

Fair Control of Uncertain Dynamical Systems under LTL Specifications

Can Zhou, Pian Yu, Thomas Parisini, Alessandro Abate, and Yulong Gao

Abstract—Uncertainties are inevitable for control of dynamical systems of practical relevance. The conventional robust control methods assume persistent worst-case realisations of uncertainties, which leads to conservative solutions. To relax this assumption, we exploit the fairness notion in formal verification to qualitatively capture realistic epistemic disturbance behaviours. We study the fair control problem, that is, to maximise the probability of satisfying an Linear Temporal Logic subject to an action-based fairness constraint for uncertain dynamical systems. The action-based fairness of epistemic disturbances can be defined flexibly: by their observed behaviour and/or by their interaction with control input, enabling both input-dependent and independent disturbance modelling. We develop a sound and complete methodology to perform correct-by-design synthesis. We show that this optimal control problem is equivalent to a reachability problem in the fairness-realizable sub-product Markov decision process. We validate our results over a case study of room temperature regulation.

I. INTRODUCTION

Linear Temporal Logic (LTL) has gained considerable attention in the synthesis of policies for dynamical systems under uncertainty. It provides a precise formalism for task specification and, when combined with probabilistic model checking, enables fully automated, correct-by-design controller synthesis [1]. LTL-based approaches have been successfully applied in domains such as autonomous navigation and robotic control [2], [3].

Practical dynamical systems are often subject to both aleatoric and epistemic uncertainty [4]: the former arises from intrinsic randomness, while the latter reflects incomplete system knowledge. Most work in LTL synthesis addresses aleatoric uncertainty, typically modelled by stochastic transitions in Markov Decision Processes (MDPs) [3]. More recent studies [2], [5] extend this address both aleatoric and epistemic uncertainty by abstracting the system as an uncertain MDP (e.g., interval MDPs) to synthesise robust policies against worst-case scenarios. However, we argue that such robustness can yield overly conservative solutions.

Example. Consider a scalar discrete-time linear system:

$$x_{t+1} = x_t + u_t + w_t + v_t,$$

where $x_t \in [-10, 10]$ is the state, $u_t \in [-1, 1]$ the control input, $w_t \in \{-2, 0, 2\}$ an epistemic disturbance from an

unknown but bounded set, and $v_t \sim \mathcal{N}(0, 1.5^2)$ an aleatoric disturbance. The objective is to maximise the probability that $x_{100} \in [-3, 3]$, starting from $x_0 = 3$. Based on 1000 Monte Carlo simulations, a robust policy assuming $w_t = 2$ at all times yields a probability of zero, whereas imposing a realistic constraint— $w_t = 0$ occurs infinitely often (i.e., the epistemic disturbance is not always at its peak)—increases the probability to 0.43.

Motivated by the above arguments and by the observation that worst-case disturbances rarely occur in practice, we introduce and extend the *action-based fairness constraint*, a concept from formal verification, to qualitatively capture realistic epistemic disturbance behaviours. Rather than adopting a purely pessimistic robust policy, we develop an *optimistic-pessimistic control policy*, termed a *fair policy*, that maximises the probability of satisfying an LTL specification under this fairness constraint. The fair behaviour of epistemic disturbances can be flexibly specified either from their own observed dynamics or through their interaction with the control input, thereby accommodating both input-independent and input-dependent disturbances. Our main contributions are as follows:

- We formulate a novel optimal LTL control problem within the framework of fair adversarial MDPs, enabling the synthesis of fair policies that formally capture realistic epistemic disturbance behaviours, both input-dependent and input-independent.
- We develop a sound and complete algorithm for synthesising optimal fair control policies, showing that the original problem reduces to a reachability problem in the fairness-realizable sub-product MDP.
- We validate the proposed approach through a case study on room temperature regulation, demonstrating the effectiveness of fairness constraints.

Related Work. *LTL-based control synthesis under uncertainty.* Control synthesis for probabilistic systems with LTL-specified tasks has recently attracted significant attention in the control community [6], [7]. This line of work enables the formal synthesis of control policies for complex specifications such as safety and liveness. *Fairness.* Fairness constraints, introduced in [8] and extended to transition systems in [9], have been systematically surveyed in [10]. In probabilistic settings, they guide resolution of nondeterminism in MDPs with LTL [11], [12], with polynomial-time analysis under strong and weak fairness proposed in [13]. Existing work, however, assumes a single action space and focuses on verification. More recent approaches interpret fairness as ω -regular winning conditions in games [14], [15].

C. Zhou, T. Parisini, and Y. Gao are with the Dept. of Electrical and Electronic Engineering, Imperial College London, United Kingdom. T. Parisini is also with the Dept. of Electronic Systems, Aalborg University, Denmark and with the Dept. of Engineering and Architecture, University of Trieste, Italy. Email: {c.zhou24, t.parisini, yulong.gao}@imperial.ac.uk

P. Yu is with the Dept. of Computer Science, University College London, United Kingdom. pian.yu@ucl.ac.uk

A. Abate is with the Dept. of Computer Science, University of Oxford, Oxford, United Kingdom. alessandro.abate@cs.ox.ac.uk

By contrast, we study a broader class of action-based fairness (unconditional, strong, weak) to capture realistic epistemic disturbance behaviours.

II. PRELIMINARIES

In this section, we provide the preliminaries on LTL specifications and on the concept of fairness.

