2024 WODES Workshop on Discrete Event Systems Rio de Janeiro, Brazil

17th IFAC Workshop on Discrete Event Systems
April 29-May 1, 2024. Rio de Janeiro, Brazil

Robust Control of Metric Discrete Event Systems Against Bounded Disturbances ⋆

Yiding Ji [*,**] **Xiang Yin** [***]

[*] *Robotics and Autonomous Systems Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China (e-mail: jiyiding@hkust-gz.edu.cn)*
[**] *Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, China*
[***] *Department of Automation, Shanghai Jiao Tong University, Shanghai, China (e-mail: yinxiang@sjtu.edu.cn).*

Abstract: This work investigates robust supervisory control problems of discrete event systems modeled as finite state automata equipped with metric functions to measure the distance between states. The system may deviate from its nominal behaviors and fail to achieve the specification under disturbances whose effects are considered to be bounded. Accordingly, the supervisor should be designed to ensure that the controlled system degrades gracefully against such adversary. We formally formulate two problems: robustness bound verification and optimal robust supervisor synthesis. For the special case of verification under constant disturbances, a control Lyapunov function approach is introduced. Then we develop a two player game framework for the general verification and synthesis problems. Specifically, a dynamic programming method is proposed on the game structure, which uniformly tackles both problems.

Keywords: Metric discrete event systems, supervisory control, robustness, two player games

1. INTRODUCTION

Ever since its initiation in the 1980s, supervisory control has been a core topic in discrete event systems (DES), see., e.g., Cassandras and Lafortune [2021]. The plant under control is usually modeled as a finite discrete structure with a specification to represent its admissible behaviors. The underlying control paradigm is to properly enable and disable certain events so that the behaviors of the controlled system are constrained within the specification.

Supervisory control has been intensively investigated in various settings of DES, including but not limited to networked control Lin et al. [2022], timed DES control Basile et al. [2021], quantitative control Ji et al. [2022], compositional control Malik et al. [2023], to name a few.

In many applications, the system usually operates in an uncertain environment and is subject to disturbances or cyber attacks. DES models are appropriate for analyzing and controlling systems against those adversaries. Specifically, the goal of robust supervisory control is to design resilient supervisors to tolerate the disturbances or errors so that the specification is still (partially) achievable, see, e.g., Wang et al. [2020], Alves et al. [2021], Meira-Góes et al. [2023] for some results. Recently, supervisory control against cyber attacks has also been extensively studied from perspectives of both the attacker and the defender, such as Ma and Cai [2021], Zheng et al. [2023], You et al. [2021], Fritz and Zhang [2023], Tai et al. [2022]. However, those works focus on logical DES models and only give a binary answer about whether system security is protected.

Given the above issues and motivated by robust control for continuous systems, we develop a framework for robust supervisory control for DES modeled by finite metric automata. First we introduce a metric function to quantify the distance between states of the system and consider a generic model for bounded disturbances. Then we define robustness of supervisors as a *topological* concept in terms of the distance between states. The control paradigm is to ensure that the degree of behavior degradation of the controlled system is *propositional* to the power of disturbances, thus no catastrophic failures are incurred. Based on those concepts, we formulate the robust supervisor verification and optimal robust supervisor synthesis problems.

First, for the special verification case where disturbances have constant bounds, we introduce *control Lyapunov functions (CLF)* on metric DES. Analysis shows that CLF induce supervisors that nominally achieve the specification under no disturbances and further provide robustness bounds for supervisors under disturbances. Second, for general verification and synthesis problems, a two-player game framework is developed under which we propose a dynamic programming approach to solve both problems.

We leverage some results of robust design in cyber physical systems, see, e.g., Majumdar et al. [2013], Girard and Eqtami [2021]. Different from symbolic synthesis methods, we consider supervisory control problems where uncontrollable events are involved and multiple events are enabled simultaneously. Some works in DES also apply ranking

⋆ The first author is supported by National Natural Science Foundation of China (NSFC) under grants 62303389 and 62373289, Guangdong Basic and Applied Basic Research Funding under grants 2024A1515012586 and 2022A151511076, Guangzhou Basic and Applied Basic Research Scheme under grant 2023A04J1067, Guangzhou Municipality-University Joint Funding under grant 2023A03J0678; and the second author is supported by NSFC under grant 62173226.

functions in supervisory control, see., e.g., Sakakibara and Ushi [2020], Sakakibara et al. [2021], whose problem settings and solution methods are incomparable with ours.

The rest of the work is organized as follows. Section 2 introduces the system model. Section 3 formulates two key problems of this work. Section 4 presents systematic solutions to both problems. Finally, Section 5 concludes the paper and lists potential future extension directions.

