2019 American Control Conference (ACC)
Philadelphia, PA, USA, July 10-12, 2019

Arithmetic-Geometric Mean Robustness for Control from Signal Temporal Logic Specifications

*Noushin Mehdipour[1], *Cristian-Ioan Vasile[2] and Calin Belta[1]

Abstract— **We present a new average-based robustness for Signal Temporal Logic (STL) and a framework for optimal control of a dynamical system under STL constraints. By averaging the scores of different specifications or subformulae at different time points, our definition highlights the frequency of satisfaction as well as how robustly each specification is satisfied. Its usefulness in control synthesis problems is illustrated through case studies.**

I. INTRODUCTION

Formal methods have been recently used to express system behavior under complex temporal requirements, verify whether the system execution meets the desired requirements, or control the system to satisfy desirable specifications [1]. Temporal Logics including Linear Temporal Logics (LTL) [2], Metric Temporal Logic (MTL) [3], Signal Temporal Logic (STL) [4] and Time Window Temporal Logic (TWTL) [5] allow precise description of system properties over time. STL is equipped with qualitative and quantitative semantics, meaning that it not only can assess whether the system execution meets the desired requirements but also provides a measure of how well requirements are met, also known as robustness. As a result, STL has been widely used for many control purposes including path planning and motion planning [6], [7] or synthesis problems [8]. Higher robustness score shows stronger satisfaction of the desired specifications. Therefore, it is desirable to maximize the robustness to improve system behavior satisfying desired temporal specifications.

The traditional robustness introduced in [9] is non-convex and non-differentiable; therefore, it is not possible to use powerful optimization techniques to maximize it. Previous works for control under STL constraints focused on using heuristic algorithms or encoding constraints as Mixed Integer Linear Programming (MILP). Heuristic optimization approaches such as Particle Swarm Optimization, Simulated Annealing and Rapidly Exploring Random Trees (RRTs) were used for synthesis, falsification and control problems [10], [11], [12]. Heuristic approaches do not require a smooth objective function; however, these algorithms do not always provide a guarantee to find the optima and have many user-defined parameters that need to be set in advance. Encoding temporal logic specifications as linear and boolean constraints was studied in [13], [14] and MILP optimization solvers such as Gurobi were used to solve the control synthesis problem. The most critical issue with MILPs is that they do not scale well as the number of variables increases, resulting in an NP-complete problem. Therefore, this approach could fail when solving problems with many variables or complex temporal constraints. Moreover, MILP implementations require all constraints (including the system dynamics in the control problem) to be linear. As a result, nonlinear dynamics must be linearized, if linearizable, which involves approximation.

Recently, there have been efforts to smooth the robustness function in order to use gradient-based optimization algorithms. In [15], [16], authors used smooth approximations of maximum and minimum functions to define a smooth robustness in order to solve a control problem. Even though these works solved the non-differentiability issue, the resulting smooth approximation had errors compared to the traditional robustness, meaning that positive robustness did not necessarily mean satisfaction of the specification unless it was greater than a pre-defined threshold. The main drawback of these works is that traditional robustness is defined by the most critical point (most satisfaction or most violation). In [17], authors defined average STL robustness for continuous-time signals and defined positive and negative robustness to solve a falsification problem. Authors in [18] described MTL as linear time-invariant filters and used the average robustness for monitoring purposes. [19] improved robustness for discrete signals by defining Discrete Average Space Robustness, and removed its nonsmoothness by approximating to a simplified version. These works refined robustness only for temporal operators while using traditional maximum and minimum functions for other operators.

Our main contribution of this paper is proposing a new average-based robustness, which we call Arithmetic-Geometric Mean (AGM) robustness. This new quantitative semantics uses arithmetic and geometric means to take into account the robustness degrees for all the subformulae and at every time point, and not just the most satisfying or violating ones. As a result, our definition rewards policies that satisfy the requirements at more time steps and with higher scores. We show that this novel definition also provides a better margin in which the specification is still satisfied when external disturbance or system perturbation exists. The advantages of our new definition are illustrated through case studies and compared to smooth approximation and MILPs methods.

II. PRELIMINARIES

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a real function. We define $[f]_+ = \begin{cases} f & f > 0 \\ 0 & \text{otherwise} \end{cases}$ and $[f]_- = -[-f]_+$, where $f = [f]_+ + [f]_-$.

*These authors contributed equally. This work was partially supported at Boston University by the National Science Foundation under grants IIS-1723995, CPS-1446151, and CMMI-1400167.[1]Noushin Mehdipour (noushinm@bu.edu), Calin Belta (cbelta@bu.edu) are with the Division of Systems Engineering at Boston University, Boston, and [2]Cristian-Ioan Vasile (cvasile@mit.edu) is with the Laboratory for Information and Decision Systems at Massachusetts Institute of Technology, Cambridge, MA, USA.

