsf{SPRE}_{\mathscr{S}}(Z) = Z \cap \mathsf{PRE}(Z) \cap \mathscr{S} \tag{10}$$

Lemma 1. *Let $\mathscr{K} := \lim_{i\to\infty} \mathsf{SPRE}^i_{\mathscr{S}}(\mathscr{X})$. Then all trajectories $x[k+D,\infty)$ such that $x[k+D] \in \mathscr{K}$ will never intersect the unsafe set $\mathscr{S}^C$.*

Proof. The Tarski fixed point theorem [17] ensures that the limit on the right hand side exists and is unique if $\mathscr{X}$ is a finite set and $\mathsf{SPRE}_{\mathscr{S}}$ is a monotone operator. Assumption 2 ensures that $\mathscr{X}$ is finite, and monotonicity of $\mathsf{SPRE}_{\mathscr{S}}$ with respect to the set containment ordering can easily be verified. Note that $\mathscr{S} = \mathsf{SPRE}^1_{\mathscr{S}}(\mathscr{X})$. Membership of state $x[k]$ in set $\mathsf{SPRE}\mathscr{S}^{i+1}(\mathscr{X})$ ensures that both $x[k], x[k+1] \in \mathscr{K}$. By induction, given $x[k+D] \in \mathsf{SPRE}^i_{\mathscr{S}}(\mathscr{X})$ and $i > 0$, trajectories from system $f \circ C$ will remain in $\mathscr{S}$ along the interval $[k+D, k+D+i)$. Because the limit set exists, $\lim_{i\to\infty} \mathsf{SPRE}^i_{\mathscr{S}}(\mathscr{X})$ is the set of points that are safe along the interval $[k+D,\infty)$. □

Building on the previous lemma, iterating SIPRE D times yields a region where all trajectories of length D are safe without coordination. The closed loop system under IND_C must never exit $\mathscr{S}$ within the interval $[k,k+D)$, and also must terminate at $x[k+D] \in \mathscr{K}$ so that the system under C can ensure safety along the infinite horizon $[k+D,\infty)$.

Proposition 1. *Let $\mathscr{K} := \lim_{i\to\infty} \mathsf{SPRE}^i_{\mathscr{S}}(\mathscr{K})$. Then $\mathsf{SIPRE}^k_{\mathscr{S}}(\mathscr{K})$ is the set of states that are safe under IND_C for $k-1$ time steps.*

Proof. Suppose $x[0] \in \mathsf{SIPRE}^k_{\mathscr{K}}(\mathscr{K})$. The set of possible states for $x[1]$ under controller IND_C is uniquely defined as $\mathsf{SIPRE}^{k-1}_{\mathscr{K}}(\mathscr{K})$ and is non-empty. By induction, a sequence $x[\cdot] = x[0]\ldots x[k]$ generated by closed loop system $f \circ \mathsf{IND}_C$ must satisfy $x[j] \in \mathsf{SIPRE}^{k-j}_{\mathscr{K}}(\mathscr{K})$ for all $j \in [0,k]$. By definition $\mathsf{SIPRE}^0_{\mathscr{K}}(\mathscr{K}) = \mathscr{K}$. □

4.2 Self-triggered coordination

It is also possible to design a scheduler for triggering communication amongst agents. Each agent maintains a countdown for the latest time communications can be initiated. As the system executes, this time is updated to provide a constantly changing upper bound on the latest time the agents need to communicate. For clarity, we assume that the connection initialization delay as in the previous section is $D = 0$.

