An STL-based Formulation of Resilience in Cyber-Physical Systems *

Hongkai Chen[1], Shan Lin[1], Scott A. Smolka[1], and Nicola Paoletti[2]

[1] Stony Brook University, Stony Brook, USA
{hongkai.chen,shan.x.lin}@stonybrook.edu
sas@cs.stonybrook.edu
[2] Royal Holloway, University of London, UK
nicola.paoletti@rhul.ac.uk

Abstract. Resiliency is the ability to quickly recover from a violation and avoid future violations for as long as possible. Such a property is of fundamental importance for Cyber-Physical Systems (CPS), and yet, to date, there is no widely agreed-upon formal treatment of CPS resiliency. We present an STL-based framework for reasoning about resiliency in CPS in which resiliency has a syntactic characterization in the form of an *STL-based Resiliency Specification* (SRS). Given an arbitrary STL formula φ, time bounds α and β, the SRS of φ, $R_{\alpha,\beta}(\varphi)$, is the STL formula $\neg\varphi\mathbf{U}_{[0,\alpha]}\mathbf{G}_{[0,\beta)}\varphi$, specifying that recovery from a violation of φ occur within time α (*recoverability*), and subsequently that φ be maintained for duration β (*durability*). These R-expressions, which are atoms in our SRS logic, can be combined using STL operators, allowing one to express composite resiliency specifications, e.g., multiple SRSs must hold simultaneously, or the system must eventually be resilient. We define a quantitative semantics for SRSs in the form of a *Resilience Satisfaction Value* (ReSV) function r and prove its soundness and completeness w.r.t. STL's Boolean semantics. The r-value for $R_{\alpha,\beta}(\varphi)$ atoms is a singleton set containing a pair quantifying recoverability and durability. The r-value for a composite SRS formula results in a set of non-dominated recoverability-durability pairs, given that the ReSVs of subformulas might not be directly comparable (e.g., one subformula has superior durability but worse recoverability than another). To the best of our knowledge, this is the first *multi-dimensional* quantitative semantics for an STL-based logic. Two case studies demonstrate the practical utility of our approach.

1 Introduction

Resiliency (*syn.* resilience) is defined as the ability to recover from or adjust easily to adversity or change [21]. Resiliency is of fundamental importance in Cyber-Physical Systems (CPS), which are expected to exhibit safety- or mission-critical behavior even in the presence of internal faults or external disturbances. Consider for example the *lane keeping* problem for autonomous vehicles (AVs), which requires a vehicle to stay within the marked boundaries of the lane it is driving in at all times. The standard temporal-logic-based notion of safety is not ideally suited for specifying the AV's behavior when it comes to lane keeping. This is because

* This is the full version of the paper under the same title accepted to FORMATS 2022.

AV technology is not perfect and driving conditions (e.g., being crowded by a neighboring vehicle) and other external disturbances may require occasional or even intermittent violations of lane keeping. Rather, the AV should behave resiliently in the presence of a lane violation, recovering from the violation in a timely fashion, and avoiding future lane departures for as long as possible. Unfortunately, there is no widely agreed notion of resiliency within the CPS community, despite several efforts to settle the issue (see Section 5).

Our Contributions. In this paper, we present an STL-based framework for reasoning about resiliency in Cyber-Physical Systems. In our approach, resiliency has a syntactic characterization in the form of an *STL-based Resiliency Specification* (SRS). Given an arbitrary STL formula φ, time bounds α and β, the SRS of φ, $R_{\alpha,\beta}(\varphi)$, is the STL formula $\neg\varphi\mathbf{U}_{[0,\alpha]}\mathbf{G}_{[0,\beta)}\varphi$, which specifies that recovery from a violation of φ occur within time α, and subsequently φ be maintained for duration β at least. The SRS of φ captures the requirement that a system quickly recovers from a violation of φ (*recoverability*) and then satisfy φ for an extended period of time (*durability*). The $R_{\alpha,\beta}(\varphi)$ expressions, which are atoms in our SRS logic, can be inductively combined using STL operators, allowing one to express composite resiliency specifications; e.g., multiple SRSs must hold simultaneously ($R_{\alpha_1,\beta_1}(\varphi_1) \wedge R_{\alpha_2,\beta_2}(\varphi_2)$), or that the system must eventually be resilient ($\mathbf{F}_I R_{\alpha,\beta}(\varphi)$).

We define a quantitative semantics for SRSs in the form of a *Resilience Satisfaction Value* (ReSV) function r. Our semantics for $R_{\alpha,\beta}(\varphi)$ atoms is a singleton set of the form $\{(rec, dur)\}$, where *rec* quantifies how early before bound α recovery occurs, and *dur* indicates for how long after bound β property φ is maintained. To the best of our knowledge, this is the first *multi-dimensional* quantitative semantics for STL.

Our approach does not make any simplifying assumption as to which of the two requirements (recoverability and durability) to prioritize or how to combine the two values. This decision can lead to a semantic structure involving two or more non-dominated (rec, dur) pairs. In such situations, we choose to retain *all* non-dominated pairs so as to provide a comprehensive, assumption-free, characterization of CPS resiliency. Thus, our semantics is a set of non-dominated (rec, dur) pairs, which is derived inductively from subformulas using Pareto optimization.

For example, consider the SRS $\psi_1 \vee \psi_2$, where the ReSV of ψ_1 (over a given signal at a particular time) is $\{(2,5)\}$ and, similarly, the ReSV of ψ_2 is $\{(3,3)\}$. The semantics of $\psi_1 \vee \psi_2$ should choose the dominant pair, but the two are non-dominated: $(3,3)$ has better recoverability, while $(2,5)$ has better durability. So we include both. We prove that our semantics is sound and complete with respect to the classic STL Boolean semantics by (essentially) showing that an SRS ψ has at least one non-dominated pair with rec, $dur > 0$ iff ψ is true.

We perform an extensive experimental evaluation of our framework centered around two case studies: UAV package delivery and multi-agent flocking. In both cases, we formulate mission requirements in STL, and evaluate their ReSV values in the context of various SRS specifications. Our results clearly demonstrate the expressive power of our framework.

2 Preliminaries

In this section, we introduce the syntax and semantics of Signal Temporal Logic (STL) [13, 18]. STL is a formal specification language for real-valued signals. We

consider n-dimensional discrete-time signals $\xi : \mathbb{T} \to \mathbb{R}^n$ where $\mathbb{T} = \mathbb{Z}_{\geq 0}$ is the (discrete) time domain.[3] $\mathbb{T}$ is the interval $[0, |\xi|]$, where $|\xi| > 0$ is the length of the signal. If $|\xi| < \infty$, we call ξ bounded. We use the words signal and trajectory interchangeably. An STL atomic predicate $p \in AP$ is defined over signals and is of the form $p \equiv \mu(\xi(t)) \geq c$, $t \in \mathbb{T}$, $c \in \mathbb{R}$, and $\mu : \mathbb{R}^n \to \mathbb{R}$. STL formulas φ are defined recursively according to the following grammar [13]:

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathbf{U}_I \varphi_2$$

where $\mathbf{U}$ is the *until* operator and I is an interval on $\mathbb{T}$. Logical disjunction is derived from $\wedge$ and $\neg$ as usual, and operators *eventually* and *always* are derived from $\mathbf{U}$ as usual: $\mathbf{F}_I\varphi = \top \mathbf{U}_I \varphi$ and $\mathbf{G}_I\varphi = \neg(\mathbf{F}_I \neg\varphi)$. The satisfaction relation $(\xi, t) \models \varphi$, indicating ξ satisfies φ at time t, is defined as follows:[4]

$$\begin{aligned}
&(\xi,t) \models p && \Leftrightarrow \quad \mu(\xi(t)) \geq c \\
&(\xi,t) \models \neg\varphi && \Leftrightarrow \quad \neg((\xi,t) \models \varphi) \\
&(\xi,t) \models \varphi_1 \wedge \varphi_2 && \Leftrightarrow \quad (\xi,t) \models \varphi_1 \wedge (\xi,t) \models \varphi_2 \\
&(\xi,t) \models \varphi_1 \mathbf{U}_I \varphi_2 && \Leftrightarrow \quad \exists\, t' \in t + I \text{ s.t. } (\xi,t') \models \varphi_2 \wedge \forall\, t'' \in [t,t'),\ (\xi,t'') \models \varphi_1
\end{aligned}$$

We call an STL formula φ *bounded-time* if all of its temporal operators are bounded (i.e., their intervals have finite upper bounds) and $|\xi|$ is large enough to determine satisfiability at time 0; i.e., $|\xi|$ is greater than the maximum over the sums of all the nested upper bounds on the temporal operators [29]. For example, if φ is $\varphi_1 \mathbf{U}_{[0,5]} \mathbf{G}_{[1,2]} \varphi_2 \wedge \mathbf{F}_{[0,10]} \mathbf{G}_{[1,6]} \varphi_2$, then a trajectory with length $N \geq \max(5+2, 10+6) = 16$ is sufficient to determine whether φ holds. In this paper, we only consider bounded-time STL formulas as in its original definition [18].

STL admits a quantitative semantics given by a real-valued function ρ such that $\rho(\varphi, \xi, t) > 0 \Rightarrow (\xi, t) \models \varphi$, and defined as follows [13]:

$$\begin{aligned}
\rho(\mu(\xi(t)) \geq c, \xi, t) &= \mu(\xi(t)) - c \\
\rho(\neg\varphi, \xi, t) &= -\rho(\varphi, \xi, t) \\
\rho(\varphi_1 \wedge \varphi_2, \xi, t) &= \min(\rho(\varphi_1, \xi, t), \rho(\varphi_2, \xi, t)) \\
\rho(\varphi_1 \mathbf{U}_I \varphi_2, \xi, t) &= \max_{t' \in t+I} \min(\rho(\varphi_2, \xi, t'), \min_{t'' \in [t, t+t')} \rho(\varphi_1, \xi, t''))
\end{aligned}$$

A ρ-value, called the *robustness satisfaction value* (RSV), can be interpreted as the extent to which ξ satisfies φ at time t. Its absolute value can be viewed as the distance of ξ from the set of trajectories satisfying or violating φ, with positive values indicating satisfaction and negative values indicating violation.

3 Specifying Resilience in STL

In this section, we introduce our STL-based resiliency specification formalism and its quantitative semantics in terms of non-dominated recoverability-durability pairs.

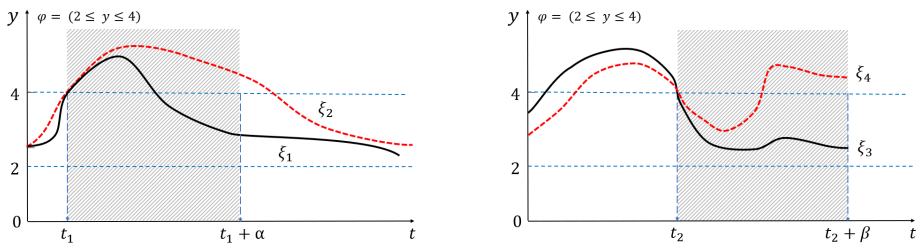
[3] Discrete-time signals over an arbitrary time step can always be mapped to signals over a unit time step.