978-1-5386-7926-5/$31.00 ©2019 AACC

A. Signal Temporal Logics (STL)

STL was introduced in [4] to monitor temporal properties of real-valued signals. Consider a discrete time sequence $\tau := \{t_k | k \in \mathbb{Z}_{\geq 0}\}$. A *signal* S is a function $S : \tau \to \mathbb{R}^n$ that maps each time point $t_k \in \tau$ to an n-dimensional vector of real values $S[t_k]$, with s_i being its ith component. Assume $[a, b]$ is the set of all $t_k \in \tau$ starting from a up to b, with $a, b \in \tau;\ b > a \geq 0$. STL Syntax is defined as:

$$\varphi := \top \mid \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathbf{U}_{[a,b]} \varphi_2, \tag{1}$$

where $\top$ is the logical *True*, μ is a *predicate*, $\neg$, $\wedge$ are the Boolean *negation* and *conjunction* operators, respectively, and $\mathbf{U}$ is the temporal *until* operator. Logical *False* is $\bot := \neg\top$. Other Boolean and temporal operators are defined as $\varphi_1 \vee \varphi_2 := \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\mathbf{F}_{[a,b]}\varphi := \top\mathbf{U}_{[a,b]}\varphi$, $\mathbf{G}_{[a,b]}\varphi := \neg\mathbf{F}_{[a,b]}\neg\varphi$. The temporal operator *Finally* or *eventually* ($\mathbf{F}_{[a,b]}\varphi$) states that "at some time point in $[a, b]$ the specification φ must be *True*"; while *globally* or *always* ($\mathbf{G}_{[a,b]}\varphi$) states that "$\varphi$ must be *True* at all times in $[a, b]$". The *until* operator ($\varphi_1\mathbf{U}_{[a,b]}\varphi_2$) states that "$\varphi_2$ must become *True* at some time point within $[a, b]$ and φ_1 must be always True prior to that". STL qualitative semantics shows *whether* a signal S satisfies a given specification φ at time t, i.e., $S[t] \models \varphi$ or not, i.e., $S[t] \not\models \varphi$, and its quantitative semantics, known as *robustness*, measures *how much* the specification is satisfied or violated. We denote the robustness for a specification φ with respect to signal S at time t as $\rho(\varphi, S, t)$ and refer to it as traditional robustness. For details on calculating traditional robustness, please refer to [9]. In [15], [16], the non-differentiable min and max functions in the traditional robustness are replaced by smooth approximations, to which we refer as approximation robustness $\tilde{\rho}$. For all robustness definitions, when time is not mentioned, satisfaction at time 0 is considered.

III. PROBLEM STATEMENT

Consider a discrete-time dynamical system given by:

$$\begin{aligned} &q[k+1] = f(q[k], u[k]), \\ &q[0] = q_0, \end{aligned} \tag{2}$$

where $q[k] \in \mathbf{Q} \subseteq \mathbb{R}^n$ is state of the system and $u[k] \in \mathbf{U} \subseteq \mathbb{R}^m$ is control input at kth time step $k \in \mathbb{Z}_{\geq 0}$; $q_0 \in \mathbf{Q}$ is the initial state and f is a function representing the dynamics of the system. Given the initial state q_0 and control sequence $u = \{u[0]u[1]...\}$, system trajectory $q = \{q[0]q[1]q[2]...\}$ is generated using (2); which we denote by $\langle q, u\rangle$. Consider a cost function $J(u[k], q[k+1])$ representing the cost of applying the control input $u[k]$. Assume system temporal requirements are given by a STL formula ϕ with a time horizon T, which is the largest time step for which signal values are needed in order to compute the robustness for the current time point. The control synthesis problem can be formulated as determining a control policy $u^* = \{u^*[0]u^*[1]...u^*[T-1]\}$ such that the system trajectory satisfies the STL specification ϕ while optimizing the associated cost:

$$\begin{aligned} u^* = \operatorname{argmin}_u &\sum_{k=0}^{T-1} J(u[k], q[k+1]), \\ \text{s.t.} \quad &\langle q, u\rangle \models \phi. \end{aligned} \tag{3}$$

Previous works used heuristic algorithms, MILP encoding, and gradient ascent to optimize (3) based on traditional and smooth robustness scores. The main shortcoming of the traditional robustness is that it only considers the robustness of the most satisfying or violating part of the specification. We address this limitation by defining a new version of robustness.

IV. ARITHMETIC-GEOMETRIC MEAN (AGM) ROBUSTNESS

We define a novel robustness η based on arithmetic and geometric means instead of the max and min functions in the traditional definition. We show that our normalized signed robustness $\eta \in [-1, 1]$ provides a better understanding of system properties, where $\eta \in (0, 1]$ corresponds to satisfaction of the specification, $\eta \in [-1, 0)$ shows violation, and $\eta = 0$ indicates inconclusiveness. Moreover, $|\eta|$ is a measure of how well the specification is satisfied or violated.

Consider a discrete time series $\tau := \{t_k | k \in \mathbb{Z}_{\geq 0}\}$. Signal S is a function $S : \tau \to \mathbb{R}^n$ that maps each time point $t_k \in \tau$ to an n-dimensional vector of real values $S[t_k]$, with s_i being its ith element. Throughout the definitions and proofs, we assume that we have bounded signals, and all their components are normalized to the interval $[-1, 1]$.

Definition 1 (AGM Robustness): Let $S : \tau \to [-1, 1]^n$ and $\varphi : s_i - \pi \geq 0$ where $\pi \in [-1, 1]$. The normalized signed AGM robustness $\eta(\varphi, S, t)$ with respect to the signal S at time t is defined as:

$$\begin{aligned} \eta(\top, S, t) &:= 1 \\ \eta(\varphi, S, t) &:= \frac{1}{2}(s_i[t] - \pi), \\ \eta(\neg\varphi, S, t) &:= -\eta(\varphi, S, t). \end{aligned} \tag{4}$$

For combination of other boolean and temporal operators in a time interval $[a, b]$, AGM robustness is recursively defined using (5) and (6), where $[a, b] = \{t_k | t_k, a, b \in \tau; a \leq t_k \leq b;\ \ b > a \geq 0\}$ and N is the number of time points in $[a, b]$. Algorithms 1-4 determine satisfaction or violation of the specification with respect to the signal S, as well as the normalized signed AGM robustness.

