

Reachability Specification Adaptation for Continuous Dynamical Systems

Zhi Zhang¹[0000–0002–3457–6919], Dalal Alrajeh¹[0000–0002–1365–8026], Yue Gu²[0000–0002–8134–5822], Roberto Metere³[0000–0001–6992–4285], Michele Sevegnani²[0000–0001–6773–9481], Kangfeng Ye³[0000–0003–2460–7926], and Poonam Yadav³[0000–0003–0169–0704]

¹ Imperial College London, UK

² University of Glasgow, UK

³ University of York, UK

Abstract. Continuous dynamical systems are fundamental in modelling a wide range of real-world phenomena, from cyber-physical systems to industrial automation. Such systems operate in dynamic environments where unanticipated situations can arise. Requirements specified at design time may become partially unachievable when the system is deployed. We propose a novel method for reachability specification degradation tailored to continuous dynamical systems. Our approach involves sampling points within the initial state space region, estimating reachability regions over time using the Lipschitz constant, and augmenting the original specification through disjunction operations. We implement our method to support how a system can adapt to requirements in real time.

Keywords: Continuous dynamical systems · Reachability · Specification degradation.

1 Introduction

Continuous dynamical systems are fundamental in modelling a wide range of real-world phenomena, from cyber-physical systems [20, 15] and communication systems [26, 16, 13] to biological processes [9] and industrial automation [21]. These systems operate in environments that are often dynamic, uncertain, or partially known, making it essential for the system requirements (specified at design time) to continually adapt as the environment changes or new information about the environment becomes available. For example, consider a workload management system for next-generation communication base stations [26]. The system must balance resource allocation across processing units and communication channels, despite fluctuating network demands and traffic patterns. Such requirement could be expressed as a safety property that states “a base station workload must never exceed 1 terabits per second (Tbps)” or formally in signal temporal logic as: $\mathbf{G}_{[a,b]}(\text{WL}_{bs} \leq 1\text{Tbps})$, where $\mathbf{G}$ is *always*, a, b represent time values, and WL_{bs} is the base station workload. However, in a major event,

the number of users connected to a station may increase dramatically, causing this requirement to be (temporarily) violated. When modelling a continuous dynamical system, this violation may manifest as a subset of trajectories from an initial state-space region (e.g., a high number of connected users) that never reach desirable states (e.g., a workload below 1 Tbps). One possible repair to the predefined requirement might be “a base station workload must never exceed 1 Tbps, or if it does, then it must reduce it to below 1 Tbps within 5 minutes”.

Several approaches have been proposed to enable adaptation of specifications (a.k.a. specification degradation) expressed in a temporal logic formalism in the face of environmental changes, such as [27, 3, 8, 6, 10]. However, these are limited in that they either (i) only handle discrete systems and thus require discretisation of the state space, causing a lack of precision in the requirements specification, or (ii) do not provide means for detecting violations of the specification for continuous dynamical systems from an initial region to trigger the adaptation.

In this paper, we propose a novel method for safety requirements’ degradation in continuous dynamical systems. Our approach to safety requirements expressed in non-nested Signal Temporal Logic (STL) [19] and considers their satisfaction checking as a reachability analysis problem [2, 14, 11]. Given an initial condition (representing a state-space region, which is the blue region of Fig. 1) for the system and a safety requirement, the approach uses the system’s Lipschitz constant to estimate the reachability regions (yellow region of Fig. 1) for the system starting from the sampling points in the initial state-space region. Since the considered STL specifications are non-nested, their satisfaction can be checked independently for each atomic temporal subformula, which simplifies the degradation process. The estimated region is then used to extract a weakening of the original specification, ensuring that the adapted specification is satisfied by all trajectories starting from this initial state-space region.

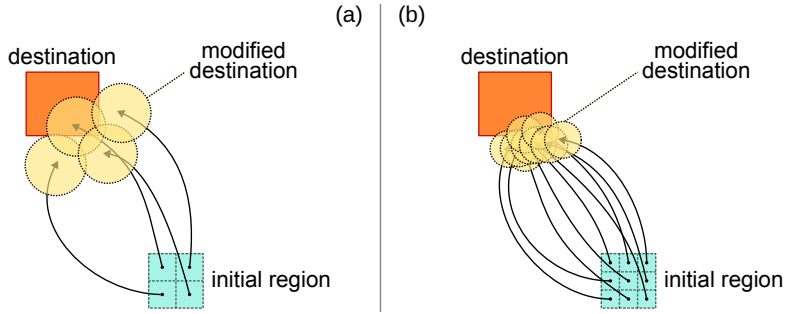


Fig. 1. The specification repair is based on the Lipschitz constant with (a) a 2×2 partition, and (b) a 3×3 partition. The yellow regions denote the reachable regions.

