A Mori-Zwanzig and MITL Based Approach to Statistical Verification of Continuous-time Dynamical Systems

Yu Wang [*,***] **Nima Roohi** [**] **Matthew West** [***]
Mahesh Viswanathan [**] **Geir E. Dullerud** [*,***]

[*] *Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, USA*
[**] *Department of Computer Science, University of Illinois at Urbana-Champaign, USA*
[***] *Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, USA*
{*yuwang8, roohi2, mwest, vmahesh, dullerud*}*@illinois.edu*

Abstract: In this work, we introduce a framework for the statistical verification of Metric Interval Temporal Logic (MITL) formulas on continuous-time dynamical systems. By considering the continuous-time Markov process associated with the dynamical system, we apply the Mori-Zwanzig method to reduce the original system to a Continuous-Time Markov Chain (CTMC). Accordingly, the MITL formulas on the original system can be reduced to MITL formulas on the CTMC. Furthermore, we propose a statistical verification algorithm for checking the MITL formulas on the CTMCand show that the original MITL formulas on the original system can be checked by this procedure.

1. INTRODUCTION

For decades, temporal logic has been a powerful tool to describe the behaviors, such as safety and liveness, of state transition systems (Clarke et al., 1986). For finite-state systems, the temporal logic formulas can be checked automatically by a library of model checkers (Manna and Pnueli, 1992). These model checkers are roughly divided into two classes: symbolic and statistical, in which only the statistical methods, based on sampling and simulation, are able to handle systems of a large number of states (Younes and Simmons, 2006).

During the past several years, various kinds of temporal logic have also been applied to the synthesis and verification of dynamical systems where they are used to specify design objectives. Here, the main challenge is that checking temporal logic formulas on infinite-state systems directly is beyond the computation capacity of the model checkers. One possible way to circumvent this problem is to reduce the dynamical systems to finite state transition systems by abstraction-based methods (Tabuada and Pappas, 2006; Kloetzer and Belta, 2008; Wongpiromsarn et al., 2010). However, finding such an abstraction is currently only possible for specific classes of dynamical systems, such as linear time-invariant systems and piecewise affine systems.

Another idea is to convert the dynamical systems into probabilistic finite state transition systems based on sampling and simulation. In (Liu et al., 2011, 2012; Zuliani et al., 2013; Zuliani, 2014), the authors reduce the dynamical systems governed by ordinary differential equations to continuous-time Markov chains (CTMC) or dynamic Bayesian networks by partitioning the state space and approximating the transition probability between the partitions by sampling. Though this method works for general dynamical systems, there is no deterministic guarantee on the error introduced in this procedure.

In this work, we propose a framework for defining and verifying a kind of temporal logic on these dynamical systems. Specifically, we consider the continuous-time Markov processes generated by a dynamical system and take metric interval temporal logic (MITL) to specify their behavior over time. This is different from current researches on approximating the dynamical systems with discrete-time Markov chains (Soudjani and Abate, 2013) and (Wang et al., 2015).

We use the Mori-Zwanzig method (Chorin et al., 2000; Beck et al., 2009) stemming from the study of model reduction of dynamical systems to reduce the infinite-state nonlinear dynamical system into CTMC. Noting that it usually needs a large number of discrete states to approximate a continuous domain, we adopt the statistical approach to check the CTMC derived and present a concrete algorithm to model check MITL formulas.

The rest of the paper is organized as follows. In Section 2, the preliminaries and the mathematical formulation of the problem are given. In Section 3, the Mori-Zwanzig method is used to reduce the Markov process to a CTMC. Accordingly, the MITL formulas on the original Markov processes are reduced to MITL formulas on the CTMC. In addition, we give the bounds of error introduced by the reduction. In Section 4, we present a statistical verification

[1] The authors acknowledge support for this work from NSF CPS grant 1329991.

algorithm for checking the MITL formulas on the CTMC and show that the result given by the algorithm are of high confidence. Finally, we conclude the main contributions in this work in Section 5.