2. SYSTEM MODEL

The discrete event system (DES) in this work is modeled as a finite-state automaton: $G = (X, E, f, X_m, x_0)$ where X is the finite state space; E is the finite set of events; $f : X \times E \to X$ is the partial transition function whose domain may be extended to $X \times E^*$ in an recursive manner as $f(x, se) = f(f(x, s), e)$ for $x \in X$, $s \in E^*$ and $e \in E$, $X_m \subseteq X$ is the set of marked states and $x_0 \in X$ is the initial state. When an event $e \in E$ is defined at a state $x \in X$, it is called *active* or *feasible* at x. *Strings* are generated when evens occur sequentially, and a set of strings constitutes a *language* which models the behaviors of the system. Specifically, $\mathcal{L}(G) = \{s \in E^* : f(x_0, s) \text{ is defined}\}$ and $\mathcal{L}_m(G) = \{s \in \mathcal{L}(G) : f(x_0, s) \in X_m\}$ stand for the language generated and marked by G, respectively. Without loss of generality, we assume that every state in G is both *accessible* and *co-accessible*.

Given G, a *metric* function is introduced to quantitatively measure the *distance* between states and it is formally defined as $d : X \times X \to \mathbb{R}_0^+$ where $\forall x, y, z \in X$, the following conditions hold: (i) $d(x, y) = 0 \Leftrightarrow x = y$ (identity); (ii) $d(x, y) = d(y, x)$ (symmetry); (iii) $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality). The distance between a state to a set of states is defined as $\forall x \in X$ and $Q \subseteq X$: $d(x, Q) = \min_{x' \in Q} d(x, x')$, i.e., the shortest distance from x to some state in Q. If G is equipped with d, it is called a metric automaton (system) and denoted by (G, d).

Example 1. Consider the metric automaton (G, d) in Figure 1. The event set is $E = \{a, b, c, d, u\}$ and the state space is $\{x_0, \cdots, x_7\}$ with a marked state x_7. The values of metric function d for every two states is summarized in the following table and the above three conditions hold.

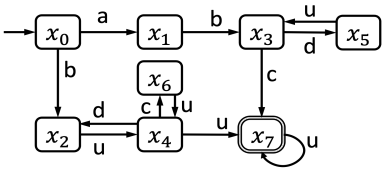

Fig. 1. The metric system (G, d)

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_0	0	4	2	5	5	5	6	6
x_1		0	3	3	3	3	3	4
x_2			0	4	4	5	5	5
x_3				0	2	3	3	1
x_4					0	2	2	2
x_5						0	2	3
x_6							0	3
x_7								0

We introduce the following notions. First, we write $f(x, e)!$ if $e \in E$ is active at $x \in X$, where ! stands for being defined. Next, for $x_1, x_2 \in X$ and $e \in E$, we write $x_1 \xrightarrow{e} x_2$ if $f(x_1, e) = x_2$. Then we define $Suc(x) = \{x' \in X : (\exists e \in E)[x \xrightarrow{e} x']\}$ as the set of *direct successor* states of x.

Then we briefly review the basic mechanism of supervisory control in DES. Given system G, a supervisor is a function $S : \mathcal{L}(G) \to \Gamma$ where $\Gamma \subseteq 2^E$ is the set of control decisions. A supervisor dynamically enables and disables events, where a control decision $\gamma \in \Gamma$ is the set of enabled events. We let $\mathbb{S}$ denote the set of supervisors. The event set E is also partitioned as $E = E_c \cup E_{uc}$, where E_c and E_{uc} represent the sets of controllable and uncontrollable events, respectively. A control decision is *admissible* if $E_{uc} \subseteq \gamma$, i.e., no uncontrollable event is disabled. By convention, we only consider admissible control decisions in this work. Additionally, S/G denotes the controlled system under S, $\mathcal{L}(S/G)$ denotes the language generated in S/G and the marked language of S/G is $\mathcal{L}_m(S/G) = \mathcal{L}(S/G) \cap \mathcal{L}_m(G)$, which is solely determined by the marking of G.

When the system does not satisfy given properties, supervisors, supervisors are designed to enforce them. In this work, the *specification* is given with respect to the *marked language* of G and is restricted to *reachability manner properties*, i.e., marked states are reached *eventually*. We also consider (weak) *liveness*, where $\forall s \in \mathcal{L}(G)$, $\exists u \in E$, such that $su \in \mathcal{L}(G)$, i.e., every state has active events defined. Given the above requirements, we let a prefix-closed language $K \subseteq \mathcal{L}_m(G)$ represent the desired behavior of G.

3. PROBLEM FORMULATION

The DES model in Section 2 describes the *nominal* dynamics of the system when no environmental *disturbances* exist. However, the given specification may no longer be achieved subject to perturbations. It is essential for the supervisors to work in a *robust* manner to *tolerate* disturbances and circumvent catastrophic failures. For that reason, we define robustness for supervisors, and formulate the two major problems to be investigated in this work.

Here we propose a simple yet general model of disturbances. The detailed mechanism such as the physical nature and origins of disturbances is beyond this work's scope. The set of disturbance is denoted by Δ and $\delta = \varepsilon$ indicates no disturbance. Given a metric automaton (G, d), function $\upsilon : X \to \mathbb{R}_0^+$ quantifies the effects of disturbances at each state. We also let $\bar{\upsilon} = \max_{x \in X_m} \upsilon(x)$ be the upper bound of disturbances at X_m. Since the specification is to reach marked states X_m and the disturbances prevent the system from achieving the specification, we also call $\bar{\upsilon}$ the upper bound of Δ when there is no confusion. Specially, the disturbance is termed *constant* if $\upsilon(x) = \upsilon(x')$ for all $x, x' \in X$, where $\upsilon(x) = \bar{\upsilon}$ naturally holds for all $x \in X$.