The fixed point computation in Proposition 1 yields a sequence of disjoint sets. Define $T : [0,F] \to 2^{\mathscr{X}}$ such that

$$T(k) = \begin{cases} \mathsf{SIPRE}^k_{\mathscr{S}}(\mathscr{K}) \setminus \mathsf{SIPRE}^{k+1}_{\mathscr{S}}(\mathscr{K}) & \text{if } k < F \\ \mathsf{SIPRE}^k_{\mathscr{S}}(\mathscr{K}) & \text{if } k = F \end{cases} \tag{11}$$

where $F \in \mathbb{N}$ is the first value where the sequence reaches a fixed point

$$F = \text{argmin}_{i \in \mathbb{N}_{\geq 0}} \mathsf{SIPRE}^{i+1}_{\mathscr{S}}(\mathscr{K}) = \mathsf{SIPRE}^{i}_{\mathscr{S}}(\mathscr{K}). \tag{12}$$

A modified inverse function $\hat{T}^{-1} : \mathscr{X} \to [0, F]$ is given by:

$$\hat{T}^{-1}(x) = \{i \in [1, F] : x \in T(i)\}. \tag{13}$$

Because the collection $T(1), \ldots, T(F)$ consists of disjoint sets, $\hat{T}^{-1}(x)$ is well defined (i.e. a singleton set) for each $x \in \mathscr{K}$. Because each agent has access to $\hat{T}$ and the state x, they can independently determine the unique value for i such that $x \in T(i)$. A countdown with initial value i is then initialized for each agent. When that value reaches $i = 0$ then the agents coordinate by selecting an action and also initialize a new countdown timer. This framework exhibits reduced communication overhead compared to a centralized architecture, while also preserving the guarantees that are otherwise impossible with a fully decentralized and coordination free controller architecture.

The self-triggered system is defined by augmenting the original system with a countdown that resets after coordination has been triggered.

Definition 3. *The system with a self-triggered communication architecture satisfies the following dynamics.*

$$x[k+1] = \begin{cases} f \circ \mathsf{IND}_C(x[k]) & \text{if } i[k] > 0 \\ f \circ C(x[k]) & \text{if } i[k] = 0 \end{cases} \tag{14}$$

$$i[k+1] = \begin{cases} i[k] - 1 & \text{if } i[k] > 0 \\ \hat{T}^{-1}(x[k+1]) & \text{if } i[k] = 0 \end{cases} \tag{15}$$

Note that when $i[k] = 0$, the counter is reset to $\hat{T}^{-1}(x[k+1])$ after the state transition from Equation (14) *occurs.*

Proposition 2. *If $x[k] \in \mathscr{K}$, then all trajectories $x[k, \infty)$ under the self-triggered communication system from Definition 3 will remain inside $\mathscr{S}$.*

5 Examples

In each of our examples, we use a modified version of the SCOTS symbolic controller synthesis toolbox [14], which takes a continuous control system and creates a finite state machine that serves as an abstract representation over which a controller is synthesized. In addition to modifications to compute Equation (4) and Equation (6), we exploit internal system dependencies to reduce the computation time of the abstraction [7]. Creating the discrete abstraction depends on parameters such as the grid size and granularity. Consider a set $\mathscr{P} = \prod_{i=1}^{N} \mathscr{P}_i$ and a discretization parameter $\eta \in \mathbb{R}^N_{>0}$. Its corresponding discretization grid is $[\mathscr{P}]_\eta := \prod_{i=1}^{N} [\mathscr{P}_i]_{\eta_i}$ where $[\mathscr{P}_i]_{\eta_i} := \{a \in \mathscr{P}_i : a = k\eta_i \text{ with } k \in \mathbb{Z}\}$ is a grid over a single dimension. A full introduction to the underlying theory appears in [16] and is beyond the scope of this paper.

5.1 Invariance in a Circle

Two agents each have control over different axes and both need to remain within a circular region.