2. PRELIMINARIES

2.1 Notations

We denote the set of *natural*, *rational*, *non-negative rational*, *real*, *positive real*, and *non-negative real* numbers, respectively by $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{Q}_{\geq 0}$, $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{R}_{\geq 0}$. For any two sets A and B, B^A is a set of elements of B indexed by elements of A. We may look at an element of B^A as a function from A to B. When n is a positive natural number, B^n is the Cartesian product of n copies of set B. Also, B^ω is exactly $B^{\mathbb{N}}$ (the set of infinite sequences of B). The power set of B is denoted by 2^B. If B is finite, the number of its elements is denoted by $|B|$. We denote the set of all intervals in $\mathbb{R}$ and $\mathbb{R}_{\geq 0}$, with rational end-points, respectively by $\mathcal{I}$ and $\mathcal{I}_{\geq 0}$. For any interval $I \in \mathcal{I}$ we denote the lower and upper bounds of I respectively by $\underline{I}$ and $\overline{I}$.

2.2 Continuous Time Markov Processes

Definition 1. A (homogeneous) continuous-time Markov process M is a tuple (Ω, f_0, L) where

- Ω is the continuous *states space*,
- f_0 is the *initial distribution* on Ω,
- L is the *Fokker-Planck operator*.

The Fokker-Planck operator is the analogy of the transition rate matrix in continuous-time Markov chains. The distribution $f(t,x)$ of the Markov process evolves by

$$\frac{\partial f(t,x)}{\partial t} = Lf(t,x). \tag{1}$$

On a finite state space, the Markov process M reduces to a continuous-time Markov chain (CTMC).

Definition 2. A continuous-time Markov chain (CTMC) C is a tuple $(\mathtt{S}, p_0, A)$ where

- $\mathtt{S}$ is the finite *state space*,
- p_0 is the *initial distribution*,
- A is the *transition rate matrix*.

The distribution $p(t)$ of the CTMC evolves by $p(t) = e^{At}p_0$. It can be computed effectively but approximately by sampling based methods such as the stochastic simulation algorithm (`SSA`) (Gillespie, 1976).

2.3 Sampling based algorithms

Since the sampling based algorithms are approximate, determining whether the distribution p satisfies the atomic proposition, namely the inequality $w \cdot p > b$ for some given $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ is only possible when $w \cdot p - b > \delta$ for some arbitrarily small $\delta > 0$. Under this condition, several algorithms have been proposed using either Chernoff bounds, or sequential probability ratio test (see Wald, 1945; Younes, 2006; Sen et al., 2005). Generally, the algorithms $\mathcal{A}_1^\delta(p, w, b, \alpha, \gamma)$ take the indifference parameter δ, and error bounds α and γ as parameters and output either `yes`, `no`, or `unknown`. The `unknown` is given when $|w \cdot p - b|$ is smaller than the given indifference region δ. The algorithms ensure that:

- $\mathbb{P}[res = \mathtt{no} \mid w \cdot p > b] \leq \alpha$
- $\mathbb{P}[res = \mathtt{yes} \mid w \cdot p < b] \leq \alpha$
- $\mathbb{P}[res = \mathtt{unknown} \mid |w \cdot p - b| \geq \delta] \leq \gamma$

The parameters δ, α, and γ can be made arbitrary small at the cost of more samples. In the case when $w \notin \{0,1\}^n$, the sequential probability ratio test is not valid; we should use the technique by Chow and Robbins (Chow and Robbins, 1965) instead.