[4] Given $t \in \mathbb{T}$ and interval I on $\mathbb{T}$, $t + I$ is used to denote the set $\{t + t' \mid t' \in I\}$.

3.1 Resiliency Specification Language

We introduce an STL-based temporal logic to reason about resiliency of STL formulas. Given an STL specification φ, there are two properties that characterize its resilience w.r.t. a signal ξ, namely, *recoverability* and *durability*: the ability to (1) recover from a violation of φ within time α, and (2) subsequently maintain φ for at least time β.

Example 1. Consider an STL specification $\varphi = (2 \leq y \leq 4)$, where y is a signal. In Figure 1(a), signals ξ_1 and ξ_2 violate φ at time t_1. Given recovery deadline α, we see that only ξ_1 satisfies recoverability of φ w.r.t. α because φ becomes true before $t_1 + \alpha$. In the case of ξ_2, φ becomes true only after $t_1 + \alpha$. In Figure 1(b), signals ξ_3 and ξ_4 recover to satisfy φ at time t_2. Given durability bound β, we observe that only ξ_3 is durable w.r.t. β.



(a) Signal ξ_1 satisfies recoverability. (b) Signal ξ_3 satisfies durability.

Fig. 1. Resilience w.r.t. an STL formula $\varphi = (2 \leq y \leq 4)$

Resilience of an STL formula φ relative to a signal should be determined by the joint satisfaction of recoverability and durability of φ. We can formalize this notion using the STL formula $R_{\alpha,\beta}(\varphi) \equiv \neg\varphi\mathbf{U}_{[0,\alpha]}\mathbf{G}_{[0,\beta)}\varphi$, which captures the requirement that the system recovers from a violation of φ within bound α and subsequently maintains φ for bound β. In what follows, we introduce SRS, an STL-based resiliency specification language that allows one to combine multiple $R_{\alpha,\beta}(\varphi)$ expressions using Boolean and temporal operators.

Definition 1 (STL-based Resiliency Specification). *The STL-based resiliency specification (SRS) language is defined by the following grammar:*

$$\psi ::= R_{\alpha,\beta}(\varphi) \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \mathbf{U}_I \psi_2$$

where φ is an STL formula, $R_{\alpha,\beta}(\varphi) \equiv \neg\varphi\mathbf{U}_{[0,\alpha]}\mathbf{G}_{[0,\beta)}\varphi$, $\alpha, \beta \in \mathbb{T}$, $\beta > 0$.

We use a semi-closed interval in the $\mathbf{G}$ operator, reflecting our requirement that φ stays true for a time interval of duration β, and may be false at the end of the interval. The piece-wise constant interpretation of discrete-time signals implies that properties between two consecutive steps remain unchanged (see Remark 5). SRS formulas are also restricted to bounded-time intervals as those in STL formulas. Boolean satisfiability of an SRS formula ψ reduces to the satisfiability of the corresponding STL formula obtained by replacing every atom $R_{\alpha,\beta}(\varphi)$ with $\neg\varphi\mathbf{U}_{[0,\alpha]}\mathbf{G}_{[0,\beta)}\varphi$. Note that satisfiability of an $R_{\alpha,\beta}(\varphi)$ atom involves satisfying the recoverability and durability requirements for φ highlighted in Example 1.

Remark 1 (Why a new logic? (1/2)). We note that SRS is equivalent to STL in the sense that any STL formula φ is equivalent to the SRS atom $R_{0,1}(\varphi)$ and, conversely, any SRS formula is an STL formula. *Then why would we introduce a new logic?* The SRS logic is explicitly designed to express specifications that combine resiliency requirements with Boolean and temporal operators. Most importantly, as we will see, our semantics for SRS is defined inductively starting from $R_{\alpha,\beta}(\varphi)$ atoms, and not from STL atomic predicates. The former are more expressive than the latter (an STL atomic predicate can be expressed via an SRS atom, but not vice versa).

Remark 2 (Resilience vs. safety, liveness, and stability). We stress that safety properties $\mathbf{G}_I\varphi$ are not sufficient to capture resilience, as they do not allow for occasional, short-lived violations. Among the liveness properties, reachability properties $\mathbf{F}_I\varphi$ do not capture the requirement that we want to satisfy φ in a durable manner. Similarly, progress properties, summarized by the template $\mathbf{G}_{I_1}(\neg\varphi \rightarrow \mathbf{F}_{I_2}\varphi)$, do not require the signal to satisfy φ for extended time periods (but they "infinitely" often lead to φ).

Thus, one might be tempted to combine safety and liveness and express resilience as a $\mathbf{F}_{[0,\alpha]}\mathbf{G}_{[0,\beta)}\varphi$, a template often called *stability* or stabilization [9]. There is a subtle but important difference between stability and our definition $\neg\varphi\mathbf{U}_{[0,\alpha]}\mathbf{G}_{[0,\beta)}\varphi$ of resilience: there could be multiple recovery episodes occurring in a trajectory, i.e., time steps where φ transitions from false to true. Stability is satisfied by *any* recovery episode within time $[0,\alpha]$ provided that φ stays true for at least time β. On the contrary, resilience is satisfied by only the *first* recovery episode, provided that φ is not violated for longer than α and stays true for at least β. This is an important difference because our resiliency semantics is defined as a recoverability-durability pair, roughly corresponding to the time it takes to recover from a violation of φ and the time for which φ subsequently remains true (see Definition 4). Since the stability pattern matches multiple recovery episodes, it is not clear which episode to use in the computation of our resiliency semantics. Our definition solves this ambiguity by considering the first episode. We remark that SRS allows us to express properties like $\mathbf{G}_I R_{\alpha,\beta}(\varphi)$ (or $\mathbf{F}_I R_{\alpha,\beta}(\varphi)$), which can be understood as enforcing the resiliency requirement $R_{\alpha,\beta}(\varphi)$ for each (for some) recovery episode within I, and whose semantics can be interpreted as the worst- (best-) case recovery episode within I.

Remark 3 (Why a new logic? (2/2)). The reader might wonder why we would be interested in using temporal operators to reason about a resiliency atom of the form $R_{\alpha,\beta}(\varphi)$ as opposed to pushing these operators into $R_{\alpha,\beta}(\varphi)$. For example, why would we consider (1) $\mathbf{G}_I R_{\alpha,\beta}(\varphi)$ instead of (2) $R_{\alpha,\beta}(\mathbf{G}_I\varphi)$ for an STL formula φ? The two expressions are fundamentally different: (1) states that the resiliency specification holds for all φ-related recovery episodes occurring in interval I; (2) states that the resiliency specification must hold in the first recovery episode relative to $\mathbf{G}_I\varphi$ (i.e., the first time $\mathbf{G}_I\varphi$ switches from false to true). Arguably, (1) is more useful than (2), even though both are reasonable SRS expressions (compound and atomic, respectively).

3.2 Semantics of Resiliency Specifications

We provide a quantitative semantics for SRS specifications in the form of a *resilience satisfaction value* (ReSV). Intuitively, an ReSV value quantifies the extent

to which recoverability and durability are satisfied. More precisely, it produces a non-dominated set of pairs $(x_r, x_d) \in \mathbb{Z}^2$, where (in the atomic case) x_r quantifies how early before bound α the system recovers, and x_d quantifies how long after bound β the property is maintained. We further demonstrate the soundness of the ReSV-based semantics w.r.t. the STL Boolean interpretation of resiliency specifications. The first step is to establish when one recoverability-durability pair is better than another.

A *set* $S \subseteq \mathbb{R}^n$ *of non-dominated tuples* is one where no two tuples x and y can be found in S such that x *Pareto-dominates* y, denoted by $x \succ y$. We have that $x \succ y$ if $x_i \geq y_i$, $1 \leq i \leq n$, and $x_i > y_i$ for at least one such i, under the usual ordering $>$.

We define a novel notion of "resilience dominance" captured by the relation $\succ_{re}$ in $\mathbb{Z}^2$. This is needed because using the standard Pareto-dominance relation $\succ$ (induced by the canonical $>$ order) would result in an ordering of ReSV pairs that is inconsistent with the Boolean satisfiability viewpoint. Consider the pairs $(-2, 3)$ and $(1, 1)$. By Pareto-dominance, $(-2, 3)$ and $(1, 1)$ are mutually non-dominated, but an ReSV of $(-2, 3)$ indicates that the system doesn't satisfy recoverability; namely it recovers two time units too late. On the other hand, an ReSV of $(1, 1)$ implies satisfaction of both recoverability and durability bounds, and thus should be preferred to $(-2, 3)$. We formalize this intuition next.

Definition 2 (Resiliency Binary Relations). *We define binary relations* $\succ_{re}$, $=_{re}$, *and* $\prec_{re}$ *in* $\mathbb{Z}^2$. *Let* $x, y \in \mathbb{Z}^2$ *with* $x = (x_r, x_d)$, $y = (y_r, y_d)$, *and* sign *is the signum function. We have that* $x \succ_{re} y$ *if either of the following conditions holds:*

1. $sign(x_r) + sign(x_d) > sign(y_r) + sign(y_d)$.
2. $sign(x_r) + sign(x_d) = sign(y_r) + sign(y_d)$, *and* $x \succ y$.

We say that x *and* y *are* mutually non-dominated, *denoted* $x =_{re} y$, *if* $sign(x_r) + sign(x_d) = sign(y_r) + sign(y_d)$ *and neither* $x \succ y$ *nor* $x \prec y$. *Under this ordering, a* non-dominated set S *is such that* $x =_{re} y$ *for every choice of* $x, y \in S$. *We denote by* $\prec_{re}$ *the dual of* $\succ_{re}$.

It is easy to see that $\succ_{re}$, $\prec_{re}$ and $=_{re}$ are mutually exclusive, and in particular, they collectively form a partition of $\mathbb{Z}^2 \times \mathbb{Z}^2$. This tells us that for any $x, y \in \mathbb{Z}^2$, either x *dominates* y (i.e., $x \succ_{re} y$), x *is dominated* by y (i.e., $x \prec_{re} y$), or the two are mutually non-dominated (i.e., $x =_{re} y$).

Lemma 1. *Relations* $\succ_{re}$ *and* $\prec_{re}$ *are strict partial orders.*

A proof can be found in Appendix C.

Definition 3 (Maximum and Minimum Resilience Sets). *Given* $P \subseteq \mathbb{Z}^2$, *with* $P \neq \emptyset$, *the* maximum resilience set *of* P, *denoted* $\max_{re}(P)$, *is the largest subset* $S \subseteq P$ *such that* $\forall x \in S$, $\forall y \in P$, $x \succ_{re} y$ *or* $x =_{re} y$. *The* minimum resilience set *of* P, *denoted* $\min_{re}(P)$, *is the largest subset* $S \subseteq P$ *such that* $\forall x \in S$, $\forall y \in P$, $x \prec_{re} y$ *or* $x =_{re} y$.

Corollary 1. *Maximum and minimum resilience sets are non-empty and non-dominated sets.*

A proof can be found in Appendix A.