2 Related Work

The closest work to ours focuses on the degradation of specifications in STL when deviations in the environment’s expected behaviour are detected [7, 6]. The primary challenge for these approaches lies in verifying whether the pre-defined specification has been violated, as it often requires checking an infinite number of trajectories. For instance, the work of [6] cannot detect specification violations when the time interval of the specification is long, and the action set is large, since this requires checking numerous trajectories generated by various combinations of actions within the specified time frame. To overcome this problem, Ghosh et al. [10] employed mixed-integer linear programming (MILP) to verify the violation of STL specifications and facilitated specification adaptation by reformulating the repair process as a model predictive control problem, for a dynamical system originating from an initial point. However, MILPs can only be used to check the violation of the specification for certain simple scenarios, since they often face infeasibility issues, such as running for extended periods without producing a solution. This is caused by large problem sizes, tight or complex constraints, or ill-conditioned problems [5].

The work of [3] presents an automated approach to the degradation of specifications expressed in a subclass of Linear Temporal Logic, for which tractable synthesis methods exist, in response to the detection of violations of environmental assumptions. Their approach considers violations of safety assumptions about the environment as a trigger for the adaptation process. As in this work, their research focuses on computing weaknesses of the original specification. Different from ours, they focus on discrete systems. In [1], a meta-learning framework is proposed to support the adaptation of learners during specification degradation tasks. They propose using a reinforcement learning agent to configure a learner that identifies the optimal degradation at runtime. However, these approaches cannot be applied to continuous dynamical systems, nor can they handle specifications without loss of accuracy.

At the same time, alternative set representations have been explored in reachability analysis, which could also be applied in the context of specification repair for continuous systems. For example, polygonal (polyhedral) sets have been used to represent the behaviour of anisotropic systems [12]. Similarly, Taylor model-based representations, which combine high-order polynomial expansions with rigorous remainder bounds, have been employed to construct flowpipes for nonlinear systems [18, 4]. While our framework is currently formulated using Lipschitz-based approximations, it could in principle be extended to incorporate these alternative representations.

Our work can be viewed as complementary to approaches to synthesising correct-by-construction control strategies for dynamical systems, as [22, 17]. These assume specifications are correct initially. Our approach complements these approaches by accounting for requirements that are too strong and cannot always be achieved, requiring their degradation.

3 Preliminary

3.1 Dynamical System

A continuous dynamical system (CDS) describes the evolution of a system's state over time, and an external dynamical environment can influence its dynamics. This system may need to satisfy certain specifications, such as safety or reachability.

Definition 1 (Continuous Dynamical System). *Given $\mathcal{S} \subseteq \mathbb{R}^n$ as the bounded state space for a system, and $\mathcal{E} \subseteq \mathbb{R}^m$ as the bounded state space for the dynamical environment, we define a continuous dynamical system as the tuple $\Sigma_s := (\mathcal{S}, \mathcal{E}, f)$, where $f : \mathcal{S} \times \mathcal{E} \rightarrow \mathcal{S}$ is a measurable function characterising the state evolution of Σ_s . Assume that this dynamical system starts from the initial state space region $B_0 \subset \mathcal{S}$. That is, given $x(t) \in \mathcal{S}$ and $e(t) \in \mathcal{E}$ at some time t , its dynamical system is defined as*

$$\dot{x}(t) = f(x(t), e(t)), \quad t \in \mathbb{R}^+, \quad (1)$$

where the dynamics of state $e(t)$ is governed by

$$\dot{e}(t) = \Theta(e(t)), \quad (2)$$

and $\Theta : \mathcal{E} \rightarrow \mathcal{E}$ is a measurable function.

If the function $e : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$ is a constant vector function, the CDS runs in an environment with a constant rate of variation. The static environment is the special case, in which $e(t)$ is always the zero vector for any $t \in \mathbb{R}^+$.