Then we illustrate how the disturbances intervene with the system's dynamics. Given an active event e in state x with $x \xrightarrow{e} y$, if $\upsilon(y) < d(x, y)$, i.e., the disturbance effect is less than the distance between x and y, then the nominal transition function $f(x, e)$ is not perturbed, thus y is reached. Otherwise, when $\upsilon(y) \geq d(x, y)$, we define

$$DR(x, e) = \{x' \in X : x' \in Suc(x),\ 0 < d(x', y) \leq \upsilon(y)\}$$

as the set of disturbed reachable stable states from x. Here x' is a direct successor state of x, and the distance

between x' and y is greater than 0 but no greater than the disturbance effect $\upsilon(y)$. A special remark is that the original target state y is excluded from $DR(x,e)$ to reflect that the dynamics has been perturbed. Also $DR(x,e)$ may not be a singleton since *nondeterminism* is triggered by disturbances. We compare the disturbance effect at the target state y with the distance $d(x,y)$. This is simply our matter of choice, as it is straightforward to reformulate the comparison w.r.t. disturbance effect $\upsilon(x)$ and $d(x,y)$.

To incorporate the disturbance effects, the *disturbed* transition function is defined as $f_d : X \times E \to 2^X$ where for all $x, y \in X$ and $e \in E$ such that $x \xrightarrow{e} y$, we have that

$$f_d(x,e) = \begin{cases} y & \text{if } \upsilon(y) < d(x,y) \\ DR(x,e) & \text{if } \upsilon(y) \geq d(x,y) \end{cases}$$

Then a metric automaton (G,d) under disturbance Δ is to replace the original transition function f by f_d.

Two key problems pertain to supervisory control subject to disturbances. First, if a supervisor nominally achieves the specification but fails to do so under disturbances, then how do we measure its robustness? Second, can we design a supervisor with the optimal robust performance? Before solving them, we first define the robustness of supervisors.

Definition 1 (σ-robust supervisor). Given a metric system (G,d), disturbance Δ with its effect function υ, and specification $K \subseteq L_m(G)$, a supervisor S is called σ-robust with respect to a positive constant σ if for $X_m^\sigma = \{x \in X : d(x, X_m) \leq \sigma \cdot \bar{\upsilon}\}$, we have that (i) $\mathcal{L}(S/G) \subseteq K$ and (ii) $\forall s \in \mathcal{L}(S/G)$, $\exists t \in E^*$ such that $f_d(x_0, st) \in X_m^\sigma$.

Definition 1 is inspired by robust control for continuous systems and the fundamental idea is that bounded disturbances only cause modest deviation from the desired behaviors. X_m^σ is an inflated set of the marked states, which includes all states within distance $\sigma \cdot \bar{\upsilon}$ from X_m. Notably, the original specification may not be achievable subject to disturbances, where a smaller σ implies that the behaviors of the controlled system deviate less substantially from the original specification. In that sense, a supervisor is deemed more robust if it is capable to drive all strings closer to X_m in the controlled system. Therefore, σ measures the robustness of supervisors. When there is no confusion, a supervisor is simply called robust if it is σ-robust.

Problem 1 (Robustness bound verification). Given metric system (G,d) and specification $K \subseteq \mathcal{L}_m(G)$, suppose that supervisor S nominally satisfies the specification, then if disturbance Δ is present and has its effect function υ, determine the smallest σ such that S is σ-robust.

Remark 1. In the settings of Problem 1, the supervisor nominally achieves the specification. The smallest bound σ implies that for any $\sigma' < \sigma$, S is not σ'-robust.

Problem 2 (Optimal robust supervisor synthesis). Given metric system (G,d) with specification $K \subseteq L_m(G)$, and disturbance Δ with its effect function υ, synthesize an optimal robust supervisor S_{opt} such that S_{opt} is σ_{min}-robust where $\sigma_{min} = \min_{S \in \mathbb{S}}\{\sigma \in \mathbb{R}^+ : S \text{ is } \sigma\text{-robust}\}$.

Remark 2. It is likely that different supervisors share the same σ_{min} as their control profiles lead the system to the same inflated set of marked states, so S_{opt} is not unique.

Example 2. Continue to consider the system in Example 1 and let $E_{uc} = \{u\}$. The specification is "reaching x_7 and staying weakly live". The disturbance effect at each state is: $\upsilon(v_0) = \upsilon(v_1) = \upsilon(v_2) = \upsilon(v_3) = \upsilon(v_4) = \upsilon(v_6) = 1$, $\upsilon(v_5) = 2$ and $\upsilon(v_7) = 4$. Consider two supervisors S_1 and S_2, where S_1 enables a at x_0; b at x_1; c at x_3; u at x_7, while S_2 enables b at x_0; u at x_2, x_6 and x_7; c, d and u at x_4. Obviously, if there are no disturbances, both supervisors achieve the specification. However, if disturbances abound, transition $x_3 \xrightarrow{c} x_7$ is perturbed since $\upsilon(x_7) > d(x_3, x_7)$, then x_7 is not reachable by S_1. Also transition $x_4 \xrightarrow{u} x_7$ is disturbed since $\upsilon(x_7) > d(x_4, x_7)$. We are going to verify the robustness for S_1 and S_2 and design an optimal robust supervisor in the next section.