Another fundamental problem is to determine statistically whether the two distributions p_1 and p_2 are close to each other. We define $\mathtt{Close}(p_1, p_2, \alpha, \delta)$ to be a (parametric) random variable with possible values `yes` and `no`. If the *total variance* $\|p_1 - p_2\|_{\mathrm{TV}} < \delta$, the variable returns `no` with probability at most α. Similarly, if $\|p_1 - p_2\|_{\mathrm{TV}} \geq \delta$, the variable returns `yes` with probability at most α. If we know p_2 exactly but can only sample from p_1, a naive (probabilistic) algorithm for `Close` would be to check if $p_1(s)$ is close $p_2(s)$ for each state s using algorithm $\mathcal{A}_1$ mentioned above. Another possible algorithm to check closeness of two distributions, which does not need to know p_2 exactly, is the one outlined in Batu et al. (2013). This algorithm is sublinear and provides the following guarantee.

Theorem 1. (Batu et al., 2013) Given parameters ζ and ϵ, and distributions p and p' over a set of n elements, there is a test which runs in time $O\left(n^{2/3}\epsilon^{-8/3}\log\left(\frac{n}{\zeta}\right)\right)$ such that if $\|p - p'\|_{\mathrm{TV}} \leq \max\left(\frac{\epsilon^{4/3}}{64\sqrt[3]{n}}, \frac{\epsilon}{8\sqrt{n}}\right)$ then the test accepts with probability at least $1 - \zeta$, and if $\|p - p'\|_{\mathrm{TV}} > \frac{\epsilon}{2}$ then the test rejects with probability at least $1 - \zeta$.

The naive algorithm is not as efficient as the sublinear algorithm on large Markov chains. On the other hand, the naive algorithm is much more efficient when most elements in p_2 are zero. In the rest of this paper, we will refer to the algorithm to check closeness of distributions, whether it is the naive one or the sublinear one, as `Close` (we only use `Close` when p_2 is known to us).

2.4 Metric Interval Temporal Logic

Definition 3. (Ouaknine and Worrell, 2008) Given a set of atomic propositions `AP`, a *signal* is a function $f \in \mathbb{R}_{\geq 0} \to 2^{\mathtt{AP}}$ mapping t to the set of atomic propositions that are true at time t.

For any $r \in \mathbb{R}_{\geq 0}$ we define $f^r(t)$ to be $f(t + r)$. Also, for any two disjoint sets of atomic propositions $\mathtt{AP}_1$ and $\mathtt{AP}_2$ and two signal functions $f_1 \in \mathbb{R}_{\geq 0} \to 2^{\mathtt{AP}_1}$ and $f_2 \in \mathbb{R}_{\geq 0} \to 2^{\mathtt{AP}_2}$ we define $(f_1 \cup f_2)(t)$ to be $f_1(t) \cup f_2(t)$. This equation implies that if we have a signal for every atomic proposition then we have a signal for the set of all atomic propositions. Note that if we let `AP` to be a set of atomic propositions defined over distributions, then C induces a signal function from $\mathbb{R}_{\geq 0}$ to `AP`.

Definition 4. (Alur et al., 1996) A MITL formula is inductively defined by the following grammar:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \mathcal{U}_{\mathcal{I}} \phi$$

where $p \in \mathtt{AP}$ is an *atomic proposition*, and $\mathcal{I}$ is a *non-singular* interval with rational endpoints. We denote the set of atomic propositions of ϕ by $\mathtt{AP}_\phi$, and may drop the subscript if it is clear from the context.

Definition 5. For a signal f and a MITL formula ϕ the satisfaction relation $f \vDash \phi$ is inductively defined as follows:

$f \vDash p$ iff $p \in f(0)$
$f \vDash \neg\phi$ iff $f \nvDash \phi$
$f \vDash \phi_1 \wedge \phi_2$ iff $f \vDash \phi_1$ and $f \vDash \phi_2$
$f \vDash \phi_1 \mathcal{U}_{\mathcal{I}} \phi_2$ iff $\exists t \in \mathcal{I} \bullet f^t \vDash \phi_2 \wedge \forall t' \in (0,t) \bullet f^{t'} \vDash \phi_1$

We define $[\![\phi]\!]$ to be the set of signals that satisfy ϕ.

