Compositional Safety Rules for Inter-Triggering Hybrid Automata

Kwesi J. Rutledge*
Univ. of Michigan, Ann Arbor, USA
krutledg@umich.edu

Glen Chou*
Univ. of Michigan, Ann Arbor, USA
gchou@umich.edu

Necmiye Ozay
Univ. of Michigan, Ann Arbor, USA
necmiye@umich.edu

ABSTRACT

In this paper, we present a compositional condition for ensuring safety of a collection of interacting systems modeled by inter-triggering hybrid automata (ITHA). ITHA is a modeling formalism for representing multi-agent systems in which each agent is governed by individual dynamics but can also interact with other agents through triggering actions. These triggering actions result in a jump/reset in the state of other agents according to a global resolution function. A sufficient condition for safety of the collection, inspired by responsibility-sensitive safety, is developed in two parts: self-safety relating to the individual dynamics, and responsibility relating to the triggering actions. The condition relies on having an over-approximation method for the resolution function. We further show how such over-approximations can be obtained and improved via communication. We use two examples, a job scheduling task on parallel processors and a highway driving example, throughout the paper to illustrate the concepts. Finally, we provide a comprehensive evaluation on how the proposed condition can be leveraged for several multi-agent control and supervision examples.

ACM Reference Format:
Kwesi J. Rutledge, Glen Chou, and Necmiye Ozay. 2021. Compositional Safety Rules for Inter-Triggering Hybrid Automata. In *24th ACM International Conference on Hybrid Systems: Computationand Control (HSCC '21), May 19–21, 2021, Nashville, TN, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3447928.3456659

1 INTRODUCTION

Proving safety or designing controllers guaranteeing safety in a multi-agent setting is a challenging task for multiple reasons. On one extreme, one can try to come up with a monolithic safety rule, which may be hard to verify due to scalability issues, and hard to follow at run-time without a central coordinator. On the other extreme, one can analyze agents individually, assuming all of the remaining agents act adversarially, in which case safety is hard to attain, if at all possible. Several frameworks have been developed between these two extremes to capture various notions of coordination, collaboration, or contracts [5, 6, 9, 14, 17, 23, 25, 28].

An alternative viewpoint is presented by responsibility-sensitive safety [7, 11, 24, 27], in the context of autonomous driving, where some hard safety constraints are replaced by a notion of not causing a crash and avoiding one whenever possible. This is particularly well-suited for scenarios where some of the agents are human-controlled, which can lead to unpredictable behaviors. From this viewpoint, we expect autonomous agents to act in a well-behaved fashion as long as others reciprocate, and we should not punish the autonomous agents for failures that are out of their control.

In this paper, we consider multi-agent systems modeled by inter-triggering hybrid automata (ITHA), a modeling framework for interacting parallel processes and multi-agent systems [20]. Inter-triggering hybrid automata consist of a collection of agents, where each agent is modeled with a discrete-time dynamical system locally. In addition to their local dynamics, agents are also equipped with triggering actions as a means to interact with other agents in the collection. In particular, these triggering actions can collectively induce a reset (i.e., jump) in the dynamics of other agents. For this class of systems, we propose a simple two part condition for each agent to follow that is shown to be sufficient to guarantee safety of the overall collection. The two parts pertain to self-safety (in the local dynamics) and responsibility (in the interactions), jointly called *responsibility-sensitive safety*. Controlled invariant sets for the individual dynamics [3] are used for both parts of the condition. Intuitively, each agent aims to remain in their corresponding invariant set, and when they use a triggering action that affects another agent, they do so in a "responsible way", by trying to ensure that the other agent's state does not leave its invariant set due to this action. The responsibility-sensitive safety conditions, being based on local invariant sets, enable us to use the same conditions either for proving the safety of other policies, for supervising existing policies, or for designing new control policies for guaranteed safety.

A central component of the inter-triggering hybrid automaton is what we call a global resolution function, which determines the reset induced on a given agent based on all agent's triggering actions. This function's value cannot be known, in general, to any of the agents at run-time. This is because the value of the global resolution function depends on the triggering actions chosen by all agents. To overcome this issue, we introduce over-approximations of the resolution functions that can be used within the responsibility rule. We also show how individual agents may compute such over-approximations, and how the conservativeness in this computation can be reduced if (local) communication between agents is allowed.

We apply our proposed framework on several multi-agent control problems, including task coordination for parallel processing on a server farm and navigation for autonomous highway driving, empirically verifying the safety guarantees of our method and also demonstrating how different over-approximations can lead to

*Both authors contributed equally to the paper. This work is supported in part by ONR grant # N00014-18-1-2501, NSF grants #1553873 and #1918123. KJR is also supported by an NSF graduate research fellowship and GC is supported by an NDSEG fellowship.

HSCC '21, May 19–21,2021, Nashville, TN, USA

ACM ISBN 978-1-4503-8339-4/21/05…$15.00
https://doi.org/10.1145/3447928.3456659

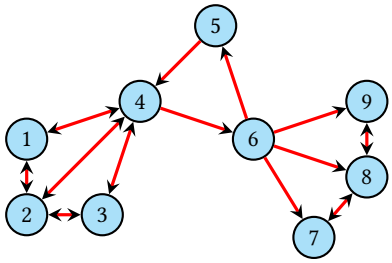

Figure 1: An example directed graph $\mathcal{G}$ that we use to demonstrate potential links between processors in a server farm.

differing levels of conservativeness. Finally, we evaluate the conservativeness of our framework by employing it as a safety supervisor on trajectories drawn from a real-world highway driving data-set [13] and demonstrate that we experience few safety overrides, suggesting that ITHA is sufficiently permissive to be used to supervise system safety without excessive intervention.

We summarize our contributions as:

- We refine the inter-triggering hybrid automata model: a flexible modeling formalism for representing multi-agent interactions, vastly expanding on the initial ideas in [20].
- We design compositional conditions which are sufficient for global safety, provided that each agent ensures it remains self-safe and responsible with respect to the agents it triggers.
- For tractability, we provide practical over-approximations of the collective triggering behavior and show how they can be made less conservative with local communication.
- We perform a comprehensive evaluation of our method on a variety of multi-agent control and supervision tasks.

2 PRELIMINARIES

In this section, we introduce the graph, dynamical system, and invariant set notation which will be used throughout the paper.

2.1 Graph notation

A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a tuple containing a set of vertices $\mathcal{V}$ and a set of directed edges $\mathcal{E}$. Note that each edge is an ordered pair $(v_1, v_2) \in \mathcal{E}$ of vertices from $\mathcal{V}$ (i.e. $v_1, v_2 \in \mathcal{V}$). We say that a vertex v_1 is *connected* to a vertex v_2 if and only if $(v_1, v_2) \in \mathcal{E}$. By the nature of directed graphs, v_1 being connected to v_2 does not mean that v_2 is connected to v_1.Within directed graphs the *inward connections* of a vertex v can be defined as follows:

$$in_{\mathcal{G}}(v) = \{v' \in \mathcal{V} \mid (v', v) \in \mathcal{E}\}.$$

The *outward connections* of a vertex v can be defined similarly:

$$out_{\mathcal{G}}(v) = \{v' \in \mathcal{V} \mid (v, v') \in \mathcal{E}\}.$$

2.2 Invariant sets

Consider a discrete-time dynamical system $\Sigma : \langle \mathcal{X}, \mathcal{U}, \mathcal{D}, f \rangle$ where $\mathcal{X}$ is the state space, $\mathcal{U}$ is the set of control inputs, $\mathcal{D}$ is the set of disturbances, and $f : \mathcal{X} \times \mathcal{U} \times \mathcal{D} \to \mathcal{X}$ is the state update function. The state of the system Σ evolves according to

$$x(t+1) = f(x(t), u(t), d(t)). \tag{1}$$

Controlled invariant sets play an important role in ensuring safety of systems with dynamics of the form (1). Formally, a robust controlled invariant set C_{inv} inside a given safe set $\mathcal{X}_{\text{safe}} \subseteq \mathcal{X}$ (i.e., $C_{\text{inv}} \subseteq \mathcal{X}_{\text{safe}}$) is a set of states that satisfies:

$$x \in C_{\text{inv}} \Rightarrow \exists u \in \mathcal{U} \;\; \forall d \in \mathcal{D} \;\; f(x, u, d) \in C_{\text{inv}}. \tag{2}$$

In words, this means that if the state $x(t)$ is in C_{inv}, there is an input $u(t)$ to ensure that $x(t+1)$ will be in C_{inv} for any disturbance within given bounds, thus allowing the states to stay in C_{inv} indefinitely.

3 A MODELING FORMALISM FOR INTERACTING SYSTEMS

We introduce *inter-triggering hybrid automata* (ITHA), a hybrid modeling formalism for collections of discrete-time hybrid systems with a special form of interaction between them. In particular, these interactions are such that they induce jumps or resets on the state evolution of individual agents.

Definition 1. *An* inter-triggering hybrid automaton *is a collection $\{\mathcal{H}_i\}_{i \in \mathcal{I}}$ of systems together with a function $\rho = (\rho_1, \ldots, \rho_{|\mathcal{I}|})$, which we refer to as a resolution function, with each $\mathcal{H}_i$, i.e., agent i, being a hybrid automaton of the form $\mathcal{H}_i = \langle \Sigma_i, \mathcal{T}_i, R_i \rangle$, where:*

- *$\Sigma_i = \langle \mathcal{X}_i, \mathcal{U}_i, \mathcal{D}_i, f_i \rangle$ are the individual dynamics for agent i;*
- *$\mathcal{T}_i$ is the set of triggering actions of agent i, including a null triggering action $\epsilon \in \mathcal{T}_i$ that indicates that agent i is not triggering a reset on any other agent;*
- *$R_i : \mathbb{N} \times \mathcal{X}_i \times \mathcal{U}_i \times 2^{\mathcal{I}} \to 2^{\mathcal{X}_i}$ is the (potentially time-varying) reset map for agent i*[1]*;*

and where each $\rho_i : \mathbb{N} \times \mathcal{T}_1 \times \cdots \times \mathcal{T}_{|\mathcal{I}|} \to 2^{\mathcal{I}}$ is the resolution function of agent i that takes the triggering inputs of the entire collection and determines the set of agents that trigger a reset on agent i.

For notational simplicity when referring to an ITHA, we will omit the resolution function and simply say "an ITHA $\{\mathcal{H}_i\}_{i \in \mathcal{I}}$". We assume the resolution function ρ satisfies the following property, which essentially says if agent i is using a null triggering action at a given time, it will not appear in the output of the resolution function of any agents at that time.