Theorem 1 (Soundness): The AGM robustness is sound, meaning that a trajectory with strictly positive robustness is satisfying the specification, and a trajectory with strictly negative robustness is violating it:

$$\begin{aligned} \eta(\varphi, S, t) > 0 \Leftrightarrow \rho(\varphi, S, t) > 0 \Rightarrow S \models \varphi, \\ \eta(\varphi, S, t) < 0 \Leftrightarrow \rho(\varphi, S, t) < 0 \Rightarrow S \not\models \varphi. \end{aligned} \tag{7}$$

Due to brevity, all proofs are omitted, but can be found in [20].

Proposition 1: Let S be a signal and ϕ a STL formula. If $\eta(\phi, S, t) = 1$, then $\eta(\varphi, S, t_k) = 1$, similarly, if $\eta(\phi, S, t) = -1$, then $\eta(\varphi, S, t_k) = -1$ for all subformulae φ of ϕ and appropriate times t_k as given by (5), (6).

A. Performance Properties

Property 1 (Smoothness and Gradient): The AGM robustness $\eta(\phi, S, t)$ is smooth in $S \in [-1, 1]^n$ almost everywhere except on the satisfaction boundaries $\rho(\varphi, S, t_k) = 0$, where φ is a subformula of ϕ, and appropriate times t_k as given in (5) and (6). Moreover, the gradient of η with respect to the elements of S that are part of ϕ's predicates is non-zero wherever it is smooth.

$$
\begin{aligned}
&\eta(\varphi_1 \wedge ... \wedge \varphi_m, S, t \mid \forall i \in [1,...,m]\ .\ \eta(\varphi_i, S, t) > 0) := \sqrt[m]{\prod_{i=1,...,m} (1 + \eta(\varphi_i, S, t))} - 1 \\
&\eta(\varphi_1 \vee ... \vee \varphi_m, S, t \mid \exists i \in [1,...,m]\ .\ \eta(\varphi_i, S, t) > 0) := \tfrac{1}{m} \sum_{i=1,...,m} [\eta(\varphi_i, S, t)]_+ \\
&\eta(\mathbf{G}_{[a,b]}\varphi, S, t \mid \forall t'_k \in [t+a, t+b]\ .\ \eta(\varphi, S, t'_k) > 0) := \sqrt[N]{\prod_{t'_k \in [t+a,t+b]} \big(1 + \eta(\varphi, S, t'_k)\big)} - 1 \\
&\eta(\mathbf{F}_{[a,b]}\varphi, S, t \mid \exists t'_k \in [t+a, t+b]\ .\ \eta(\varphi, S, t'_k) > 0) := \tfrac{1}{N} \sum_{t'_k \in [t+a,t+b]} [\eta(\varphi, S, t'_k)]_+
\end{aligned}
\tag{5}
$$

$$
\begin{aligned}
&\eta(\varphi_1 \wedge ... \wedge \varphi_m, S, t \mid \exists i \in [1,...,m]\ .\ \eta(\varphi_i, S, t) \le 0) := \tfrac{1}{m} \sum_{i=1,...,m} [\eta(\varphi_i, S, t)]_- \\
&\eta(\varphi_1 \vee ... \vee \varphi_m, S, t \mid \forall i \in [1,...,m]\ .\ \eta(\varphi_i, S, t) \le 0) := -\sqrt[m]{\prod_{i=1,...,m} (1 - \eta(\varphi_i, S, t))} + 1 \\
&\eta(\mathbf{G}_{[a,b]}\varphi, S, t \mid \exists t'_k \in [t+a, t+b]\ .\ \eta(\varphi, S, t'_k) \le 0) := \tfrac{1}{N} \sum_{t'_k \in [t+a,t+b]} [\eta(\varphi, S, t'_k)]_- \\
&\eta(\mathbf{F}_{[a,b]}\varphi, S, t \mid \forall t'_k \in [t+a, t+b]\ .\ \eta(\varphi, S, t'_k) \le 0) := -\sqrt[N]{\prod_{t'_k \in [t+a,t+b]} \big(1 - \eta(\varphi, S, t'_k)\big)} + 1
\end{aligned}
\tag{6}
$$

Algorithm 1: AGM ROBUSTNESS FOR AND

Input: STL Formula $\phi = \varphi_1 \wedge \varphi_2 \wedge ...\varphi_m$; Signal S
Output: AGM Robustness $\eta(\phi, S)$

Find $\eta(\varphi_i, S)$ for $i = \{1, 2, ..., m\}$ using (4);
If $ANY(\eta(\varphi_i, S) \le 0)$, then $S \not\models \phi$,
$\eta(\phi, S | S \not\models \phi) := \frac{1}{m} \sum_{i=1,...,m} [\eta(\varphi_i, S)]_-$;
Else: $S \models \phi$,
$\eta(\phi, S | S \models \phi) := \sqrt[m]{\prod_{i=1,...,m} (1 + \eta(\varphi_i, S))} - 1.$