By selecting a sufficiently small time step h_t and an initial time t_0 , Eq. (1) can be approximated by the Euler method, as follows,

$$x(k+1) = x(k) + h_t f(x(k), e(k)), \quad (3)$$

where $k = \frac{t-t_0}{h_t}$. For any time $t^* \in \mathbb{R}^+$, the time interval $[t_0, t^*]$ can be represented as $[0, k^*]$, where $k^* = \frac{t^*-t_0}{h_t}$, $k^* \in \mathbb{Z}^+$. Its initial condition $x(0)$ denotes x_0 and x_k denotes $x(k)$. In the following discussion, we only consider this form for the CDS. Thus, the evolution of the system described in Eq. (1) is characterised by the trajectories of its states and measurements.

In this paper, we assume that the models of Eq. (1) and (2) are all known and autonomous dynamical systems. In the following sections, we will study the specification adaptation, such as the properties of safety and reachability.

3.2 Signal Temporal Logic

Signal Temporal Logic (STL) was originally developed as an extension of Metric Interval Temporal Logic (MITL), designed to analyse and reason about the behaviour of real-valued, dense-time signals [19]. Thus, STL can be used to specify the time-varying behaviour of a system based on the real value signal. In this

paper, we focus on the STL with non-nested temporal operators fragment, which includes all the key temporal operators of STL except for their nesting.

STL is consisted of predicates h which is **true** if and only if a function $h : \mathbb{R}^n \rightarrow \mathbb{R}$ is such that $h(x) \geq 0$. The syntax for an STL formula ϕ is defined as

$$\phi := \text{true} \mid h(x) \geq 0 \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \mathbf{F}_{[a,b]}\phi \mid \mathbf{G}_{[a,b]}\phi \mid \phi_1 \mathbf{U}_{[a,b]}\phi_2,$$

where ϕ_1, ϕ_2 are STL formulas and $a, b \in \mathbb{R}_{\geq 0}$ with $a \leq b$. $\mathbf{U}$ is the until operator, and $\phi_1 \mathbf{U}_{[a,b]}\phi_2$ means ϕ_1 must hold until ϕ_2 becomes true within a time interval $[a, b]$. The temporal operator $\mathbf{F}$ and $\mathbf{G}$ are *eventually* and *always* operator. Let $\models$ denote the satisfaction relation between a time-state pair (x, t) and an STL formula ϕ , i.e., $(x, t) \models \phi$ which means ϕ holds true in the signal x at the time moment t , where the function $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$.

Definition 2. For a signal x the satisfaction of a formula ϕ as described in [19], as follows

$$\begin{aligned} (x, t) \models h &\iff h(x(t)) \geq 0 \\ (x, t) \models \neg h &\iff \neg((x, t) \models h) \\ (x, t) \models \phi_1 \wedge \phi_2 &\iff (x, t) \models \phi_1 \wedge (x, t) \models \phi_2 \\ (x, t) \models \phi_1 \mathbf{U}_{[a,b]}\phi_2 &\iff \exists t_1 \in [t + a, t + b] \cdot (x, t_1) \models \phi_2 \wedge \forall t_2 \in [t, t_1] \cdot (x, t_2) \models \phi_1 \\ (x, t) \models \mathbf{F}_{[a,b]}\phi &\iff \exists t_1 \in [t + a, t + b] \cdot (x, t_1) \models \phi \\ (x, t) \models \mathbf{G}_{[a,b]}\phi &\iff \forall t_1 \in [t + a, t + b] \cdot (x, t_1) \models \phi \end{aligned}$$

4 Problem Statement

We are interested in checking the trajectory of a system whose behaviour is prescribed by Eq. (1), starting from the initial region B_0 , against a reachability property specified as an STL formula.

Definition 3. (Regional Satisfaction) Given a specification ϕ and its time interval $[0, t^*]$, the system described in Eq. (1) can satisfy this specification ϕ , when starting from initial region B_0 , i.e., for all $x(k)$ generated from Eq. (3), the set $\mathcal{X}_\phi := \{x_0 \in B_0 \mid \exists k \in \{0, \dots, k^*\}, x_k \models \phi\}$ is equal to B_0 . We say this system has a Regional Satisfaction w.r.t. ϕ , when starting from the initial region B_0 .

Definition 4 (Partially Regional Satisfaction). Given a specification ϕ and its time interval $[0, t^*]$, the system described in Eq. (1) can partially satisfy this specification ϕ , when starting from the given region B_0 , i.e., the set $\mathcal{X}_\phi \subset B_0$. We say this system has a partially regional satisfaction w.r.t. ϕ , when starting from the given region B_0 .