A. LTL and Limit-Deterministic Büchi Automata

We use LTL to formally specify the desired system properties and state-based fairness constraints. LTL consists of path formulae over a general alphabet AP comprising atomic propositions, defined by the following syntax [1]:

$$\varphi ::= \text{true} \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \mathbf{U}\varphi_2,$$

where $a \in AP$ is an atomic proposition. $\bigcirc\varphi$ holds if φ is true at the next time step. $\varphi_1 \mathbf{U}\varphi_2$ holds if φ_1 remains true until φ_2 eventually becomes true. The temporal operators “eventually” ($\Diamond$) and “always” ($\Box$) are derived as $\Diamond\varphi := \text{true} \mathbf{U} \varphi$ and $\Box\varphi := \neg\Diamond\neg\varphi$. Due to space constraints, we omit the full LTL semantics and refer the reader to [1] for details. Any LTL formula can be equivalently represented as a nondeterministic Büchi automaton (NBA) [16]. Recently, a class of NBAs known as Limit-Deterministic Büchi Automata (LDBAs) [17] has been considered as particularly appropriate for the quantitative analysis of probabilistic systems [17], and we also utilise the LDBAs in this paper.

Definition 1 (LDBA [17]). An LDBA $\mathcal{A}$ is defined as a tuple $\mathcal{A} = (\Sigma, Q, q_0, \Delta, Acc)$ where Σ is an alphabet, $Q = Q_i \uplus Q_{acc}$ is the set of states partitioned into two disjoint sets Q_i and Q_{acc} . $q_0 \in Q_i$ is the initial state. $\Delta = \Delta_i \uplus \Delta_{acc} \uplus \Delta_j$ where: $\Delta_i : Q_i \times \Sigma \rightarrow Q_i$, $\Delta_{acc} : Q_{acc} \times \Sigma \rightarrow Q_{acc}$, $\Delta_j : Q_i \times \{\epsilon\} \rightarrow 2^{Q_{acc}}$, and $Acc \subseteq Q_{acc} \times \Sigma \times Q_{acc}$ is the set of accepting transitions.

B. Action-based Fairness Constraint

Following [1], we adopt three fairness notions¹: unconditional, strong, and weak. Unconditional fairness requires a behaviour to occur infinitely often; strong fairness requires that if a behaviour occurs infinitely often, so must the desired one; and weak fairness requires the desired behaviour to occur infinitely often only if a required action continuously enabled after a certain time point. We then formalise it as:

Definition 2 (Action-based Fairness Constraints [1]). An action-based fairness for Act is a triple:

$$\mathcal{F} = (\mathcal{F}_{ucond}, \mathcal{F}_{strong}, \mathcal{F}_{weak})$$

with $\mathcal{F}_{ucond}, \mathcal{F}_{strong}, \mathcal{F}_{weak} \subseteq 2^{Act}$. An infinite execution $\rho = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \dots$ is fair if for all $A \in \mathcal{F}_{ucond}$, there are infinitely many i , s.t. $\alpha_i \in A$; for all $A \in \mathcal{F}_{strong}$, it holds that $(\exists i. Act(s_i) \cap A \neq \emptyset) \implies (\exists i. \alpha_i \in A)$; and for all $A \in \mathcal{F}_{weak}$, it holds that $(\forall i. Act(s_i) \cap A \neq \emptyset) \implies (\exists i. \alpha_i \in A)$. Here, $\exists i$ denotes “there exist infinitely many i ”, and $\forall i$ denotes “for all except finitely many i ”. $Act(s)$ denotes the set of available actions at state s .

¹These notions apply to general transition systems, including MDPs.

III. MATHEMATICAL MODEL

Consider the discrete-time dynamical model

$$x_{t+1} = f(x_t, u_t, w_t) + v_t \quad (1)$$

where $x_t \in X$ is the system state, $u_t \in U$ is the control input, $w_t \in W$ is an unknown, bounded control noise, v_t is a stochastic disturbance with probability density function ϕ , $f : X \times U \times W \rightarrow X$ represents the known system dynamics.

Abstraction. We adopt the Adversary Markov Decision Process (AMDP) model from [18] to abstract the system dynamics. AMDP differs from the standard MDP in a key aspect: it incorporates two action spaces—control input u and disturbance w —to govern the model.

Definition 3 (AMDP [18]). An AMDP is a tuple $\mathcal{M} = \langle S, s_0, Act_u, Act_w, \delta, AP, L \rangle$, where S is a finite, non-empty set of states with an initial state $s_0 \in S$; Act_u, Act_w are two finite action spaces; $\delta : S \times Act_u \times Act_w \rightarrow \text{Dist}(S)$ is a (partial) probabilistic transition function; AP is an atomic proposition set; and $L : S \rightarrow 2^{AP}$ is a labelling function.

Furthermore, since this paper considers fairness behaviour specification in the system, we extend the AMDP model to the fair AMDP, formally defined as follows:

Definition 4 (Fair Adversary Markov Decision Process (FAMDP)). A FAMDP is an AMDP in which action spaces Act_u and Act_w are following a given (general) action-based fairness specification.

To obtain the FAMDP abstraction of (1), the state space X is partitioned into a finite set of disjoint convex regions $S := \{s_1, \dots, s_N\}$, with $\bigcup_{i=1}^N s_i = X$. Given a point $x \in X$, define the map $Q : X \rightarrow S$ as $Q(x) := \{s_i \in S : x \in s_i\}$, which maps a point x to the unique partition s_i that contains it. Then it holds that $L(x) = L(x'), \forall x, x' : Q(x) = Q(x')$ where L is the labelling function given in Definition 3. This condition guarantees that all points within the same partition share the same true subset of AP . The state space decomposition can be computed exactly or approximately using various existing methods [19]. Similarly, the action space U and the disturbance space W are partitioned into finite sets of discrete subsets, denoted as $Act_u = \{U_a\}_{a \in Act_u}$ and $Act_w = \{W_a\}_{a \in Act_w}$. Then, the probabilistic transition function is given by: $\delta(s, a_u, a_w)(s') := \int_{x' \in s'} \phi(x'; f(x, u, w)) dx'$ where ϕ is the given probability density function.