Example 2. Let $P = \{(-1,2),(1,-2),(2,-1)\}$. Then, we have $\max_{re}(P) = \{(-1,2),(2,-1)\}$ because $(-1,2) =_{re} y$ for all $y \in P$ and $(2,-1) \succ_{re} (1,-2)$, $(2,-1) =_{re} (1,-2)$, and $(2,-1) =_{re} (2,-1)$. In contrast, $(1,-2)$ is not in $\max_{re}(P)$ because $(1,-2) \prec_{re} (2,-1)$. Similarly, we have $\min_{re}(P) = \{(-1,2),(1,-2)\}$. We also note that (as per Corollary 1) the elements of $\max_{re}(P)$ and $\min_{re}(P)$ are mutually non-dominated, i.e., $(-1,2) =_{re} (2,-1)$ and $(-1,2) =_{re} (1,-2)$, respectively.

Now we are ready to introduce the semantics for our SRS logic. Its definition makes use of maximum and minimum resilience sets in the same way as the traditional STL robustness semantics (see Section 2) uses max and min operators in compound formulas. Hence, by Corollary 1, our semantics produces non-dominated sets, which implies that all pairs in such a set are equivalent from a Boolean satisfiability standpoint. This is because $x_r > 0$ ($x_d > 0$) in Definition 2 implies Boolean satisfaction of the recoverability (durability) portion of an $R_{\alpha,\beta}(\varphi)$ expression. This property will be useful in Theorem 1, where we show that our semantics is sound with respect to the Boolean semantics of STL.

Definition 4 (Resilience Satisfaction Value). *Let ψ be an SRS specification and $\xi : \mathbb{T} \to \mathbb{R}^n$ a signal. We define $r(\psi,\xi,t) \subseteq \mathbb{Z}^2$, the* resilience satisfaction value *(ReSV) of ψ with respect to ξ at time t, as follows.*

- *For ψ an SRS atom of the form $R_{\alpha,\beta}(\varphi)$, φ an STL formula,*

$$r(\psi,\xi,t) = \{(-t_{rec}(\varphi,\xi,t) + \alpha, t_{dur}(\varphi,\xi,t) - \beta)\} \tag{1}$$

where

$$t_{rec}(\varphi,\xi,t) = \min\left(\{d \in \mathbb{T} \mid (\xi, t+d) \models \varphi\} \cup \{|\xi| - t\}\right) \tag{2}$$

$$t_{dur}(\varphi,\xi,t) = \min\left(\{d \in \mathbb{T} \mid (\xi, t'+d) \models \neg\varphi\} \cup \{|\xi| - t'\}\right), \tag{3}$$

$$t' = t + t_{rec}(\varphi,\xi,t)$$

- *The ReSV of a composite SRS formula is defined inductively as follows.*

$$\begin{aligned}
r(\neg\psi,\xi,t) &= \{(-x,-y) : (x,y) \in r(\psi,\xi,t)\} \\
r(\psi_1 \wedge \psi_2,\xi,t) &= \min_{re}(r(\psi_1,\xi,t) \ \cup \ r(\psi_2,\xi,t)) \\
r(\psi_1 \vee \psi_2,\xi,t) &= \max_{re}(r(\psi_1,\xi,t) \ \cup \ r(\psi_2,\xi,t)) \\
r(\mathbf{G}_I\psi,\xi,t) &= \min_{re}(\cup_{t' \in t+I}\, r(\psi,\xi,t')) \\
r(\mathbf{F}_I\psi,\xi,t) &= \max_{re}(\cup_{t' \in t+I}\, r(\psi,\xi,t')) \\
r(\psi_1 \mathbf{U}_I \psi_2,\xi,t) &= \max_{re} \cup_{t' \in t+I} \min_{re}(r(\psi_2,\xi,t') \cup \\
&\qquad \min_{re} \cup_{t'' \in [t,t+t')}\, r(\psi_1,\xi,t''))
\end{aligned}$$

In the base case, $t' = t + t_{rec}(\varphi,\xi,t)$ is the first time φ becomes true starting from and including t. If recovery does not occur along ξ (and so the first set in Eq. (2) is empty), then $t' = |\xi|$ (the length of the trajectory). Similarly, $t' + t_{dur}(\varphi,\xi,t)$ is the first time φ is violated after t'. Thus $t_{dur}(\varphi,\xi,t)$ quantifies the maximum time duration φ remains true after recovery at time t'. If φ is true for the entire duration of ξ, then $t_{rec}(\varphi,\xi,t) = 0$ and $t_{dur}(\varphi,\xi,t) = |\xi| - t$. If φ is false for the duration of ξ, then $t_{rec}(\varphi,\xi,t) = |\xi| - t$; therefore $t' = |\xi|$ and $t_{dur}(\varphi,\xi,t) = 0$.

Therefore, the semantics for an SRS atom $R_{\alpha,\beta}(\varphi)$ is a singleton set $\{(x_r, x_d)\}$, where x_r quantifies how early before time bound α recovery occurs, and x_d indicates for how long after time bound β the property is maintained. Thus, x_r and x_d quantifies the satisfaction extent (in time) of recoverability and durability, respectively. An important property follows from this observation: similar to traditional STL robustness, a positive (pair-wise) ReSV value indicates satisfaction of recoverability or durability, a negative ReSV value indicates violation, and larger ReSV values indicate better resiliency, i.e., shorter recovery times and longer durability.

The ReSV semantics for composite SRS formulas is derived by computing sets of maximum/minimum recoverability-durability pairs in a similar fashion to STL robustness: for the $\wedge$ and $\mathbf{G}$ operators, we consider the minimum set over the ReSV pairs resulting from the semantics of the subformulas; for $\vee$ and $\mathbf{F}$, we consider the maximum set. We remark that our semantics induces sets of pairs (rather than unique values); our $\succ_{re}$ and $\prec_{re}$ relations used to compute maximum and minimum resilience sets are therefore partial in nature. This is to be expected because any reasonable ordering on multi-dimensional data is partial by nature. For example, given pairs $(2, 5)$ and $(3, 3)$, there is no way to establish which pair dominates the other: the two are indeed non-dominated and, in particular, the first pair has better durability but worse recoverability than the second pair. In such a situation, our semantics would retain both pairs.

Algorithm for computing the ReSV function r. The algorithm to compute r is a faithful implementation of Definition 4. It takes an SRS formula ψ, signal ξ, and time t, produces a syntax tree representing ψ, and computes the r-values of the subformulas of ψ in a bottom-up fashion starting with the SRS atomic R-expressions at the leaves. Each leaf-node computation amounts to evaluating satisfaction of the corresponding STL formula. The complexity of this operation is $O(|\xi|^{2l})$, for a trajectory ξ and an STL formula with at most l nested *until* operators [13, 16]. Let m be the number of R-expressions in ψ. Given that we need to evaluate the R-expressions at each time point along ξ, the time needed to compute the r-values of all SRS atoms in ψ is $O(m\,|\xi|^{2l+1})$.

Every node v of the tree has an associated result set P_v of (*rec*, *dur*) pairs. When v is an interior node of the tree, P_v is determined, using Definition 4, by computing the maximum or minimum resilience set of the result set(s) of v's children. The complexity for computing P_v can thus be shown to be quadratic in the size of its input. In particular, the size of the input at the root of ψ's syntax tree is bounded by $O(|\xi|^{2L})$, where L is the maximum number of nested *until* operators in ψ.[5] Furthermore, the ReSV of the *until* operator can be computed in a manner similar to how STL robustness is computed for the *until* operator. Therefore, the complexity of computing the root node's ReSV is $O((|\xi|^{2L})^2 + |\xi|^{2L})$. Consequently, the total complexity of computing the ReSV of ψ is $O(m\,|\xi|^{2l+1} + |\xi|^{4L})$.

Remark 4 (Bounded-time SRS formulas). We say that an SRS formula is *bounded-time* if all of its temporal operators are bounded and the STL formulas serving as SRS atoms are bounded-time. When a bounded signal ξ is not long enough to

[5] This is because the size of an interior node v's input is bounded (in the case of the *until* operator) by $|\xi|^2$ times the sum of the sizes of the result sets of v's children. The size of the root node's input is thus $O(|\xi|^{2L})$.

evaluate a bounded-time formula, we extend it to the required length simply by repeating its terminal value $\xi(T)$. For example, let $\varphi = \mathbf{G}_{[0,10]}$ $(2 \leq y \leq 4)$ and consider ξ_3 from Figure 1(b), where $|\xi_3| = t_2 + \beta$. We extend $|\xi_3|$ to $t_2 + \beta + 10$, the required length to determine if $(\xi_3, t + \beta) \models \varphi$ (it does). Note that the signal extension might result in an overestimate of recoverability or durability.

Remark 5 (Relation to time robustness). A notion of (right) time robustness of an STL atomic proposition p on a trajectory ξ at time t is given by [13, 26]:

$$\theta^+(p, \xi, t) = \chi(p, \xi, t) \cdot \max\{d \geq 0 \; s.t. \; \forall t' \in [t, t + d], \chi(p, \xi, t') = \chi(p, \xi, t)\}$$

where $\chi(p, \xi, t) = +1$ if $(\xi, t) \models p$, and -1 otherwise. Intuitively, $|\theta^+(p, \xi, t)|$ measures how long after t property p (or $\neg p$) remains satisfied. One might be tempted to use θ^+ in our definition $t_{rec}(\varphi, \xi, t)$ by setting it to $\max\{0, \theta^+(\neg\varphi, \xi, t)\}$, i.e., the maximum time duration for which φ is violated (or 0 if φ holds at t). This, however, implies that $t' = t + t_{rec}(\varphi, \xi, t)$ now represents the last time point for which φ is false. In our definition, we want instead t' to be the first time point φ becomes true. This difference is important, especially for discrete-time signals where the distance between two consecutive time points is non-negligible.

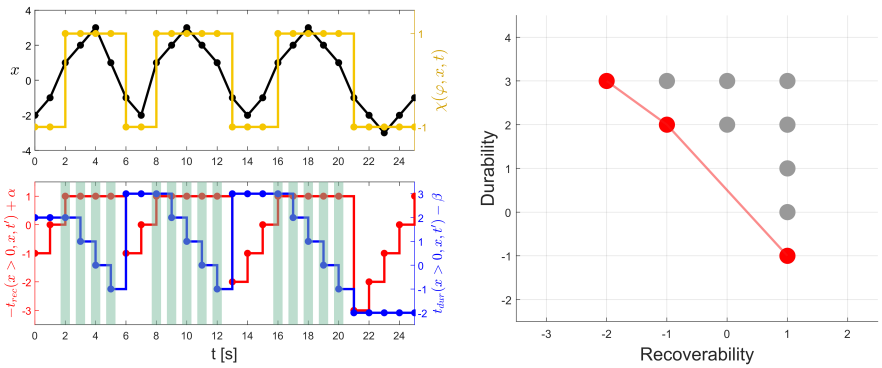
Moreover, time robustness may not quite handle some common corner cases. Consider a proposition p, and two signals ξ_1 and ξ_2 such that $(\xi_1, t) \models p$ and $(\xi_1, t') \not\models p$, $t' > t$, and $(\xi_2, t) \not\models p$ and $(\xi_2, t') \models p$, $t' > t$. The two signals have opposite behaviors in terms of satisfying p. In discrete-time settings (where a discrete-time signal is interpreted in the continuous domain as a piece-wise constant function), we have that ξ_1 (ξ_2) satisfies p ($\neg p$) throughout the interval $[t, t+1)$ (i.e., for "almost" 1 time unit). However, time robustness cannot distinguish between the two signals, namely, $\theta^+(p, \xi_1, t) = \theta^+(p, \xi_2, t) = 0$. Thus, if we used θ^+ to define t_{rec} it would be impossible to disambiguate between the first case (where no violation occurs at t) and the second case (where a violation occurs at t followed by a recovery episode at the next step). Our definition of t_{rec} in Eq. (2) correctly assign a value of 0 to ξ_1 (p is already satisfied, no recovery at t) and a value of 1 to ξ_2 (from t, it takes 1 time unit for p to become true).