Definition 5 (Strong Regional Unsatisfaction). Given a specification ϕ , the system described in Eq. (1) cannot satisfy this specification ϕ , when starting from the given region B_0 , i.e., for all $x(0) \in B_0$, the set $\mathcal{X}_\phi$ is empty. We say this system has a strong regional unsatisfaction with respect to ϕ , when starting from the given region $B_0 \subset S$.

Remark 1. It is easy to obtain that: if the system described in Eq. (1) fails to achieve regional satisfaction with respect to ϕ , then its trajectories from the initial region B_0 either satisfy ϕ only partially or are strongly unsatisfactory.

Definition 6 (Specification Repair). *Given a specification ϕ , the system described in Eq. (1) whose initial region is B_0 has strong regional unsatisfactory w.r.t. ϕ . Then, a new generated specification ϕ^* is such that for Eq. (3) the set $\mathcal{X}_{\phi^*} := \{x_0 \in B_0 \mid \exists k \in \{0, \dots, k^*\}, x_k \models \phi^*\}$ is equal to B_0 . We say this specification ϕ^* is the specification repair of ϕ for this system described in Eq. (1) whose initial region is B_0 .*

Problem. Let the system described in Eq. (1), whose initial region is B_0 , not have regional satisfaction with respect to an STL formula ϕ . Based on the dynamics of this system and the parameters of STL, generate a new specification ϕ^* such that this system, whose initial region is B_0 , can have regional satisfaction w.r.t. the new specification ϕ^* .

In this problem, we face two key challenges:

C1 Determining when a system described in Eq. (1) whose initial region is B_0 , does not have regional satisfaction w.r.t. specification ϕ ?

C2 Determining an adaptation to the specification that ensures regional satisfaction w.r.t. specification ϕ ?

For **C1**, since the initial region B_0 for the system described in Eq. (1) is a dense and uncountable set, it is impossible to check the trajectory of this system starting from every point in B_0 . Additionally, it is difficult to degrade the specification based on the limited information (**C2**). The following section will try to address these challenges.

5 Methodology

We propose here a method to verify whether a system described in Eq. (1) whose initial region is B_0 has regional satisfaction w.r.t. specification ϕ , and to degrade when it does not.

For Eq. (1), the satisfaction of the specification can be verified by evaluating the function h values corresponding to the system's state variables. Typically, for systems with a continuous state space, evaluating the function h can be reduced to a reachability analysis. This involves determining whether the state variables can enter some specified value ranges while meeting the given temporal requirements. In this section, we adopt the Lipschitz constant $\mathcal{L}$ of the vector field of Eq. (1) to verify the specification.

For the dynamical system described in Eq. (1), it has a Lipschitz constant $\mathcal{L}$, such that for any x and x^* in $\mathcal{S}$, $\|f(x_k) - f(x_k^*)\| \leq \mathcal{L}\|x_k - x_k^*\|$. The Lipschitz

constant $\mathcal{L}$ can be used to quantify the distance between two trajectories of Eq. (1), each starting from a different initial point. According to Eq. (3),

$$\begin{aligned} \|x_{k+1} - x_{k+1}^*\| &\leq h_t \|f(x_k) - f(x_k^*)\| + \|x_k - x_k^*\| \\ &\leq (\mathcal{L}h_t + 1)\|x_k - x_k^*\| \leq (\mathcal{L}h_t + 1)^{k+1}\|x_0 - x_0^*\|. \end{aligned} \quad (4)$$

This property is useful for quickly estimating the reachable region of Eq. (1) by sampling a limited number of trajectories from a small set of initial points, as shown in Algorithm 1. In detail, we can partition the initial region into many cells, as shown in Step 1, and then sample the central point of each cell as the initial point, as indicated by the blue region in Fig. 1. For each initial point, the system described in Eq. (1) generates the associated trajectory. According to the specification ϕ , we find the point in each trajectory, as shown in Step 5, satisfying that the value of the function h at this point is the closest to the value region of the same function in the specification ϕ . Alongside, the relevant maximum time moment is recorded. Using the maximum distance between any point in a cell and its centre in Step 2, the inequality described in Eq. (4) is applied to estimate the reachable region, which calculation can be represented as the Step 9, as illustrated by the yellow circle in Fig. 1. By comparing the overlap between the specification region and the estimated region, it is possible to quickly determine whether a system starting from the initial region B_0 has regional satisfaction (i.e., **C1** can be solved). Through Step 7 in Algorithm. 1, the new specification is generated, ensuring that trajectories from B_0 pass through the estimated region while meeting the time constraints of specification ϕ , which **C2** is addressed.