Path. A path ρ of $\mathcal{M}$ is a sequence $\rho = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \dots$ where $\alpha_j = (a_j^u, a_j^w) \in Act_u \times Act_w$. We define $FPaths$ ($FPaths_s$) and $IPaths$ ($IPaths_s$) as the sets of all finite and infinite paths of $\mathcal{M}$ (starting from state s), respectively. A path is fair if it satisfies the given fairness constraint. We denote $FairPaths^{\mathcal{F}}$ ($FairPaths_s^{\mathcal{F}}$) as the sets of action-based fair infinite paths of $\mathcal{M}$ (starting from state s). The available action sets at s are given by $Act_u(s) = \{a_u \in Act_u \mid \exists a_w \in Act_w, s' \in S, \delta(s, (a_u, a_w))(s') > 0\}$ with $Act_w(s)$ defined analogously.

Policy. A deterministic finite-memory policy π for $\mathcal{M}$ is a function $\pi : FPaths \rightarrow (Act_u \times Act_w)$. The policy π

is memoryless if $\rho[-1] = \rho'[-1]$, then $\pi(\rho) = \pi(\rho')$. For a deterministic memoryless policy, we denote $\pi(s)$ as $\pi(\rho)$ whenever $\rho[-1] = s$. Furthermore, a fair policy:

Definition 5 (Fair Policy [1]). Let $\mathcal{F}$ be an action-based fairness constraint. A policy $\pi \in \Pi$ for the model $\mathcal{M}$ is said to be *fair* if, for all states $s \in S$, it almost surely induces fair infinite paths, i.e., $\Pr_s^\pi \{\rho \in \text{FairPaths}_s^\pi\} = 1$. We denote the set of all fair policies of $\mathcal{M}$ by $\text{fair}(\Pi)$. And, $\mathcal{F}$ is *realisable* in $\mathcal{M}$ if there exists at least one fair policy.

The probability that $\mathcal{M}$ satisfies φ under fairness constraint $\mathcal{F}$ with fair policy π is $\Pr^\pi(\mathcal{M} \models_{\mathcal{F}} \varphi) = \Pr_{\mathcal{M}}^\pi(s_0 \models_{\mathcal{F}} \varphi)$, where $\Pr_{\mathcal{M}}^{\pi \in \Pi}(s \models_{\mathcal{F}} \varphi) = \Pr_{\mathcal{M}}^{\pi \in \text{fair}(\Pi)}(s \models_{\mathcal{F}} \varphi)$.

IV. GENERAL ACTION-BASED FAIRNESS FRAMEWORK AND PROBLEM FORMULATION

A. General Action-based Fairness Framework

In Section II-B, we present the action-based fairness framework (Definition 2) from [1]. This framework assumes a single action space, which is inadequate for FAMDP in two respects: (1) FAMDP comprises two distinct action spaces, Act_u and Act_w ; and (2) it cannot capture dependencies between action spaces, which are essential for modelling input-dependent disturbances under both strong and weak fairness. To overcome these limitations, we extend the framework to FAMDP as follows:

Definition 6 (General Action-based Fairness). An action-based fairness constraint for a FAMDP $\mathcal{M} = \langle S, s_0, Act_u, Act_w, \delta, AP, L \rangle$ is defined as a triple: $\mathcal{F} = (\mathcal{F}_{ucond}, \mathcal{F}_{strong}, \mathcal{F}_{weak})$ with $\mathcal{F}_{ucond}, \mathcal{F}_{strong}, \mathcal{F}_{weak} \in \{(enabled, taken) \mid enabled \in 2^{Act_m}, taken \in 2^{Act_n}\}$ where $m, n \in \{u, w\}$. For an infinite execution $\rho = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \dots$ in $\mathcal{M}$, where $\alpha_j = (a_j^u, a_j^w) \in Act_u \times Act_w$, an enabled action space is $A_e \subseteq 2^{Act_m}$ and a taken action space is $A_t \subseteq 2^{Act_n}$, with $m, n \in \{u, w\}$: i) ρ is unconditionally fair if for all $(\emptyset, A_t) \in \mathcal{F}_{ucond}$, $\exists^\infty j. a_j^n \in A_t$; ii) ρ is strongly fair if for all $(A_e, A_t) \in \mathcal{F}_{strong}$, $(\exists^\infty i. Act_m(s_i) \cap A_e \neq \emptyset) \implies (\exists^\infty j. a_j^n \in A_t)$; iii) ρ is weakly fair if for all $(A_e, A_t) \in \mathcal{F}_{weak}$, $(\forall^\infty i. Act_m(s_i) \cap A_e \neq \emptyset) \implies (\exists^\infty j. a_j^n \in A_t)$.

B. Problem Formulation

Problem 1. Given a discrete-time dynamic system abstracted as FAMDP $\mathcal{M}$, a LTL task specification φ and a general action-based fairness constraint $\mathcal{F}$ as defined in Definition 6, synthesise the optimal fair policy $\pi = (\pi_u, \pi_w)$. That is, solve the following optimisation problem:

$$\max_{\pi} \Pr^\pi(\mathcal{M} \models_{\mathcal{F}} \varphi) \quad (2)$$

V. MAIN RESULTS

We now present our solution to Problem 1 for a given FAMDP $\mathcal{M}$, LTL formula φ , and general action-based fairness constraint $\mathcal{F}$. Our approach follows an automata-based method that reduces the problem to a reachability

analysis. However, the presence of the fairness constraint introduces additional challenges. The solution consists of three key steps: 1) Construct the product MDP, denoted as $\mathcal{M}^\times$, by combining the FAMDP $\mathcal{M}$ with the LDBA $\mathcal{A}_\varphi$ derived from the LTL specification φ ; 2) If the given fairness constraint $\mathcal{F}$ is realisable, construct the fairness-realisable sub-product MDP $\mathcal{M}_{\mathcal{F}}^\times$. The original problem is then reduced to a reachability problem over $\mathcal{M}_{\mathcal{F}}^\times$; 3) Synthesise the fair optimal policy of the reachability problem within the sub-product MDP $\mathcal{M}_{\mathcal{F}}^\times$.