Algorithm 2: AGM ROBUSTNESS FOR OR

Input: STL Formula $\phi = \varphi_1 \vee \varphi_2 \vee ...\varphi_m$; Signal S
Output: AGM Robustness $\eta(\phi, S)$

Find $\eta(\varphi_i, S)$ for $i = \{1, 2, ..., m\}$ using (4);
If $ANY(\eta(\varphi_i, S) > 0)$, then $S \models \phi$,
$\eta(\phi, S | S \models \phi) := \frac{1}{m} \sum_{i=1,...,m} [\eta(\varphi_i, S)]_+$;
Else: $S \not\models \phi$,
$\eta(\phi, S | S \not\models \phi) = -\sqrt[m]{\prod_{i=1,...,m} (1 - \eta(\varphi_i, S))} + 1.$

Algorithm 3: AGM ROBUSTNESS FOR GLOBALLY

Input: STL Formula $\phi = \mathbf{G}_{[a,b]}\varphi$; Signal S
Output: AGM Robustness $\eta(\phi, S)$

Find $\eta(\varphi, S[t'_k])$ for time points $t'_k \in [a, b]$ using (4);
If $ANY(\eta(\varphi, S[t'_k]) \le 0)$, then $S \not\models \phi$,
$\eta(\phi, S | S \not\models \phi) := \frac{1}{N} \sum_{t'_k \in [a,b]} [\eta(\varphi, S[t'_k])]_-$;
Else: $S \models \phi$,
$\eta(\phi, S | S \models \phi) := \sqrt[N]{\prod_{t'_k \in [a,b]} \big(1 + \eta(\varphi, S[t'_k])\big)} - 1.$

Algorithm 4: AGM ROBUSTNESS FOR EVENTUALLY

Input: STL Formula $\phi = \mathbf{F}_{[a,b]}\varphi$; Signal S
Output: AGM Robustness $\eta(\phi, S)$

Find $\eta(\varphi, S[t'_k])$ for time points $t'_k \in [a, b]$ using (4);
If $ANY(\eta(\varphi, S[t'_k]) > 0)$, then $S \models \phi$,
$\eta(\phi, S | S \models \phi) := \frac{1}{N} \sum_{t'_k \in [a,b]} [\eta(\varphi, S[t'_k])]_+$;
Else: $S \not\models \phi$,
$\eta(\phi, S | S \not\models \phi) := -\sqrt[N]{\prod_{t'_k \in [a,b]} \big(1 - \eta(\varphi, S[t'_k])\big)} + 1.$

Property 2 (Arithmetic and Geometric Means): Comparison between traditional and AGM robustness demonstrates the advantage of our average-based definition. Consider a signal $S \in [0, 1]$ and three subformulae $\varphi_1, \varphi_2, \varphi_3$ with $\rho(\varphi_1, S) = \rho(\varphi_2, S) = \eta(\varphi_1, S) = \eta(\varphi_2, S) = 1$ and $\rho(\varphi_3, S) = \eta(\varphi_3, S) = 0.2$. While traditional robustness uses max function and returns $\rho(\varphi_1 \vee \varphi_2, S) = \rho(\varphi_1 \vee \varphi_3, S) = 1$; AGM definition returns $\eta(\varphi_1 \vee \varphi_3, S) = 0.6$, which is positive showing that the specification is satisfied, but the robustness is less than 1 (highest satisfaction), which is attainable only when both subformulae are maximally satisfied, i.e., $\eta(\varphi_1 \vee \varphi_2, S) = 1$. We now assume $\rho(\varphi_1, S) = \rho(\varphi_2, S) = \eta(\varphi_1, S) = \eta(\varphi_2, S) = 0.2$ and $\rho(\varphi_3, S) = \eta(\varphi_3, S) = 1$. While traditional robustness uses min function and returns $\rho(\varphi_1 \wedge \varphi_2, S) = \rho(\varphi_1 \wedge \varphi_3, S) = 0.2$; AGM definition returns $\eta(\varphi_1 \wedge \varphi_2, S) = 0.2$, which is positive showing the specification is satisfied, but the score is less than $\eta(\varphi_1 \wedge \varphi_3, S) = 0.55$, which shows a stronger satisfaction. For temporal operators, we first examine traditional and AGM robustness for $\phi_1 = \mathbf{F}_{[1,4]}(S > 0.5)$. Using max function in the traditional definition, $\rho(\phi_1, S_i) = 0.5$ for $i = 1, 2, 3$ in Fig. 1 (Left). However, AGM robustness takes a time average over the formula horizon considering all the times the predicate is satisfied; therefore, it returns higher robustness $\eta(\phi_1, S_1) = 0.5$ for S_1 and lower robustness $\eta(\phi_1, S_2) = 0.25$ and $\eta(\phi_1, S_3) = 0.125$ for S_2,S_3, respectively. Basically, AGM robustness for $F_{[a,b]}\varphi$ can be interpreted as "*eventually satisfy φ with the maximum possible satisfaction as early as possible and for as long as possible*". For signals in Fig. 1 (Right) and $\phi_2 = \mathbf{G}_{[0,4]}(S > 0.5)$, traditional robustness with min function returns $\rho(\phi_2, S_1) = 0.5$ for S_1, while giving same robustness $\rho(\phi_2, S_i) = 0.1$ for S_2, S_3. On the other hand,

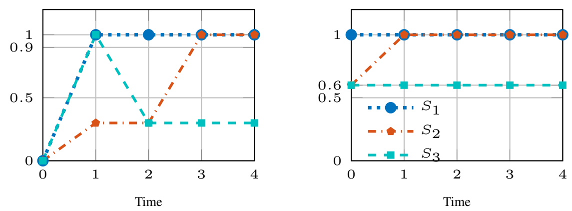

Fig. 1. Signals for traditional and AGM robustness comparison.