Assumption 1. *For all i, for all t such that $\tau_i(t) = \epsilon$, we have $i \notin \rho_j(t, \ldots, \tau_i(t), \ldots)$ for all j, which implies, as a special case, $\rho_i(t, \epsilon, \ldots, \epsilon) = \emptyset$.*

Definition 2 (Execution of an ITHA). *Given sequences of control inputs $\mathbf{u}_i = u_i(0), u_i(1) \ldots$ and triggering inputs $\boldsymbol{\tau}_i = \tau_i(0), \tau_i(1) \ldots$ for each agent i, an* execution *of $\{\mathcal{H}_i\}_{i \in \mathcal{I}}$ is a collection of sequences $\{\mathbf{e}_i\}_{i \in \mathcal{I}}$, each a sequence of alternating states and actions $\mathbf{e}_i = x_i(0), u_i(0), \tau_i(0), x_i(1), u_i(1), \tau_i(1), \ldots$ such that:*

$$x_i(0) \in \mathcal{X}_i \quad \forall i \in \mathcal{I} \tag{3a}$$

$$(u_i(t), \tau_i(t)) \in \mathcal{U}_i \times \mathcal{T}_i \quad \forall i \in \mathcal{I}, \forall t \geq 0 \tag{3b}$$

$$x_i(t+1) \in \begin{cases} f_i(x_i(t), u_i(t), \mathcal{D}_i) & \text{if } \rho_i(t) = \emptyset, \\ R_i(t, x_i(t), u_i(t), \rho_i(t)) & \text{otherwise.} \end{cases} \tag{3c}$$

where $\rho_i(t) \triangleq \rho_i(t, \tau_1(t), \tau_2(t), \ldots, \tau_{|\mathcal{I}|}(t))$.

[1] With a slight abuse of notation, the last argument of the reset map is shown as an index set, but the actual reset value depends on the state, input, and triggering action of the agents in that set.

For an element $\mathbf{e}_i$ of an execution, we denote the corresponding state trajectory by $\mathbf{x}_i = x_i(0), x_i(1), \ldots$. We consider problems related to safety of an execution of an ITHA, where we require the state trajectory of each agent $\mathcal{H}_i$ to remain in a safe set $\mathcal{X}_{i,safe} \subseteq \mathcal{X}_i$ for all times. We use an execution $\{\mathbf{e}_i\}_{i\in\mathcal{I}}$ remaining in a collection of sets $\{\mathcal{X}_{i,safe}\}_{i\in\mathcal{I}}$ inter-changeably with the corresponding state trajectories $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$ remaining in the same collection.

To make the definition of inter-triggering hybrid automaton concrete, we present two examples used throughout the paper.

Example 1 (Parallel Processors on a Server Farm). *A collection of processors in a server farm can be treated as a collection of agents where each agent's state is the number of jobs it has left to compute. In other words, agent i's state $x_i \in \mathbb{N}$, where there is a limit of jobs, $n_{overflow}$, over which the processor will create a stack overflow and fail. External jobs d_i for processor i are passed into the server according to a protocol that blocks new jobs from coming in if $x_i \geq n_{throttle}$ where $n_{throttle} < n_{overflow}$ and the processor always can take the action to address a job in its queue or do nothing. Thus, the individual dynamics can be visualized as shown in Fig. 2. In addition, the processors can be recruited by other processors according to a directed graph $\mathcal{G}$ that indicates which processors can send jobs to which other processors, i.e. processor i can recruit processor j if $(i, j) \in \mathcal{E}$. An example of such a graph is shown in Fig. 1. This scenario can be modeled with the representation $\{\mathcal{H}_i\}_{i\in\mathcal{I}}$, where each agent $\mathcal{H}_i = \langle \Sigma_i, \mathcal{T}_i, R_i\rangle$ is composed of the following parts:*

- *Dynamics Σ_i where $\mathcal{X}_i = \mathbb{N}$ is the queue of jobs to be done by agent i; $\mathcal{U}_i = \{0, -1\}$ represents processor i's choice to do nothing (i.e. $u_i(t) = 0$) or to address one of the jobs in its queue (i.e. $u_i(t) = -1$); $\mathcal{D}_i \in \{0, 1, 2\}$ represents the number of external jobs passed into processor i, and f_i, given by*

$$f_i(x_i, u_i, d_i) = \begin{cases} x_i + u_i & \text{if } x_i \geq n_{throttle} \\ x_i + u_i + d_i & \text{otherwise,} \end{cases}$$

  describes how the queue of jobs is changing for agent i in the absence of it being recruited;
- *$\mathcal{T}_i = 2^{out_{\mathcal{G}}(i)}$ represents the possible sets of agents that agent i recruits to help it with its queue, with null element $\epsilon = \emptyset$, according to its outgoing edges in $\mathcal{G}$;*
- *Reset map R_i describes how the queue for agent i changes if it was recruited by or recruited another agent. To each agent in the recruit set (i.e. $\forall j \in \tau_i(t) \in \mathcal{T}_i$), agent i sends 1 job from its queue to that processor:*

$$R_i(t, x_i(t), u_i(t), S) = \begin{cases} 0 & \text{if } s_i(t) \leq 0, \\ s_i(t) & \text{otherwise,} \end{cases} \tag{4}$$

  where $s_i(t) = x_i(t) - |\tau_i(t)| + u_i(t) + |S| + d_i(t)$, where S is the set of agents recruiting agent i.

Furthermore, we can write the i^{th} component of the resolution function $\rho_i(t) = \{j \in \mathcal{I} \mid i \in \tau_j(t)\}$; that is, the set of agents triggering agent i at time t is the set of agents which contain i in its triggering action at time t. Concretely, for agent 1 in Fig. 1, $\rho_1(t) \subseteq \{2, 4\}$, for all t. If $\tau_2(t) = \epsilon$ and $\tau_4(t) = \{1, 6\}$, then $\rho_1(t) = \{4\}$, regardless of the triggering actions of the remaining agents.

Note that each processor can avoid the overflow states indefinitely if it is within a robust control invariant set that is completely contained in the safe set $\{x \in \mathbb{N} \mid x < n_{overflow}\}$. Under the individual dynamics, the maximal control invariant set in the safe set (i.e. $C \subseteq \{x \in \mathbb{N} \mid x < n_{overflow}\}$) can be quickly shown to be $C = \{x \in \mathbb{N} \mid x \leq n_{overflow} - 1\}$. Depending on the objectives of the processors, i.e. maximizing throughput, each processor may need to trigger other agents, and without adequate precaution the triggering can reset the states of some processors above $n_{overflow}$, leading to unsafe behavior.

Example 2 (Highway Driving). *Consider a collection of vehicles travelling in the same direction on a highway (see Fig. 3). This collection can be represented by an inter-triggering hybrid automaton $\{\mathcal{H}_i\}_{i\in\mathcal{I}}$ where each agent $\mathcal{H}_i = \langle \Sigma_i, \mathcal{T}_i, R_i\rangle$ is composed of:*

- *Dynamics Σ_i with $\mathcal{X}_i = [0, v_{max}] \times [0, \infty) \times [0, v_{max}]$, where state $x_i = [v_i, h_i, v_i^L]^\intercal$ contains v_i (velocity of current agent i, henceforth referred to as the ego vehicle), h_i (headway between this agent and the nearest car in front of it on the same lane, henceforth referred to as the lead vehicle), and v_i^L (the velocity of the lead vehicle), $\mathcal{U}_i$ is the set of allowed inputs, with input $u_i = a_i$ being the acceleration of the ego car, $\mathcal{D}_i$ is the set of allowable disturbances, with disturbance $d_i = a_i^L$ being the acceleration of the lead vehicle, and the system dynamics $f_i : \mathcal{X}_i \times \mathcal{U}_i \times \mathcal{D}_i \to \mathcal{X}_i$ is such that*

$$f_i(x_i, u_i, d_i) = \begin{bmatrix} 1 & 0 & 0 \\ -\Delta t & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} x_i + \begin{bmatrix} \Delta t \\ 0 \\ 0 \end{bmatrix} u_i + \begin{bmatrix} 0 \\ 0 \\ \Delta t \end{bmatrix} d_i \tag{5}$$

  where Δt is the sampling time;
- *$\mathcal{T}_i = \{stay, left, right\}$ is the set of possible lane change decisions, with null element $\epsilon = stay$;*
- *A reset $R_i(t, x_i(t), u_i(t), \rho_i(t))$ is triggered on agent i if one of the following happens (1) $\tau_i(t) \neq \epsilon$, (2) if their lead car is j and $\tau_j(t) \neq \epsilon$ and/or (3) a car becomes the current agent's lead car (i.e. lead car was j at time $t-1$ but the lead car becomes $k \neq j$ at time t). The value of the reset depends on the triggering actions of all agents $\rho_i(t)$, which can possibly affect agent i at time t. This determines the new lead car and hence the new values of h_i and v_i^L, while v_i evolves according to individual dynamics in* (5).

Controlled invariant sets for the system defined in (5) *can be computed using polyhedral set computation methods such as those discussed in [18, 25]. An example of such a set is shown in Fig. 4.*

To further illustrate how the reset map is defined for highway driving, the value of R_i will be explained for some vehicles in Fig. 3. First, suppose that the current time is t and vehicles E and F_2 have states $x_E = [v_E, h_E, v_E^L]^\intercal$ and $x_{F_2} = [v_{F_2}, h_{F_2}, v_{F_2}^L]^\intercal$, respectively, and apply control inputs u_E and u_{F_2}. Suppose that the ego vehicle makes a left lane change at time t; this action triggers a reset of its own continuous state as well as that of F_2 (leading to changes in headway and lead car velocity for both vehicles). Formally, if $\tau_E(t) = \{left\}$, then

$$R_E(t, x_E, u_E, \{E\}) = \begin{bmatrix} v_E + \Delta t u_E \\ \infty \\ v_{max} \end{bmatrix}$$

and

$$R_{F_2}(t, x_{F_2}, u_{F_2}, \{E\}) = \begin{bmatrix} v_{F_2} + \Delta t u_{F_2} \\ h_{F_2}^{rel} + (v_E - v_{F_2})\Delta t \\ v_E + \Delta t u_E \end{bmatrix}$$

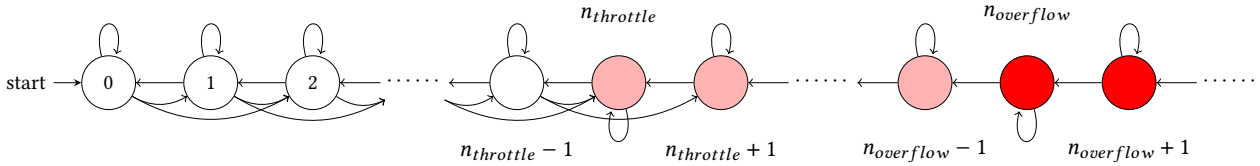

Figure 2: The individual dynamics for a processor in the collection specified in Example 1. The processor is unable to accept as many jobs when $x_i(t) \geq n_{throttle}$ and it experiences a stack overflow (i.e. it fails) if $x_i(t) \geq n_{overflow}$.

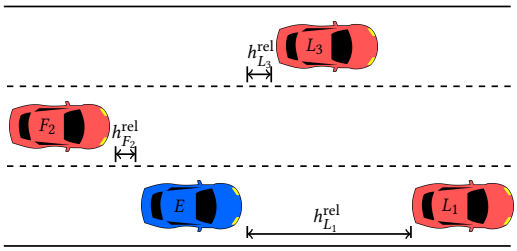

Figure 3: A collection of vehicles on the highway, as described in Example 2. The ego vehicle E is marked in blue, and longitudinal distances between the ego vehicle and car i are marked as h_i^{rel}.