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \end{aligned} \tag{16}$$

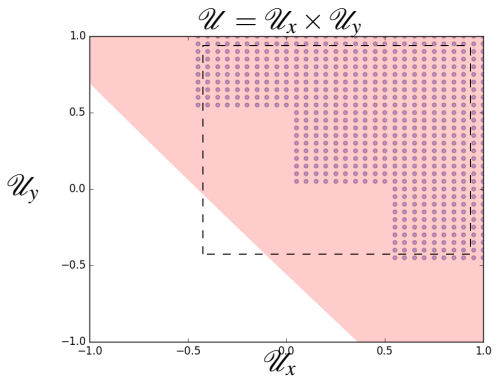


Figure 4: Individual dots represent the synthesized safe control set from SCOTS under $C(x)$ at point $x = (x_1, x_2) = (-.62, -.5)$. Without discretization, the true safe action space would be the shaded region in red. The dashed box shows the possible coordination-free actions $\mathsf{IND}_C(x)$, which is not contained in the safe action space. Importantly, the synthesized safe inputs are a subset of the true set. Note that $||x||_2 \approx .796$, which is near the boundary of $\mathscr{S}$.

Let $\mathscr{X} = \mathscr{U} = [-1,1] \times [-1,1]$. Although the dynamics are independent, the safety region is a circle with a radius 0.8 so $\mathscr{S} = \{(x_1, x_2) : x_1^2 + x_2^2 \leq .64\}$ so both agents must coordinate with one another to avoid exiting $\mathscr{S}$ near the boundary. It is clear that the system can always enforce safety within $\mathscr{S}$ simply by picking a control input $(u_1, u_2) := -(x_1, x_2)$.

A discretization of the system dynamics is constructed with a sampling period of $t = .01$. The state space grid $[\mathscr{X}]_\eta$ is constructed with $\eta = [.01, .01]$ and input space grid is $[\mathscr{U}]_\varepsilon$ with $\varepsilon = [.05, .05]$. Figure 4 depicts all safe control inputs at $(x_1, x_2) = (-.62, .5)$ which is near the boundary of $\mathscr{S}$. The staircase shape of the boundary between the safe and unsafe inputs is due to the discretization of the dynamics. Inputs towards the upper right move the state to the interior of $\mathscr{S}$, while safe inputs at the lower left hug the boundary between $\mathscr{S}$ and $\mathscr{S}^C$. If both systems jointly pick low values for u_1 and u_2 then a violation occurs, however both agents can pick $u_1, u_2 = -1$ if the other agent concedes and chooses a higher value.

Figure 5 depicts the propagation at various time steps of the coordination-free region via the SIPRE operator in Section 3. Figure 5 shows that a system beginning at the origin can experience an uncoordinated collision is possible after 29 discrete time steps which under sampling period $t = .29$ corresponds to an interval of length .29 in continuous time. However for the continuous system the worst case time step is roughly twice as much $.8/\sqrt{2} \approx .565$, which is the case when $u_1, u_2 \in \{-1, 1\}$ and maintain constant values over time. This is mainly due to the discretization errors that arise when abstracting the continuous system to a discrete one. Note that the discretization error does not jeopardize the safety guarantee. Rather, the discrete case underestimates how much time is available for agents to avoid communication, thus providing a more conservative guarantee.

5.2 Intersection Collision Avoidance

Consider two vehicles that are approaching an intersection with no stop sign or a traffic signal. They are controlled independently but each are equipped with V2V radios and may communicate with one another. They also are equipped with enough sensors to identify the position and velocity of all vehicles