Remark 2. In this paper, we restrict our analysis to the known-model setting. Nonetheless, the framework naturally extends to the unknown-model case, where an upper bound on the LC can be estimated directly from data using methods based on extreme value theory [25, 24]. This upper bound may result in an estimated region that is excessively large.

6 Case Study

We present two case studies to demonstrate the capability of our proposed method in estimating the reachability region based on the given specification and initial region, and subsequently degrading the specification. Our first case analyses a workload management system for next-generation communication base stations with dynamic loads. Our second case examines the dynamics of a nonlinear car-like model in response to environmental changes. Both use the Lipschitz constant to weaken specifications and ensure compliance. The computations are performed on a laptop with an Apple M1 Pro CPU and 32 GB RAM, running MATLAB R2024a. The source code can be found at <https://github.com/zcode-pa/SRofCDS>.

Example 1 (Workload management system).

Algorithm 1 Automated Weakening Specification of Continuous Dynamical System

Require: Initial region B_0 , STL specification ϕ , time interval $I := [t_{\text{start}}, t_{\text{end}}]$, partition parameter (n, m) , time step h_t , LC $\mathcal{L}$

- 1: Partition B_0 into $n \times m$ cells, ensuring each cell $B_0^i, i \in \{1, \dots, n \times m\}$ has equal width and height
- 2: Quantify $r := \max_{x \in B_0^i} \|x - x_c^i\|$, where x_c^i is central point of B_0^i
- 3: **for** $i = 1 : n \times m$ **do**
- 4: Simulate the trajectory starting from x_c^i
- 5: Locate the trajectory point $x_c^{i,*}$ whose distance to specification ϕ is minimised and record the moment
- 6: **end for**
- 7: Take the maximum moment t^*
- 8: Calculate $(\mathcal{L}h_t + 1)^{k^*}$, where $k^* = \frac{t^* - t_{\text{start}}}{h_t}$
- 9: At each point $x_c^{i,*}, i = 1, \dots, n \times m$, generate a circle of radius $r(\mathcal{L}h_t + 1)^{k^*}$; denote this circle by $\mathcal{B}_i$
- 10: Generate the new region $\mathcal{B}$ as the union of all circles $\mathcal{B}_i, i = 1, \dots, n \times m$, where $\mathcal{B} := \bigcup_{i=1}^{n \times m} \mathcal{B}_i$.
- 11: Compare with the region of specification ϕ , denoted this region by $\mathcal{B}_\phi$
- 12: Replace $\mathcal{B}_\phi$ with the new region $\mathcal{B}$

Consider a workload management system for next-generation Base Stations (BSs) [26]. Assume each BS in active mode has the same load dynamics. The dynamics of each BS in active mode of the wireless network are

$$\dot{l}_i = f(l_i) + \sum_{j=1, j \neq i}^N a_{ij} g(l_i, l_j), \quad (5)$$

where N is the number of BSs, $l_i := \frac{\sum_{u=1}^{U_i(t)} \hat{B}_{u,i}}{B_i(t)}$ is the load of BS station i at time t and continuous-time variable, $B_i(t)$ is the number of physical resource blocks (PRBs) of BS i can assign to users at time t , $\hat{B}_{u,i} := Q_u / w(P_{u,i}, N_0)$ is the minimum number of PRBs that should be assigned to user u , Q_u is the minimum data rate that needs to be provided, $w(P_{u,i}, N_0)$ is the rate function, $P_{u,i}$ is the power that user u receives from BS i and N_0 is the noise power. f_i is the load dynamics in BS i , and $g(\cdot, \cdot)$ is the offloading dynamics. The stable state is $l_i = 1$.

Here, we consider the model with two BSs: $\dot{l}_1 = (1 - l_1^2) + (l_1 - l_2)$, $\dot{l}_2 = (1 - l_2^2) + (l_2 - l_1)$. The original specification can be the requirement $\mathbf{F}_{[t_0, t_c]}(1 - \epsilon \leq l_1 \leq 1 + \epsilon \wedge 1 - \epsilon \leq l_2 \leq 1 + \epsilon)$, where t_0 is the moment when many users are connecting to the network. This specification dictates that the system must converge to the stable state's neighbourhood within the time interval $[t_0, t_c]$, ensuring eventual stabilisation..