A. Product MDP

Following the standard automata-based approach, we first construct the MDP $\mathcal{M}^\times$ by composing the FAMDP $\mathcal{M}$ with the LDBA $\mathcal{A}$ derived from the LTL specification φ . This construction is defined as follows:

Definition 7 (Product MDP). Given a FAMDP $\mathcal{M} = \langle S, s_0, Act_u, Act_w, \delta, AP, L \rangle$ and an LDBA $\mathcal{A}_\varphi = \langle \Sigma, Q, q_0, \Delta, Acc \rangle$ corresponding to the LTL formula φ , the product MDP $\mathcal{M}^\times$ is defined as: $\mathcal{M}^\times = \mathcal{M} \otimes \mathcal{A}_\varphi = \langle S^\times, s_0^\times, Act^\times, \delta^\times, L^\times, Acc^\times \rangle$, where: $S^\times = S \times Q$ is the state space, with initial state $s_0^\times = (s_0, \Delta_{\mathcal{A}_\varphi}(q_0, L(s_0)))$ and $Act^\times = Act_u \times Act_w$; $\delta^\times((s, q), (a_u, a_w))((s', q')) = \delta(s, a_u, a_w)(s')$ if $q' = \Delta(q, L(s'))$, and 0 otherwise; The labelling function is given by $L^\times((s, q)) = L(s)$; The accepting condition $Acc^\times$ is defined as: $Acc^\times = \{((s, q), (a_u, a_w), (s', q')) \in S^\times \times (Act_u \times Act_w) \times S^\times \mid (q, L(s), q') \in Acc\}$.

B. Construction of Fairness-Realisable sub-Product MDP

Without fairness constraints, standard automata-based planning methods for MDPs with LTL specifications, typically involve computing maximal end components (MECs) in the product MDP and identifying those that are accepting. These accepting MECs (AMECs) are then treated as goal states in the corresponding reachability planning problems. However, as illustrated in Problem 1, our focus is on synthesising a fair policy—one that satisfies the given action-based fairness constraint. Notably, such a fair policy may not always exist within the model $\mathcal{M}^\times$ [1]. To this end, and inspired by [13], verifying the realisability of the specified fairness condition $\mathcal{F}$ is a crucial step. If the constraint is realisable, we then proceed to construct the fairness-realisable sub-product MDP, which consists of all states reachable under fair policies. This sub-MDP then serves as the foundation for subsequent fair policy synthesis.

To illustrate following techniques, we first introduce the definitions and properties of sub-MDP, (accepting) MEC (AMEC), (accepting) fair MECs ((A)FMECs) and fairness-realisable sub product MDPs. Following this, Algorithm 1 outlines the overall procedure for constructing a fairness-realisable sub product MDP if the given fairness constraint is realisable.

Definition 8 (sub-MDP). A sub-MDP of an MDP $\mathcal{M}$ is a pair (T, A) , where $T \subseteq S$ is non-empty and $A : T \rightarrow 2^{Act}$ is a function such that (1) $\forall s \in T, A(s) \neq \emptyset$ and $A(s) \subseteq$

$Act(s)$; (2) if $s \in T$ and $\alpha \in A(s)$, then $Post(s, \alpha) = \{t \in S \mid \delta(s, \alpha)(t) > 0\} \subseteq T$.

Definition 9 (End Component). An **end component** of an MDP $\mathcal{M}$ is a sub-MDP (T, A) such that the digraph $M_{T,A}$ induced by (T, A) is strongly connected. A **maximal EC** is an EC (T, A) that is not strictly contained in any other EC (T', A') , i.e., there exists no larger EC such that $(T, A) \subsetneq (T', A')$. The set of MECs in an MDP is finite and can be uniquely identified using the algorithm presented in [20]. A **fair MEC** (FMEC) is an MEC that satisfies the given fairness constraint $\mathcal{F}$, i.e. there exists a policy π in MEC such that: $\Pr^\pi(\text{MEC} \models \mathcal{F}) = 1$. Additionally, an **accepting (F)MECs** an (F)MEC that satisfies Büchi acceptance conditions as defined II-A.

Definition 10 (Fairness Realisable sub-Product MDP). Given a union state set of FMECs $S_{\mathcal{F}}$ in product MDP $\mathcal{M}^\times$, a fairness realisable sub product MDP is a sub product MDP $\mathcal{M}_{\mathcal{F}}^\times = \langle S_{\mathcal{F}}^\times, s_0^\times, Act_{\mathcal{F}}^\times, \delta_{\mathcal{F}}^\times, L^\times, Acc^\times \rangle$, where $\forall s \in S_{\mathcal{F}}^\times = S_{\mathcal{F}} \subseteq S^\times$ can reach to at least one FMECs, $Act_{\mathcal{F}}^\times(s) \subseteq Act^\times(s)$ is non-empty for all $s \in S_{\mathcal{F}}^\times$, and $\delta_{\mathcal{F}}^\times(s, \alpha) = \delta^\times(s, \alpha)$ for $s \in S_{\mathcal{F}}^\times, \alpha \in Act_{\mathcal{F}}^\times(s)$ with $\text{Supp}(\delta^\times(s, \alpha)) \subseteq S_{\mathcal{F}}^\times$.

Lemma 1 (FMEC Properties. Lem. 2 of [13] and Thm. 3.1 and 4.2 of [20]). There exists a finite-memory policy π such that i) visits every state-action pair in FMEC infinitely often with probability 1; ii) for all states $s \in \text{FMEC}$, $\Pr_s^\pi\{\rho \in \text{FairPaths}_s^\pi\} = 1$; and iii) stays in FMEC forever.