AGM definition calculates $\eta(\phi_2, S_1) = 0.5$ for S_1, and lower scores $\eta(\phi_2, S_2) = 0.41$ for S_2 and $\eta(\phi_2, S_3) = 0.1$ for S_3. Thus, the AGM definition for $G_{[a,b]}\varphi$ can be interpreted as "*always satisfy* φ *with the maximum possible satisfaction for all time points in* $[a, b]$".

Property 3 (Performance Under Disturbance): The AGM robustness provides a better satisfaction margin in the presence of disturbance. Consider the specification $\phi_3 = F_{[1,4]}(S > 0.9)$ and signals S_1, S_3 in Fig. 1 (Left), satisfying ϕ_3 with same traditional and AGM robustness $\rho(\phi_3, S_3) = \eta(\phi_3, S_1) = 0.1$. In traditional robustness definition, S_3 only satisfies ϕ_3 at a single time point, i.e., at $t = 1$ with $\rho(\phi_3, S_3[1]) = 0.1$; while for S_1 to have the same score using AGM robustness, $\eta(\phi_3, S_1[t_k]) = 0.1$ for all $t_k \in [1, 4]$. It can be easily shown that applying any disturbance $d > 0.1$ at $t = 1$ to S_3 results in violation of ϕ_3. However, ϕ_3 is still satisfied in S_1 under the same disturbance d, although the satisfaction would become weaker. Therefore, at a same score for the traditional and AGM robustness, satisfaction would hold for larger disturbance using the AGM definition.

B. Normalization

The normalization is not restrictive but is desired to provide a meaningful understanding about satisfaction or violation of a specification, especially when comparing robustness in a formula with predicates defined over different properties or scales. Consider the following specification:

$$\varphi_1 = x_R > 5,\ \varphi_2 = \textit{Battery} > 30,\ \phi = \varphi_1 \wedge \varphi_2.$$

$x_R \in [0, 10]$ is robot position and *Battery* $\in [0, 100]$ shows its battery level. Without normalization, at $x_R = 6$, *Battery* $= 80$, robustness $\rho(\varphi_1, x_R) = 1$ and $\rho(\varphi_2, \textit{Battery}) = 50$. Since the variables are in different scales, unnormalized robustness ρ is not a meaningful measure of the specification ϕ, i.e., we have $\rho(\phi, (x_R, \textit{Battery})) = 1$ for $x_R = 6$, and any *Battery* $= \{31, 32, \ldots, 100\}$. Therefore, not only normalization is not limiting, but is actually essential in practice.

V. CONTROL USING THE AGM ROBUSTNESS

To solve the control synthesis problem (3), we need to find trajectories which satisfy the specification ϕ. A positive robustness provides a margin in which any perturbation up to η does not change satisfaction of the specification. Therefore, we can maximize robustness over all possible control inputs to find not just a satisfying trajectory, but one that has the strongest satisfaction of the specification:

$$\begin{aligned} u^* &= \text{argmax}_u\, \eta(\phi, \langle q, u\rangle) \\ \text{s.t.} &\quad \eta(\phi, \langle q, u\rangle) > 0. \end{aligned} \tag{8}$$

Assume the system dynamics f in (2) is smooth. Based on *Property 1*, we can use advanced optimization algorithms such as gradient ascent techniques to maximize the AGM robustness η, rather than using heuristic methods or MILP encoding. Gradient ascent is an iterative optimization algorithm for finding maximum of a function $F(x)$ by taking steps proportional to its gradient at each iteration i:

$$x^{i+1} \leftarrow x^i + \alpha^i\, \nabla F, \tag{9}$$

where $\nabla F = \frac{\partial F}{\partial x}$ and α is step size. Despite heuristic optimization algorithms which have so many parameters to be set, gradient methods only need to tune the step size α. Due to non-smoothness in η at the satisfaction boundaries, we use proximal stochastic gradient ascent or sub-gradient ascent method with diminishing step size [21]. To initialize gradient ascent, a random control input sequence $u^0 \in \mathbf{U}$ is generated, and the resulting trajectory starting from initial state q_0 is found using system dynamics, which may violate the state constraints or STL specification. The gradient ascent optimization then finds optimal control policy u^* which maximizes AGM robustness function η for given STL constraints ϕ with respect to the system execution $\langle q, u\rangle$. Combining (8) and (3), we can solve a relaxed problem in which we maximize the robustness as much as possible as well as minimizing the penalized cost. The combined fitness function is defined as:

$$\begin{aligned} u^* = \text{argmax}_u &(\eta(\phi, \langle q, u\rangle) - \lambda \sum_{k=0}^{T-1} J(u[k], q[k+1])), \\ \text{s.t.} &\quad \eta(\phi, \langle q, u\rangle) > 0, \\ &q[k+1] = f(q[k], u[k]), \\ &q[0] = q_0, \\ &q[k] \in \mathbf{Q} \subseteq \mathbb{R}^n, \\ &u[k] \in \mathbf{U} \subseteq \mathbb{R}^m, \end{aligned} \tag{10}$$

where λ penalizes the trade-off between maximizing robustness to get the highest STL satisfaction and minimizing the associated cost. Assuming the cost function J is also smooth, gradient ascent can be used to solve the constrained nonlinear optimization problem (10).