As shown in the phase portrait of system (5) in Fig. 2 (a) and (b), the estimated reachability regions do not have any overlaps with the destination region of $\mathbf{F}_{[0, 0.4]}(\bigwedge_{i=1}^2 0.9 \leq l_i \leq 1.1)$, i.e., the trajectories of system (5) starting from

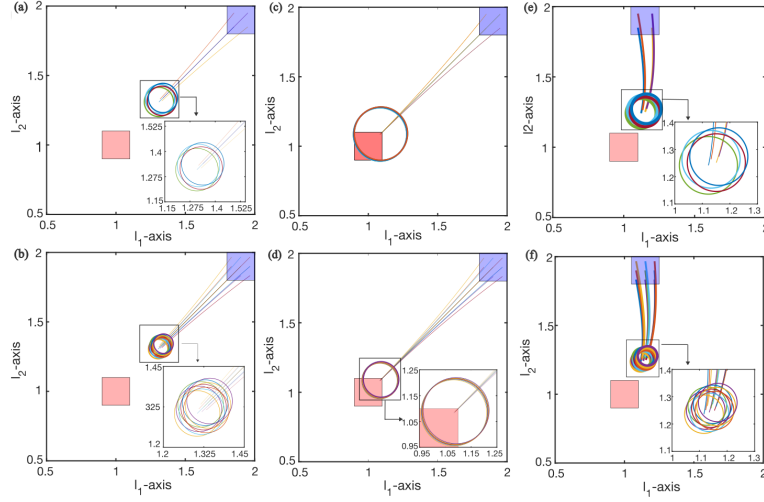


Fig. 2. Reachability regions estimated from different initial regions under different partitions: 2×2 (a), (c), (e), and 3×3 (b), (d), (f), where the purple regions represent the initial regions, the red regions denote the destination regions, and the circles represent the estimated regions.

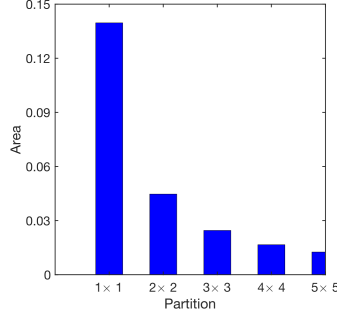


Fig. 3. The area of the estimated reach regions under varying partitions of the initial region.

initial region $B_0 = [1.8, 2] \times [1.8, 2]$ cannot satisfy this specification. Thus, the destination region specified in the original specification needs to be combined with the estimated reachability region using a disjunction, depicted as circles in Fig. 2 (a) and (b), where the radii are 0.1054 and 0.0703, respectively. This ensures that the trajectory from every point within the initial region B_0 satisfies the updated specification. For another initial region $B_0^* = [1.05, 1.25] \times [1.8, 2]$, the similar results can be found in Fig. 2 (e) and (f). In addition, the estimated reachability region based on the specification with large time interval $\mathbf{F}_{[0,1]}(\wedge_{i=1}^2 0.9 \leq l_i \leq 1.1)$ can be found in Fig. 2 (c) and (d). Besides that, in Fig. 3, under finer partitions of the initial region, the area of the estimated region becomes smaller and smaller.

Example 2 (Non-linear car model).

Consider the non-linear car-like model as presented in [23]:

$$\begin{aligned}\dot{x} &= v \cos \theta, \quad \dot{y} = v \sin \theta, \\ \dot{\theta} &= v\kappa, \quad \dot{v} = u_v, \quad \dot{\kappa} = u_\kappa,\end{aligned}\tag{6}$$

where $(x, y, \theta, v, \kappa)^T \in \mathbb{R}^5$ is state vector, consisting of its planar position (x, y) [m], its orientation θ [rad], speed v [m/s], and curvature κ [m⁻¹]. The control input $u = (u_v, u_\kappa)^T$ is two-dimensional and consists of the derivatives of speed and curvature, respectively. In our simulation, we let the control input $u = (0, 0)^T$, i.e., v and κ are constant, respectively. The initial value of the speed and curvature decides the derivative of orientation.

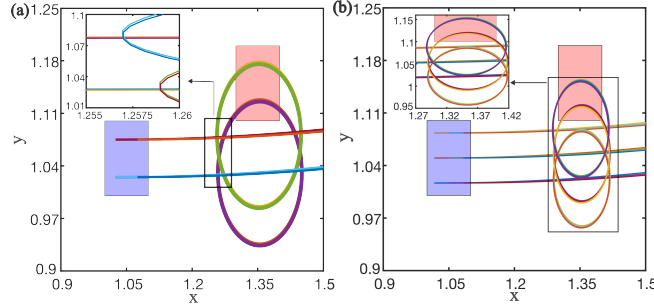


Fig. 4. Reachability regions estimated from the initial condition with different partitions: (a) 2×2 and (b) 3×3 .