Algorithm 1 Build Fairness Realisable Sub Product MDP

Input: A general action-based fairness constraint $\mathcal{F}$; A product MDP $\mathcal{M}^\times = \langle S^\times, s_0^\times, Act^\times, \delta^\times, L^\times, Acc^\times \rangle$

Output: A fairness-realizable sub-product MDP $\mathcal{M}_{\mathcal{F}}^\times$; A set of FMECs

```

1:  $\mathcal{E} \leftarrow \text{compute\_MECs}(\mathcal{M}^\times)$ 
    $\triangleright$  Step 1: Identify all MECs in  $\mathcal{M}^\times$ 
2:  $\text{FMECs} \leftarrow \emptyset$ 
    $\triangleright$  Step 2: Identify FMECs with Algorithm 2
3: for each  $(T, A) \in \mathcal{E}$  do
4:   if is_fair_MEC $((T, A), \mathcal{F})$  then
5:      $\text{FMECs} \leftarrow \text{FMECs} \cup \{(T, A)\}$ 
    $\triangleright$  Step 3: Identify states that can reach an FMEC
6:  $S_{\mathcal{F}} \leftarrow \text{find\_ancestors}(\mathcal{M}^\times, \text{FMECs})$ 
    $\triangleright$  Step 4: Determine fairness realisability
7: if  $s_0^\times \in S_{\mathcal{F}}$  then
8:    $\mathcal{M}_{\mathcal{F}}^\times$  constructed by following Definition 10 with  $S_{\mathcal{F}}$ .
9:   return  $\mathcal{M}_{\mathcal{F}}^\times, \text{FMECs}$ 
10: else
11:   return None,  $\emptyset$ 
```

In Algorithm 1, we first compute the MECs of the product MDP. Then, adjusting the recursive algorithm proposed in [13], presented as Algorithm 2, we identify the set of fair MECs by verifying whether each MEC $\models \mathcal{F}$. The soundness and completeness of Algorithm 2 follow the structure of

Algorithm 2 Recursive Algorithm: **is_fair_MEC** $((T, A), \mathcal{F})$. (Adjusted from Algorithm 1 from [13])

Input: A general action-based fairness constraint $\mathcal{F}$; An MEC (T, A) .

Output: True if the given MEC is fair, otherwise False.

```

1:  $(\mathcal{F}_{ucond}, \mathcal{F}_{strong}, \mathcal{F}_{weak}) \leftarrow \mathcal{F}$ 
2:  $A_u \times A_w \leftarrow A$ 
3: for all  $(\emptyset, A_t) \in \mathcal{F}_{ucond}$  do  $\triangleright$  Unconditional
4:   if  $(A_u \cup A_w) \cap A_t = \emptyset$  then
5:     return False
6: for all  $(A_e, A_t) \in \mathcal{F}_{strong}$  do  $\triangleright$  Strong
7:   if  $(A_u \cup A_w) \cap A_e \neq \emptyset$  and  $(A_u \cup A_w) \cap A_t = \emptyset$  then
8:      $\mathcal{E}' \leftarrow \text{compute\_MECs}((T, A) \ominus A_e)$   $\triangleright$ 
        $((T, A) \ominus A_e)$  denotes the subgraph where  $\forall t \in T$  with
        $A'(t) = A(t) \setminus A_e$ .
9:      $\mathcal{F}' \leftarrow (\mathcal{F}_{ucond}, \mathcal{F}_{strong} \setminus (A_e, A_t), \mathcal{F}_{weak})$ 
10:    is_fair  $\leftarrow$  False
11:    for all  $(T', A') \in \mathcal{E}'$  do
12:      if is_fair_MEC $((T', A'), \mathcal{F}')$  then
13:        is_fair  $\leftarrow$  True
14:    if not is_fair then
15:      return False
16: for all  $(A_e, A_t) \in \mathcal{F}_{weak}$  do  $\triangleright$  Weak
17:   if  $(A_u \cup A_w) \cap A_t = \emptyset$  and  $[(A_e \subseteq Act_u) \wedge (A_u \setminus A_e = \emptyset)]$  or  $[(A_e \subseteq Act_w) \wedge (A_w \setminus A_e = \emptyset)]$  then
18:     return False
19: return True
```

Alg. 1 in [13] and are omitted due to space constraints. The properties of FMECs in Lemma 1 enable reducing an MDP planning problem with LTL under fairness constraints to a reachability problem over the product MDP. However, as previously noted, the fairness constraint may not always be realisable. Therefore, inspired by Theorem 1 in [13], we perform a reachability analysis in $\mathcal{M}^\times$ to determine the set of states $S_{\mathcal{F}}$ that can reach at least one FMEC. If the initial state $s_0 \in S_{\mathcal{F}}$, then the given action-based fairness constraint is realisable, allowing us to construct a fairness-realizable sub-product MDP $\mathcal{M}_{\mathcal{F}}^\times$. Since the sub-product $\mathcal{M}_{\mathcal{F}}^\times$ produced by Algorithm 1 includes all states reachable under all fair policies, as shown in Lemma 2, Theorem 1 then follows.

Lemma 2. The product MDP $\mathcal{M}^\times$ and the fairness-realizable sub-product MDP $\mathcal{M}_{\mathcal{F}}^\times$ constructed by Algorithm 1 have the same set of fair policies.