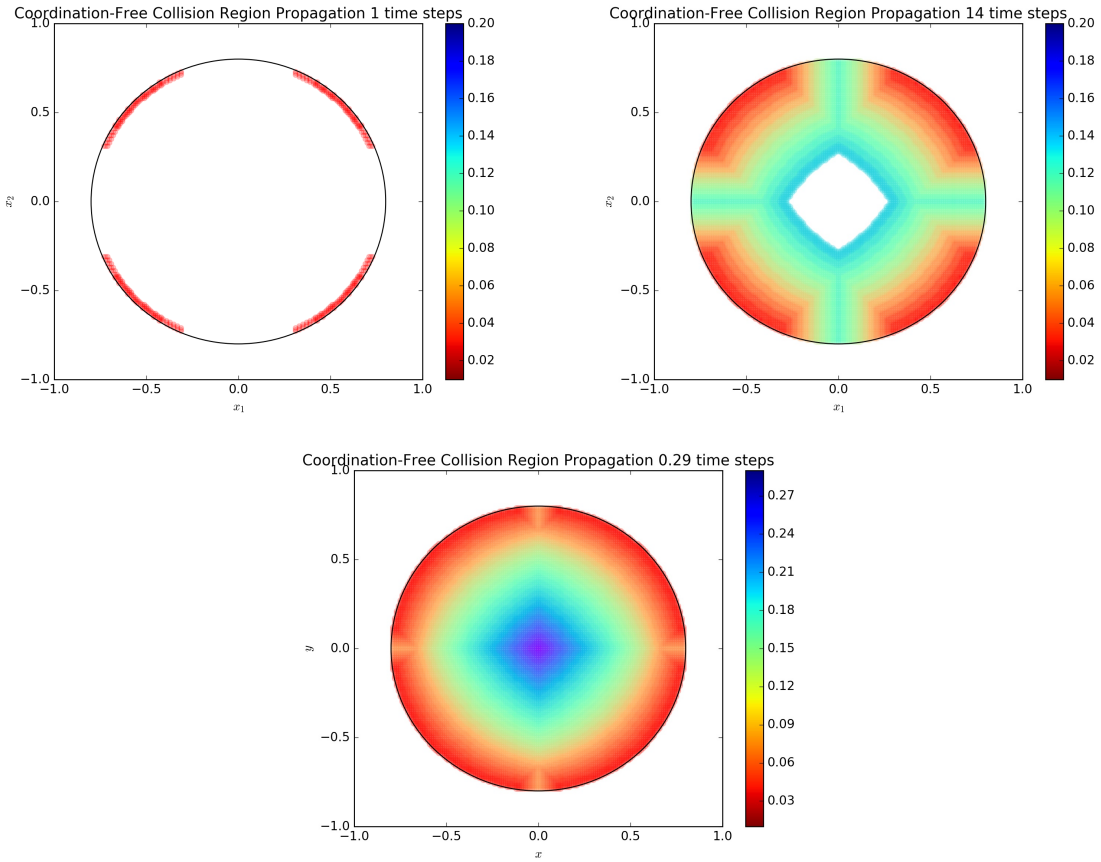


Figure 5: Multiple snapshots at $i = 1, 14, 29$ as the region $\mathscr{K} \setminus \mathsf{SPRE}^i_{\mathscr{S}}(\mathscr{K})$ grows. One can alternatively visualize $\mathsf{SPRE}^i_{\mathscr{S}}(\mathscr{K})$ as a shrinking interior white region as the length of the communication-free interval grows. Red regions represent areas where the system will imminently exit $\mathscr{S}$ unless the two agents coordinate their actions, while blue regions in the interior are only unsafe if the agents do not coordinate for a prolonged period. A fixed point was reached at $i = 29$.