2.5 Timed Automata

Definition 6. (Timed Automata). A timed automaton A is a tuple $(\mathtt{Q}, \mathtt{X}, \Sigma, \mathtt{L}, \mathtt{I}, \mathtt{E}, \mathtt{Q}^{\mathtt{init}}, \mathtt{Q}^{\mathtt{final}})$ such that:

- $\mathtt{Q}$ is a finite non-empty set of *locations*,
- $\mathtt{X}$ is a finite non-empty set of *clocks*,
- Σ is a finite non-empty set of *labels*,
- $\mathtt{L} \in \mathtt{Q} \to \Sigma$ maps each location to the *label* of that location,
- $\mathtt{I} \in \mathtt{Q} \to \mathcal{I}^{\mathtt{X}}_{\geq 0}$ maps each location to its *invariant* which is the set of possible values of variables in that location.
- $\mathtt{E} \subseteq \mathtt{Q} \times \mathtt{Q} \times 2^{\mathtt{X}}$ is a finite set of *edges* of the form (s, d, j), where (1) s is the *source*, (2) d is *destination*, and (3) j is the set of clocks that are reset by the edge. Given an edge e, we denote the components by $\mathtt{S}e$, $\mathtt{D}e$, and $\mathtt{J}e$.
- $\mathtt{Q}^{\mathtt{init}} \subseteq \mathtt{Q}$ is a set of *initial locations*.
- $\mathtt{Q}^{\mathtt{final}} \subseteq \mathtt{Q}$ is a set of *final locations*.

A configuration of a timed automaton A at every time instance is completely determined by its control location and valuation of variables at that time. Each clock takes values in $\mathbb{R}_{\geq 0}$. The set of all possible valuations is $\mathbb{R}^{\mathtt{X}}_{\geq 0}$ which we denote it by $\mathtt{V}$. Also for any $t \in \mathbb{R}_{\geq 0}$, $\nu \in \mathtt{V}$, and $j \subseteq \mathtt{X}$, we define (1) $(\nu + t)(x)$ to be $\nu(x) + t$, and (2) $\nu[j := 0](x)$ to be 0 if $x \in j$ and $\nu(x)$ otherwise.

Definition 7. (Alur et al., 1996) A *run* ρ of a timed automaton A is an infinite sequence

$$\underset{\nu_0}{\rightarrow} (q_0, I_0) \xrightarrow[\nu_1]{j_1} (q_1, I_1) \xrightarrow[\nu_2]{j_2} (q_2, I_2) \xrightarrow[\nu_3]{j_3} \cdots$$

where for all $i \in \mathbb{N}$ we have

(1) $q_i \in \mathtt{Q}$,
(2) $I_i \in \mathcal{I}_{\geq 0}$,
(3) $j_{i+1} \subseteq \mathtt{X}$,
(4) $\nu_i \in \mathtt{V}$,
(5) $q_0 \in \mathtt{Q}^{\mathtt{init}}$,
(6) $(q_i, q_{i+1}, j_{i+1}) \in \mathtt{E}$,
(7) $\forall i \in \mathbb{N} \bullet \bar{I}_i \leq \underline{I}_{i+1}$,
(8) $\nu_{i+1} = (\nu_i + \bar{I}_i - \underline{I}_i)[j_{i+1} := 0]$,
(9) $\forall t \in \mathbb{R}_{\geq 0} \bullet \exists i \in \mathbb{N} \bullet t \in I_i \wedge \forall j \in \mathbb{N} \bullet t \in I_j \Rightarrow i = j$, For all $i \in \mathbb{N}$ and $t \in I_i$, the location $\mathtt{Q}_\rho(t)$ is defined to be q_i, and the valuation $\nu_\rho(t)$ is defined to be $\nu_i + t - \underline{I}_i$.
(10) $\forall t \in \mathbb{R}_{\geq 0} \bullet \nu_\rho(t) \in \mathtt{I}(\mathtt{Q}_\rho(t))$.