VI. CASE STUDIES

We show the applicability and efficacy of our framework for control synthesis problems in linear and nonlinear systems with and without external disturbance, and compare our results with traditional and approximation robustness. To emphasize the differences between the proposed robustness and the traditional and approximation ones, we set $\lambda = 0$ in (10). Gradient ascent simulations are coded in MATLAB and MILP is implemented in Gurobi. Maximum number of iterations for gradient ascent is set to 300.

A. AGM Robustness Versus Traditional Robustness

Example 1: Consider a nonholonomic dynamical system:

$$\begin{aligned} x[k+1] &= x[k] + \cos\theta[k]v[k], \\ y[k+1] &= y[k] + \sin\theta[k]v[k], \\ \theta[k+1] &= \theta[k] + w[k], \end{aligned} \tag{11}$$

and the desired task "*Always* stay in the *Init* for 5 steps *and eventually* visit *Reg1* between $[6, 10]$ steps *and eventually* visit *Reg2* between $[11, 15]$ steps *and Always* avoid *Obs*", formally specified as STL formula:

$$\begin{aligned} \phi_1 = (&\mathbf{G}_{[1,5]}\ \textit{Init}) \wedge (\mathbf{F}_{[6,10]}\ \textit{Reg1}) \\ &\wedge (\mathbf{F}_{[11,15]}\ \textit{Reg2}) \wedge (\mathbf{G}_{[0,15]}\ \neg \textit{Obs}), \end{aligned} \tag{12}$$

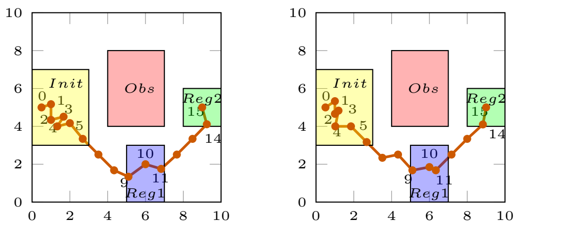

Fig. 2. Trajectories with same maximum traditional robustness $\rho = 1$.

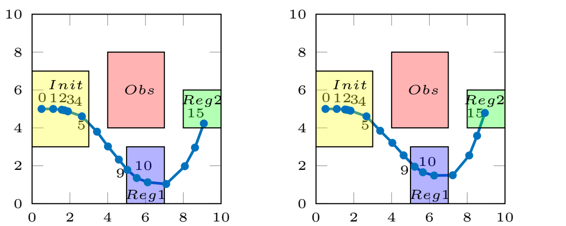

Fig. 3. Trajectory with positive AGM robustness $\eta = 0.138$ (Left) and after more gradient ascent iterations with $\eta = 0.144$ (Right).

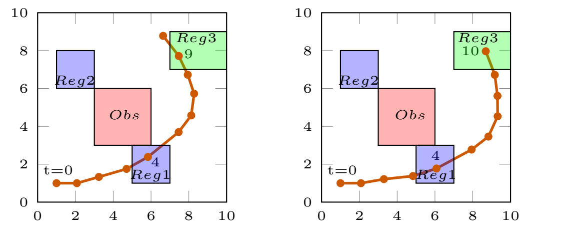

Fig. 4. Trajectory with positive approximation robustness $\tilde{\rho} = 0.424$ ($\rho = 0.461$) (Left) and after more iterations $\tilde{\rho} = 0.731$ ($\rho = 0.778$) (Right)

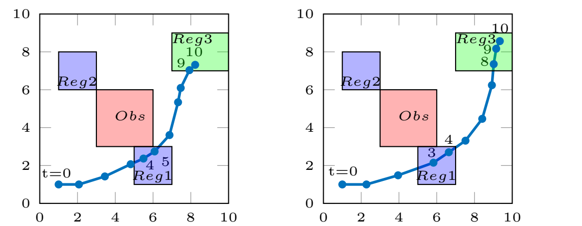

Fig. 5. Trajectory with positive AGM robustness $\eta = 0.130$ (Left) and after more gradient ascent iterations with $\eta = 0.171$ (Right).

where state vector $q = [x, y, \theta]$ indicates the robot position and orientation and $u = [v, w]$ is the input vector.

To maximize the traditional robustness ρ using the MILP implementation, we need to linearize the dynamics. We use feedback linearization to convert the nonlinear dynamics (11) in to a discrete double integrator dynamics [14]. By linearizing dynamics, we can only control x and y directly and robot orientation θ is controlled indirectly. Two optimal trajectories maximizing traditional robustness for the linearized system found by Gurobi with same maximum traditional robustness $\rho = 1$ are shown in Fig. 2. The MILP implementation for STL constraints in ϕ_1 with time horizon $T = 15$ has 95 continuous and 70 integer (binary) variables. It is shown in [15] that MILP does not scale well with the number of integer variables. Therefore, MILP is not applicable for complex specifications with many $\vee$ and $\mathbf{F}$ operators (that must be encoded as binary variables) and long time horizons.