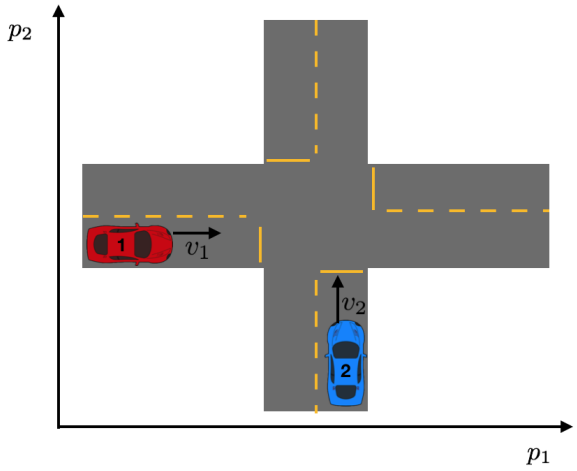


Figure 6: Intersection Collision Avoidance

near the intersection. We consider a simple set of system dynamics given by

$$\dot{p}_i = v_i \tag{17}$$
$$\dot{v}_i = u_i - Kv^2 \tag{18}$$

with some constant $K = .2$. A higher value for k signifies higher air drag. Let $\mathscr{P}_1, \mathscr{P}_2 = [-10, 10]$ and $\mathscr{V}_1, \mathscr{V}_2 = [0, 3]$. The state space is $\mathscr{X} := \prod_{i=1}^{2} (\mathscr{P}_i \times \mathscr{V}_i)$ and $\mathscr{U} := \prod_{i=1}^{2} [-1, 1]$. The invariant region is the region where at least one vehicle is outside the intersection and no collision has occurred and is succinctly encoded as the set

$$\mathscr{S} := \{x : (|p_1| \geq 2) \vee (|p_2| \geq 2)\}. \tag{19}$$

We use the SCOTS toolbox to synthesize a supervisory controller C and compute its corresponding invariance region $\mathscr{K}$ with the procedure in Section 4.1. The system dynamics discretization used a sampling period of $t = .2$, state space grid $[\mathscr{X}]_\eta$ parameter $\eta = [.1, .1, .1, .1]$ and input space grid $[\mathscr{U}]_\varepsilon$ parameter $\varepsilon = [.1, .1]$.

After synthesizing controller C, its decomposed counterpart IND_C is analyzed. Within $\mathscr{K}^C$ even a centralized controller is unable to guarantee that a collision will *not* occur. This unsafe region is to be avoided and communication is necessary to avoid it. Section 5.2 depicts the 3D projection of $\mathscr{K}^C$ and the evolution of the unsafe region $(\mathsf{SIPRE}^D_{\mathscr{S}}(\mathscr{K}))^C$ with no communication.

5.3 Self-Triggered Coordination in a 2D Gridworld

Let there be $N = 2$ agents navigating a 2D grid. Both agents have identical dynamics to Equation (20) as shown below with superscripts $i = 1, 2$ as indexes for each agent.

$$\begin{aligned} \dot{x}_1^i &= u_1^i \\ \dot{x}_2^i &= u_2^i \end{aligned} \tag{20}$$