Based on Definition 7, initial clock valuations could be anything that satisfies the invariant of the initial location of that run.

For any run ρ, we define $\mathtt{inf}(\rho)$ to be the set of locations q such that the set of states that appear infinitely often in ρ. Furthermore, $\mathtt{Lang}(A)$ is defined to be the set of signals induced by A. Formally, a signal $f \in \mathbb{R}_{\geq 0} \to \Sigma$ is in $\mathtt{Lang}(A)$ iff there exists a run ρ such that $\mathtt{inf}(\rho) \cap \mathtt{Q}^{\mathtt{final}} \neq \emptyset$ and for all $t \in \mathbb{R}_{\geq 0}$ we have $f(t) = \mathtt{L}(\mathtt{Q}_\rho(t))$. We refer to the problem of checking $\mathtt{Lang}(A) = \emptyset$ the *emptiness problem*.

Theorem 2. (Alur et al., 1996) Timed automata are closed under intersection. Furthermore, for every timed automata A_1 and A_2, a timed automaton A can be effectively constructed such that $\mathtt{Lang}(A) = \mathtt{Lang}(A_1) \cap \mathtt{Lang}(A_2)$.

Theorem 3. (Alur and Dill, 1994) The emptiness problem is decidable for timed automata.

Alur et al. (1996) showed how to convert a MITL formula ϕ into a timed automaton T_ϕ such that $[\![\phi]\!] = \mathtt{Lang}(T_\phi)$. Therefore, ϕ is satisfiable iff $\mathtt{Lang}(T_\phi) \neq \emptyset$. In Section 4 we use statistical model checking to construct a timed automaton $T_{C,\mathtt{AP}_\phi}$ for a CTMC C and a MITL formula ϕ that with high probability over-approximates the signal function induced by C and $\mathtt{AP}_\phi$. Intersection of $\mathtt{Lang}(T_\phi)$ and $\mathtt{Lang}(T_{C,\mathtt{AP}_\phi})$ is empty iff no signal induced by $T_{C,\mathtt{AP}_\phi}$ satisfies ϕ. Similarly intersection of $\mathtt{Lang}(T_{\neg\phi})$ and $\mathtt{Lang}(T_{C,\mathtt{AP}_\phi})$ is empty iff no signal induced by $T_{C,\mathtt{AP}_\phi}$ violates ϕ. In the first case, we know that with high probability $C \nvDash \phi$, and in the second case we know that with high probability $C \vDash \phi$. If none of those intersections are empty, the result is not known and one need to construct $T_{C,\mathtt{AP}_\phi}$ with smaller error bounds. By Theorem 2 we know how to compute a timed automaton that models these intersections, and by Theorem 3 we know that the emptiness problem of the result automata is decidable.

2.6 Problem Formulation

Consider a continuous-time dynamical system

$$\dot{x}(t) = F(x(t)) \tag{2}$$

where $x = [x_1, \ldots, x_m]^{\mathrm{T}} \in \Omega$. The state space $\Omega \subseteq \mathbb{R}^m$ is assumed to be compact. Given an initial distribution $f(0, x)$ on Ω, the original system corresponds to a continuous-time Markov process on the state space Ω, in which the Fokker-Planck operator is given by

$$L = -\sum_{i=1}^{m} F_i(x) \frac{\partial}{\partial x_i}, \tag{3}$$

In this work, we assume $f(t, x)$ is square-integrable on ω.

On the Markov process, we specify the atomic definitions of MITL by the integration with respect to a weight function.

Definition 8. The atomic proposition $\mathtt{AP}$ takes the form

$$\mathtt{AP} : \int_\Omega w(x) f(t, x) \mathrm{d}x > c, \tag{4}$$

and given an execution $E : [0, \infty) \to \mathtt{AP}$,

$$E \vDash \mathtt{AP} \iff \int_\Omega w(x) f(0, x) \mathrm{d}x > c, \tag{5}$$

where $c \in [0, 1]$ is a constant and $w(x)$ is a *weight function*.