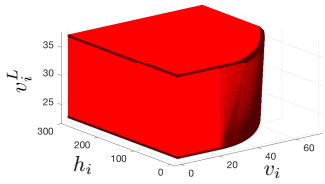

Figure 4: The controlled invariant set for the car-following system defined in (5) for parameters used in [18, 25].

where we assume by convention that resetting an agent to have no lead car results in a headway of ∞ and a lead car velocity of v_{max}.

As concrete examples, we overview a subset of possible values that the resolution function may take in the highway setting. We can write $\rho_E(t, \tau_E(t) = left, \tau_{L_1}(t) = \epsilon, \tau_{F_2}(t) = \epsilon, \tau_{L_3}(t) = right) = \{E, L_3\}$, as both the ego vehicle and vehicle L_3 changing lanes resets the continuous state of the ego vehicle. Likewise, we can write $\rho_E(t, \tau_E(t) = \epsilon, \tau_{L_1}(t) = \epsilon, \tau_{F_2}(t) = right, \tau_{L_3}(t) = \epsilon) = \emptyset$, as F_2 will not trigger a state reset on the ego vehicle. Vehicle L_1 can also trigger a reset on the state of the ego vehicle by changing lanes, i.e. $\rho_E(t, \tau_E(t) = \epsilon, \tau_{L_1}(t) = left, \tau_{F_2}(t) = \epsilon, \tau_{L_3}(t) = right) = \{L_1\}$. A final, more complicated case occurs when vehicles E and L_1 both make a left lane change, while L_3 makes a right lane change at the same time: $\rho_E(t, \tau_E(t) = left, \tau_{L_1}(t) = left, \tau_{F_2}(t) = \epsilon, \tau_{L_3}(t) = right) = \{E, L_3\}$; in this case, L_1 does not end up factoring into the resolution function as L_3 becomes the ego vehicle's lead car instead.

4 COMPOSITIONAL SAFETY RULES

In general, not all executions of an inter-triggering hybrid automaton are safe. To render the executions safe, control policies may need to restrict the possible control inputs and triggering actions of individual agents. In this section, we develop sufficient conditions on local control policies that collectively guarantee safety. To do this, robust controlled invariant sets are found for the inter-triggering hybrid automaton's individual dynamics and then responsibility-sensitive safe controllers are defined with respect to these individual invariant sets. This section shows that safety can be guaranteed when all agents in the collection use such responsibility-sensitive safe controllers and then discusses how conservativeness can be further reduced via a communication scheme.

4.1 Control Policies and Safety Control Problem for ITHA

At run-time each agent i picks its control inputs $u_i(t)$ and triggering actions $\tau_i(t)$ based on the information available to it by time t. Formally, for a given set $\mathcal{Y}_i$ of possible observations of agent i, a memoryless local controller (or, control policy) is a function $\gamma_i : \mathcal{Y}_i \to \mathcal{U}_i \times \mathcal{T}_i$. Similarly, a local controller with memory is a function $\gamma_i : \mathcal{Y}_i^+ \to \mathcal{U}_i \times \mathcal{T}_i$, where the superscript + denotes finite non-zero repetition. If agent i's decisions only depend on its own state or the state of all agents, we have $\mathcal{Y}_i = \mathcal{X}_i$ or $\mathcal{Y}_i = \mathcal{X}_1 \times \ldots \times \mathcal{X}_{|\mathcal{I}|}$, respectively. Also, if an agent can access a (potentially time-varying) subset of other agents' states, we have $\mathcal{Y}_i = \bigcup_{\mathcal{I}' \subset \mathcal{I}} \{\mathcal{X}_j\}_{j \in \mathcal{I}'}$. In addition to states, $\mathcal{Y}_i$ can incorporate observations of actions of the other agents, which would be relevant when introducing the communication scheme in section 4.3.2.

Definition 3 (Controlled Execution of an ITHA). *Given a collection of controllers $\{\gamma_i\}_{i \in \mathcal{I}}$, $\{\gamma_i\}_{i \in \mathcal{I}}$-controlled executions of $\{\mathcal{H}_i\}_{i \in \mathcal{I}}$ are the set of executions where control inputs $\mathbf{u}_i$ and triggering inputs $\boldsymbol{\tau}_i$ are produced according to the function γ_i for all i.*

Given a collection of local safe sets $\{\mathcal{X}_{i,safe}\}_{i \in \mathcal{I}}$ and information $\mathcal{Y}_i$ available to each agent, synthesizing local controllers for each agent to guarantee global safety is a distributed synthesis problem [21]. Verifying existence of such controllers is known to be undecidable in general even when the sets $\mathcal{X}_i, \mathcal{U}_i, \mathcal{D}_i$ are finite [4, 8]. Any architecture defining the information flow in a distributed synthesis problem can be captured by choosing some $\mathcal{I}_i'$ and setting $\mathcal{Y}_i = \{\mathcal{X}_j\}_{j \in \mathcal{I}_i'}$, therefore synthesis in the ITHA setting cannot be easier. Given this hardness result, we instead search for sufficient conditions on local controllers under which global safety is guaranteed. These conditions can be checked locally and instantaneously in time. Moreover, instead of working with a fixed observation structure, we will deduce the sets $\mathcal{Y}_i$ each local agent should have access to in order to be able to comply with the conditions.

4.2 Responsibility-Sensitive Safety

Responsibility-sensitive safety consists of two rules. The first rule handles safety of the individual dynamics and the second rule handles safety during triggering interactions. Consider the first rule:

Definition 4 (Self-Safety). *A control policy γ_i renders agent $\mathcal{H}_i$* self-safe *on a set $\mathcal{X}_{i,c} \subseteq \mathcal{X}_i$ if for all states in $\mathcal{X}_{i,c}$, the control input guarantees $\mathcal{H}_i$'s own safety assuming a reset will not happen*

in the next step. In math, for all t, if $x_i(t) \in \mathcal{X}_{i,c}$, then $u_i(t)$ produced by γ_i is such that

$$f_i(x_i(t), u_i(t), \mathcal{D}_i) \subseteq \mathcal{X}_{i,c}. \tag{6}$$

It is clear from (2) that for the existence of a self-safe controller on $\mathcal{X}_{i,c}$, $\mathcal{X}_{i,c}$ should be a robust controlled invariant set. Moreover, the controller would only need information on the agent's own state to be in $\mathcal{Y}_i$. Though even when $\mathcal{X}_{i,c}$'s are robust controlled invariant sets for Σ_i's, adopting a controller that renders $\mathcal{X}_{i,c}$ invariant, an agent $\mathcal{H}_i$ cannot be guaranteed to remain in $\mathcal{X}_{i,c}$ because its state trajectories depend on both f_i and R_i. To incorporate the potential resets of agent $\mathcal{H}_i$ into an invariance condition, any agent j contributing to a reset on agent i, i.e., $j \in \rho_i(t)$, should somehow make guarantees about R_i on the same set. While it may be problematic to expect agents to know ρ in a distributed setting, an over-approximation of the value of ρ at each time step, as defined next, can be obtained locally in many practical scenarios.

Definition 5 (Resolution Over-approximation). *A function $\hat{\rho}_i : \mathbb{N} \times \mathcal{T}_1 \times \mathcal{T}_2 \times \cdots \times \mathcal{T}_{|\mathcal{I}|} \to 2^{2^{\mathcal{I}}}$ is an* over-approximation *of the i^{th} component of the resolution function if and only if:*

$$\rho_i(t, \tau_1', \tau_2', \cdots, \tau_{|\mathcal{I}|}') \in \hat{\rho}_i(t, \tau_1', \tau_2', \cdots, \tau_{|\mathcal{I}|}').$$

for all values of t and all $\tau_i' \in \mathcal{T}_i$ for all i. Similarly, we say $\hat{\rho}$ is an over-approximation of the resolution function ρ, denoted as $\hat{\rho} \supseteq \rho$, if and only if each $\hat{\rho}_i$ is an over-approximation of corresponding ρ_i.

For notation convenience, when the triggering action arguments of $\hat{\rho}$ are clear from the context or are irrelevant, we simply write $\hat{\rho}_i(t)$. We discuss how resolution over-approximations can be obtained locally by each agent in Section 4.3. In general, different agents j might have different resolution over-approximations $\hat{\rho}^{(j)} \supseteq \rho$ depending on their local information. With this in mind, to enable safety through resets, we define a responsibility rule that uses such over-approximations.

Definition 6 ($\hat{\rho}$-Responsibility). *Given an over-approximation $\hat{\rho}$ of the resolution function and a collection $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$ of sets, a controller γ_j renders an agent $\mathcal{H}_j$ $\hat{\rho}$-*responsible *with respect to the sets $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$ if, when agent j triggers a reset on other agents, agent j's triggering action does not lead to safety violations for any other agent that it could induce a reset on according to $\hat{\rho}$, possibly including itself. In math, the controller γ_j renders $\mathcal{H}_j$ $\hat{\rho}$-*responsible*, if for all t, $\tau_j(t)$ and $u_j(t)$ produced by the controller are such that if $\tau_j(t) \neq \epsilon$ and $x_i(t) \in \mathcal{X}_{i,c}$ $\forall i \in \mathcal{I}$, then for all $i \in \mathcal{I}$ and $S \in \hat{\rho}_i(t)$ with $j \in S$ we have:*

$$\begin{cases} R_i(t, x_i(t), \mathcal{U}_i, S) \subseteq \mathcal{X}_{i,c} & \text{if } i \neq j, \text{and} \\ R_i(t, x_i(t), u_i(t), S) \subseteq \mathcal{X}_{i,c} & \text{if } i = j. \end{cases} \tag{7}$$

We use controller being $\hat{\rho}$-responsible (or, self-safe), agent being $\hat{\rho}$-responsible (or, self-safe) and controller rendering an agent $\hat{\rho}$-responsible (or, self-safe), interchangeably. With all of the above the following theorem can be stated, which provides a recursive safety guarantee.

Theorem 1. *Consider an inter-triggering hybrid automaton $\{\mathcal{H}_i\}_{i\in\mathcal{I}}$, an accompanying collection of sets $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$ that are robustly controlled invariant for respective Σ_i's in their respective safe sets $\{\mathcal{X}_{i,safe}\}_{i\in\mathcal{I}}$ and a collection $\{\hat{\rho}^{(i)}\}_{i\in\mathcal{I}}$ of resolution over-approximations. Then,*

(1) there exists local controllers γ_i for each agent $\mathcal{H}_i$ that render them self-safe and $\hat{\rho}^{(i)}$-responsible with respect to the sets $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$ and
(2) if each agent uses a controller γ_i that renders itself self-safe and $\hat{\rho}^{(i)}$-responsible with respect to the sets $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$, the state trajectories corresponding to any $\{\gamma_i\}_{i\in\mathcal{I}}$-controlled execution of $\{\mathcal{H}_i\}_{i\in\mathcal{I}}$ beginning in $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$ always remain within these sets.