Proof. The state space $S_{\mathcal{F}}$ of the fairness-realizable sub-product MDP $\mathcal{M}_{\mathcal{F}}^\times$, as constructed by Algorithm 1, contains all states in $\mathcal{M}^\times$ that can reach at least one FMEC. Therefore, for all states $s \in S' = S^\times \setminus S_{\mathcal{F}}$, we have $s \not\models \exists \Diamond \text{FMEC}$. This implies that, under any policy π in $\mathcal{M}^\times$, $\Pr_s^\pi(\rho \in \text{FairPaths}_s^\pi) = 0$. Consequently, under any fair policy $\pi_{\mathcal{F}}$ for $\mathcal{M}^\times$, the induced infinite paths will never visit any state in S' . It follows that the fair behaviours of $\mathcal{M}^\times$ are entirely contained within $\mathcal{M}_{\mathcal{F}}^\times$, and thus both have

the same set of fair policies. $\square$

Theorem 1. Given an FAMDP $\mathcal{M}$, an LTL formula φ , and a general action-based fairness constraint $\mathcal{F}$, the maximal probability of satisfying φ under the fairness constraint $\mathcal{F}$ is equivalent to the maximal probability of reaching the set $\mathcal{E}_{\mathcal{F}}^{\times}$ in the fairness-realizable sub-product MDP $\mathcal{M}_{\mathcal{F}}^{\times}$. Here, $\mathcal{E}_{\mathcal{F}}^{\times}$ and $\mathcal{E}^{\times}$ denote the unions of the state sets of all AFMECs in $\mathcal{M}_{\mathcal{F}}^{\times}$ and all AMECs in $\mathcal{M}^{\times}$, respectively.

$$\max_{\pi \in \Pi} \Pr^{\pi}(\mathcal{M} \models_{\mathcal{F}} \varphi) = \max_{\pi^{\times} \in \Pi^{\times}} \Pr^{\pi^{\times}}(\mathcal{M}^{\times} \models_{\mathcal{F}} \Diamond \mathcal{E}^{\times}) \quad (3)$$

$$= \max_{\pi_{\mathcal{F}}^{\times} \in \text{fair}(\Pi^{\times})} \Pr^{\pi_{\mathcal{F}}^{\times}}(\mathcal{M}^{\times} \models \Diamond \mathcal{E}^{\times}) \quad (4)$$

$$= \max_{\pi_{\mathcal{F}}^{\times} \in \text{fair}(\Pi_{\mathcal{F}}^{\times})} \Pr^{\pi_{\mathcal{F}}^{\times}}(\mathcal{M}_{\mathcal{F}}^{\times} \models \Diamond \mathcal{E}_{\mathcal{F}}^{\times}) \quad (5)$$

$$= \max_{\pi^{\times} \in \Pi_{\mathcal{F}}^{\times}} \Pr^{\pi^{\times}}(\mathcal{M}_{\mathcal{F}}^{\times} \models \Diamond \mathcal{E}_{\mathcal{F}}^{\times}) \quad (6)$$

Proof. By the standard automata-based planning approach and the properties of FMECs presented in Lemma 1, we obtain Equation (3). Equation (4) follows directly from the definition of $\models_{\mathcal{F}}$. By Lemma 2, we know that $\text{fair}(\Pi^{\times}) = \text{fair}(\Pi_{\mathcal{F}}^{\times})$, and that $\mathcal{M}_{\mathcal{F}}^{\times}$ preserves all paths and transitions that could be taken by an optimal fair policy in $\mathcal{M}^{\times}$. This yields Equation (5). Finally, following Lem. 10.131 in [1], we show that the equality in Equation (6) holds. We have $\max_{\pi_{\mathcal{F}}^{\times} \in \text{fair}(\Pi_{\mathcal{F}}^{\times})} \Pr^{\pi_{\mathcal{F}}^{\times}}(\mathcal{M}_{\mathcal{F}}^{\times} \models \Diamond \mathcal{E}_{\mathcal{F}}^{\times}) \leq \max_{\pi^{\times} \in \Pi_{\mathcal{F}}^{\times}} \Pr^{\pi^{\times}}(\mathcal{M}_{\mathcal{F}}^{\times} \models \Diamond \mathcal{E}_{\mathcal{F}}^{\times})$, since fair policies form a subset of all policies. Let $\hat{\pi} \in \Pi^{\times}$ be a policy achieving the maximum probability on the right-hand side. Define $\hat{\pi}_{\mathcal{F}}^{\times}$ to behave as $\hat{\pi}$ for all finite histories states until reaching $\mathcal{E}_{\mathcal{F}}^{\times}$, after which it follows a round-robin policy. By Lemma 1, the system remains within the FMEC, and for all $s \in \text{FMEC}$, $\Pr_s^{\hat{\pi}_{\mathcal{F}}^{\times}}\{\rho \in \text{FairPaths}_s^{\hat{\pi}_{\mathcal{F}}^{\times}}\} = 1$. Paths with a finite prefix to $\mathcal{E}_{\mathcal{F}}^{\times}$ are thus fair. Hence, $\hat{\pi}_{\mathcal{F}}^{\times}$ achieves the same maximal probability as $\hat{\pi}$ while satisfying fairness constraint, i.e., $\hat{\pi}_{\mathcal{F}}^{\times} \in \text{fair}(\Pi_{\mathcal{F}}^{\times})$, justifying the equality in Equation (6). $\square$

Thanks to Theorem 1, Problem 1 now reduces to the problem of maximising reachability probabilities for AFMECs $\mathcal{E}_{\mathcal{F}}^{\times}$ in fairness-realizable sub-MDP $\mathcal{M}_{\mathcal{F}}^{\times}$ under all policies.

C. Optimal Control Policy under Fairness Constraints

The maximum probability of reaching AFMECs $\mathcal{E}_{\mathcal{F}}^{\times}$ in $\mathcal{M}_{\mathcal{F}}^{\times}$ can be computed using value iteration. We define $\mathcal{N}_e$ as the set of states from which $\mathcal{E}_{\mathcal{F}}^{\times}$ is unreachable under any policy. For each state $s_k \in \mathcal{S}_{\mathcal{F}}^{\times}$, the value function $v(k)$ represents the probability of eventually reaching $\mathcal{E}_{\mathcal{F}}^{\times}$. The initial values are assigned as: $v_0(k) = 1$ if $s_k \in \mathcal{E}_{\mathcal{F}}^{\times}$, else 0. For all states $s_k \in \mathcal{S}_{\mathcal{F}}^{\times} \setminus (\mathcal{E}_{\mathcal{F}}^{\times} \cup \mathcal{N}_e)$, the values are updated iteratively according to:

$$v_{t+1}(k) = \max \left\{ \sum_{s_i \in \mathcal{S}_{\mathcal{F}}^{\times}} \delta_{\mathcal{F}}^{\times}(s_k, \alpha)(s_i) v_t(i) \mid \alpha \in \text{Act}_{\mathcal{F}}^{\times}(s_k) \right\}, \quad (7)$$