4. VERIFICATION AND SYNTHESIS METHODS

In this section, we sequentially solve Problem 1 and Problem 2. For robustness verification subject to disturbances with a constant bound, we first introduce *ranking functions* and extend the concept to *control Lyapunov functions*, based on which we determine the supervisor's robustness measure and partially solve Problem 1. Then we propose a game theoretic framework to tackle robustness verification under state-dependent disturbances and supervisor synthesis issues . Specifically, a bipartite transition structure is defined and both problems are reformulated as a *two-player game* between the protagonist supervisor and the antagonist environment. A dynamic programming algorithm is proposed to completely solve two problems.

4.1 Verification under disturbances with constant bounds

Given a metric system (G,d), a function $g : X \to \mathbb{R}$ is called *Lipschitz continuous* if there exists a constant $K > 0$ such that $\forall x, y \in X$, $|g(x) - g(y)| \leq M \cdot d(x,y)$, where M is the *Lipschitz constant* of g and d is the distance metric. Since the domain of g is finite, any real valued function g is Lipschitz continuous. Next we define ranking functions.

Definition 2 (Ranking functions). Given a metric system (G,d), $\rho : X \to \mathbb{R}_0^+$ is a ranking function if (i) $\rho(x) = 0 \Leftrightarrow x \in X_m$; (ii) there exists a monotonically increasing function $\alpha : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ such that $\alpha(0) = 0$ and $\alpha(d(x, X_m)) \leq \rho(x)$ for all $x \in X$.

A ranking function is defined on the finite state space of G and is naturally Lipschitz continuous. By definition, a ranking function has larger values at states further from marked states of G. Then we impose some extra conditions on ranking functions to define their "control" versions.

Definition 3 (Control Lyapunov functions (CLF)). Given metric system (G,d), a ranking function ρ is called a CLF if there exists a monotonically increasing function $h : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ where $h(0) = 0$ and $\forall x \in X$, $\forall e \in \Gamma(x)$:

$[e \in E_{uc} \Rightarrow \rho(f(x,e)) - \rho(x) \leq -h(d(x, X_m))] \wedge$

$[e \in E_c \Rightarrow \exists e' \in \Gamma(x) : \ \rho(f(x,e')) - \rho(x) \leq -h(d(x, X_m))]$

The implications of Definition 3 are two-fold. First, all uncontrollable active events lead the CLF to decrease or retain its values. Second, if an active event is controllable, then there exists at least one feasible event (either controllable or uncontrollable) leading the CLF to decrease or remain unchanged. The occurrence of new events can be viewed as control inputs as they are enabled by the

supervisor. In that sense, Definition 3 is analogous with its counterpart in nonlinear control, see, e.g., Sontag [1998].

Definition 4 (CLF Induced Supervisors). Given a metric system (G, d) and CLF ρ, a supervisor S is induced by ρ if $\forall s \in \mathcal{L}(S/G)$ such that $f(x_0, s) = x$, then $S(s) = \{e \in E_c : \rho(f(x, e)) - \rho(x) \leq -h(d(x, X_m))\} \cup E_{uc}$

If a supervisor S is induced by a CLF r, then S enables all controllable events satisfying the inequality of Definition 3 and all feasible uncontrollable events. That is, the supervisor always issues commands following the direction of not increasing the values of r. The next theorem offers a necessary and sufficient condition for supervisors to nominally satisfy the specification under no disturbances, whose proof is omitted here due to space limitation.

Theorem 1. Given metric system (G, d) with specification $K \subseteq \mathcal{L}_m(G)$, supervisor S achieves K if and only if there exists a control Lyapunov function ρ that induces S.

Example 3. We continue to discuss the system in Example 2 and temporarily ignore the disturbances. We introduce a rank function ρ whose values at each state $x_i \in X$ is listed in the following table, together with the distance from x_i to the marked state x_7. It can be verified that ρ is a control Lyapunov function and induces supervisor S_1 in Example 2. For a monotonically increasing function $h(x) = 2x$, inequalities $\rho(x_1) - \rho(x_0) \leq -h(d(x_0, x_7))$, $\rho(x_3) - \rho(x_1) \leq -h(d(x_1, x_7))$ and $\rho(x_7) - \rho(x_3) \leq -h(d(x_3, x_7))$ hold, which implies that the control decisions of S_1 (enabling a at x_0; b at x_1; c at x_3; u at x_7) follow the direction of value decreasing of ρ at a rate controlled by h.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$d(x_i, x_7)$	6	4	5	1	2	3	3	0
$\rho(x_i)$	24	11	16	3	4	6	6	0

Then Theorem 2 (proof omitted) describes the degradation of the controlled system's behaviors under disturbances with a constant bound. The deviation from nominal outcomes is propositional to the power of disturbances.