Remark 6. We focus on discrete-times signals, thus a discrete-time SRS framework and discrete-time ReSV semantics. However, we expect that our approach can be extended to continuous time in a straightforward manner, because STL is well-defined over continuous-time signals. We leave this extension for future work.

Example 3. In Figure 2, we consider a (one-dimensional) signal x (Figure 2(a) top) and the proposition $x > 0$ (whose Boolean satisfaction value w.r.t. x is plotted on top of x). Consider the SRS formula $\psi_1 = \mathbf{G}_{[0,20]} R_{\alpha,\beta}(x > 0)$ with $\alpha = 1$, $\beta = 2$. Following Definition 4, the ReSV of ψ_1 can be written as:

$$r(\psi_1, x, 0) = \min_{re} \left(\cup_{t' \in [0,20]} r(R_{\alpha,\beta}(x > 0), x, t') \right)$$

where $r(R_{\alpha,\beta}(x > 0), x, t') = \{(-t_{rec}(x > 0, x, t') + \alpha, t_{dur}(x > 0, x, t') - \beta)\}$. Figure 2(a) bottom shows how the $-t_{rec}$ and the t_{dur} values evolve over time. These values are also displayed in Figure 2(b) on a recoverability-durability plane, to better identify the Pareto-optimal values that constitute the ReSV of ψ_1, i.e., the elements of the minimum resilience set in the RHS of the above equation. This



(a) (b)

Fig. 2. (a) Signal x and its Boolean semantics w.r.t. $x > 0$ (top). Recoverability (red) and durability (blue) values for $x > 0$ over signal x (bottom). Green bars indicate $x > 0$. (b) Red dots represent the Pareto front for the recoverability-durability pairs of $R_{\alpha,\beta}(x > 0)$ over time interval $[0, 20]$.

is equal to $\{(-1, 2), (1, -1), (-2, 3)\}$, representing the recoverability and durability values of x w.r.t. $R_{\alpha,\beta}(x > 0)$ at times $t' = 0, 5, 13$.

On the other hand, we obtain a different ReSV if we consider the SRS formula $\psi_2 = R_{\alpha,\beta}(\mathbf{G}_{[0,20]}x > 0)$, where the $\mathbf{G}$ temporal operator is pushed inside the resiliency atom. Since x never satisfies $\mathbf{G}_{[0,20]}x > 0$ (thus, it violates the property at time 0 and never recovers), we have $t_{rec}(\mathbf{G}_{[0,20]}x > 0, x, 0) = |x| = 25$ and $t_{dur}(\mathbf{G}_{[0,20]}x > 0, x, 0) = 0$, resulting in $r(\psi_2) = \{(-24, -2)\}$.

Proposition 1. *The ReSV $r(\psi, \xi, t)$ of an SRS formula ψ w.r.t. a signal ξ at time t is a non-dominated set.*

A simple proof is given in Appendix B, showing that any ReSV is either a maximum or minimum resilience set and thus, a non-dominated set as per Corollary 1. We are now ready to state the main theorem of our work, which establishes the soundness and completeness of SRS logic; i.e., our semantics is consistent from a Boolean satisfiability standpoint.

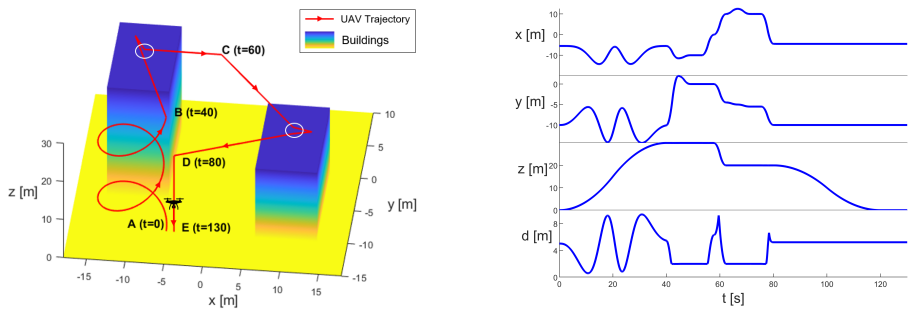
Theorem 1 (Soundness and Completeness of SRS Semantics). *Let ξ be a signal and ψ an SRS specification. The following results at time t hold:*
1) $\exists\, x \in r(\psi, \xi, t)$ *s.t.* $x \succ_{re} \mathbf{0} \implies (\xi, t) \models \psi$
2) $\exists\, x \in r(\psi, \xi, t)$ *s.t.* $x \prec_{re} \mathbf{0} \implies (\xi, t) \models \neg\psi$
3) $(\xi, t) \models \psi \implies \exists\, x \in r(\psi, \xi, t)$ *s.t.* $x \succ_{re} \mathbf{0}$ *or* $x =_{re} \mathbf{0}$
4) $(\xi, t) \models \neg\psi \implies \exists\, x \in r(\psi, \xi, t)$ *s.t.* $x \prec_{re} \mathbf{0}$ *or* $x =_{re} \mathbf{0}$

A proof is given in Appendix D. Since every ReSV $r(\psi, \xi, t)$ is a non-dominated set, then $\exists\, x \in r(\psi, \xi, t)$ s.t. $x \succ_{re} \mathbf{0}$ implies that $x \succ_{re} \mathbf{0}$ holds for all $x \in r(\psi, \xi, t)$ (see the proof of Theorem 1 for further details).

4 Case Study

In this section, we demonstrate the utility of our SRS logic and ReSV semantics on two case studies. We use Breach [12] for formulating STL formulas and evaluating their STL Boolean semantics. Experiments were performed on an Intel Core i7-8750H CPU @ 2.20GHz with 16GB of RAM and Windows 10 operating system. Our resiliency framework has been implemented in MATLAB; our implementation along with our case studies can be found in a publicly-available library.[6]

4.1 UAV Package Delivery



(a) (b)

Fig. 3. (a) UAV trajectory and layout of buildings. (b) UAV coordinates and distance to closest building.

Our first case study involves package delivery via an unmanned aerial vehicle (UAV). We consider a quadrotor UAV model with a proportional–integral–derivative (PID) controller [22] and use $\mathbf{p} = (x, y, z)$ to denote the UAV's 3-D position vector. The UAV is tasked with dropping off packages on the rooftops of two adjacent buildings; a simulated trajectory of the UAV along with various points of interest along the trajectory are given in Figure 3(a). As can be seen, the segment of the trajectory from A to B is spiral-like/unstable, and can be attributed to the disturbance caused by the load it is carrying (a disturbance that dissipates once the UAV reaches the building's rooftop). The coordinates of the UAV and its distance to the closest building are plotted over time in Figure 3(b). The (discrete) trajectory is 130 seconds long, i.e., $T = 130$ secs, with a time-step of $1/20$ secs. We use the following STL formulas to specify the UAV's mission.

Height regulation: the UAV should remain below $H_{max} = 120$ meters [1].

$$\varphi_1 = \mathbf{G}_{[0,T]}\, \varphi_1', \qquad \varphi_1' = (z \leq H_{max})$$

Package delivery: the UAV needs to hover at a delivery location for a specified period of time (delivery locations are closed $\mathcal{L}^2$-balls with radius $\epsilon = 1$ centered at C_1, C_2). The locations are set to $C_1 = (-10, 0, 30)$, $C_2 = (10, -5, 20)$, and $||\cdot||$ denotes the $\mathcal{L}^2$-norm.

$$\varphi_2 = \mathbf{F}_{[0,43]}\mathbf{G}_{[0,1]}\, \varphi_2', \qquad \varphi_2' = (||\mathbf{p} - C_1|| \leq \epsilon)$$
$$\varphi_3 = \mathbf{F}_{[0,65]}\mathbf{G}_{[0,3]}\, \varphi_3', \qquad \varphi_3' = (||\mathbf{p} - C_2|| \leq \epsilon)$$

[6] See https://github.com/hongkaichensbu/resiliency-specs

Collision avoidance: the UAV should maintain a minimum distance $d_{min} = 1.5$ meters to the closest building. It is violated repeatedly in the spiral ascent from A to B.

$$\varphi_4 = \mathbf{G}_{[0,T]}\, \varphi_4', \qquad \varphi_4' = (d \geq d_{min})$$

We compute the ReSV values of the above STL formulas when both φ_i and φ_i' are used as SRS atoms. In particular, in Table 1, we consider expressions of the form $R_{\alpha,\beta}(\varphi_i)$, i.e., where the temporal operators appear inside the SRS atoms, while in Table 2, nested φ_i' expressions are replaced by $R_{\alpha,\beta}(\varphi_i')$ expressions in φ_i, i.e., temporal operators appear outside the SRS atoms. Syntactically, this difference is subtle, but as we show, it is of significant importance semantically. We assume the UAV state is unchanged after the trajectory ends, when a longer trajectory is needed to determine satisfiability (see Remark 4). We choose $\alpha, \beta = 4$ for the SRS atoms so that we can illustrate repeated property violation and recovery.

SRS formula	$r(\psi_i, \xi, 0)$	Exec. time (sec)
$\psi_1 = R_{4,4}(\varphi_1)$	{(4, 126)}	14.69
$\psi_2 = R_{4,4}(\varphi_2)$	{(0.1, 45.5)}	12.74
$\psi_3 = R_{4,4}(\varphi_3)$	{(-0.2, 65.05)}	12.53
$\psi_4 = R_{4,4}(\varphi_4)$	{(-73.3, 48.7)}	14.18

Table 1. SRS expressions of the form $R_{\alpha,\beta}(\varphi_i)$ for UAV properties φ_i. All r-values are computed using trajectory ξ of Figure 3(a) at time 0.

SRS formula	$r(\psi_i', \xi, 0)$	Corresponding SRS atoms	Exec. time (sec)
$\psi_1' = \mathbf{G}_{[0,T]} R_{4,4}(\varphi_1')$	{(4, -4)}	$r(R_{4,4}(\varphi_1'), \xi, 130)$	8.50
$\psi_2' = \mathbf{F}_{[0,43]}\mathbf{G}_{[0,1]} R_{4,4}(\varphi_2')$	{(3.95, -3.5), (4, -3.55)}	$r(R_{4,4}(\varphi_2'), \xi, 42.85)$, $r(R_{4,4}(\varphi_2'), \xi, 42.95)$	18.84
$\psi_3' = \mathbf{F}_{[0,65]}\mathbf{G}_{[0,3]} R_{4,4}(\varphi_3')$	{(3.95, -1.35), (4, -1.4), (-0.2, 3.05)}	$r(R_{4,4}(\varphi_3'), \xi, 61.5)$, $r(R_{4,4}(\varphi_3'), \xi, 61.6)$, $r(R_{4,4}(\varphi_3'), \xi, 65)$	26.70
$\psi_4' = \mathbf{G}_{[0,T]} R_{4,4}(\varphi_4')$	{(-0.25, 5.8), (4,-4)}	$r(R_{4,4}(\varphi_4'), \xi, 8.4)$, $r(R_{4,4}(\varphi_4'), \xi, 130)$	8.93

Table 2. Nested φ_i' expressions replaced by $R_{\alpha,\beta}(\varphi_i')$ expressions in UAV properties φ_i.