The value iteration algorithm converges to the value function denoted by v^* . This function represents the maximum

probability of reaching the accepting set $\mathcal{E}_{\mathcal{F}}^{\times}$ in the fair realisable sub-product MDP $\mathcal{M}_{\mathcal{F}}^{\times}$ under a fair policy $\hat{\pi}_{\mathcal{F}}^{\times}$. Once v^* is obtained, we can synthesise an optimal fair policy $\hat{\pi}_{\mathcal{F}}^{\times} : \mathcal{S}_{\mathcal{F}}^{\times} \rightarrow \text{Act}_{\mathcal{F}}^{\times}$ for each state $s_k \in \mathcal{S}_{\mathcal{F}}^{\times} \setminus (\mathcal{E}_{\mathcal{F}}^{\times} \cup \mathcal{N}_e)$ by computing $v^*(k) = \sum_{s_i \in \mathcal{S}_{\mathcal{F}}^{\times}} \delta_{\mathcal{F}}^{\times}(s_k, \alpha)(s_i) v_t(i)$, and selecting the action α that satisfies Equation (7) for the state s_k . For states in $\mathcal{E}_{\mathcal{F}}^{\times}$, the round-robin policy is applied, while for those in $\mathcal{N}_e$, actions can be chosen arbitrarily, as they do not affect the probability of reaching $\mathcal{E}_{\mathcal{F}}^{\times}$. Following this procedure, we obtain a state-based optimal fair control policy $\hat{\pi}_{\mathcal{F}}^{\times}$ that maximises the probability of reaching the accepting FMECs in the fair realisable sub-product MDP $\mathcal{M}_{\mathcal{F}}^{\times}$.

VI. CASE STUDY

In this section, we present a case study to illustrate the effectiveness of our method.² We consider the room temperature regulation benchmark [21], which models the control of indoor temperatures through cooling water circulation. We abstracted the dynamic system into a FAMDP using the software tool **FAUST**² [22]. The room and cooling water temperature ranges were defined as [19.5, 20.5] and [4.5, 5.5], respectively, with the input space $U = [0, 1]$.

A. Effectiveness of Fairness Constraints

We analyse the impact of fairness on room temperature dynamics with disturbance $w \in [-0.3, 0.3]$ and abstraction error 4, yielding a FAMDP $\mathcal{M}$ with 81 states, 10 control partitions, and 6 disturbance partitions. The objective is $\Diamond S_{goal1}$, where $S_{goal1} = \{x_1 \in [19.5, 19.72], x_2 \in [4.94, 5.05]\}$. Unconditional fairness requires all disturbance partitions except extremes to recur infinitely often: $F_{ucond} = \{(\emptyset, \{a_w^1\}), (\emptyset, \{a_w^2\}), (\emptyset, \{a_w^3\}), (\emptyset, \{a_w^4\})\}$, preventing persistence of extreme disturbances. Using the methods in Section V, we compute optimal fair policies and evaluate them via 1,000 Monte Carlo runs of 1,000 steps under four distributions of w . Figure 1 compares robust, fairness-constrained, and simulated satisfaction probabilities. Results show fairness reduces conservatism and increases satisfaction probability (e.g. from nearly 0% to ~25% in state region $x_1 \in [20.17, 20.28], x_2 \in [5.39, 5.5]$).

B. Fairness in Input-Dependent Disturbance

We next consider input-dependent disturbance with $w \in [-0.5, 0.5]$ and the same abstraction error, giving 81 states and 10 partitions for both u and w . The objective is $\Diamond S_{goal2}$, where $S_{goal2} = \{x_1 \in [19.94, 20.05], x_2 \in [4.83, 5.05]\}$. Strong fairness requires that if the valve is infinitely often in low range (a_u^{0-4}), then small disturbances also recur infinitely often: $F_{strong} = \{(\{a_u^{0-4}\}, \{a_w^3\}), (\{a_u^{0-4}\}, \{a_w^4\}), (\{a_u^{0-4}\}, \{a_w^5\}), (\{a_u^{0-4}\}, \{a_w^6\})\}$. Figure 2 shows optimal fair policies and satisfaction probabilities under robust, fairness-constrained, and simulated scenarios. Strong fairness captures input-dependent disturbance more effectively, again yielding higher satisfaction probabilities.

²All simulations were performed on a Surface Laptop 5 equipped with a 2.7 GHz Intel Core i7 processor and 16 GB of RAM.

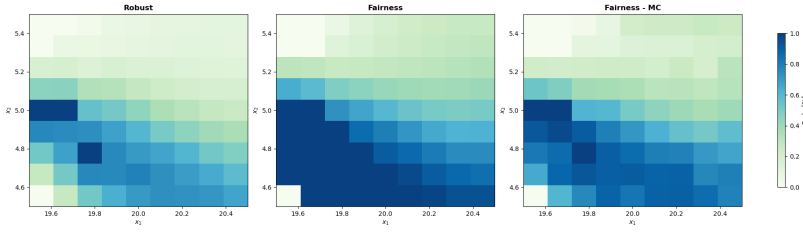


Fig. 1: Effectiveness of Fairness Constraints: comparison of maximal satisfying probabilities under the optimal robust policy (left), the optimal fair policy (middle), and the Monte Carlo satisfying probabilities (right) obtained by fixing the optimal fair policy u and randomly sampling the disturbance w . The x - and y -axes represent x_1 and x_2 , respectively.