Proof. To prove statement (1), consider, for each $\mathcal{H}_i$, a controller that produces an input $u_i(t) \in \{u \mid f(x_i(t), u, \mathcal{D}) \subseteq \mathcal{X}_{i,c}\}$ and the triggering action $\tau_i(t) = \epsilon$ for all time t for which $x_i(t) \in \mathcal{X}_{i,c}$. With the triggering action $\tau_i(t) = \epsilon$, the controller γ_i trivially satisfies the definition of $\hat{\rho}$-responsibility. Also, $\mathcal{X}_{i,c}$ being a robust controlled invariant set guarantees $u_i(t)$ exists whenever $x_i(t) \in \mathcal{X}_{i,c}$ and with this $u_i(t)$ the controller satisfies (6).

To show statement (2), we use induction on time. In the base case ($t = 0$), by assumption, all agents satisfy $x_i(0) \in \mathcal{X}_{i,c}$. Assume at time $t = k$, each agent's state $x_i(k)$ is in its corresponding set $\mathcal{X}_{i,c}$. The controller γ_i either produces (i) $\tau_i(k) = \epsilon$ or (ii) $\tau_i(k) \neq \epsilon$.

First, consider case (i). Since controller γ_i renders $\mathcal{H}_i$ self-safe, $u_i'(k) \in \mathcal{U}_i$ produced by it satisfies (6). With this choice of $u_i'(k)$, there are two possibilities for state evolution. If $\rho_i(k) = \emptyset$, the state x_i evolves with the first line of Eq. (3c) and we have $x_i(k+1) \in \mathcal{X}_{i,c}$ by (6). If $\rho_i(k) \neq \emptyset$, state x_i evolves with the second line of Eq. (3c), that is, $x_i(k+1) \in R_i(k, x_i(k), u_i'(k), \rho_i(k))$. Let $j \in \rho_i(k) \in \hat{\rho}_i^{(j)}(k)$. By Assumption 1, $\tau_j(k) \neq \epsilon$. By agent j being $\hat{\rho}^{(j)}$-responsible with $\tau_j(k) \neq \epsilon$, for any $S \in \hat{\rho}_i^{(j)}(k)$ with $j \in S$, and, in particular for $S = \rho_i(k)$, the first line of (7) is satisfied. Since $R_i(k, x_i(k), u_i'(k), \rho_i(k)) \subseteq R_i(k, x_i(k), \mathcal{U}_i, \rho_i(k))$ and $j \in \rho_i(k)$ was arbitrary, $x_i(k+1) \in \mathcal{X}_{i,c}$ follows.

Now, consider case (ii). By assumption, the controller γ_i produces $\tau_i^*(k) \neq \epsilon$ and $u_i^*(k)$ such that both conditions in (7) and condition (6) are satisfied. Then, if $\rho_i(k) = \emptyset$, the state x_i evolves with the first line of Eq. (3c) and we have $x_i(k+1) \in \mathcal{X}_{i,c}$ by (6). If $\rho_i(k) \neq \emptyset$, state x_i evolves with the second line of Eq. (3c), that is, $x_i(k+1) \in R_i(k, x_i(k), u_i^*(k), \rho_i(k))$. Let $j \in \rho_i(k) \in \hat{\rho}_i^{(j)}(k)$. If $j \neq i$, the reasoning in case (i) above holds. If $j = i \in \rho_i(k)$, by assumption, $u_i^*(k)$ also satisfies the second line of (7) for any $S \in \hat{\rho}_i^{(j)}(k)$ with $i \in S$, and in particular for $S = \rho_i(k)$. Therefore, $x_i(k+1) \in R_i(k, x_i(k), u_i^*(k), \rho_i(k)) \subseteq \mathcal{X}_{i,c}$. □

This theorem essentially says for ITHA, existence of controlled invariant sets for individual dynamics is a sufficient condition for ensuring global safety. However, this is not a necessary condition and our results do not apply to the cases where the only way to ensure safety is via triggering. The next result relates the self-safety and responsibility conditions to "not being at fault" as in [24] in the sense that if an agent's control policy is self-safe and $\hat{\rho}$-responsible, there exists controllers for the remaining agents such that the overall system stays safe.

Corollary 1. *Consider an inter-triggering hybrid automaton $\{\mathcal{H}_i\}_{i\in\mathcal{I}}$, an accompanying collection of sets $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$ that are robustly controlled invariant for respective Σ_i's in their respective safe sets $\{\mathcal{X}_{i,safe}\}_{i\in\mathcal{I}}$ and a collection $\{\hat{\rho}^{(i)}\}_{i\in\mathcal{I}}$ of resolution over-approximations. If some subset $\{\mathcal{H}_j\}_{j\in\mathcal{I}'}$ with $\mathcal{I}' \subset \mathcal{I}$ of agents*

have controllers γ_j which are both self-safe and $\hat{\rho}^{(j)}$-responsible on the sets $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$, then there exists controllers γ_i for all of the other agents $\{\mathcal{H}_i\}_{i\in\mathcal{I}\setminus\mathcal{I}'}$ such that the state trajectories corresponding to any $\{\gamma_i\}_{i\in\mathcal{I}}$-controlled execution of $\{\mathcal{H}_i\}_{i\in\mathcal{I}}$ beginning in $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$ always remain within these sets.

Proof. For each $i \in \mathcal{I} \setminus \mathcal{I}'$, consider the controller γ_i that produces an input $u_i(t) \in \{u \mid f(x_i(t), u, \mathcal{D}) \subseteq \mathcal{X}_{i,c}\}$, which exists by $\mathcal{X}_{i,c}$ being robust controlled invariant, and triggering action $\tau_i(t) = \epsilon$ for all time t for which $x_i(t) \in \mathcal{X}_{i,c}$. As shown in the proof of Theorem 1 statement (1), such γ_i is self-safe and $\hat{\rho}^{(i)}$-responsible on $\mathcal{X}_{i,c}$. Since γ_j for $j \in \mathcal{I}'$ are given to be self-safe and $\hat{\rho}^{(j)}$-responsible on $\mathcal{X}_{j,c}$, with the above choice of controllers for agents in $\mathcal{I} \setminus \mathcal{I}'$, all the controllers are self-safe and $\hat{\rho}$-responsible, which by statement (2) of Theorem 1 ensures safety of the executions. □

One can try to verify self-safety and responsibility for given sets $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$, controllers γ_i and resolution over approximations $\hat{\rho}$. Conditions (6) and (7) can also be used to synthesize controllers that render an ITHA self-safe and $\hat{\rho}$-responsible or to supervise existing controllers at run-time. The latter two are the use cases we demonstrate in Section 5 using robust controlled invariant sets for $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$. Given $\hat{\rho}$, the basic idea is to construct the set of all triggering actions and control inputs that together satisfy conditions (6) and (7). This set is always non-empty when $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$ are robust controlled invariant sets and it can be constructed at run-time. Then, for synthesis, a pair (u_j, τ_j) is picked from this set and for supervision, we check if the controller's u_j, τ_j is in this set or not. A few comments are in order as to what information, in general, is needed to construct this set, which also prescribes what observations should be included in $\mathcal{Y}_j$ to implement a controller γ_j constructed this way. In general, the states of all agents i, for which j appears in the sets in $\hat{\rho}_i(t)$ should be included in $\mathcal{Y}_j$. However, we note that the reset maps together with the collection $\{\mathcal{X}_{i,c}\}_{i\in\mathcal{I}}$ of sets in practice have more structure that can simplify checking for $\hat{\rho}$-reponsibility or the amount of observations needed. For instance, for the processor example, for all $S \subseteq \mathcal{I}$, $R_i(\cdot,\cdot,\cdot,S) \subseteq \mathcal{X}_{i,c}$ implies for all S' with $|S'| \leq |S|$, $R_i(\cdot,\cdot,\cdot,S') \subseteq \mathcal{X}_{i,c}$. In words, if the processor i is safe when recruited by a number of other processors, it will be safe when recruited by a smaller number of processors. This implies that it is enough to check the condition (7) only for the largest cardinality S containing j instead of all such sets. Similarly, for the highway example, for all $S \subseteq \mathcal{I}$, there is an $S^*_{t,i} \subseteq S$ such that $R_i(t,\cdot,\cdot,S^*_{t,i}) \subseteq \mathcal{X}_{i,c}$ implies $R_i(t,\cdot,\cdot,\tilde{S}) \subseteq \mathcal{X}_{i,c}$ for all non-empty $\tilde{S} \subseteq S$. This is because there is a "worst-case" lane switching leading to a "worst-case" reset. In a sense, it does not matter what switching actions an arbitrary agent takes; only agents close to agent j matter. Thus, an ego vehicle can reason over the set of agents switching lanes nearest to itself while still being able to guarantee safety. It is also worth remarking that when either R_i or $\mathcal{X}_{i,c}$ is not known exactly, an over-approximation of R_i and an under-approximation of $\mathcal{X}_{i,c}$ can be used in (7) while still guaranteeing overall safety of the ITHA per Theorem 1. Moreover, as discussed in the next section, $\hat{\rho}^{(i)}$ can be constructed on the fly, meaning $\hat{\rho}^{(i)}$ is only known up to $\hat{\rho}^{(i)}(t)$ at a given time t but this is enough to construct a controller that is self-safe and $\hat{\rho}^{(i)}$-responsible at time t.

4.3 Finding Resolution Over-Approximations

Both complexity and conservativeness can be exacerbated if the over-approximation $\hat{\rho}^{(i)}$ is "far" from the true ρ. The task of identifying proper over-approximations of ρ is thus a vitally important one. Insights into the structure of the problem can be used to generate good over-approximations.

We start this section with a relatively easy to compute, yet possibly conservative, over-approximation. Then, we define an order between agents through which they can communicate and substantially reduce conservatism. Along the way, we also discuss how these over-approximations look for our running examples.

4.3.1 Trivial Over-Approximations: If we assume each agent knows the resolution function ρ, a trivial over-approximation of the resolution function can be locally computed at each time step by considering every possible choice of triggering inputs for all other agents. In math, agent i computes the j^{th} component of a trivial over-approximation $\rho^{(i)}_j$ as:

$$\hat{\rho}^{(i)}_j(t) = \hat{\rho}^{(i)}_j(t, \tau_i(t)) = \left\{ \rho' \in 2^{\mathcal{I}} \;\middle|\; \begin{array}{l} \exists \tau_k \in \mathcal{T}_k \;\; \forall k \in \mathcal{I} \setminus \{j\} : \\ \rho' = \rho_j(t, \tau_1, \ldots, \tau_i(t), \ldots, \tau_{|\mathcal{I}|}) \end{array} \right\} \tag{8}$$

It can be easily shown that $\hat{\rho}^{(i)}$, components of which are constructed as above is an over-approximation of ρ.