Theorem 2. Given metric system (G, d) with specification $K \subseteq \mathcal{L}_m(G)$, and disturbance Δ with constant bound $\overline{\upsilon}$, if supervisor S is induced by control Lyapunov function r with Lipschitz constant M, then S/G is ensured to reach $\{x \in X : h(d(x, X_m)) \leq M\overline{\upsilon}\}$ and S is $h^{-1}(M\overline{\upsilon})/\overline{\upsilon}$-robust.

4.2 Game-theoretic supervisor verification and synthesis

For robustness bound verification and optimal robust supervisor synthesis under state dependent disturbances, we formulate a two-player game between the supervisor and the environment (disturbances), where winning conditions are properly defined. Then we analyze the game to propose a dynamic programming approach to search for supervisor's optimal winning strategies and robustness bound.

We first define *metric state estimate* to integrate logical state information and distance information. A metric state estimate contains two components: one tracks the system's current state estimate under control and disturbances; the other tracks the distance from the state estimate to X_m.

Definition 5 (Metric State Estimate). Given a metric automaton (G, d), a metric state estimate is a tuple

$$q = ((x_1, \cdots, x_n), \max_{i \leq n} d(x_i, X_m)) \in 2^X \times \mathbb{R}_0^+$$

where $\mathcal{E}(q)$ and $\mathcal{D}(q)$ denote the state estimate and distance value components of q, respectively.

Given a metric state estimate q, the distance value $\mathcal{D}(q)$ is set to be the maximum distance between some state in $\mathcal{E}(q)$ to X_m. This reflects the worst possible deviation from marked states caused by disturbances. Based on Definition 5, we introduce *metric bipartite transition systems.*

Definition 6 (Metric Bipartite Transition System). A metric bipartite transition system w.r.t. (G, d) and disturbance Δ is $T = (Q_Y, Q_Z, f_{yz}, f_{zy}, E, \Gamma, d, q_0^y)$ where

- $Q_Y \subseteq 2^X \times \mathbb{R}_0^+$ is the set of metric state estimates;
- $Q_Z \subseteq 2^X \times \mathbb{R}_0^+ \times \Gamma \times \{0, 1\}$ is the set of *augmented* metric state estimates and for $q^z \in Q_Z$, $\mathcal{M}(q^z)$, $\Gamma(q^z)$ and $\mathcal{B}(q^z)$ denote the metric state estimate, control decision and disturbance indicator of q^z, respectively;
- $f_{yz} : Q_Y \times \Gamma \to Q_Z$ is the transition from Q_Y to Q_Z states, which satisfies $\forall q^y \in Q_Y$, $\forall \gamma \in \Gamma$ and $\forall q^z \in Q_Z$:

  $f_{yz}(q^y, \gamma) = q^z \Rightarrow [\mathcal{M}(q^z) = q^y] \wedge [\Gamma(q^z) = \gamma] \wedge$
  $[\mathcal{B}(q^z) = \mathbb{I}((\exists e \in \gamma \wedge \exists x \in \mathcal{E}(q^y) \wedge \exists x' \in X) \Rightarrow (x \xrightarrow{e} x'$
  $\wedge \upsilon(x') \geq d(x, x'))]$ where $\mathbb{I}$ is the indicator function;

- $f_{zy} : Q_Z \times E \to Q_Y$ is the transition from Q_Z to Q_Y states, which satisfies $\forall q^z \in Q_Z$, $\forall e \in E$ and $\forall q^y \in Q_Y$:

  $f_{zy}(q^z, e) = q^y \Rightarrow [e \in \Gamma(q^z)] \wedge [\mathcal{E}(q^y) = \{x' \in X : (\exists x \in$
  $\mathcal{M}(\mathcal{E}(q^z)))[x' \in f_d(x, e)]\}] \wedge [\mathcal{D}(q^y) = \max_{x' \in \mathcal{E}(q^y)} d(x', X_m))]$

- E is the set of events of G;
- Γ is the set of admissible control decisions;
- d is the distance metric of G;
- $q_0^y = \{x_0, d(x_0, X_m)\} \in Q_Y$ is the initial state.

In general, a metric bipartite transition system (MBTS) is a two player game arena where the supervisor plays against the environment (disturbances). A Q_Y-state, aka Y-state, is a metric state estimate where the supervisor issues control decisions. A Q_Z-state, aka Z-state, is a metric state estimate augmented with an admissible control decision and a disturbance indicator. A transition from a Y-state q^y to a Z-state q^z "remembers" the last issued control command and indicates whether the transition will be disturbed. At this stage, the metric state estimate has not been updated yet, thus $\mathcal{M}(q^z) = q^y$. Additionally, $\Gamma(q^z) = \gamma$ is the control decision made at the preceding Y-state and the indicator $\mathcal{B}(q^z) = 1$ if the transition $x \xrightarrow{e} x'$ is disturbed since the disturbance effect $\upsilon(x')$ is no less than the distance $d(x, x')$, otherwise $\mathcal{B}(q^z) = 0$, as discussed in Section 3. A transition from a Z-state q^z to a Y-state q^y represents the update of metric state estimates under control decisions and disturbances. To be more specific, the state estimate of q^y is the set of states reachable via enabled event e from some state in the state estimate of q^z, following the disturbed transition function f_d. The distance value $\mathcal{D}(q^y)$ is also calculated by Definition 5.