In Table 1, $r(\psi_1, \xi, 0) = \{(4, 126)\}$ reflects the UAV's resilience w.r.t. φ_1 (height regulation) as it holds in $[0, T]$, thus reaching its maximum *rec* and *dur*. Entry $r(\psi_3, \xi, 0) = \{(-0.2, 65.05)\}$ considers the resilience of φ_3: it is false at time 0 but becomes true at time 4.2, making recovery 0.2 secs slower than $\alpha = 4$; φ_3 then remains true until time 73.25, resulting a durational period 65.05 secs longer than $\beta = 4$.

Table 2 includes an extra column (the third one) showing the SRS atoms corresponding to each (*rec*, *dur*) pair in the formula's ReSV (second column). In the first row, the ReSV of ψ_1' is $r(\psi_1', \xi, 0) = \{(4, -4)\}$. Because of the outermost $\mathbf{G}_{[0,T]}$ operator in ψ_1', $(4, -4)$ represents the worst-case recovery episode (relative to the STL property φ_1') within the interval $[0, T]$, meaning that $(4, -4)$ is dominated by every other (rec, dur) pair in $[0, T]$. The third column tells us that such episode happens at time $t = 130$. The (rec, dur) values in entry $r(\psi_3', \xi, 0) = \{(3.95, -1.35), (4, -1.4), (-0.2, 3.05)\}$ represent the best-case episodes (due to the outermost $\mathbf{F}$ operator) within $t \in [0, 65]$ of the ReSV of

$\mathbf{G}_{[0,3]} R_{4,4}(\varphi'_3)$, which in turn, gives us the worst-case episodes (due to the inner $\mathbf{G}$ operator) of the ReSV of $R_{4,4}(\varphi'_3)$ within $[t, t+3]$.

Even though there are some (relatively small) negative values in Table 2, overall, the results of Table 2 are consistent with those of Table 1, thereby reflecting the overall resiliency of the UAV package-delivery mission. Let us remark the difference between the results of Table 1 and 2, i.e., between the ReSVs of ψ_i and ψ'_i. The former are SRS atoms relative to a composite STL formula φ_i, and so their ReSVs are singleton sets representing the first recovery episode of φ_i. The latter are composite SRS formulas relative to an atomic STL formula φ'_i. As such, their ReSVs are obtained inductively following the structure of the SRS formulas and thus, they may include multiple non-dominated pairs.

We observe that execution times are largely affected by the size of the intervals in the temporal operators appearing in the SRS and STL formulas: computing ψ_2 and ψ_3 in Table 1 (ψ'_2 and ψ'_3 in Table 2) is more efficient than computing ψ_1 and ψ_4 (ψ'_1 and ψ'_4), even though the former expressions involve nested temporal operators. Indeed, the interval size directly affects both the number of subformula evaluations and the size of the ReSV set. Moreover, our ReSV algorithm uses an implementation of the *always* and *eventually* operators that makes them particularly efficient when applied to (atomic) subformulas that exhibit few recovery episodes (e.g., see the $\mathbf{G}_{[0,T]}$ operator in ψ'_1 in Table 2).

4.2 Multi-Agent Flocking

We consider the problem of multi-agent flock formation and maintenance in the presence of external disturbances. We use the rule-based Reynolds flocking model (see Appendix E) involving boids $\mathcal{B} = \{1, \ldots, n\}$ in m-dimensional space. Boid i's position is $x_i \in \mathbb{R}^m$, $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbb{R}^{m \cdot n}$ is a global configuration vector, and $\xi = [\mathbf{x}(1), \ldots, \mathbf{x}(k)] \in \mathbb{R}^{m \cdot n \cdot k}$ is a k-step trajectory.

We consider a 500-second simulation (trajectory) of a 30-boid flock with a time-step of 0.1 sec in a 2-D plane; so, $T = 500$ secs. We subject 20 of the boids to an intermittent uniformly random displacement [15] from $[0, M] \times [0, 2\pi]$, where $M = 20$ meters and 2π are the maximum magnitude and direction of the displacement, respectively. The subset of 20 boids is chosen uniformly at random during the intervals $[100, 150]$, $[250, 300]$, and $[400, 450]$ in seconds. The simulation starts with the boids at random positions with random velocities, both sampled within some bounded intervals.

The relevant STL specifications for the flocking mission are the following.

Flock formation: a cost function $J(\mathbf{x})$ consisting of a cohesion and a separation term determines whether the boids form a flock [20]:

$$J(\mathbf{x}) = \frac{1}{|\mathcal{B}| \cdot (|\mathcal{B}| - 1)} \cdot \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}, i<j} ||x_{ij}||^2 + \omega \cdot \sum_{(i,j) = \mathcal{E}(\mathbf{x})} \frac{1}{||x_{ij}||^2}$$

where $x_{ij} = x_i - x_j$, $\omega = 1/100$, and $\mathcal{E}(\mathbf{x})$ is the set of neighboring boid pairs within an interaction radius $r_c = 25$ meters: $\mathcal{E}(\mathbf{x}) = \{(i,j) \in \mathcal{B}^2 \mid ||x_{ij}|| < r_c, i \neq j\}$. $J(\mathbf{x}) \leq \delta$, $\delta = 500$, implies that flock formation has been obtained.

$$\varphi_1 = \mathbf{G}_{[0,500]}\, \mathbf{F}_{[0,60]}\, \varphi'_1, \qquad \varphi'_1 = (J(\mathbf{x}) \leq \delta)$$

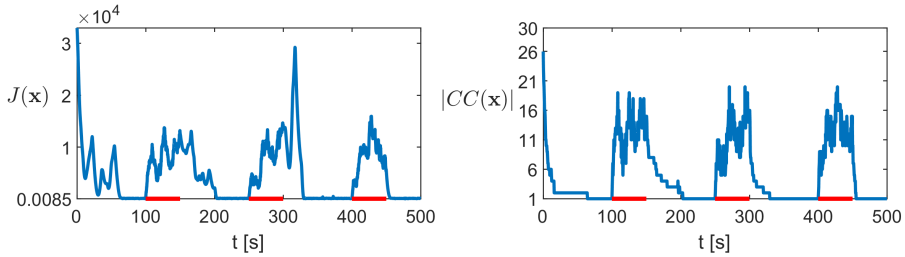


Fig. 4. $J(\mathbf{x})$ and $|CC(\mathbf{x})|$ for 30 boids. Red portions of x-axes indicate intervals of random displacement.

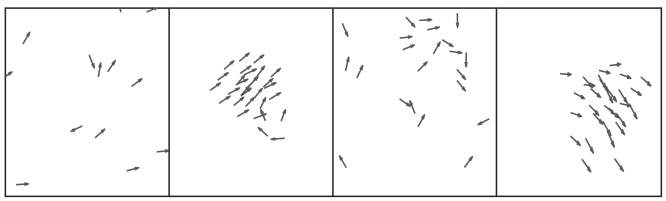


Fig. 5. Flock simulation snapshots at times 0, 75, 300, and 350 (left to right).

SRS formula	$r(\psi_i, \xi, 0)$	Exec. time (sec)
$\psi_1 = R_{30,30}(\varphi_1)$	{(-239.4, 200.6)}	40.47
$\psi_2 = R_{30,30}(\varphi_2)$	{(-239.3, 200.7)}	39.75

Table 3. SRS expressions of the form $R_{\alpha,\beta}(\varphi_i)$ for flocking properties φ_i. All r-values are computed using trajectory ξ of Figure 4 at time 0.

Over the whole trajectory, the flock formation should always be obtained in a timely fashion. This is to be expected in the present of recurrent disturbances to the flock.

Connected components: the number of connected components $|CC(\mathbf{x})|$ of the proximity net $\mathcal{G}(\mathbf{x}) = (\mathcal{B}, \mathcal{E}(\mathbf{x}))$ represents potential fragmentation of the flock . Ideally, $|CC(\mathbf{x})|$ should remain at 1 after flock formation.

$$\varphi_2 = \mathbf{G}_{[0,500]}\, \mathbf{F}_{[0,60]}\, \varphi_2', \qquad \varphi_2' = (|CC(\mathbf{x})| = 1)$$

The functions $J(\mathbf{x})$ and $|CC(\mathbf{x})|$ are plotted in Figure 4. Figure 5 shows flock formation at times 75 and 350 where $(J(\mathbf{x}),\ t) \models \varphi_1'$ and $(|CC(\mathbf{x})|,\ t) \models \varphi_2'$, and times 0 and 300 where $(J(\mathbf{x}),\ t) \not\models \varphi_1'$ and $(|CC(\mathbf{x})|,\ t) \not\models \varphi_2'$. Similar to Section 4.1, we consider two cases of SRS specifications. As in Section 4.1, we compute the ReSV values of φ_1, φ_2 in Table 3 and Table 4. We select time bounds that are consistent with the timescales of flock formation, i.e., $\alpha, \beta = 30$.

As shown in Table 3, $r(\psi_2, \xi, 0) = \{(-239.3, 200.7)\}$, meaning that φ_2 is false at time 0 but becomes true at time 269.3 (i.e., 239.3 seconds later than $\alpha = 30$) and remains true until the end of the trajectory (i.e., interval $[269.3, T]$ is 200.7 seconds longer than $\beta = 30$). In Table 4, $r(\psi_2', \xi, 0) = \{(-42.7, 17.2), (30, -30)\}$; i.e., the (rec, dur) pairs at times 130.2 and 500, representing the worst recovery episodes in $t \in [0, T]$, which are also the best episodes in the interval $[t, t + 60]$. Overall, our results show the resilience of Reynolds flocking model to repeated random disturbances.

SRS formula	$r(\psi_i', \xi, 0)$	Corresponding SRS atoms	Exec. time (sec)
$\psi_1' = \mathbf{G}_{[0,500]}(\mathbf{F}_{[0,60]} R_{30,30}(\varphi_1'))$	{(-42.1, 17.8), (-19, 13.4), (-21.8, 15.3), (30,-30)}	$r(R_{30,30}(\varphi_1'), \xi, 130.4)$, $r(R_{30,30}(\varphi_1'), \xi, 280.4)$, $r(R_{30,30}(\varphi_1'), \xi, 402.9)$, $r(R_{30,30}(\varphi_1'), \xi, 500)$	291.72
$\psi_2' = \mathbf{G}_{[0,500]}(\mathbf{F}_{[0,60]} R_{30,30}(\varphi_2'))$	{(-42.7,17.2), (30, -30)}	$r(R_{30,30}(\varphi_2'), \xi, 130.2)$, $r(R_{30,30}(\varphi_2'), \xi, 500)$	279.54

Table 4. Nested φ_i' expressions replaced by $R_{\alpha,\beta}(\varphi_i')$ expressions in flocking properties φ_i.

5 Related Work

Resiliency has been studied in different system engineering contexts including computer hardware [17], communication [28], distributed systems [23], cybersecurity [30], and model checking [27].