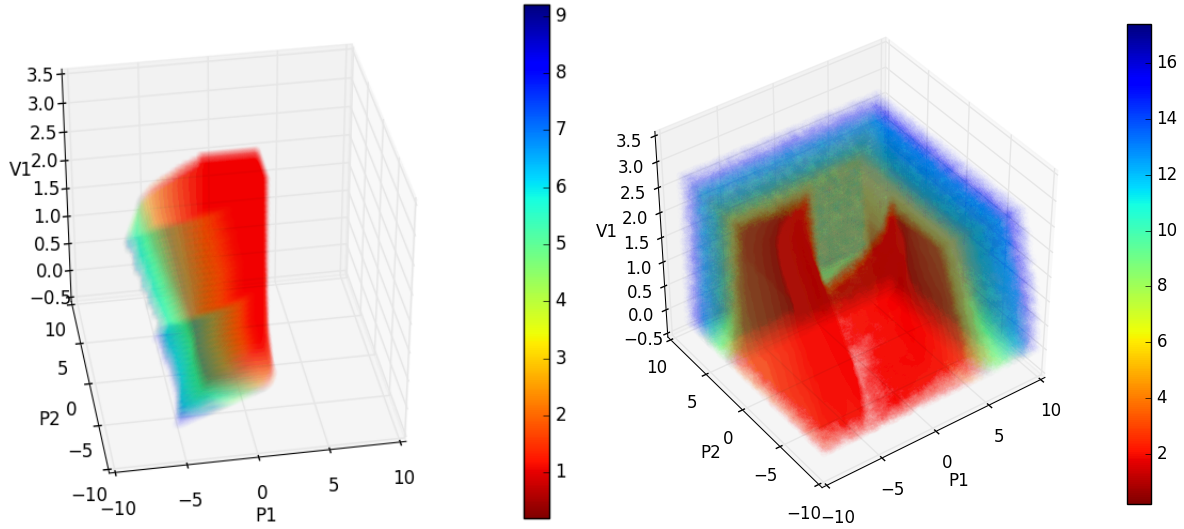


Figure 7: (Left) Three dimensional projection of the four dimensional unsafe region $\mathscr{K}^C$ for centralized controller C with $v_2 = 2.8$ held constant. Color scale shows the earliest potential collision time. (Right) Figure shows the unsafe action region $(\mathsf{SIPRE}^D_{\mathscr{S}}(\mathscr{K}))^C$ for the system $f \circ \mathsf{IND}_C(x)$ expand as communication delay D increases.

The sets $\mathscr{X}^i = [-.2,.2] \times [-.2,.2]$ and $\mathscr{U}^i = [-1,1] \times [-1,1]$ for both $i = 1,2$. A collision has occurred between both agents in the region

$$\mathscr{S}^C = \{(x^1,x^2) \in \mathscr{X}^1 \times \mathscr{X}^2 : \max(|x^1_1 - x^2_1|, |x^1_2 - x^2_2|) < 0.1\}. \tag{21}$$

SCOTS is again used to synthesize a centralized controller for the system. The discrete abstraction was constructed with sampling period $\tau = .01$, state space grid $[\mathscr{X}]_\eta$ with parameter $\eta = [.01,.01,.01,.01]$, and input space grid $[\mathscr{U}]_\varepsilon$ with parameter $\varepsilon = [.2,.2,.2,.2]$. Figure 8 shows the trajectory of the system with the self-triggering implementation and how $\hat{T}^{-1}(x[k])$ as defined in Equation (13) varies with respect to time.

6 Conclusion

We have presented a method to analyze when communication is necessary in order for a distributed controller to satisfy a safety requirement. While the current implementation deals with memoryless controllers future work will look into control policies with memory, time varying connectivity, and an application to richer specifications including those expressible in temporal logic.

References

[1] Federal Aviation Administration (2017): *Code of Federal Regulations, Title 14*.

[2] Florian D Brunner, TMP Gommans, WPMH Heemels & Frank Allgöwer (2015): *Communication Scheduling in Robust Self-Triggered MPC for Linear Discrete-Time Systems*. *IFAC-PapersOnLine* 48(22), pp. 132–137, doi:10.1016/j.ifacol.2015.10.319.

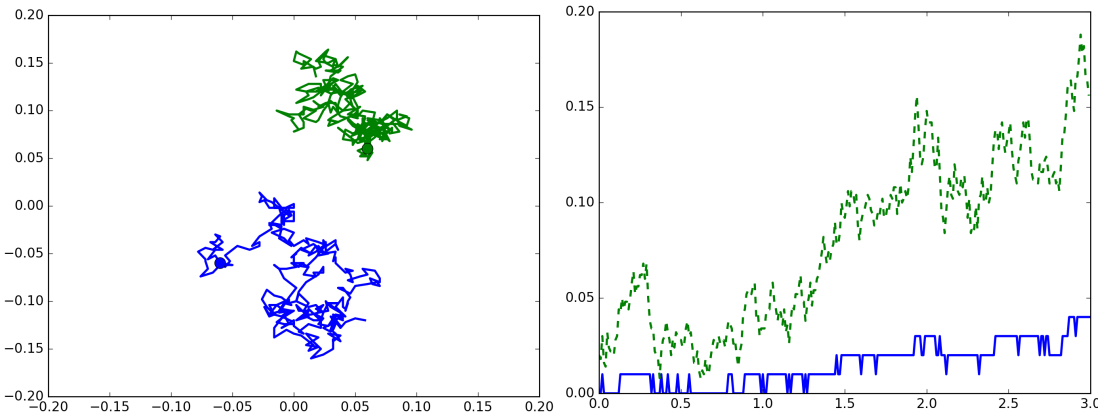


Figure 8: (Left) Trajectories of both systems (Right) The solid line is the value for $\hat{T}^{-1}(x[k])$ which underapproximates the actual time to when a collision is inevitable $\mathscr{K}^C$. Because Equation (20) is fully actuated, the safe set $\mathscr{K}$ and the invariance region $\mathscr{S}$ are identical.