Given an MBTS T, we let $Post(q^y) = \{q^z \in Q^Z : (\exists \gamma \in \Gamma)[f_{yz}(q^y, \gamma) = q^z]\}$ and $Post(q^z) = \{q^y \in Q^Y : (\exists e \in E)[f_{zy}(q^z, e) = q^y]\}$ be the set of direct successors states of a Y-state q^y and a Z-state q^z, respectively. The set of runs in T is denoted by $Run(G)$ and a run is of the form: $r = q_1^y \xrightarrow{\gamma_1} q_1^z \xrightarrow{e_1} q_2^y \cdots \xrightarrow{\gamma_n} q_n^z \xrightarrow{e_n} q_{n+1}^y$. And we

also denote by $Run_y(T)$ (respectively $Run_z(T)$) the set of runs ending with a Y-state (respectively Z-state).

We also require that a MBTS be *complete*, which essentially means that at least one control decision is defined at every Y-state and all enabled events are allowed to occur from any given Z-state. In addition, a Z-state q^z is called *deadlock-free* if $(\forall x \in \mathcal{E}(\mathcal{M}(q^z)))(\exists e \in \Gamma(q^z))[f_d(x,e)!]$, otherwise, q^z is a *deadlock* state. This condition guarantees that some enabled event in $\Gamma(q^z)$ is always defined at any state x of the state estimate $\mathcal{E}(\mathcal{M}(q^z))$. Following a similar proof as Lemma V.1 in Yin and Lafortune [2016], we can show that a supervisor induced by an MBTS T is (weakly) live if and only if all Z-states of T are deadlock-free.

Both players on a MBTS possess *strategies* to selects its next step's transition at its positions. A control strategy works in the same way as a standard supervisor, thus we will use terms "supervisor" and "supervisor's strategy (control strategy)" interchangeably. Moreover, we call a strategy *positional* if its decisions only depend on the current (augmented) metric state estimate. Existing results in Apt and Grädel [2011] show that positional strategies suffice for a player to win reachability games, so we will restrict our attention to positional strategies aftermath.

Next, Algorithm 1 constructs the largest complete MBTS, where being the largest is in a graph merging sense. The structure is a tuple $T_m = (Q_Y^m, Q_Z^m, E, \Gamma, f_{yz}^m, f_{zy}^m, q_0^y)$, where for all complete MBTSs T, $T \sqsubseteq T_m$ holds.

Algorithm 1 Build T_m

Input: G, d, Δ
Output: $T_m = (Q_Y^m, Q_Z^m, E_o, \Gamma, f_{yz}^m, f_{zy}^m, q_0^y)$

1: $Q_Y^m = \{q_0^y\} = \{x_0, d(x_0, X_m)\}$, $Q_Z^m = \emptyset$;
2: $T_m^{pre} = DoDFS(q_0^y, G, d, \Delta)$;
3: **while** there exist Y-states without successors **do**
4: remove such states and their predecessor Z-states;
5: **return** T_m;
6: **procedure** $DoDFS(q^y, G, d, \Delta)$
7: **for** $\gamma \in \Gamma$ **do**
8: $q^z = f_{yz}(q^y, \gamma)$ by Definition 6;
9: **if** q^z is deadlock-free **then**
10: add transition $q^y \xrightarrow{\gamma} q^z$ to f_{yz}^m;
11: **if** $q^z \notin Q_Z^m$ **then**
12: $Q_Z^m = Q_Z^m \cup \{q^z\}$;
13: **for** $e \in \gamma$ **do**
14: $\tilde{q}^y = f_{zy}(q^z, e)$ by Definition 6;
15: add transition $q^z \xrightarrow{e} \tilde{q}^y$ to f_{zy}^m;
16: **if** $\tilde{q}^y \notin Q_Y^m$ **then**
17: $Q_Y^m = Q_Y^m \cup \{\tilde{q}^y\}$;
18: $DoDFS(\tilde{q}^y, G, d, \Delta)$;

Algorithm 1 consists of two major steps. First, a depth-first search of procedure $DoDFS$ is initiated from the initial state q_0^y to add new states and transitions to the structure. Within this step, Line 9 checks whether a newly added Z-state is deadlock-free. If it is the case, we proceed to add all successor Y-states to the structure. $DoDFS$ is called recursively until no new states are added, which eventually results in an MBTS T_m^{pre}. Since the above process may result in Y-states without successors, the next step from Line 3 is to remove such states and their predecessor Z-states since enabled observable events should not be blocked from occurring. The algorithm will converge after a finite number of steps since the state space of T_m is finite due to the finite number of Y-states and Z-states.