In the context of cyber-physical systems, the literature on resilience is diverse, both in the approach taken and terminology used [2,6,19,31]. Among the logic-based approaches, standard STL robustness provides a notion of the extent to which a signal can be perturbed in space before affecting property satisfaction, which is seen by some authors as a form of resilience [19]. STL time robustness [13, 26] is the equivalent notion when perturbations in time (forward or backward) are considered. There are similarities between (right) time robustness and our notion of recoverability, but the two semantics are fundamentally different as our ReSVs include a second dimension, durability. See Remark 5 for more details on this comparison. Aksaray et al. [2] propose a (time-) "shifting" version of STL and a resilient controller to maximize the robustness value of the shifted formula as fast as possible. This approach, however, only supports **GF**-formulas and does not provide a dedicated "shifting" semantics.

Control-theoretic characterizations of resiliency includes Bouvier et al. [6], which defines a resilience measure (optimized by the controller) based on the additional time it will take for a system to reach a target if under malfunctions. The control framework of Zhu et al. [31] defines a resilient system as one that can restore its state after extreme events caused by specific perturbation classes. In contrast, we answer the question "What is resilience in CPS?" from a temporal-logic perspective. Namely, we provide a syntax for CPS resilience using STL, and a corresponding quantitative semantics. Using our framework in optimal control is not the focus of the present work, but it is a natural continuation.

Another related line of work involves policy and parameter synthesis under multi-objective temporal-logic specifications [4, 7, 8, 14]. Even though we similarly consider multiple requirements (namely recoverability and durability), there is an important difference: we use the satisfaction degrees of recoverability and durability to define the semantics of an STL-based logic, with the semantics of composite formulas derived via Pareto optimization. On the other hand, the above mentioned related work adopt, for each requirement, the usual (Boolean, probabilistic, or quantitative) semantics.

Finally, parametric STL [3, 5] has, akin to our work, a set-based semantics. In this case, the semantics is given by the set of parameter evaluations p for which the resulting concrete formula (instantiated with p) is satisfied by the given signal. We remark that, in our approach, the specification is fixed, non-parametric, and the

set-based semantics arises from the fact that our recoverability-durability pairs might not be directly comparable, i.e., mutually non-dominated.

6 Conclusion

In this paper, we presented a logic-based framework to reason about CPS resiliency. We define resiliency of an STL formula φ as the ability of the system to recover from violations of φ in a timely and durable manner. These requirements represent the atoms of our formally defined SRS logic, which allows to combine such resiliency statements using temporal and Boolean operators. We also introduced ReSV, the first multi-dimensional semantics for an STL-based logic. Under this semantics, an SRS formula is interpreted as a set of non-dominated (rec, dur) pairs, which respectively quantify how quickly the underlying system recovers from a property violation and for how long it satisfies the property thereafter. Importantly, we proved that our ReSV semantics is sound and complete w.r.t. the Boolean semantics of STL. We illustrated our new resiliency framework with two case studies: UAV package delivery and flock formation. Collectively, our results demonstrate the expressive power and flexibility of our framework in reasoning about resiliency in CPS.

In summary, the contribution of our work is not just establishing theoretical foundations of CPS resiliency but also providing a method to equip temporal logics with multi-dimensional semantics, an approach that in the future could be extended to support arbitrary multi-requirement specifications beyond resiliency.

Acknowledgments. We thank the anonymous reviewers for their valuable feedback and suggestions for improving the quality of this paper. Research supported in part by NSF CNS-1952096, CNS-1553273 (CAREER), OIA-2134840, OIA-2040599, CCF-1918225, and CPS-1446832.

References

1. Airspace 101 — Rules of the Sky. https://www.faa.gov/uas/recreational_fliers/where_can_i_fly/airspace_101/ (Aug 2021)
2. Aksaray, D.: Resilient satisfaction of persistent and safety specifications by autonomous systems. In: AIAA Scitech 2021 Forum. pp. 1124–1134 (Jan 2021). https://doi.org/10.2514/6.2021-1124
3. Asarin, E., Donzé, A., Maler, O., Nickovic, D.: Parametric identification of temporal properties. In: International Conference on Runtime Verification. pp. 147–160. Springer (2011)
4. Bakhirkin, A., Basset, N., Maler, O., Jarabo, J.I.R.: Paretolib: a Python library for parameter synthesis. In: International Conference on Formal Modeling and Analysis of Timed Systems. pp. 114–120. Springer (2019)
5. Bakhirkin, A., Ferrère, T., Maler, O.: Efficient parametric identification for STL. In: 21st International Conference on Hybrid Systems: Computation and Control. pp. 177–186 (2018)
6. Bouvier, J., Xu, K., Ornik, M.: Quantitative resilience of linear driftless systems. In: Proceedings of the Conference on Control and its Applications. pp. 32–39. SIAM (Jul 2021). https://doi.org/10.1137/1.9781611976847

7. Calinescu, R., Češka, M., Gerasimou, S., Kwiatkowska, M., Paoletti, N.: Efficient synthesis of robust models for stochastic systems. Journal of Systems and Software **143**, 140–158 (Sep 2018). https://doi.org/10.1016/j.jss.2018.05.013
8. Chen, T., Forejt, V., Kwiatkowska, M., Simaitis, A., Wiltsche, C.: On stochastic games with multiple objectives. In: Proceedings pf International Symposium on Mathematical Foundations of Computer Science (MFCS 2013). pp. 266–277. Springer (Aug 2013). https://doi.org/10.1007/978-3-642-40313-2_25
9. Cook, B., Fisher, J., Krepska, E., Piterman, N.: Proving stabilization of biological systems. In: International Workshop on Verification, Model Checking, and Abstract Interpretation. pp. 134–149. Springer (Jan 2011). https://doi.org/10.1007/978-3-642-18275-4_11
10. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge University Press (May 2002). https://doi.org/10.1017/CBO9780511809088
11. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc., New York, NY, USA (2001)
12. Donzé, A.: Breach, a toolbox for verification and parameter synthesis of hybrid systems. In: Proceedings of International Conference on Computer Aided Verification. pp. 167–170. Springer (Jul 2010). https://doi.org/10.1007/978-3-642-14295-6_17
13. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: Proceedings of International Conference on Formal Modeling and Analysis of Timed Systems. pp. 92–106. Springer (Sep 2010). https://doi.org/10.1007/978-3-642-15297-9_9
14. Etessami, K., Kwiatkowska, M., Vardi, M.Y., Yannakakis, M.: Multi-objective model checking of Markov decision processes. In: Proceedings of International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2007). pp. 50–65. Springer (Mar 2007). https://doi.org/10.1007/978-3-540-71209-1_6
15. Grosu, R., Lukina, A., Smolka, S.A., Tiwari, A., Varadarajan, V., Wang, X.: V-formation via model predictive control. arXiv preprint arXiv:2002.08955 (2020)
16. Haghighi, I., Jones, A., Kong, Z., Bartocci, E., Gros, R., Belta, C.: SpaTeL: A novel spatial-temporal logic and its applications to networked systems. In: Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control (HSCC '15). pp. 189–198 (2015). https://doi.org/10.1145/2728606.2728633
17. Hari, S.K.S., Tsai, T., Stephenson, M., Keckler, S.W., Emer, J.: SASSIFI: An architecture-level fault injection tool for GPU application resilience evaluation. In: Proceedings of International Symposium on Performance Analysis of Systems and Software (ISPASS). pp. 249–258. IEEE (Apr 2017). https://doi.org/10.1109/ISPASS.2017.7975296
18. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: Proceedings of Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, pp. 152–166. Springer (Sep 2004). https://doi.org/10.1007/978-3-540-30206-3_12
19. Mehdipour, N.: Resilience for Satisfaction of Temporal Logic Specifications by Dynamical Systems. Ph.D. thesis, Boston University (2021), https://open.bu.edu/handle/2144/41871
20. Mehmood, U., Paoletti, N., Phan, D., Grosu, R., Lin, S., Stoller, S.D., Tiwari, A., Yang, J., Smolka, S.A.: Declarative vs rule-based control for flocking dynamics. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing. pp. 816–823 (Apr 2018). https://doi.org/10.1145/3167132.3167222
21. Merriam-Webster Dictionary. https://www.merriam-webster.com/dictionary/resiliency
22. Michael, N., Mellinger, D., Lindsey, Q., Kumar, V.: The GRASP multiple micro-UAV testbed. IEEE Robotics & Automation Magazine **17**(3), 56–65 (Sep 2010). https://doi.org/10.1109/MRA.2010.937855

23. Prokhorenko, V., Babar, M.A.: Architectural resilience in cloud, fog and edge systems: A survey. IEEE Access **8**, 28078–28095 (Feb 2020). https://doi.org/10.1109/ACCESS.2020.2971007
24. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques. pp. 25–34. ACM (Aug 1987). https://doi.org/10.1145/37401.37406
25. Reynolds, C.W.: Steering behaviors for autonomous characters. In: Proceedings of Game Developers Conference. pp. 763–782. Citeseer (1999)
26. Rodionova, A., Lindemann, L., Morari, M., Pappas, G.J.: Time-robust control for STL specifications. In: Proceedings of IEEE Conference on Decision and Control. IEEE (Dec 2021). https://doi.org/10.1109/CDC45484.2021.9683477
27. Selyunin, K., Jaksic, S., Nguyen, T., Reidl, C., Hafner, U., Bartocci, E., Nickovic, D., Grosu, R.: Runtime monitoring with recovery of the SENT communication protocol. In: Part I, Proceedings of International Conference on Computer Aided Verification. pp. 336–355. Springer (Jul 2017). https://doi.org/10.1007/978-3-319-63387-9_17
28. Tan, S., Wu, Y., Xie, P., Guerrero, J.M., Vasquez, J.C., Abusorrah, A.: New challenges in the design of microgrid systems: Communication networks, cyber-attacks, and resilience. IEEE Electrification Magazine **8**(4), 98–106 (Dec 2020). https://doi.org/10.1109/MELE.2020.3026496
29. Yaghoubi, S., Fainekos, G.: Worst-case satisfaction of STL specifications using feedforward neural network controllers: A Lagrange multipliers approach. In: Proceedings of Information Theory and Applications Workshop, ITA. pp. 127:1–127:20. IEEE (Feb 2020). https://doi.org/10.1109/ITA50056.2020.9244969
30. Yuan, H., Xia, Y., Yang, H.: Resilient state estimation of cyber-physical system with multichannel transmission under DoS attack. IEEE Transactions on Systems, Man, and Cybernetics: Systems **51**(11), 6926–6937 (Nov 2020). https://doi.org/10.1109/TSMC.2020.2964586
31. Zhu, Q., Başar, T.: Robust and resilient control design for cyber-physical systems with an application to power systems. In: Proceedings of IEEE Conference on Decision and Control and European Control Conference. pp. 4066–4071. IEEE (Dec 2011). https://doi.org/10.1109/CDC.2011.6161031

A Proof of Corollaries

Corollary 1. *Maximum and minimum resilience sets are non-empty and non-dominated sets.*

Proof. Let S be the maximum resilience set of set P with $P \neq \emptyset$.

We prove that S is non-empty by induction.