Example 3 (Cont'd Example 1). *For the server farm in Fig. 1, we revisit the case where $\tau_2(t) = \epsilon$ and $\tau_4(t) = \{1, 6\}$, which leads to $\rho_1(t) = \{4\}$, regardless of the triggering actions of the remaining agents. Using the trivial over-approximation, processor 1 has $\hat{\rho}^{(1)}_1(t) = \{\emptyset, \{2\}, \{4\}, \{2, 4\}\}$, processor 2 has $\hat{\rho}^{(2)}_1(t) = \{\emptyset, \{4\}\}$ and processor 4 has $\hat{\rho}^{(4)}_1(t) = \{\{4\}, \{2, 4\}\}$. Note that no estimates depend on $\tau_1(t)$, as agent 1 cannot recruit itself.*

Example 4 (Cont'd Example 2). *Consider the trivial over-approximation from the perspective of the ego agent, in the case where $\tau_E(t) = \epsilon$: the only possible resets depend on if L_1 does or does not trigger a reset on E; that is, $\hat{\rho}^{(E)}_E(t) = \{\emptyset, \{L_1\}\}$. If instead $\tau_E(t) =$ left, it is more complicated: $\hat{\rho}^{(E)}_E(t) = \{\{E\}, \{E, L_3\}, \{E, L_1\}\}$. The first case occurs if neither L_1 nor L_3 changes to the center lane simultaneously, the second case occurs if L_3 changes to the center lane, regardless of the triggering action of L_1, and the last case occurs if L_1 makes a lane change and L_3 does not.*

4.3.2 Ordered Actions. In some situations, agents in an inter-triggering hybrid automaton $\{\mathcal{H}_i\}_{i\in\mathcal{I}}$ can communicate their planned actions with one another. If such communication is done in an "orderly" manner, it can allow agents to obtain much refined over-approximations as they no longer need to consider all possible actions of the other agents.

We again assume each agent knows the resolution function ρ. Moreover, we assume at each time that there is a total order $\succeq_t$ among the agents that all agents know and use to communicate their planned triggering inputs.[2]

[2]The assumption of $\succeq_t$ being a total order can be relaxed. In particular, agents can still get an over-approximation for a class of partial orders $\succeq_t$ for which the Hasse diagram of the partially ordered set $(\{\mathcal{H}_i\}_{i\in\mathcal{I}}, \succeq_t)$ is a rooted forest at each time, i.e., an agent does not receive information from two incomparable agents at a given time.

Given such an order, we propose Algorithm 1 for each agent to construct their resolution over-approximation at each time t. With abuse of notation and without loss of generality, we assume that the automaton $\{\mathcal{H}_i\}_{i\in\mathcal{I}}$ is (re)ordered/(re)indexed (by keeping track of their associated overestimates computed so far) at each time t so that $\{\mathcal{H}_i\}_{i\in\mathcal{I}} = \{\mathcal{H}_1, \mathcal{H}_2, \cdots, \mathcal{H}_{|\mathcal{I}|}\}$ and the index of each agent represents its ranking according to $\succeq_t$. Then, Algorithm 1 is called starting with $\mathcal{H}_1$ to compute resolution over-approximations $\hat{\rho}^{(1)}$ and to choose a triggering action $\tau_1(t)$ for which a self-safe and $\hat{\rho}^{(1)}$-responsible control input $u_1(t)$ exists. Then, $\tau_1(t)$ is shared with the next agent and agent $\mathcal{H}_2$ calls the algorithm with $\{\tau_1(t)\}$, and so on, until all agents compute their triggering actions for time t. Then these actions are executed and time progresses.

Algorithm 1: Resolution over-approximation construction with ordered actions

Result: $\hat{\rho}^{(j)}(t,\tau_j), \tau_j(t)$
Input: $\mathcal{H}_j$, $\{\tau_i(t)\}_{i=1}^{j-1}$, $\{x_i(t)\}_{i\in\mathcal{I}}$
$\widetilde{\mathcal{T}}_j \leftarrow \emptyset$
for $\tau_j \in \mathcal{T}_j$ **do**
  $\hat{\rho}^{(j)}(t,\tau_j) \leftarrow \emptyset$
  for $(\tau_{j+1}, \cdots, \tau_{|\mathcal{I}|}) \in \mathcal{T}_{j+1} \times \cdots \times \mathcal{T}_{|\mathcal{I}|}$ **do**
    $S \leftarrow \rho(t, \tau_1(t), \tau_2(t), \cdots, \tau_{j-1}(t), \tau_j, \cdots, \tau_{|\mathcal{I}|})$
    if $\forall u_j \in \mathcal{U}_j,\ R_j(t, x_j(t), u_j, S) \not\subseteq \mathcal{X}_{j,c}$ **then**
      continue;
    if $\exists i \neq j$ *s.t.* $R_i(t, x_i(t), \mathcal{U}_i, S) \not\subseteq \mathcal{X}_{i,c}$ **then**
      continue;
    $\hat{\rho}^{(j)}(t,\tau_j) \leftarrow \hat{\rho}^{(j)}(\tau_j) \cup \{S\}$
    $\widetilde{\mathcal{T}}_j \leftarrow \widetilde{\mathcal{T}}_j \cup \{\tau_j\}$;
$\tau_j(t) \in \widetilde{\mathcal{T}}_j$

This scheme, and particularly the method for constructing $\hat{\rho} = \{\hat{\rho}_i\}_{i\in\mathcal{I}}$ can be shown to produce an over-approximation.

Lemma 1. *Calling Algorithm 1 at each time step according to order $\succeq_t$ produces functions $\hat{\rho}^{(j)}$, each of which is an over-approximation of ρ at every time step.*

Proof. Note that $\rho(t) = \{\rho_i(t, \tau_1(t), \tau_2(t), \cdots, \tau_{|\mathcal{I}|}(t))\}_{i\in\mathcal{I}}$. Thus, when $\{\tau_i(t)\}_{i\in\mathcal{I}}$ is completely known, we explicitly know $\rho(t)$. At the $|\mathcal{I}|^{th}$ call of Algorithm 1 at time t, after $\tau_{|\mathcal{I}|}(t)$ is chosen then $\{\tau_i(t)\}_{i\in\mathcal{I}}$ is completely known. This indicates that $\rho(t) \in \hat{\rho}^{(|\mathcal{I}|)}(t)$ for all t.

Now, consider an arbitrary call k of the Algorithm 1 at time t. By definition, $\hat{\rho}_i^{(k)}(t)$ contains all resolutions $\rho_i(t, \tau_1(t), \tau_2(t), \cdots, \tau_{k-1}(t), \tau'_k, \tau'_{k+1}, \cdots, \tau'_{|\mathcal{I}|})$ where $\tau'_k, \tau'_{k+1}, \cdots$ are arbitrarily chosen. Similarly, $\hat{\rho}_i^{(k-1)}$ contains all resolutions $\rho_i(t, \tau_1(t), \tau_2(t), \cdots, \tau_{k-2}(t), \tau'_{k-1}, \tau'_k, \tau'_{k+1}, \cdots, \tau'_{|\mathcal{I}|})$ where $\tau'_{k-1}, \tau'_k, \cdots$ are arbitrarily chosen. By observation one can see that $\hat{\rho}_i^{(k-1)} \supseteq \hat{\rho}_i^{(k)}$. Therefore, by induction $\rho_i(t) \in \hat{\rho}_i^{(k)}$ for any k and any i. Therefore, the functions $\hat{\rho}^{(j)}$ produced by Algorithm 1 are an over-approximation of ρ at every time step. □

Example 5 (Order in Highway Example). *In the highway example, one can assume that any vehicle (e.g. vehicle E in Fig. 3) on the highway sees the actions of the vehicles in front of it (or on a slower lane when two agents are aligned) and can use those to inform its own lane change decisions. In this way, a time-varying ordering is implemented where $\mathcal{H}_i \succeq_t \mathcal{H}_j$ if and only if $\mathcal{H}_i$ is in front of $\mathcal{H}_j$ or they are aligned and $\mathcal{H}_j$ is on a slower lane compared to $\mathcal{H}_i$. This gives a total order across agents at each time t.*

For example, in Fig. 3 the ordering is $\mathcal{H}_{L_1} \succeq \mathcal{H}_{L_3} \succeq \mathcal{H}_E \succeq \mathcal{H}_{F_2}$. This choice is motivated by the intuition that the ego vehicle $\mathcal{H}_E$ is behind vehicles $\mathcal{H}_{L_1}$ and $\mathcal{H}_{L_3}$, so it can see their actions. As concrete examples of what Algorithm 1 outputs in this case, if $\tau_E(t) = \epsilon$, then $\hat{\rho}_E^{(E)}(t) = \{L_1\}$ when $\tau_{L_1}(t) \neq \epsilon$ and $\hat{\rho}_E^{(E)}(t) = \emptyset$ otherwise; that is, $\hat{\rho}_E^{(E)}(t)$ is not conservative, as it will observe the triggering action of L_1. Similarly, if $\tau_E(t) =$ left, $\hat{\rho}_E^{(E)}(t) = \{\rho_E(t)\}$, since E sees the triggering actions of both L_3 and L_1 and thus there is no conservativeness.

5 EXPERIMENTS

We evaluate the flexibility and applicability of ITHA by using it to perform single-agent control in the highway driving scenario (Section 5.1) and multi-agent control in the parallel processing scenario (Section 5.2). Finally, we evaluate the conservativeness of the ITHA-based responsibility-sensitive safety rules on a real highway driving data-set (Section 5.3). Our software implementation is published at [1].

5.1 Single-agent control: highway driving

We demonstrate ITHA on the highway driving scenario, as described in Example 2, where only the ego vehicle is controlled and seeks to remain safe and responsible with respect to the uncontrolled vehicles. The ego vehicle E seeks to track a nominal velocity $v_{\text{nom}} = 15m/s$, formally solving the following receding horizon control problem at each time-step:

$$
\begin{aligned}
\min_{\substack{x_E, u_E,\\ \tau_E(t_0)}} \ & \sum_{t=t_0+1}^{t_0+H} \|v(t) - v_{\text{nom}}\|_2^2 \\
\text{s.t.} \ & x_E(t+1) = f_E(x_E(t), u_E(t), \tilde{d}(t)), && t = t_0+2, \ldots, t_0+H-1 \\
& x_E(t_0+1) \in \mathcal{X}_{E,c} \\
& x_E(t_0+1) = f_E(x_E(t_0), u_E(t_0), \tilde{d}(t_0)), && \text{if } \tau_E(t_0) = \epsilon \\
& x_E(t_0+1) = R_E^l(t_0, x_E(t_0), u_E(t_0), \bigcup\{\hat{\rho}_j(t_0)\}), && \text{if } \tau_E(t_0) = l \\
& R_j^l(t_0, x_j(t_0), u_j(t_0), \bigcup\{\hat{\rho}_j(t_0)\}) \in \mathcal{X}_{j,c}, \\
& \quad \forall u_j(t_0) \in \mathcal{U}_j, \forall j \in \hat{\rho}_{-E}(t_0), && \text{if } \tau_E(t_0) = l \\
& x_E(t_0+1) = R_E^r(t_0, x_E(t_0), u_E(t_0), \bigcup\{\hat{\rho}_j(t_0)\}), && \text{if } \tau_E(t_0) = r \\
& R_j^r(t_0, x_j(t_0), u_j(t_0), \bigcup\{\hat{\rho}_j(t_0)\}) \in \mathcal{X}_{j,c}, \\
& \quad \forall u_j(t_0) \in \mathcal{U}_j, \forall j \in \hat{\rho}_{-E}(t_0), && \text{if } \tau_E(t_0) = r
\end{aligned} \tag{9}
$$

and executing $u_E(t_0)$, where the prediction horizon $H = 25$, the predicted disturbance $\tilde{d}(t) = 0$ if $t > t_0$ and $\tilde{d}(t) = -10$ if $t = t_0$, and the continuous dynamics $f_E(\cdot,\cdot,\cdot)$ are as in (5), where $\Delta t = 0.1$. Furthermore, $\mathcal{U}_i = [-10, 10]$ for all agents i, and we define $\hat{\rho}_{-E}(t) = \{i \in \mathcal{I} \mid \exists j \in \hat{\rho}_i(t), j \cap \{E\} \neq \emptyset\}$ as an over-approximation of the set of all agents that E can trigger at time t. We will shortly describe the specific $\hat{\rho}$ that we use in our experiments. Finally, we abuse notation to define $R_j^l(\cdot,\cdot,\cdot,\cdot)$ and $R_j^r(\cdot,\cdot,\cdot,\cdot)$ as functions which output the reset state upon making a left and right lane change, respectively.