3. MODEL REDUCTION

Let μ be the Borel measure on the state space Ω. For any (Borel) measurable set $s \subseteq \Omega$, let $\mu(s)$ be the (Borel) measure of s.

Definition 9. $S = \{s_1, s_2, \ldots, s_n\}$ is called a (nontrivial) measurable partition of a state space X if

(1) each s_i is measurable and $\mu(s_i) > 0$,
(2) $\bigcup_{i=1}^{n} s_i = X$,
(3) $s_i \cap s_j = \phi$ for any $i \neq j$.

Given the partition, we can use the Mori-Zwanzig method, which is mathematically a kind of Galerkin projection method, to reduce the original dynamical system to a finite-state continuous-time Markov chain.

3.1 Mori-Zwanzig Projection

Based on the partition, we define a projection Π by

$$p(t,x) = \Pi f(t,x) = \sum_{i=1}^{n} p_i(t)\mathbf{U}_{s_i}(x), \tag{6}$$

where

$$p_i(t) = \int_{s_i} f(t,x)\mathrm{d}x, \tag{7}$$

$$\mathbf{U}_{s_i}(x) = \begin{cases} 1, & \text{if } x \in s_i \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

It maps a probability distribution function on the continuous state space Ω to a probability distribution function on the finite set S.

Lemma 4. Let $S = \{s_1, \ldots, s_n\}$ be a measurable partition of X. Then the corresponding projection operator Π is contractive, namely

$$\int |p_1(t,x) - p_2(t,x)|\mathrm{d}x \leq \int |f_1(t,x) - f_2(t,x)|\mathrm{d}x. \tag{9}$$

where $p_i(t,x) = \Pi f_i(t,x)$ for $i = 1, 2$.

As shown in Figure 1, the projection Π induces a projection on the evolution operator of the Markov chain on Ω. Specifically, if we take the first representation, then $\Phi(t,x;s,y)$ reduces to $\phi(t,x;s,y)$ by

$$\phi(t,x;s,y) = \sum_{i=1}^{n}\sum_{j=1}^{n} \phi_{ij}(t,s)\mathbf{U}_{s_i}(x)\mathbf{U}_{s_j}(y) \tag{10}$$

where

$$\phi_{ij}(t,s) = \int_{s_i \times s_j} \frac{\Phi(t,x;s,y)}{\mu(s_j)}\mathrm{d}x\mathrm{d}y. \tag{11}$$

If we take the second representation, then L reduces to A by

$$A\left(\sum_{j=1}^{n} p_j \mathbf{U}_{s_j}\right) = \left(\frac{\int_{s_i} L(\mathbf{U}_{s_j}(x))\mathrm{d}x}{\mu(s_j)}\right)\mathbf{U}_{s_i} \tag{12}$$

The projection Π also reduces the atomic formulas. If the weigh function

$$w(x) = \sum_{i=1}^{n} w_i \mathbf{U}_{s_i}(x) \tag{13}$$

is a simple function supported on S, then

$$\int_X w(x)f(t,x)\mathrm{d}x > c \Longleftrightarrow \sum_{i=1}^{n} w_i p_i(t) > c. \tag{14}$$

$$\begin{array}{ccc} f(s,y) & \xrightarrow{\quad\Phi\quad} & f(t,x) \\ \Pi \big\downarrow & & \big\downarrow \Pi \\ p(s,y) & \xrightarrow{\quad\phi\quad} & p(t,x) \end{array}$$

Fig. 1. Diagram for single-step projection

where $\mu(s_i)$ is the Borel measure of s_i. Therefore, we can reduce MITL formulas by replacing the integrations with the summations.