1. Assume P is a singleton set. It is trivial that S is non-empty.
2. For $P \neq \emptyset$, let $S \neq \emptyset$ be the maximum resilience set of P and $r \in \mathbb{Z}^2$. Let S' be the maximum resilience set of $P \cup \{r\}$. If $r \in P$, it is trivial that $S' = S$, and hence $S' \neq \emptyset$. If $r \notin P$, then:
   - If exists $s \in S$ s.t. $s \succ_{re} r$ or $s =_{re} r$, then we have $s \in S'$ (given that S is the maximum resilience set of P, then $s \succ_{re} p'$ or $s =_{re} p'$ for all $p' \in P \cup \{r\}$).
   - Otherwise, we have $s \prec_{re} r$ for all $s \in S$. In this case, for all $p \in P$ and $s \in S$, it holds that either (1) $p \prec_{re} s$, which, by transitivity of $\prec_{re}$, implies that $p \prec_{re} r$; or (2) $p =_{re} s$, in which case $p \prec_{re} r$ or $p =_{re} r$ (i.e., $p \not\succ_{re} r$): indeed if $p \succ_{re} r$ held, then by transitivity of $\succ_{re}$, we would have $p \succ_{re} s$, which contradicts the assumption. In either case, $r \in S'$.
   - Therefore, S' is non-empty.
3. By induction, S is non-empty for any $P \neq \emptyset$.

We then prove that S is non-dominated. Let $s_1, s_2 \in S$.

1. Because $s_1 \in S$ and $s_2 \in S \subseteq P$, we have $s_1 \succ_{re} s_2$ or $s_1 =_{re} s_2$. Similarly, we have $s_2 \succ_{re} s_1$ or $s_2 =_{re} s_1$. Therefore, $s_1 =_{re} s_2$.
2. By definition, S is a non-dominated set.

Minimum resilience sets can be proven to be non-empty and non-dominated sets in a similar manner. □

B Proof of Propositions

Proposition 1. *The ReSV $r(\psi, \xi, t)$ of an SRS formula ψ w.r.t. a signal ξ at time t is a non-dominated set.*

Proof. We show that for any SRS ψ and signal ξ, $r(\psi, \xi, t)$ is either a maximum or minimum resilience set of a non-empty set. By Corollary 1, this implies that $r(\psi, \xi, t)$ is a non-empty and non-dominated set.

If ψ is an SRS atom, then $r(\psi, \xi, t)$ is a singleton set, and thus it is trivially both the maximum and minimum resilience set of itself. When ψ is of the form $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2$, or $\psi_1 \mathbf{U}_I \psi_2$, its semantics $r(\psi, \xi, t)$ is defined as either a maximum or minimum resilience set (see Definition 4). The only remaining case is $\psi = \neg\psi_1$. Let $x, y \in \mathbb{Z}^2$ with $x = (x_r, x_d), y = (y_r, y_d)$. If $r(\psi_1, \xi, t)$ is a non-dominated set, then $r(\psi, \xi, t) = \{(-x_r, -x_d) \mid x \in r(\psi_1, \xi, t)\}$ is a non-dominated set also. This is because $x =_{re} y$ iff $(-x_r, -x_d) =_{re} (-y_r, -y_d)$ for any x, y. □

C Proof of Lemmas

Lemma 1. *Relations $\succ_{re}$ and $\prec_{re}$ are strict partial orders.*

Proof. Let $x, y, z \in \mathbb{Z}^2$ be different with $x = (x_r, x_d)$, $y = (y_r, y_d)$, $z = (z_r, z_d)$.
We prove that $\succ_{re}$ has irreflexivity and transivity properties.

- Irreflexivity: we can see that $x \succ_{re} x$ never holds because $sign(x_r)+sign(x_d) = sign(x_r) + sign(x_d)$ but x does not Pareto-dominate itself.
- Transivity: Assume that $x \succ_{re} y$ and $y \succ_{re} z$. Using the transivity of Pareto-dominance $\succ$ [11], one of the two following statements is true:
  - $sign(x_r) + sign(x_d) > sign(z_r) + sign(z_d)$.
  - $sign(x_r) + sign(x_d) = sign(z_r) + sign(z_d)$ and x Pareto-dominates z.

  Therefore $x \succ_{re} z$.

Irreflexivity and transitivity together imply asymmetry; so $\succ_{re}$ is a strict partial order, and the dual of a strict partial order, $\prec_{re}$, is also a strict partial order [10]. □

D Proof of Theorems

Theorem 1 (Soundness and Completeness of SRS Semantics). *Let ξ be a signal and ψ an SRS specification. The following results at time t hold:*
1) $\exists\, x \in r(\psi, \xi, t)$ *s.t.* $x \succ_{re} \mathbf{0} \Longrightarrow (\xi, t) \models \psi$
2) $\exists\, x \in r(\psi, \xi, t)$ *s.t.* $x \prec_{re} \mathbf{0} \Longrightarrow (\xi, t) \models \neg\psi$
3) $(\xi, t) \models \psi \Longrightarrow \exists\, x \in r(\psi, \xi, t)$ *s.t.* $x \succ_{re} \mathbf{0}$ *or* $x =_{re} \mathbf{0}$
4) $(\xi, t) \models \neg\psi \Longrightarrow \exists\, x \in r(\psi, \xi, t)$ *s.t.* $x \prec_{re} \mathbf{0}$ *or* $x =_{re} \mathbf{0}$

Proof. We first establish the following results because they are used in the main proof. Let $x, y \in \mathbb{Z}^2$ with $x = (x_r, x_d)$, $y = (y_r, y_d)$.

i) $x \succ_{re} \mathbf{0}$ is equivalent to $x_r, x_d \geq 0 \wedge x \neq \mathbf{0}$.
ii) $x =_{re} \mathbf{0}$ is equivalent to $x_r\, x_d < 0 \vee x = \mathbf{0}$.
iii) By i) and ii), $x \succ_{re} \mathbf{0} \vee x =_{re} \mathbf{0}$ is equivalent to $x_r, x_d \geq 0 \vee x_r\, x_d < 0$.
iv) If $x \succ_{re} \mathbf{0}$ and $y \succ_{re} x \vee y =_{re} x$, then $y_r, y_d \geq 0 \wedge y \neq \mathbf{0}$ holds, which, by i), is equivalent to $y \succ_{re} \mathbf{0}$.
v) If $x \succ_{re} \mathbf{0} \vee x =_{re} \mathbf{0}$ and $y =_{re} x$, then it holds that $y_r, y_d \geq 0 \vee y_r\, y_d < 0$, which by iii), is equivalent to $y \succ_{re} \mathbf{0} \vee y =_{re} \mathbf{0}$.
vi) If $x \succ_{re} \mathbf{0} \vee x =_{re} \mathbf{0}$ and $y \succ_{re} x$, then it holds that $y_r, y_d > 0 \vee y_r\, y_d < 0 \vee (y_r = 0 \wedge y_d > 0) \vee (y_r > 0 \wedge y_d = 0)$ is true, thus $y \succ_{re} \mathbf{0}$ or $y =_{re} \mathbf{0}$.[7]
vii) Let φ be an STL formula with α,$\beta > 0$; let $\psi = R_{\alpha,\beta}(\varphi)$ be an SRS atom. Since $(\xi, t) \models \psi$ is equivalent from a Boolean satisfaction standpoint to $\neg\varphi \mathbf{F}_{[0,\alpha]} \mathbf{G}_{[0,\beta)} \varphi$, we can conclude that $(\xi, t) \models \psi$ holds iff $\exists\, 0 \leq t_1 \leq \alpha$ s.t. $\mathbf{G}_{[t,t+t_1)} \neg\varphi \wedge \mathbf{G}_{[t+t_1,t+t_1+\beta)}\, \varphi$ is true.

[7] Given $x =_{re} \mathbf{0}$, the relation $y \succ_{re} x$ implies that 1) $y_r\, y_d < 0 \wedge y \succ x$ holds (when $x_r\, x_d < 0$) or 2) $y_r, y_d > 0 \vee (y_r = 0 \wedge y_d > 0) \vee (y_r > 0 \wedge y_d = 0)$ holds (when $x_r\, x_d < 0$ or $x_r, x_d = 0$). The disjunction of 1) and 2) is equivalent to $(y_r\, y_d < 0 \wedge y \succ x) \vee (y_r, y_d \geq 0 \wedge y \succ \mathbf{0})$, which implies $y_r\, y_d < 0 \vee y_r, y_d \geq 0$, i.e., $y \succ_{re} \mathbf{0} \vee y =_{re} \mathbf{0}$.

viii) Assume $\exists\, x \in r(\psi, \xi, t)$ s.t. $x \succ_{re} \mathbf{0}$. By Proposition 1, we have $y =_{re} x$ holds for all $y \in r(\psi, \xi, t)$. Then we have $y_r, y_d \geq 0 \wedge y \neq \mathbf{0}$, i.e., $y \succ_{re} \mathbf{0}$.[8] Similarly, we can prove that $\exists\, x \in r(\psi, \xi, t)$ s.t. $x =_{re} \mathbf{0}$ implies that $x =_{re} \mathbf{0}$ holds for all $x \in r(\psi, \xi, t)$.

We prove statements 1) and 3) of Theorem 1. Statements 2) and 4) can be proven analogously. The proofs follow the inductive structure used in the definition of ReSV. We first prove statement 1) of Theorem 1 as follows.

1) $\exists\, x \in r(\psi, \xi, t)$ s.t. $x \succ_{re} \mathbf{0} \Longrightarrow (\xi, t) \models \psi$