To interpret (9), we note that the first constraint enforces the continuous dynamics from the second timestep onwards, and the second constraint enforces self-safety. The third constraint enforces

the continuous dynamics at the first timestep if no triggering action is taken, while the fourth and sixth constraints enforce an appropriate state reset if the ego agent performs a left or right lane change, respectively. Finally, the fifth and seventh constraints enforce that all agents that are triggered by the ego vehicle's lane change action can remain safe by applying any control input.

Note that (9) can be represented as a mixed integer quadratic program, where $\tau_E(t_0) \in \{\epsilon, \text{left}, \text{right}\}$ can be modeled with an integer decision variable $z \in \{0, 1, 2\}$ used within a big-M formulation [2] to determine the lane change choice. To improve performance, (9) only seeks to enforce self-safety and responsibility at the first time-step (the system remains safe as only the input from first time-step is executed; hence, only safe actions are applied).

The uncontrolled vehicles are simulated using the Intelligent Driver Model (IDM) [26]:

$$\begin{aligned} x_i(t+1) = \;& x_i(t) + \Delta t v_i(t) \\ v_i(t+1) = \;& v_i(t) + \Delta t a \left(1 - \left(\frac{v_i(t)}{\tilde{v}_i}\right)^{\delta} - \right. \\ & \left. \left(\frac{s_0 + v_i(t) T_H + v_i(t)(v_i(t) - v_i^{\text{lead}}(t))}{2\sqrt{ab}}\right)^2\right) \end{aligned} \tag{10}$$

with parameters $\delta = 4, s_0 = 5, T_H = 1.5, a = b = 10$ and randomly sampled nominal velocities $\tilde{v}_i$. Here, $v_i^{\text{lead}}(t)$ refers to the velocity of agent i's lead car. We execute for 500 time-steps, solving (9) at each time-step; see Fig. 5 for a visualization of an example execution.

To illustrate the impact of different over-approximations of ρ on conservativeness, we compare control performance under these $\hat{\rho}$:

(A) The trivial over-approximation (8)
(B) The ordered over-approximation described in Algorithm 1, with the time-varying ordering as described in Example 5

A video showing the behavior of the $\hat{\rho}$-responsible ego vehicle when using the two different over-approximations is available here: https://youtu.be/a5IULWQYVzM. Under over-approximation (A), the ego vehicle travels 351.9 meters, while it travels 415.3 meters under over-approximation (B), averaged over 25 random initializations of the uncontrolled vehicles. In all simulations for both over-approximations, we did not experience any unsafe behavior, as guaranteed by the theory. We note that the performance improvement of the second over-approximation is a result of it being less conservative than the first. For instance, consider the example in Fig. 5. The simulation remains the same for both over-approximations up until time $t = 86$. At time $t = 87$, the ego agent is unable to make an advantageous left lane change to lane 3 under over-approximation (A), because if the car in lane 4 also changes to lane 3 simultaneously, it would lead to a safety violation. However, under over-approximation (B), the ego agent can safely make that lane change because the ordering implies that the car in lane 4 observes and should yield to the triggering action of the ego vehicle. A similar event occurs at time $t = 201$: under (A), the ego vehicle cannot make an advantageous switch to lane 4, because if the car in lane 5 is to simultaneously switch to lane 4, it would result in safety violations. By the end, the ego vehicle travels 161 meters further under (B) than under (A) (Fig. 5, bottom).

Finally, we note that while (B) outperforms (A) on average, (A) can still possibly outperform (B) for specific assignments of the uncontrolled vehicles. This occurs because (9) is restricted to a one-step plan for triggering actions, so the ego vehicle can make extra lane changes under (B) that can cause it to get trapped behind a slow car without realizing that it can free itself using a long sequence of lane changes; planning triggering actions over a longer horizon would aid in escaping from these "local optima".

	Avg. no. accepted jobs	Avg. safety violations
ρ estimate (A)	268.6	0
ρ estimate (B)	262.2	147.64
ρ estimate (C)	302.72	0

Table 1: Parallel processor statistics, averaged over 25 runs.

Overall, this experiment suggests we can use ITHA-based controllers to control an agent in a multi-agent environment with safety guarantees under limited communication, and that conservativeness of $\hat{\rho}$ can affect control performance.

5.2 Multi-agent control: parallel processors

We demonstrate ITHA on the parallel processor scenario, as described in Example 1, where we control all agents (processors). Our task is to maximize the number of accepted jobs over a finite horizon. At each time-step, we indirectly achieve this in a decentralized, receding-horizon fashion by computing control inputs and triggering inputs individually for each agent, which greedily minimize the number of remaining jobs for that agent. Formally, for each agent i, we solve the following integer program at each time-step t:

$$\begin{aligned} \min_{u_i(t), \tau_i(t)} \quad & x_i(t+1) \\ \text{s.t.} \quad & x_i(t+1) = x_i(t) - u_i(t) + d_i(t) - \textstyle\sum_{j=1}^{|\mathcal{I}|} \tau_i^j(t) \\ & x_i(t+1) \in \mathcal{X}_{i,c} \\ & x_i(t+1) \geq 0 \\ & \tau_i^j(t) = 0, \; \forall j \notin out_{\mathcal{G}}(i) \\ & R_j(t, x_j(t), u_j(t), \textstyle\bigcup\{\hat{\rho}_j(t)\}) \in \mathcal{X}_{j,c}, \\ & \qquad \forall u_j(t) \in \mathcal{U}_j, \forall j : \tau_i^j(t) = 1 \end{aligned} \tag{11}$$

where $u_i(t) \in \{0, 1\}$ and $\tau_i(t) \in \{0, 1\}^{|\mathcal{I}|}$. Here, $|\mathcal{I}| = 10$, $n_{\text{throttle}} = 3$, and $n_{\text{overflow}} = 5$. We define an over-approximation of the set of agents that agent i can trigger at time t, $\hat{\rho}_{-i}(t)$, in the same way as in the highway example. Similar to the highway example, we will compare performance between three ρ estimates:

(A) The trivial over-approximation (8). Here, $\hat{\rho}_i(t) = 2^{\text{in}_{\mathcal{G}}(i)}$.
(B) An *under-approximation* $\rho_{-j}^{\text{bad}}(t) \subseteq \{i\}$, that is, when planning $\tau_i(t)$, agent i assumes no other agent will recruit j.
(C) The ordered over-approximation described in Algorithm 1, with a time-invariant priority order sorted by processor index, i.e. $\mathcal{H}_1 \succ \ldots \succ \mathcal{H}_{\mathcal{I}}$.

We simulate 25 runs, each over a horizon of 50 time-steps, and report the performance statistics in Table 1. In each run, we generate a random undirected connectivity graph $\mathcal{G}$, where an edge between agents i and j exists if a sample uniformly drawn from $[0, 1]$ is greater than or equal to 0.1. Disturbances $d_i(t)$ are also generated randomly. Note that using the ρ estimate (A) leads to conservative performance, since there is no communication; thus, for agent i to recruit agent j, it must guarantee that agent j can remain safe if the rest of agent j's neighbors also trigger it. This overall leads to few recruitment actions, and thus many jobs are

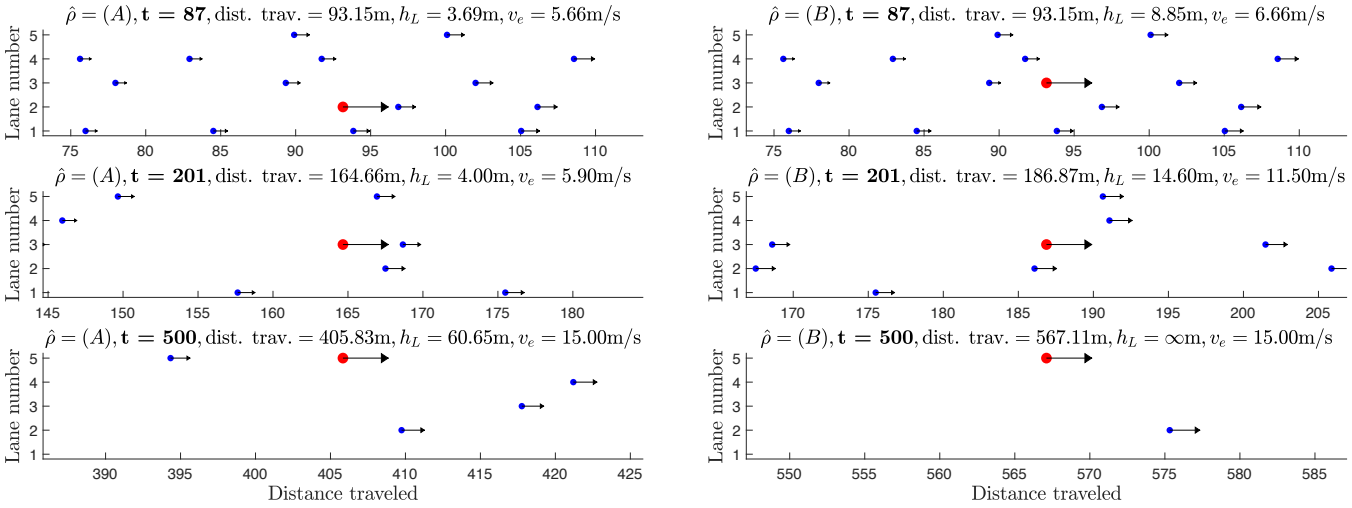

Figure 5: Highway driving example. *Red*: ego vehicle. *Blue*: uncontrolled vehicles. Arrow magnitudes are proportional to agent velocity. *Top row*: the ego vehicle can make an advantageous change to lane 3 under (B), but not under (A) due to a hypothetical simultaneous lane change from the lane 4 agent to lane 3. *Middle row*: the ego vehicle cannot make an advantageous lane change to lane 4 under (A) due to a hypothetical simultaneous lane change from the lane 5 agent to lane 4. *Bottom row*: At the end of the simulation, there is a large performance gap between using over-estimates (A) and (B).

rejected. On the other hand, (B) is less conservative, but as it is an unsafe estimate of ρ (since it is an under-approximation), safety violations can occur, such as when many other processors recruit one processor during the same time-step, leading to the recruited processor exceeding n_{overflow}. As a side effect, the average number of accepted jobs is also lower under (B) since many processors are often over the throttle limit, limiting the number of incoming jobs. With the ordered contract, we can avoid this mismanagement, more efficiently allocating jobs among the processors and preventing jobs from being unnecessarily rejected while remaining safe.