We next maximize AGM robustness η for the nonlinear dynamics (11) using gradient ascent. Fig. 3 shows two trajectories satisfying STL constraints in ϕ_1 obtained in different iterations of gradient ascent. Although both methods generate satisfying trajectories, our proposed approach generates a more smooth trajectory by controlling both robot position and orientation. Moreover, maximum traditional robustness using MILP is obtained when trajectory visits each region with maximum robustness at a single time point ($Reg1$ at $t = 10$, $Reg2$ at $t = 15$) without rewarding the frequency of satisfaction while using the AGM robustness, trajectory with higher robustness visits $Reg1$ *as early as possible and for as long as possible* ($t = 9, 10$); with all subformuale having *maximum possible robustness* (trajectory is toward the center of regions) while always avoiding obstacle.

B. AGM Robustness Versus Approximation Robustness

In [15] authors used Sequential Quadratic Programming (SQP) on the smooth approximation robustness $\tilde{\rho}$ of MTL specifications and showed it was more time efficient than MILP approach. However, smooth approximation was within a predefined error δ of the traditional robustness, i.e., $|\rho - \tilde{\rho}| \leq \delta$. As a result, a positive approximation robustness $\tilde{\rho}$ did not necessarily correspond to a trajectory satisfying the specification and it was required to add $\tilde{\rho} \geq \delta$ as a constraint in the optimization problem. We compare the results for maximizing approximation robustness $\tilde{\rho}$ and AGM robustness η and show the advantage of our approach, both in accuracy (removing errors due to soft minimum/maximum approximations) and satisfaction performance.

Example 2: Consider the nonlinear dynamical system:

$$\begin{aligned} x[k+1] &= x[k] + \cos\theta[k]v[k], \\ y[k+1] &= y[k] + \sin\theta[k]v[k], \\ \theta[k+1] &= \theta[k] + v[k]w[k], \end{aligned} \tag{13}$$

and the desired task "*Eventually* visit *Reg1 or Reg2* between $[1, 5]$ steps *and eventually* visit *Reg3* between $[6, 10]$ steps *and Always* avoid *Obs*", formally specified as STL formula:

$$\phi_2 = (\mathbf{F}_{[1,5]}\ (Reg1 \vee Reg2))\ \wedge (\mathbf{F}_{[6,10]}\ Reg3) \wedge (\mathbf{G}_{[0,10]}\ \neg Obs). \tag{14}$$

Fig. 4 and Fig. 5 show trajectories satisfying STL constraints in ϕ_2 obtained using gradient ascent maximizing the approximation robustness $\tilde{\rho}$ and the AGM robustness η, achieved up to the termination criteria. Although both methods generate trajectories satisfying specification ϕ_2, the trajectory with higher approximation robustness visits $Reg1$ ($t = 4$) and $Reg3$ ($t = 10$) at a single time point while using the AGM robustness, trajectory with higher robustness visits $Reg1$ ($t = 3, 4$) and $Reg3$ ($t = 8, 9, 10$) *as early as possible and for as long as possible*; forcing trajectory to move toward the center of each region while always avoiding the obstacle. Note that due to the approximation errors resulted from approximating max and min functions, traditional robustness ρ and approximation robustness $\tilde{\rho}$ have different values for the same trajectory.

C. Performance Under Disturbance

We demonstrate the advantage of maximizing the AGM robustness, rather than the traditional robustness, in a control synthesis problem under external disturbance.

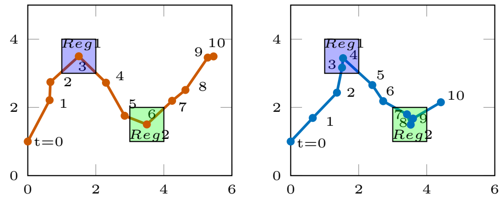

Fig. 6. Trajectories with maximum approximation robustness $\tilde{\rho} = 0.292$ (ρ=0.5) (Left) and positive AGM robustness $\eta = 0.173$ (Right).

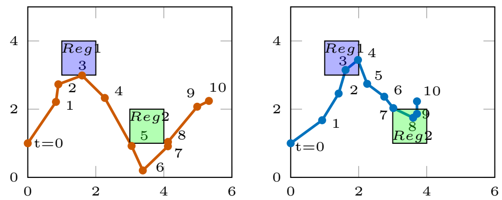

Fig. 7. A trajectory generated by disturbed $u^*_{\tilde{\rho}}$ violating ϕ_3 with negative approximation robustness $\tilde{\rho} = -0.123$ ($\rho = -0.083$) (Left) and by disturbed u^*_η satisfying ϕ_3 with positive AGM robustness $\eta = 0.166$ (Right).

Example 3: Consider a linear dynamical system:

$$\begin{aligned} x[k+1] &= x[k] + u_x[k], \\ y[k+1] &= y[k] + u_y[k], \end{aligned} \tag{15}$$

where $q = [x, y]$ is the state vector indicating robot position and $u = [u_x, u_y]$ is the input. The desired task is "*Eventually* visit *Reg1* between $[1, 5]$ steps *and eventually* visit *Reg2* between $[6, 10]$ steps", formally specified as STL formula:

$$\phi_3 = (\mathbf{F}_{[1,5]}\ Reg1) \wedge (\mathbf{F}_{[6,10]}\ Reg2). \tag{16}$$