[3] Randal E Bryant (1992): *Symbolic Boolean manipulation with ordered binary-decision diagrams*. *ACM Computing Surveys (CSUR)* 24(3), pp. 293–318, doi:10.1145/136035.136043.

[4] Mo Chen, Sylvia L Herbert, Mahesh S Vashishtha, Somil Bansal & Claire J Tomlin (2018): *A general system decomposition method for computing reachable sets and tubes*. *IEEE Transactions on Automatic Control*, doi:10.1109/TAC.2018.2797194.

[5] Eric Dallal & Paulo Tabuada: *Decomposing Controller Synthesis for Safety Specifications*. In: *CDC2016*, doi:10.1109/CDC.2016.7799148.

[6] T.M.P. Gommans & W.P.M.H. Heemels (2015): *Resource-aware MPC for constrained nonlinear systems: A self-triggered control approach*. *Systems and Control Letters* 79, pp. 59 – 67, doi:10.1016/j.sysconle.2015.03.003. Available at `http://www.sciencedirect.com/science/article/pii/S0167691115000481`.

[7] Felix Gruber, Eric S Kim & Murat Arcak (2017): *Sparsity-Aware Finite Abstraction*. In: *CDC2017*, doi:10.1109/CDC.2017.8263995.

[8] WPMH Heemels, Karl Henrik Johansson & Paulo Tabuada (2012): *An introduction to event-triggered and self-triggered control*. In: *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, IEEE, pp. 3270–3285, doi:10.1109/CDC.2012.6425820.

[9] Eric S Kim, Murat Arcak & Sanjit A Seshia (2015): *Compositional controller synthesis for vehicular traffic networks*. In: *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, IEEE, pp. 6165–6171, doi:10.1109/CDC.2015.7403189.

[10] Pierre-Jean Meyer, Antoine Girard & Emmanuel Witrant (2017): *Compositional abstraction and safety synthesis using overlapping symbolic models*. *IEEE Transactions on Automatic Control*, doi:10.1109/TAC.2017.2753039.

[11] Pierluigi Nuzzo, Alberto L Sangiovanni-Vincentelli, Davide Bresolin, Luca Geretti & Tiziano Villa (2015): *A platform-based design methodology with contracts and related tools for the design of cyber-physical systems*. *Proceedings of the IEEE* 103(11), pp. 2104–2132, doi:10.1109/JPROC.2015.2453253.

[12] Gunther Reissig, Alexander Weber & Matthias Rungger (2017): *Feedback refinement relations for the synthesis of symbolic controllers*. *IEEE Transactions on Automatic Control* 62(4), pp. 1781–1796, doi:10.1109/TAC.2016.2593947.

[13] Matthias Rungger & Majid Zamani (2016): *Compositional Construction of Approximate Abstractions of Interconnected Control Systems*. IEEE Transactions on Control of Network Systems, doi:10.1109/TCNS.2016.2583063.

[14] Matthias Rungger & Majid Zamani (2016): *SCOTS: A tool for the synthesis of symbolic controllers*. In: Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, ACM, pp. 99–104, doi:10.1145/2883817.2883834.

[15] Sadra Sadraddini, János Rudan & Calin Belta (2017): *Formal synthesis of distributed optimal traffic control policies*. In: Proceedings of the 8th International Conference on Cyber-Physical Systems, ACM, pp. 15–24, doi:10.1145/3055004.3055011.

[16] Paulo Tabuada (2009): *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, doi:10.1007/978-1-4419-0224-5.

[17] Alfred Tarski (1955): *A lattice-theoretical fixpoint theorem and its applications*. Pacific journal of Mathematics 5(2), pp. 285–309, doi:10.2140/pjm.1955.5.285.