Overall, this experiment suggests that we can also use ITHA to control multiple agents in a decentralized fashion with safety guarantees, and that it is vital to select an appropriate ρ estimate to ensure safety and good performance.

5.3 Supervision: Evaluation of ITHA on data

Several frameworks for autonomous driving have sought to supervise a performance controller with a safety supervisor, which overrides when the performance controller may lead the system to an unsafe state. These supervisors often use invariant sets or control barrier functions to detect these safety violations. However, the usefulness of a safety supervisor is often dependent on its conservativeness, i.e. it should not unnecessarily override, as the jerkiness of changing controllers may annoy or frighten the user. To empirically demonstrate that the ITHA framework can serve as a high-quality safety supervisor that provides rigorous safety guarantees while remaining sufficiently permissive, we demonstrate that an ITHA-based supervisor achieves low override rates when supervising on a real world highway driving data-set [13].

The HighD data-set consists of trajectories of each driver's position with annotations, such as the vehicle lane and vehicle class (motorcycle, truck, or car), with data recorded at six different locations on the German Autobahn at various times of day. We use 110516 trajectories from the HighD data-set, containing a total of around 4×10^7 (x, u, d) data-point tuples. The state, input, and disturbance trajectories for each car in the data-set under the dynamics (5) (the v_i, h_i, v_i^L, and d_i trajectories) are generated as follows. v_i, h_i, and v_i^L are directly provided in the HighD data-set; we compute d_i via finite-differencing using the lead car velocities and $\Delta t = 0.04$ seconds, which is the provided time discretization of the data-set.

We hold out 20% of the data and use the remaining 80% as a "training set" to compute disturbance bounds. These disturbance bounds are used to compute invariant sets which are well-calibrated to the driving behavior observed in the data-set. Let the set of all disturbance trajectories in the training data-set for dynamics (5) be denoted $\Xi_d \doteq \{\xi_{d,i}\}_{i=1}^N$, where N is the number of trajectories in the data-set. While the data-set can be noisy, we do not perform any denoising in this step, and instead process outliers when computing the disturbance bounds. Specifically, we process Ξ_d for outliers by only keeping the data between the 0.025- and 0.975-sample quantiles $\hat{d}_{0.025}, \hat{d}_{0.975}$; that is, we concatenate Ξ_d, sort the result in increasing order (i.e. obtain the order statistics of Ξ_d, denoted $\Xi_{d,(1)}, \Xi_{d,(2)}, \ldots$), and remove all disturbances belonging in the first 2.5 and last 2.5 percent of Ξ_d (this is possible, since d is scalar in (5)). Formally, we define the y-sample quantile, $y \in (0, 1)$, as $\hat{d}_y = \Xi_{d,(\lceil yN \rceil)}$, where N is the number of elements in Ξ_d. Let this modified data-set be denoted $\hat{\Xi}_d \doteq \{d \in \Xi_d \mid d \in [\hat{d}_{0.025}, \hat{d}_{0.975}]\}$. We compute an invariant set assuming disturbances d satisfy $d \in \mathcal{D} = \mathcal{D}_i = [\hat{d}_{0.025}, \hat{d}_{0.975}]$, and use these invariant sets $\mathcal{X}_{i,c} = C_{\text{inv}}$ within an ITHA-based supervisor.

To quantify the conservativeness of using ITHA-based responsibility-sensitive safety rules to supervise highway driving,

Case	Override percentage
Self-safety	10.09%
Responsibility: Trivial $\hat{\rho}$	28.03%
Responsibility: Prioritized order $\hat{\rho}$	2.30%

Table 2: Override statistics for HighD data-set supervision.

we calculate the number of times our supervisor overrides the human control input on the trajectories observed in the data-set. Specifically, we calculate the fraction of datapoints in which self-safety (as defined in Definition 4) and $\hat{\rho}$-responsibility (as defined in Definition 6) are violated; we denote this the *override rate*, reported in Table 2. A low override rate indicates that our supervisor will not frequently engage and is not excessively conservative, which is desirable since there are no crashes in the data-set. As mentioned, we compare between two different $\hat{\rho}$ over-approximations to evaluate responsibility: the first uses the trivial over-approximation (8), while the second uses the ordering contract where any agent j behind agent i in the direction of travel must yield to the triggering actions of agent i (see Example 5 for more details).

Analyzing the override percentages in Table 2, we observe that supervising self-safety with ITHA results in relatively low override percentages, while the ordered contract outperforms the trivial contract substantially. This is to be expected, since using the trivial over-approximation leads to overrides being counted if the ego car is changing lanes and there exists another car adjacent to the new lane with similar longitudinal position as the ego car. This is common behavior (i.e. many cars may simultaneously be at similar longitudinal positions on the highway in different lanes). Note that these override rates can be further improved by employing context-dependent invariant sets generated with disturbance bounds computed on different clusters of data (contexts), i.e. only on trajectories recorded in the fast lane, or only on trajectories recorded at rush hour. Further investigation of the impact of context-dependence on the conservativeness of ITHA-based supervisor rules is an interesting direction for future work.

Overall, this experiment suggests that an ITHA-based supervisor can obtain low override rates on a real driving dataset, indicating that driving data-sets can be used to calibrate an ITHA-based safety supervisor and that such supervisors are permissive enough to avoid excessive overrides (10% for self-safety and 2% for an appropriate responsibility contract) and act as a useful safety supervisor.

6 DISCUSSION

In this section, we provide a few remarks on limitations and simple extensions of our framework:

- ITHA is appropriate in modeling systems whose individual dynamics are decoupled but have additional triggering actions for interaction. Our self-safety and responsibility rules utilize this structure to provide sufficient conditions for global safety. In comparison, existing compositional frameworks, such as [5, 9, 14, 17, 19, 22, 23, 25], that give sufficient conditions for global safety allow agents' dynamics to be coupled but do not allow for triggering actions.
- We note that our approach can be extended to guarantee safety for settings in which individual systems may have communication delays or sensor noise by leveraging recent advances in invariant set computation [10, 12, 15, 29] for systems with these imperfections.
- The increased complexity of our method over other responsibility-sensitive safety frameworks for driving (i.e. [7]) can be attributed in part to analyzing "second-order" triggers, i.e. reasoning about the set of agents which can have their feasible triggering set modified by the triggering action of another agent. In the two-lane highway driving setting, such behavior does not exist (which is what is considered in most existing responsibility-sensitive highway driving frameworks) since there are no "second-order" neighbors; however, to guarantee safety when there are more than two lanes of traffic, it is vital to consider second-order behavior.
- While we show communication and introducing an order for triggering action selection can reduce conservativeness, it can be further reduced by communicating control inputs. We assume that agents evaluate responsibility using all inputs u that the triggered agents can apply (see Eq. (7)); however, if agents can communicate their state and input to the triggered agent, we can relax this "for all inputs" condition to "there exists an input" according to a similar priority.
- Finally, we note our framework allows for priorities between agents which are not fixed a priori and dynamic orders can be chosen to improve performance. For instance, in the processor example, agents within the same connected component of the communication graph can mutually communicate their state, and agents with the most remaining jobs can be reassigned to have higher priority in recruiting other agents, as they are closer to the throttle threshold. So, there is a potential to employ distributed algorithms to select such dynamic orders.

7 CONCLUSIONS

In this paper, we introduce a novel modeling paradigm for multi-agent systems, the inter-triggering hybrid automaton, and apply it to two very different systems: parallel processing and autonomous driving. We derive an approach for proving safety of these systems using the notions of self-safety and $\hat{\rho}$-responsibility, which we show both theoretically and empirically result in guaranteed safe execution of the entire collection of agents. We also describe methods for generating practical approximations of the resolution function, and how local communication can be leveraged to improve these approximations. Finally, we demonstrate our approach on single- and multi-agent control in the parallel processing and highway driving scenarios, and furthermore evaluate the conservativeness of our approach on a safety supervisor task using real highway driving data. In future work, we wish to use assume-guarantee contracts between different agents in an ITHA to allow more coordination during resets. Also, we would like to extend our analysis to handle collections of agents whose triggering actions take multiple time steps to complete (i.e. non-instantaneous lane changes in the highway example) or have a delayed effect on the rest of the collection. Finally, we conjecture that ITHA is a special (less expressive) type of discrete-time hybrid I/O automaton [16] where each individual agent and the resolution function are hybrid I/O automata. This connection will be further investigated.

REFERENCES

[1] Compositional safety rules for inter-triggering hybrid automata (codeocean). https://doi.org/10.24433/CO.3247007.v1.

[2] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1st edition, 1997.

[3] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[4] K. Chatterjee, T. A. Henzinger, J. Otop, and A. Pavlogiannis. Distributed synthesis for ltl fragments. In *2013 Formal Methods in Computer-Aided Design*, pages 18–25. IEEE, 2013.

[5] Y. Chen, J. Anderson, K. Kalsi, S. H. Low, and A. D. Ames. Compositional set invariance in network systems with assume-guarantee contracts. In *2019 American Control Conference (ACC)*, pages 1027–1034. IEEE, 2019.

[6] E. Dallal and P. Tabuada. Decomposing controller synthesis for safety specifications. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 5720–5725. IEEE, 2016.

[7] R. De Iaco, S. L. Smith, and K. Czarnecki. Universally safe swerve manoeuvres for autonomous driving. *arXiv preprint arXiv:2001.11159*, 2020.

[8] B. Finkbeiner and S. Schewe. Uniform distributed synthesis. In *20th Annual IEEE Symposium on Logic in Computer Science (LICS'05)*, pages 321–330. IEEE, 2005.

[9] K. Ghasemi, S. Sadraddini, and C. Belta. Compositional synthesis via a convex parameterization of assume-guarantee contracts. In *HSCC '20: 23rd ACM International Conference on Hybrid Systems: Computation and Control, Sydney, New South Wales, Australia, April 21-24, 2020*, pages 16:1–16:10. ACM, 2020.

[10] T. Gurriet, P. Nilsson, A. Singletary, and A. D. Ames. Realizable set invariance conditions for cyber-physical systems. In *2019 American Control Conference (ACC)*, pages 3642–3649. IEEE, 2019.

[11] M. Hekmatnejad, S. Yaghoubi, A. Dokhanchi, H. B. Amor, A. Shrivastava, L. Karam, and G. Fainekos. Encoding and monitoring responsibility sensitive safety rules for automated vehicles in signal temporal logic. In *Proceedings of the 17th ACM-IEEE International Conference on Formal Methods and Models for System Design*, MEMOCODE '19, pages 6:1–6:11, New York, NY, USA, 2019. ACM.

[12] M. Jankovic. Control barrier functions for constrained control of linear systems with input delay. In *2018 Annual American Control Conference (ACC)*, pages 3316–3321. IEEE, 2018.

[13] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *ITSC*, 2018.