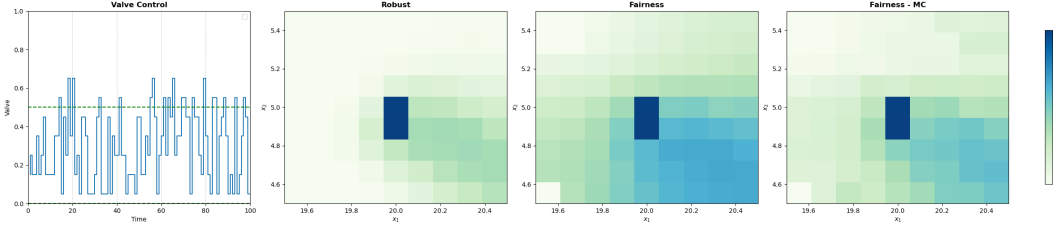


Fig. 2: Fairness in input-dependent disturbance: optimal fair policy for u (left), probabilities under the optimal robust policy (middle-left) and the optimal fair policy (middle-right), and Monte Carlo estimates with the optimal fair policy (right).

VII. CONCLUDING REMARKS

This paper addresses the fair control of uncertain dynamical systems with LTL specifications under action-based fairness constraints. The key contribution is to replace worst-case assumptions in robust control with fairness-based assumptions, thereby avoiding undue conservatism. We extend the action-based fairness framework from formal verification to formally capture realistic fairness behaviours of epistemic disturbances, grounded in their observed dynamics and interaction with control inputs. A sound and complete algorithm for correct-by-design, optimally fair policy synthesis is proposed and evaluated in a case study on room temperature regulation.

REFERENCES

- [1] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT press, 2008.
- [2] E. M. Wolff, U. Topcu, and R. M. Murray, “Robust control of uncertain markov decision processes with temporal logic specifications,” in *Proceedings of 51st IEEE Conference on Decision and Control*. IEEE, 2012, pp. 3372–3379.
- [3] M. Lahijanian, S. B. Andersson, and C. Belta, “A probabilistic approach for control of a stochastic system from ltl specifications,” in *Proceedings of the 48th IEEE Conference on Decision and Control held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 2236–2241.
- [4] E. Hüllermeier and W. Waegeman, “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods,” *Machine learning*, vol. 110, no. 3, pp. 457–506, 2021.
- [5] T. Badings, L. Romao, A. Abate, D. Parker, H. A. Poonawala, M. Stoelinga, and N. Jansen, “Robust control for dynamical systems with non-gaussian noise via formal abstractions,” *Journal of Artificial Intelligence Research*, vol. 76, pp. 341–391, 2023.
- [6] W. Ren, R. M. Jungers, and D. V. Dimarogonas, “Zonotope-based symbolic controller synthesis for linear temporal logic specifications,” *IEEE Transactions on Automatic Control*, 2024.
- [7] R. Oura, T. Ushio, and A. Sakakibara, “Bounded synthesis and reinforcement learning of supervisors for stochastic discrete event systems with ltl specifications,” *IEEE Transactions on Automatic Control*, 2024.
- [8] E. W. Dijkstra, “Cooperating sequential processes,” in *The origin of Concurrent Programming: from Semaphores to Remote Procedure Calls*. Springer, 2002, pp. 65–138.
- [9] J.-P. Queille and J. Sifakis, “Fairness and related properties in transition systems—a temporal logic to deal with fairness,” *Acta informatica*, vol. 19, pp. 195–220, 1983.
- [10] M. Z. Kwiatkowska, “Survey of fairness notions,” *Information and Software Technology*, vol. 31, no. 7, pp. 371–386, 1989.
- [11] S. Hart, M. Sharir, and A. Pnueli, “Termination of probabilistic concurrent program,” *ACM Transactions on Programming Languages and Systems*, vol. 5, no. 3, pp. 356–380, 1983.
- [12] M. Y. Vardi, “Automatic verification of probabilistic concurrent finite state programs,” in *Proceedings of 26th Annual Symposium on Foundations of Computer Science*. IEEE, 1985, pp. 327–338.
- [13] C. Baier, M. Groesser, and F. Ciesinski, “Quantitative analysis under fairness constraints,” in *Proceedings of International Symposium on Automated Technology for Verification and Analysis*, vol. 5799. Springer, 2009, p. 135.
- [14] K. Chatterjee and N. Fijalkow, “A reduction from parity games to simple stochastic games,” *arXiv preprint arXiv:1106.1232*, 2011.
- [15] T. Banerjee, R. Majumdar, K. Mallik, A.-K. Schmuck, and S. Soudjani, “Fast symbolic algorithms for omega-regular games under strong transition fairness,” *TheoretCS*, vol. 2, 2023.
- [16] M. Y. Vardi and P. Wolper, “Reasoning about infinite computations,” *Information and computation*, vol. 115, no. 1, pp. 1–37, 1994.
- [17] S. Sickert, J. Esparza, S. Jaax, and J. Křetínský, “Limit-deterministic büchi automata for linear temporal logic,” in *Proceedings of International Conference on Computer Aided Verification*. Springer, 2016, pp. 312–332.
- [18] T. Wongpiromsarn and E. Frazzoli, “Control of probabilistic systems under dynamic, partially known environments with temporal logic specifications,” in *Proceedings of IEEE 51st IEEE Conference on Decision and Control*. IEEE, 2012, pp. 7644–7651.
- [19] J.-C. Latombe, “Robot motion planning, vol. 124,” *Springer Science & Business Media, Doi*, vol. 10, pp. 978–1, 2012.
- [20] L. De Alfaro, *Formal verification of probabilistic systems*. stanford university, 1998.
- [21] A. Fehnker and F. Ivančić, “Benchmarks for hybrid systems verification,” in *Proceedings of International Workshop on Hybrid Systems: Computation and Control*. Springer, 2004, pp. 326–341.
- [22] Z. Esmail, S. Soudjani, C. Gevaerts, and A. Abate, “Faust 2: formal abstractions of uncountable-state stochastic processes,” in *Proceedings of 21st International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Newcastle University, 2015.