In addition, if the external environment, such as wind conditions, varies, the model is presented as follows:

$$\begin{aligned}\dot{x} &= v \cos \theta + w_x, \quad \dot{y} = v \sin \theta + w_y, \\ \dot{\theta} &= v\kappa + w_\theta, \quad \dot{v} = u_v, \quad \dot{\kappa} = u_\kappa,\end{aligned}\tag{7}$$

where variable $w_x : \mathbb{R}^+ \rightarrow \mathbb{R}$, $w_y : \mathbb{R}^+ \rightarrow \mathbb{R}$, and $w_\theta : \mathbb{R}^+ \rightarrow \mathbb{R}$. The evolution of w_x , w_y , and w_θ is governed by system (2). In this example, we only consider the constant wind, $w_x = 0.3$, $w_y = 0.1$, and $w_\theta = 0.1$.

The estimated reachability regions shown in Fig. 4 and 5 are for specifications:

$$\begin{aligned}\mathbf{F}_{[0,1]} &(1.3 \leq x \leq 1.4 \wedge 1.1 \leq y \leq 1.2) \\ \mathbf{F}_{[0,0.4]} &(1.3 \leq x \leq 1.4 \wedge 1.1 \leq y \leq 1.2)\end{aligned}$$

These two specifications indicate that the non-linear car eventually reaches the region defined by $1.3 \leq x \leq 1.4$ and $1.1 \leq y \leq 1.2$ within time intervals $[0, 1]$ and $[0, 0.4]$, respectively. They all demonstrate that the estimated regions (i.e., circles) obtained with finer partitions are smaller and more precise compared to

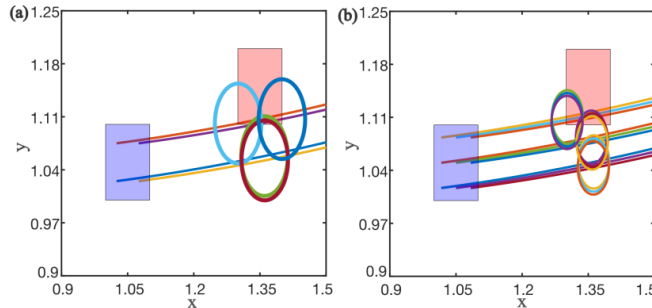


Fig. 5. Reachability regions estimated from the initial condition under wind condition, with different partitions: (a) 2×2 and (b) 3×3 .

those from coarser partitions. By applying a disjunction operation to the estimated regions, the newly modified specification can be satisfied by all trajectories starting from the relevant initial region.

Remark 3. The two examples demonstrate how reachability estimation using Lipschitz constants can be applied to achieve specification degradation. This approach ensures that system trajectories, starting from an initial state-space region, can satisfy the newly modified specifications, even in the face of environmental changes.

7 Conclusion and Future Works

We proposed a novel method for gracefully degrading reachability specifications of continuous dynamical systems operating in the dynamical environment, which has not been studied before. The proposed approach enables systems to adapt their requirements dynamically by sampling points within the initial state space, estimating reachability regions using the system's Lipschitz constant, and employing a disjunction operation to augment the original specification. Validation of a workload management system for next-generation base stations, along with a non-linear car-like model, demonstrates the effectiveness and practicality of the method. In the future, we aim to develop a formal framework that integrates online reachability analysis with specification transformation, enabling provably correct and minimally conservative requirement adaptations in real time. We also plan to extend this framework to control continuous dynamical systems with dense feasible control inputs and to handle more complex specifications, including reach-avoid tasks and specifications featuring nested temporal operators.

8 Acknowledgement

The EPSRC supports this work through the Communications Hub for Empowering Distributed Cloud Computing Applications and Research (CHEDDAR) under grants EP/X040518/1 and EP/Y037421/1.