[14] L. Liebenwein, W. Schwarting, C.-I. Vasile, J. DeCastro, J. Alonso-Mora, S. Karaman, and D. Rus. Compositional and contract-based verification for autonomous driving on road networks. In *Robotics Research*, pages 163–181. Springer, 2020.

[15] Z. Liu, L. Yang, and N. Ozay. Scalable computation of controlled invariant sets for discrete-time linear systems with input delays. In *2020 American Control Conference, ACC 2020, Denver, CO, USA, July 1-3, 2020*, pages 4722–4728. IEEE, 2020.

[16] N. Lynch, R. Segala, and F. Vaandrager. Hybrid i/o automata. *Information and computation*, 185(1):105–157, 2003.

[17] P.-J. Meyer, A. Girard, and E. Witrant. Compositional abstraction and safety synthesis using overlapping symbolic models. *IEEE Transactions on Automatic Control*, 63(6):1835–1841, 2017.

[18] P. Nilsson, O. Hussien, A. Balkan, Y. Chen, A. D. Ames, J. W. Grizzle, N. Ozay, H. Peng, and P. Tabuada. Correct-by-construction adaptive cruise control: Two approaches. *TCST*, 24(4):1294–1307, 2016.

[19] P. Nilsson and N. Ozay. Synthesis of separable controlled invariant sets for modular local control design. In *2016 American Control Conference (ACC)*, pages 5656–5663. IEEE, 2016.

[20] N. Ozay. Inter-triggering hybrid automata: a formalism for responsibility-sensitive safety. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, pages 1–2, 2020. poster.

[21] A. Pneuli and R. Rosner. Distributed reactive systems are hard to synthesize. In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, pages 746–757. IEEE, 1990.

[22] S. Sadraddini and C. Belta. Distributed robust set-invariance for interconnected linear systems. In *2018 Annual American Control Conference (ACC)*, pages 1274–1279. IEEE, 2018.

[23] A. Saoud, A. Girard, and L. Fribourg. Contract-based design of symbolic controllers for safety in distributed multiperiodic sampled-data systems. *IEEE Transactions on Automatic Control*, 2020.

[24] S. Shalev-Shwartz, S. Shammah, and A. Shashua. On a formal model of safe and scalable self-driving cars, 2017.

[25] S. W. Smith, P. Nilsson, and N. Ozay. Interdependence quantification for compositional control synthesis with an application in vehicle safety systems. In *CDC*, pages 5700–5707. IEEE, 2016.

[26] M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, Aug 2000.

[27] S. Vaskov, S. Kousik, H. Larson, F. Bu, J. R. Ward, S. Worrall, M. Johnson-Roberson, and R. Vasudevan. Towards provably not-at-fault control of autonomous robots in arbitrary dynamic environments. In *Robotics: Science and Systems XV, University of Freiburg, Freiburg im Breisgau, Germany, June 22-26, 2019*, 2019.

[28] L. Wang, A. D. Ames, and M. Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674, June 2017.

[29] L. Yang and N. Ozay. Efficient safety control synthesis with imperfect state information. In *Conference on Decision and Control (CDC) 2020*. IEEE, 2020.

A DEFINING THE RESETS AND RESOLUTION FUNCTION FOR THE HIGHWAY EXAMPLE

In order to define the reset map and the resolution function for the highway example, we need to introduce global states in a global coordinate system for the overall system. Both the reset map and the resolution function can be expressed as time-invariant functions of these global states. On the other hand, we model the states of individual vehicles in ITHA in some local coordinates in (5), Global-state dependent reset maps and resolution functions can be converted to time-varying variants on the individual states. Since the self-safety and responsibility conditions are defined for time-varying reset maps and resolution functions that depend on individual states, our safety results are still applicable with this conversion. This section clarifies what information each vehicle would need at a given time to compute $\hat{\rho}$ and to comply with condition (7).

Let agent i's global state be defined as $\bar{x}_i = \begin{bmatrix} v_i & p_i & \ell_i \end{bmatrix}^\top \in \bar{\mathcal{X}}_a = [0, v_{max}] \times [0, \infty) \times \{1, ..., n_\ell\}$ where $\ell_i \in \{1, ..., n_\ell\}$ is the current lane's number, p_i is the longitudinal position in the current lane, and v_i is the current longitudinal velocity in the direction of the current lane (i.e. the same value in (5)). Here we take the state spaces $\bar{\mathcal{X}}_a$ of the vehicles to be identical for simplicity. Also, by convention, we take the rightmost lane to have value 1. Let $\{\bar{x}_i\}_{i=1}^{|\mathcal{I}|}$ be the collection of all agents' global states, with $\bar{\mathcal{X}}_g \triangleq \bar{\mathcal{X}}_a^{|\mathcal{I}|}$ denoting this state space, and $\bar{\mathcal{X}}_S$ denoting the restriction of this space to agents $S \subset \mathcal{I}$.

The dynamics of the global state can be written as:

$$\begin{aligned} \bar{x}_i(t+1) &= \begin{bmatrix} 1 & 0 & 0 \\ \Delta t & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bar{x}_i(t) + \begin{bmatrix} \Delta t \\ 0 \\ 0 \end{bmatrix} u_i(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_{\tau_i}(t), \\ &\triangleq \bar{f}_i(\bar{x}_i(t), u_i(t), u_{\tau_i}(t)), \end{aligned} \tag{12}$$

where u_i is the acceleration, i.e., the same control input in the ITHA representation, and

$$u_{\tau_i}(t) = \begin{cases} 1 & \tau_i(t) = \text{left}, \\ 0 & \tau_i(t) = \text{stay}, \\ -1 & \tau_i(t) = \text{right}. \end{cases}$$

Note that it is more intuitive to define control invariant sets over the local state space defined in (5) rather than in the global state space of the global coordinates since safety depends only on inter-vehicle distances. Thus, local reasoning is sufficient for the proofs of this paper. However, to define the resets, local states are not sufficient.

Let us define the following nearest leader of agent i operator $L_i : \bar{\mathcal{X}}_a \times \bigcup_{S \subset \mathcal{I}} \bar{\mathcal{X}}_S \to \mathcal{I} \cup \emptyset$. It is defined by the following simple optimization:

$$\begin{aligned} L_i(\bar{x}_i, \{\bar{x}_j\}_{j \in S}) = \quad & \underset{j \in \mathcal{S} \setminus \{i\}}{\arg\min} \quad |p_i - p_j| \\ & \text{subject to} \quad p_i \le p_j \\ & \qquad\qquad \ell_j = \ell_i \end{aligned} \tag{13}$$

In words, it finds the closest lead car within the set S to agent i on the same lane with it. Assuming that two cars cannot occupy the same point on the highway (in this case, the same longitudinal position and lane), the value of argmin will always be either a singleton or the empty set. However, even if it is not a singleton, picking any minimizer works for the purposes of the next lemma, which relates the global states to individual (local) ones.

Lemma 2. *There exists a unique mapping from the global state $\{\bar{x}_i\}_{i=1}^{|\mathcal{I}|}$ to the states $\{x_i\}_{i=1}^{|\mathcal{I}|}$ of the individual vehicles in the ITHA representation.*

Proof. To map the global state $\{\bar{x}_i\}_{i=1}^{|\mathcal{I}|}$ to the ITHA states $\{x_i\}_{i=1}^{|\mathcal{I}|}$, consider the mapping for one agent $\mathcal{H}_i$:

$$x_i = \begin{cases} \begin{bmatrix} v_i \\ \infty \\ v_{max} \end{bmatrix} & L_i = \emptyset \\ \begin{bmatrix} v_i \\ p_{L_i} - p_i \\ v_{L_i} \end{bmatrix} & \text{otherwise} \end{cases} \tag{14}$$

where L_i is the abbreviation of $L_i(\bar{x}_i, \{\bar{x}_j\}_{j \in \mathcal{I}})$. □

We define a localized version of the mapping (14), denoted $\mathcal{G}_S : \bar{\mathcal{X}}_S \to \prod_{i \in S} \mathcal{X}_i$, to be the mapping when L_i is taken to be $L_i(\bar{x}_i, \{\bar{x}_j\}_{j \in S})$. We denote by $\mathcal{G}_S^i : \bar{\mathcal{X}}_S \to \mathcal{X}_i$ the component of $\mathcal{G}_S$ corresponding to agent $\mathcal{H}_i$.

Now we will define the resolution functions $\rho_i : \bar{\mathcal{X}}_g \times \mathcal{T}_1 \times \cdots \times \mathcal{T}_{|\mathcal{I}|} \to 2^{\mathcal{I}}$ and reset maps $R_i : \bar{\mathcal{X}}_g \times \mathcal{X}_i \times \mathcal{U}_i \times 2^{\mathcal{I}} \to 2^{\mathcal{X}_i}$ that depend on the global quantities[3]. We have

$$\begin{aligned} \rho_i : (\{\bar{x}_i\}_{i=1}^{|\mathcal{I}|}, \tau_1, \cdots, \tau_{|\mathcal{I}|}) \mapsto \\ \{j \in \mathcal{I} \mid \tau_j \neq \text{stay}, \ell_j = \ell_i, p_j \ge p_i\} \cup \\ \{j \in \mathcal{I} \mid \tau_j \neq \text{stay}, \ell_j + u_{\tau_j} = \ell_i + u_{\tau_i}, p_j + v_j \Delta t \ge p_i + v_i \Delta t\}. \end{aligned}$$

In words, the resolution function ρ_i maps the global state and the triggering actions to the set of all agents that use a non-null triggering action that are on the same lane with agent i and ahead of it and those whose triggering action will put them on the same lane with agent i ahead of it in the next step. Then, the trivial resolution over-approximation $\hat{\rho}_i(t)$ used in our experiments considers, in the worst-case, all agents ahead of agent i on the same lane at time t and all agents that can be on the same lane with agent i ahead of it at time $t+1$. However, many of these combinations S of reset-triggering agents appearing as an output of $\hat{\rho}_i(t)$ lead to the same reset value, as defined by:

$$R_i(\{\bar{x}_i\}_{i=1}^{|\mathcal{I}|}, x_i, u_i, S) = \mathcal{G}_{S'}^i(\{\bar{f}_j(\bar{x}_j, u_j, u_{\tau_j})\}_{j \in S'}),$$

where $S' = \{j \in \mathcal{I} \mid \ell_j + u_{\tau_j} = \ell_i + u_{\tau_i}, p_j + v_j \Delta t \ge p_i + v_i \Delta t\}$. By construction, $\tau_j = \text{stay}$ for $j \in S' \setminus S$. Therefore, the triggering actions of the agents in $S' \cap S$ is sufficient for estimating R_i at time t, in addition to the estimates of other arguments. Note that the first set in the ρ_i definition does not directly seem to contribute to the reset map but it captures the agents leaving in front of agent i, which in turn affect who the lead car will be in the next step. At run-time, the local controller γ_i does not need the knowledge of the entire global states but needs to know the lanes and relative positions $(p_i(t) - p_j(t))$ and relative velocities $(v_i(t) - v_j(t))$ of agents j for which $i \in \hat{\rho}_j(t)$.

[3] To be precise, the actual value the state is reset to, depends on the triggering actions, states, and inputs of agents in ρ_i rather than the agents' indices as mentioned in footnote 1.