Example 4. Reconsider the system in Example 2. We follow Algorithm 1 to construct T_m in Figure 2, where square states and oval states represent Y-states and Z-states, respectively. The algorithm starts from q_0^y where four control decisions ranging from γ_1 to γ_4 are available. If the supervisor opts for γ_1, then a Z-state is reached and the disturbance takes turns to play. The remaining structure is interpreted in a similar manner. Note that the three red Z-states are deadlock, thus not included in T_m.

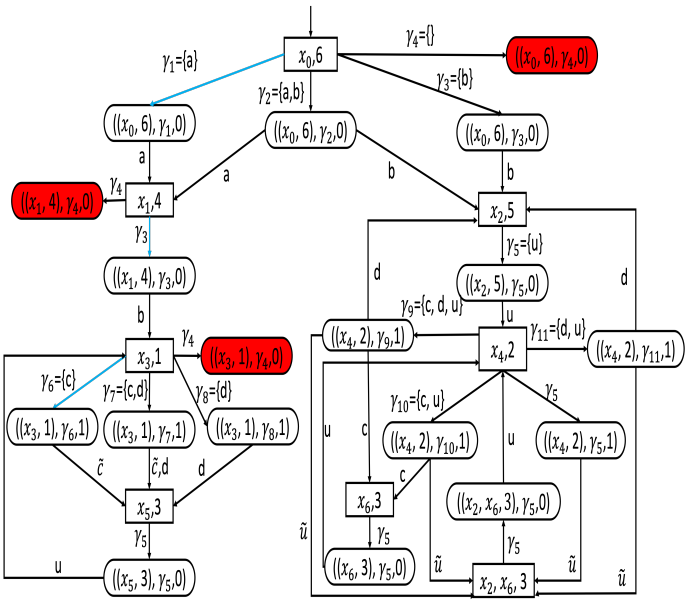

Fig. 2. MBTS T_m (without the three red deadlock states)

Then a dynamic programming approach is proposed to determine the optimal robustness bound for each state in T_m. We introduce a sequence of monotonic operators V_i for $i \geq 0$, which are defined recursively as: for all $q^y \in Q_Y^m$,

$$V_{i+1}(q^y) = \min\{V_i(q^y), \min_{q^z \in Post(q^y)} \max_{\bar{q}^y \in Post(q^z)} V_i(\bar{q}^y)\}$$

and $V_0(q^y) = \mathcal{D}(q^y)$ for all $q^y \in Q_Y^m$. Also the above equation immediately implies that for all $q^y \in Q_Y^m$,

$$V_1(q^y) = \min\{\mathcal{D}(q^y), \min_{q^z \in Post(q^y)} \max_{\bar{q}^y \in Post(q^z)} \mathcal{D}(\bar{q}^y)\}$$

Therefore $V_1(q^y)$ encodes the shortest distance to reach the marked states via at most one step of action, i.e., staying at q^y or going to its successor states. The supervisor aims to minimize the distance to marked states, while disturbance always takes actions to force the controlled system to move as far away as possible from marked states. In other words, the supervisor and the environment are playing a *min-max game* with respect to the distance to marked states. The supervisor has to consider the "*worst case*" scenario caused by disturbances, which is reflected by *max* operator in the above equations. By iterating the sequence of operators, we are expected to reach a *fixed point* defined as:

$$V^*(q^y) = \min\{\mathcal{D}(q^y), \min_{r \in Run_z(q^y)} \max_{\bar{q}^y \in Post(Last_Z(r))} \mathcal{D}(\bar{q}^y)\}$$

The fixed-point of the game is also the optimal achievable robustness bound of supervisors (control strategies)

induced by T_m. Since the state space of T_m is always finite, the fixed point is guaranteed to be calculated after a finite number of iterations. Therefore, the complexity for calculating the fixed point is linear in the size of T_m.

Now we ultimately solve Problem 1 as follows. For a supervisor S, we first retrieve all the control decisions of S and locate a control strategy in T_m that works in the same manner with S, i.e., S is induced by T_m (under disturbances). Aftermath we isolate a deterministic MBTS T_S from T_m by specifying a unique control decision at each Y-state and guarantees that T_S only induces S. Then we analogously define the sequence of operators V_i for $i \geq 0$ on T_S, and calculate its fixed point $V^*(q_0^y)$ for the initial state q_0^y of T_S, which returns that S is $V^*(q_0^y)/\bar{v}$-robust.

The fixed point of the min-max game in T_m indicates that the optimal robustness bound for supervisors induced by T_m is $\sigma_{min} = V^*(q_0^y)/\bar{v}$. Then the optimal winning control strategy is determined by the "optimal path" leading to $V^*(q_0^y)$. Finally we realize the optimal supervisor in its automaton form and completely solve Problem 2.