References

1. Alrajeh, D., Benjamin, P., Uchitel, S.: Adaptation 2: Adapting specification learners in assured adaptive systems. In: 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE). pp. 1347–1352. IEEE (2021)
2. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. *Theoretical computer science* **138**(1), 3–34 (1995)
3. Buckworth, T., Alrajeh, D., Kramer, J., Uchitel, S.: Adapting specifications for reactive controllers. In: 2023 IEEE/ACM 18th Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). pp. 1–12. IEEE (2023)
4. Chen, X., Abraham, E., Sankaranarayanan, S.: Taylor model flowpipe construction for non-linear hybrid systems. In: 2012 IEEE 33rd Real-Time Systems Symposium. pp. 183–192. IEEE (2012)
5. Chinneck, J.W.: Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods, vol. 118. Springer Science & Business Media (2007)
6. Chu, S., Koe, J., Garlan, D., Kang, E.: Integrating graceful degradation and recovery through requirement-driven adaptation. In: Proceedings of the 19th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. pp. 122–132 (2024)
7. Chu, S., Shedden, E., Zhang, C., Meira-Góes, R., Moreno, G.A., Garlan, D., Kang, E.: Runtime resolution of feature interactions through adaptive requirement weakening. In: 2023 IEEE/ACM 18th Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). pp. 115–125. IEEE (2023)
8. D’Ippolito, N., Braberman, V., Kramer, J., Magee, J., Sykes, D., Uchitel, S.: Hope for the best, prepare for the worst: multi-tier control for adaptive systems. In: Proceedings of the 36th International Conference on Software Engineering. pp. 688–699 (2014)
9. DiStefano III, J.: Dynamic systems biology modeling and simulation. Academic Press (2015)
10. Ghosh, S., Sadigh, D., Nuzzo, P., Raman, V., Donzé, A., Sangiovanni-Vincentelli, A.L., Sastry, S.S., Seshia, S.A.: Diagnosis and repair for synthesis from signal temporal logic specifications. In: Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control. pp. 31–40 (2016)
11. Giorgetti, N., Pappas, G.J., Bemporad, A.: Bounded model checking of hybrid dynamical systems. In: Proceedings of the 44th IEEE Conference on Decision and Control. pp. 672–677. IEEE (2005)
12. Greenstreet, M.R., Mitchell, I.: Reachability analysis using polygonal projections. In: International workshop on hybrid systems: Computation and control. pp. 103–116. Springer (1999)
13. Grindrod, P., Higham, D.J.: A dynamical systems view of network centrality. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **470**(2165), 20130835 (2014)

14. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What's decidable about hybrid automata? In: Proceedings of the twenty-seventh annual ACM symposium on Theory of computing. pp. 373–382 (1995)
15. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Movement imitation with nonlinear dynamical systems in humanoid robots. In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292). vol. 2, pp. 1398–1403. IEEE (2002)
16. Jacobsson, K.: Dynamic modeling of Internet congestion control. Ph.D. thesis, KTH (2008)
17. Jagtap, P., Soudjani, S., Zamani, M.: Formal synthesis of stochastic systems via control barrier certificates. *IEEE Transactions on Automatic Control* **66**(7), 3097–3110 (2020)
18. Makino, K., Berz, M.: Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics* **6**, 239–316 (2003)
19. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems. pp. 152–166. Springer (2004)
20. Qu, Z.: Cooperative control of dynamical systems: applications to autonomous vehicles. Springer Science & Business Media (2009)
21. Rigatos, G.G.: Modelling and control for intelligent industrial systems. *Adaptive Algorithms in Robotics and Industrial Engineering* (2011)
22. Tabuada, P.: Verification and control of hybrid systems: a symbolic approach. Springer Science & Business Media (2009)
23. Webb, D., Berg, J., Kinodynamic, R.: Optimal motion planning for systems with linear differential constraints [J]. *arXiv preprint arXiv: 1205.5088* (2012)
24. Weng, T.W., Zhang, H., Chen, P.Y., Yi, J., Su, D., Gao, Y., Hsieh, C.J., Daniel, L.: Evaluating the robustness of neural networks: An extreme value theory approach. In: International Conference on Learning Representations (2018), <https://openreview.net/forum?id=BkUHMZ0b>
25. Wood, G.R., Zhang, B.: Estimation of the Lipschitz constant of a function. *Journal of Global Optimization* **8**(1), 91–103 (1996)
26. Zou, M., Guo, W.: Cascade network stability of synchronized traffic load balancing with heterogeneous energy efficiency policies. In: GLOBECOM 2024 - 2024 IEEE Global Communications Conference. pp. 1083–1088 (2024). <https://doi.org/10.1109/GLOBECOM52923.2024.10901638>
27. Zudaire, S.A., Nahabedian, L., Uchitel, S.: Assured mission adaptation of UAVs. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* **16**(3-4), 1–27 (2022)