We first find control policies u^* maximizing $\tilde{\rho}$ and η using gradient ascent. Optimal trajectories satisfying the specification ϕ_3 are shown in Fig. 6. It is clear that maximum approximation robustness $\tilde{\rho}$ is achieved when center of each region is visited at a single time ($Reg1$ at $t = 3$, $Reg2$ at $t = 6$). However, by maximizing the AGM robustness η, *not only the center of each region (maximum satisfaction) is visited at least once, but also each region is visited for more time points* ($Reg1$ at $t = 3, 4$, $Reg2$ at $t = 7, 8, 9$). Next, we perturb system by adding a gaussian noise $\mathcal{N}(0, \sigma^2)$ to the optimal control policies found:

$$\begin{aligned} u^*_{\tilde{\rho}} &\leftarrow u^*_{\tilde{\rho}} + \mathcal{N}(0, \sigma^2), \\ u^*_\eta &\leftarrow u^*_\eta + \mathcal{N}(0, \sigma^2) \end{aligned} \tag{17}$$

We apply the disturbed control policies (17) to the system (15) and find resulting trajectories for different values of σ over 100 simulations. Results show that disturbed control policy found by maximizing approximation robustness fails to satisfy the specification in 58% of the times, while by maximizing the AGM robustness, specification fails for an average of 41%. Fig. 7 (Left) illustrates a resulting trajectory by applying disturbed optimal policy $u^*_{\tilde{\rho}}$ violating ϕ_3; and (Right) a resulting trajectory by applying disturbed optimal policy u^*_η, still satisfying ϕ_3 but at different time points and with a smaller robustness η. Therefore, empirically, the control policy found by maximizing the AGM robustness performs better when disturbance is added after designing the control input.

VII. CONCLUSION AND FUTURE WORK

We presented a novel average-based robustness for STL by considering not just the critical subformula or critical time point but all subformulae at all appropriate time points. We demonstrated that AGM robustness provides a better score in control problems compared to the traditional score. We also showed that empirically, system under external disturbance has on average a better performance when maximizing AGM robustness rather than the traditional one.

REFERENCES

[1] C. Belta, B. Yordanov, and E. A. Gol, *Formal methods for discrete-time dynamical systems*. Springer, 2017, vol. 89.

[2] A. Pnueli, "The temporal logic of programs," in *18th Annual Symposium on Foundations of Computer Science*, Oct 1977, pp. 46–57.

[3] R. Koymans, "Specifying real-time properties with metric temporal logic," *Real-time systems*, vol. 2, no. 4, pp. 255–299, 1990.

[4] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. Springer, 2004, pp. 152–166.

[5] C.-I. Vasile, D. Aksaray, and C. Belta, "Time window temporal logic," *Theoretical Computer Science*, vol. 691, pp. 27–54, 2017.

[6] C. I. Vasile, V. Raman, and S. Karaman, "Sampling-based Synthesis of Maximally-Satisfying Controllers for Temporal Logic Specifications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, pp. 3840–3847.

[7] Y. Shoukry, P. Nuzzo, A. Balkan, I. Saha, A. L. Sangiovanni-Vincentelli, S. A. Seshia, G. J. Pappas, and P. Tabuada, "Linear temporal logic motion planning for teams of underactuated robots using satisfiability modulo convex programming," in *Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 1132–1137.

[8] V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia, "Reactive synthesis from signal temporal logic specifications," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*. ACM, 2015, pp. 239–248.

[9] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *International Conference on Formal Modeling and Analysis of Timed Systems*. Springer, 2010, pp. 92–106.

[10] N. Mehdipour, D. Briers, I. Haghighi, C. M. Glen, M. L. Kemp, and C. Belta, "Spatial-temporal pattern synthesis in a network of locally interacting cells," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 3516–3521.

[11] H. Abbas and G. Fainekos, "Convergence proofs for Simulated Annealing falsification of safety properties," in *Allerton Conference on Communication, Control, and Computing*, 2012, pp. 1594–1601.

[12] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[13] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conference on Decision and Control*, Dec 2014, pp. 81–87.

[14] S. Saha and A. A. Julius, "An MILP approach for real-time optimal controller synthesis with metric temporal logic specifications," in *American Control Conference (ACC)*. IEEE, 2016, pp. 1105–1110.

[15] Y. V. Pant, H. Abbas, and R. Mangharam, "Smooth operator: Control using the smooth robustness of temporal logic," in *IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 1235–1240.

[16] X. Li, Y. Ma, and C. Belta, "A policy search method for temporal logic specified reinforcement learning tasks," in *Annual American Control Conference (ACC)*. IEEE, 2018, pp. 240–245.

[17] T. Akazaki and I. Hasuo, "Time robustness in mtl and expressivity in hybrid system falsification," in *International Conference on Computer Aided Verification*. Springer, 2015, pp. 356–374.

[18] A. Rodionova, E. Bartocci, D. Nickovic, and R. Grosu, "Temporal Logic as Filtering," in *19th International Conference on Hybrid Systems: Computation and Control*. ACM, 2016, pp. 11–20.

[19] L. Lindemann and D. V. Dimarogonas, "Robust control for signal temporal logic specifications using discrete average space robustness," *Automatica*, vol. 101, pp. 377–387, 2019.

[20] N. Mehdipour, C.-I. Vasile, and C. Belta, "Arithmetic-Geometric Mean Robustness for Control from Signal Temporal Logic Specifications," *2019 American Control Conference(ACC)*, philadelphia, USA, (Accepted) available at: https://arxiv.org/abs/1903.05186.

[21] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.