Example 5. We continue Examples 3 and 4. By iterating the operators V_i for $i \geq 0$ on T_m, we have that $V^*(q^y) = 3$ for all Y-state q^y and the optimal control strategy is: choosing γ_1, γ_3, then any of $\{\gamma_6, \gamma_7, \gamma_8\}$ and repeat the above choices. The detailed calculation process is omitted here. We highlight an optimal control strategy by blue lines in Figure 2. The induced optimal robust supervisor is exactly S_1 in Example 3 and S_1 is shown in Figure 3. Due to the disturbances, x_5 instead of x_7 is reached after c is enabled at x_3. Since $d(x_3, x_7) = 1$ and $d(x_5, x_7) = 3$, S_1 guarantees to reach states within distance 3 from X_m. Since $d(x_3, x_7) = 1$ and $d(x_5, x_7) = 3$, supervisor S_1 guarantees to reach states within distance 3 from x_7. Further analysis shows that supervisor S_2 in Example 3 only guarantees to reach states within distance 5 from x_7.

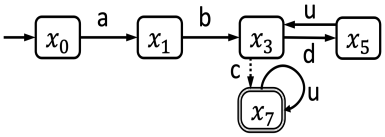

Fig. 3. An optimal robust supervisor with $\sigma_{min} = 3$

5. CONCLUSION

This work considered robust supervisory control on metric automaton model for the first time in DES. After introducing the system and disturbance models, we formulated robustness verification and optimal robust supervisor synthesis problems. A special case of verification was investigated first. Then we developed a game theoretic framework and proposed a dynamic programming method to search for fixed points which turn out to provably tackle both problems. For future extensions, we will consider robust supervisory control with more complex specifications than reachability, e.g., those expressed as linear temporal logics.

REFERENCES

Alves, M.V., da Cunha, A.E., Carvalho, L.K., Moreira, M.V., and Basilio, J.C. (2021). Robust supervisory control of discrete event systems against intermittent loss of observations. *Intl. J. of Con.*, 94(7), 2008–2020.

Apt, K.R. and Grädel, E.E. (2011). *Lectures in game theory for computer scientists*. Cambridge Uni. Press.

Basile, F., Cordone, R., and Piroddi, L. (2021). Supervisory control of timed discrete-event systems with logical and temporal specifications. *IEEE Transactions on Automatic Control*, 67(6), 2800–2815.

Cassandras, C.G. and Lafortune, S. (2021). *Introduction to discrete event systems – 3rd Edition*. Springer.

Fritz, R. and Zhang, P. (2023). Detection and localization of stealthy cyber attacks in cyber-physical discrete event systems. *IEEE Transactions on Automatic Control*.

Girard, A. and Eqtami, A. (2021). Least-violating symbolic controller synthesis for safety, reachability and attractivity specifications. *Automatica*, 127, 109543.

Ji, Y., Yin, X., and Lafortune, S. (2022). Local mean payoff supervisory control for discrete event systems. *IEEE Transactions on Automatic Control*, 67(5), 2282–2297.

Lin, L., Zhu, Y., Tai, R., Ware, S., and Su, R. (2022). Networked supervisor synthesis against lossy channels with bounded network delays as non-networked synthesis. *Automatica*, 142, 110279.

Ma, Z. and Cai, K. (2021). Optimal secret protections in discrete-event systems. *IEEE Transactions on Automatic Control*, 67(6), 2816–2828.

Majumdar, R., Render, E., and Tabuada, P. (2013). A theory of robust omega-regular software synthesis. *ACM Trans. on Embedded Computing Systems*, 13(3), 1–27.

Malik, R., Mohajerani, S., and Fabian, M. (2023). A survey on compositional algorithms for verification and synthesis in supervisory control. *Discrete Event Dynamic Systems: Theory and Applications*, 33(3), 279–340.

Meira-Góes, R., Kang, E., Lafortune, S., and Tripakis, S. (2023). On tolerance of discrete systems with respect to transition perturbations. *Discrete Event Dynamic Systems: Theory and Applications*, 33, 395–424.

Sakakibara, A., Urabe, N., and Ushio, T. (2021). Finite-memory supervisory control of discrete event systems for LTL[$\mathcal{F}$] specifications. *IEEE Transactions on Automatic Control*, 67(12), 6896–6903.

Sakakibara, A. and Ushi, T. (2020). On-line permissive supervisory control of discrete event systems for scLTL specifications. *IEEE Control Sys. Letters*, 4(3), 530–535.

Sontag, E.D. (1998). *Mathematical control theory: deterministic finite dimensional systems*. Springer.

Tai, R., Lin, L., Zhu, Y., and Su, R. (2022). Synthesis of the supremal covert attacker against unknown supervisors by using observations. *IEEE Transactions on Automatic Control*, 68(6), 3453 – 3468.

Wang, X., Hu, H., and Zhou, M. (2020). Discrete event approach to robust control in automated manufacturing systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(1), 123–135.

Yin, X. and Lafortune, S. (2016). A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Transactions on Automatic Control*, 61(8), 2140–2154.

You, D., Wang, S., Zhou, M., and Seatzu, C. (2021). Supervisory control of petri nets in the presence of replacement attacks. *IEEE Transactions on Automatic Control*, 67(3), 1466–1473.

Zheng, S., Shu, S., and Lin, F. (2023). Modeling and control of discrete event systems under joint sensor-actuator cyber attacks. *IEEE Trans. on Con. of Network Sys.*