- $\psi = R_{\alpha,\beta}(\varphi)$. By Definition 4, $r(\psi, \xi, t)$ is a singleton set; $x \succ_{re} \mathbf{0}$ implies $x_r, x_d \geq 0 \wedge x \neq \mathbf{0}$, which then implies $x_r, x_d \geq 0$. By transitivity of implication, we can equivalently prove that $x_r, x_d \geq 0$ implies $(\xi, t) \models \psi$. By Eq. (1), $x_r \geq 0$ implies $0 \leq t_{rec}(\varphi, \xi, t) \leq \alpha$. Let $t_1 = t_{rec}(\varphi, \xi, t)$. By Eq. (2), we have that $\mathbf{G}_{[t,t+t_1)} \neg\varphi$ holds. By Eq. (1), $x_d \geq 0$ implies $t_{dur}(\varphi, \xi, t + t_1) \geq \beta$, and thus, by Eq. (3), $\mathbf{G}_{[t+t_1,t+t_1+\beta)}\ \varphi$ holds. Therefore $x_r, x_d \geq 0$ implies $\exists\, 0 \leq t_1 \leq \alpha$ s.t. $\mathbf{G}_{[t,t+t_1)} \neg\varphi \wedge \mathbf{G}_{[t+t_1,t+t_1+\beta)}\ \varphi$; by result vii), $(\xi, t) \models \psi$.
- $\psi = \neg\psi_1$. By Definition 4, $r(\psi, \xi, t) = \{(-x_r, -x_d) : x \in r(\psi_1, \xi, t)\}$. Thus, by the implicant in i), we have that $\exists\, x \in r(\psi_1, \xi, t)$ s.t. $x \prec_{re} \mathbf{0}$ holds. From the induction hypothesis, we have $(\xi, t) \models \neg\psi_1$; thus $(\xi, t) \models \psi$.
- $\psi = \psi_1 \wedge \psi_2$. By Definition 4, $r(\psi_1 \wedge \psi_2, \xi, t) = \min_{re}(r(\psi_1, \xi, t) \cup r(\psi_2, \xi, t))$. By Definition 3 and result iv), we have $x' \succ_{re} x$ or $x' =_{re} x$, thus $x' \succ_{re} \mathbf{0}$ for all $x' \in r(\psi_1, \xi, t) \cup r(\psi_2, \xi, t)$. From the induction hypothesis, we have $(\xi, t) \models \psi_1$ and $(\xi, t) \models \psi_2$; thus $(\xi, t) \models \psi_1 \wedge \psi_2$.
- $\psi = \psi_1 \vee \psi_2$. By Definition 4, $r(\psi_1 \vee \psi_2, \xi, t) = \max_{re}(r(\psi_1, \xi, t) \cup r(\psi_2, \xi, t))$. By Definition 3, we have $x \in r(\psi_1, \xi, t) \cup r(\psi_2, \xi, t)$; therefore $x \in r(\psi_1, \xi, t)$ or $x \in r(\psi_2, \xi, t)$. From the induction hypothesis, we have $(\xi, t) \models \psi_1$ or $(\xi, t) \models \psi_2$; thus $(\xi, t) \models \psi_1 \vee \psi_2$.
- $\psi = \psi_1 \mathbf{U}_I \psi_2$. By Definition 4, $r(\psi_1 \mathbf{U}_I \psi_2, \xi, t) = \max_{re} \cup_{t' \in t+I} \min_{re}(r(\psi_2, \xi, t') \cup \min_{re} \cup_{t'' \in [t,t+t')} r(\psi_1, \xi, t''))$. By Definition 3 and result iv), we have that $\exists\, t_1 \in t + I$ s.t. $x_1 \succ_{re} \mathbf{0}$ for all $x_1 \in r(\psi_2, \xi, t_1) \cup \min_{re} \cup_{t'' \in [t,t+t_1)} r(\psi_1, \xi, t'')$. From the induction hypothesis, we have $(\xi, t_1) \models \psi_2$. Similarly, we have $x_2 \succ_{re} \mathbf{0}$ for all $x_2 \in r(\psi_1, \xi, t'')$ and all $t'' \in [t, t + t_1)$. From the induction hypothesis, $\forall\, t'' \in [t, t + t_1)$ $(\xi, t'') \models \psi_1$. Together with $(\xi, t_1) \models \psi_2$ we can conclude that $(\xi, t) \models \psi_1 \mathbf{U}_I \psi_2$.

We then prove statement 3) of Theorem 1 as follows.

3) $(\xi, t) \models \psi \Longrightarrow \exists\, x \in r(\psi, \xi, t)$ s.t. $x \succ_{re} \mathbf{0}$ or $x =_{re} \mathbf{0}$

- $\psi = R_{\alpha,\beta}(\varphi)$. By result (vii), $(\xi, t) \models \psi$ implies that $\exists\, 0 \leq t_1 \leq \alpha$ s.t. $\mathbf{G}_{[t,t+t_1)} \neg\varphi \wedge \mathbf{G}_{[t+t_1,t+t_1+\beta)}\ \varphi$ is true, which results in $x_r, x_d \geq 0$ by Eqs. (2) and (3). This is equivalent to $x \succ_{re} \mathbf{0}$ (when at least one of $x_r, x_d \geq 0$ holds strictly) or $x =_{re} \mathbf{0}$ (when $x_r = x_d = 0$).

[8] Indeed, for $y =_{re} x$ to hold, we need $sign(y_d) + sign(y_r) = sign(x_d) + sign(x_r) \geq 1$, which can only happen if $y_r, y_d \geq 0 \wedge y \neq \mathbf{0}$.

- $\psi = \neg\psi_1$. From the induction hypothesis, $(\xi, t) \models \neg\psi_1$ implies that $\exists\, x \in r(\neg\psi_1, \xi, t)$ s.t. $x \succ_{re} \mathbf{0}$ or $x =_{re} \mathbf{0}$. Then, by Definition 4, we have that $\exists\, x \in r(\psi_1, \xi, t)$ s.t. $x \prec_{re} \mathbf{0}$ or $x =_{re} \mathbf{0}$ (i.e., $x_r, x_d \leq 0 \vee x_r\, x_d < 0$ is true). By Definition 4, $(-x_r, -x_d) \in r(\psi, \xi, t)$; thus $(-x_r, -x_d) \succ_{re} \mathbf{0}$ (when at least one of $x_r, x_d \leq 0$ holds strictly) or $(-x_r, -x_d) =_{re} \mathbf{0}$ (when $x_r = x_d = 0$ or $x_r\, x_d < 0$ is true).
- $\psi = \psi_1 \wedge \psi_2$. This is equivalent to $(\xi, t) \models \psi_1 \wedge (\xi, t) \models \psi_2$ holds. From induction hypothesis, we have $\exists\, x_1 \in r(\psi_1, \xi, t)$ s.t. $x_1 \succ_{re} \mathbf{0}$ or $x_1 =_{re} \mathbf{0}$; and $\exists\, x_2 \in r(\psi_2, \xi, t)$ s.t. $x_2 \succ_{re} \mathbf{0}$ or $x_2 =_{re} \mathbf{0}$. By result viii), we have $x \succ_{re} \mathbf{0}$ or $x =_{re} \mathbf{0}$ holds for all $x \in r(\psi_1, \xi, t)$ and $x \in r(\psi_2, \xi, t)$. By Definition 3, we have $x \succ_{re} \mathbf{0}$ or $x =_{re} \mathbf{0}$ holds for all $x \in \min_{re}(r(\psi_1, \xi, t) \cup r(\psi_2, \xi, t))$, i.e., by Definition 4, for all $x \in r(\psi, \xi, t)$.
- $\psi = \psi_1 \vee \psi_2$. By definition, $(\xi, t) \models \psi_1 \vee \psi_2$ holds implies that $(\xi, t) \models \psi_1 \vee (\xi, t) \models \psi_2$ holds. From induction hypothesis, we have $\exists\, x_1 \in r(\psi_1, \xi, t)$ s.t. $x_1 \succ_{re} \mathbf{0}$ or $x_1 =_{re} \mathbf{0}$; or $\exists\, x_2 \in r(\psi_2, \xi, t)$ s.t. $x_2 \succ_{re} \mathbf{0}$ or $x_2 =_{re} \mathbf{0}$. Let w.l.o.g. the former holds; then we have $x_1 \in r(\psi_1, \xi, t) \cup r(\psi_2, \xi, t)$. By Definition 3, $x \succ_{re} x_1$ or $x =_{re} x_1$ for all $x \in \max_{re}(r(\psi_1, \xi, t) \cup r(\psi_2, \xi, t))$. By results v) and vi), $x \succ_{re} \mathbf{0}$ or $x =_{re} \mathbf{0}$. By Definition 4, $r(\psi, \xi, t) = \max_{re}(r(\psi_1, \xi, t) \cup r(\psi_2, \xi, t))$.
- $\psi = \psi_1 \mathbf{U}_I \psi_2$. By definition, $(\xi, t) \models \psi_1 \mathbf{U}_I \psi_2$ holds implies that $\exists\, t' \in t + I$ s.t. $(\xi, t') \models \psi_2 \wedge \forall\, t'' \in [t, t'), (\xi, t'') \models \psi_1$. From induction hypothesis, we have $\exists\, x_2 \in r(\psi_2, \xi, t')$ s.t. $x_2 \succ_{re} \mathbf{0}$ or $x_2 =_{re} \mathbf{0}$; and $\exists\, x_1 \in r(\psi_1, \xi, t'')$ s.t. $x_1 \succ_{re} \mathbf{0}$ or $x_1 =_{re} \mathbf{0}$ for all $t'' \in [t, t')$. By result (viii), we have that $x \succ_{re} \mathbf{0}$ or $x =_{re} \mathbf{0}$ holds for all $x \in r(\psi_2, \xi, t')$ and for all $x \in r(\psi_1, \xi, t'')$, for all $t'' \in [t, t')$. By Definition 3, we evince that $\exists\, t' \in t + I$ s.t. $x \succ_{re} \mathbf{0}$ or $x =_{re} \mathbf{0}$ holds for all $x \in \min_{re}(r(\psi_2, \xi, t') \cup \min_{re} \cup_{t'' \in [t, t+t')} r(\psi_1, \xi, t''))$. Note that the latter is a subset of $\cup_{t' \in t+I} \min_{re}(r(\psi_2, \xi, t') \cup \min_{re} \cup_{t'' \in [t, t+t')} r(\psi_1, \xi, t''))$, which implies, together with Definition 3 and results v) and vi), that $x \succ_{re} \mathbf{0}$ or $x =_{re} \mathbf{0}$ holds for all $x \in \max_{re} \cup_{t' \in t+I} \min_{re}(r(\psi_2, \xi, t') \cup \min_{re} \cup_{t'' \in [t, t+t')} r(\psi_1, \xi, t''))$, which is $r(\psi, \xi, t)$ by Definition 4.

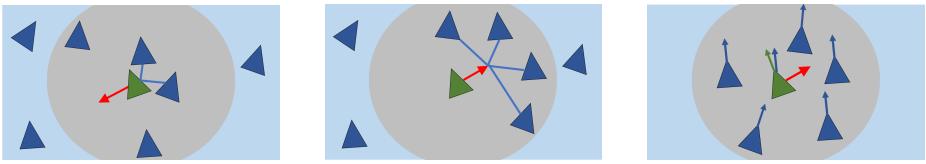
This completes the proof. □

E Reynolds flocking model

We use Reynolds rule-based model [24, 25] to describe the dynamics of flocking. There are three steering behaviors used to determine how the agents, which are called boids, maneuver based on the positions and velocities of their nearby flockmates. *Separation* causes a boid to move away from its nearby flockmates in crowded situations. *Cohesion* drives a boid to move towards the average location of its nearby flockmates. *Alignment* causes a boid to steer in the direction of the average heading of its nearby flockmates.

Figure 6 illustrates these rules. We consider a set of boids $\mathcal{B} = \{1, \ldots, n\}$ in m-dimensional space that move in discrete time as follows:

$$\mathbf{x}(k+1) = \mathbf{x}(k) + dt \cdot v_i(k) + d_i(k)$$
$$v_i(k+1) = v_i(k) + dt \cdot a_i(k)$$



(a) Separation. (b) Cohesion. (c) Alignment.

Fig. 6. Reynolds model rules. Gray circles represent the interaction region of the green boid.

where $\mathbf{x}(k) = [x_1(k), \ldots, x_n(k)] \in \mathbb{R}^{m \cdot n}$ is the configuration of all boids, $\mathbf{v}(k) = [v_1(k), \ldots, v_n(k)] \in \mathbb{R}^{m \cdot n}$ and $\mathbf{a}(k) = [a_1(k), \ldots, a_n(k)] \in \mathbb{R}^{m \cdot n}$. Vectors $x_i(k)$, $v_i(k)$, $a_i(k)$, $d_i(k) \in \mathbb{R}^m$ are the position, velocity, acceleration, and random displacement of boid i at step k. Time interval dt is the duration of a time step.