3.2 Error Estimation

The error of the projection operator Π is given by

$$\Delta(\Pi) = \max_{i\in[n]} \int_{\Omega} \frac{|L(\mathbf{U}_{s_i}) - A(\mathbf{U}_{s_i})|}{\mu(s_i)}\mathrm{d}x. \tag{15}$$

For the whole reduction procedure, the error function $\delta(t,x)$ gives the difference at time t between evolving by the original system and evolving by the reduced system by

$$\delta(t,x) = (\Pi e^{tL} - e^{tA}\Pi)f(0,x), \tag{16}$$

where $f(0,x)$ is the initial distribution.

Lemma 5. When the initial distribution $f(0,x)$ is supported on the partition S, namely $f(0,x) = \Pi f(0,x)$, the error function $\delta(t,x)$ satisfies

$$\delta(t,x) = \Pi \int_{[0,t]} e^{(t-s)L}(L - A)p(s,x)\mathrm{d}t \tag{17}$$

Definition 10. The operator L is γ-contractive ($\gamma > 0$) if

$$\int_{\Omega} |e^{tL}f_1(x) - e^{tL}f_2(x)|\mathrm{d}x \leq \int_{\Omega} e^{-\gamma t}|f_1(x) - f_2(x)|\mathrm{d}x, \tag{18}$$

for any two probability distribution function $f_1(x), f_2(x) \in L_2$.

Theorem 6. If the operator L is γ-contractive, then the error function $\delta(t,x)$ satisfies $\int_{\Omega} |\delta(t,x)|\mathrm{d}x \leq \Delta(\Pi)/\gamma$.

Theorem 7. Given a measurable partition $S = \{s_1, \ldots, s_n\}$ and the corresponding projection operator P, a Markov process (T, μ_0) and its reduction (τ, p_0) satisfies the following equations:

$$\sum_{i=1}^{n} w_i p_i > c \Longrightarrow \int_{\Omega} w(x)f(t,x)\mathrm{d}x > c - m,$$

$$\sum_{i=1}^{n} w_i p_i < c \Longrightarrow \int_{\Omega} w(x)f(t,x)\mathrm{d}x < c + m.$$

where the margin $m = \Delta(\Pi)\max_{i,j\in[n]} |w_i - w_j|/2\gamma$.

4. STATISTICAL MODEL CHECKING OF MITL

In this section we show that for a CTMC C and a MITL formula ϕ with atomic propositions $\{P_i\}$, how to statistically construct a timed automaton $T_{C,\mathtt{AP}_\phi}$ such that reachable locations of this automaton at time t are labeled by the subset of atomic propositions in ϕ that are true in C at time t. By $[\![C, \mathtt{AP}_\phi]\!]$ we denote the singleton set containing the unique signal induced by C and ϕ. Our algorithm guarantees that with high probability $[\![C, \mathtt{AP}_\phi]\!] \subseteq$

$[\![T_{C,AP_\phi}]\!]$. Therefore, if $[\![T_{C,AP_\phi}]\!] \cap [\![\phi]\!] = \emptyset$, we know that with high probability $C \not\models \phi$ and our algorithm returns "no". Similarly, if $[\![T_{C,AP_\phi}]\!] \cap [\![\neg\phi]\!] = \emptyset$, we know that with high probability $C \not\models \phi$ and our algorithm returns "yes". Otherwise, the algorithm returns "unknown" and one need to construct $T_{C,\mathtt{AP}_\phi}$ with smaller error bounds. Due to space limitation we assume $\mathtt{AP}_\phi$ is singleton. Generalization to the case where this assumption does not hold is straightforward from below. Therefore, in the rest of this section we focus only on constructing $T_{C,\{P\}}$ for an atomic formula $P := w \cdot \mu > b$ and bounding its error.

Note that C induces exactly one signal for P, but since we are doing statistical model checking, we cannot always determine the signal exactly. Thus in our construction $[\![T_{C,\{P\}}]\!]$ might have more than just one signal, and this is the error that we need to bound. Let $p(t) := e^{At}p_0$ be the probability distribution induced by C at time t. Let $f(t) := \mathtt{if}\ w \cdot p(t) > b\ \mathtt{then}\ \{P\}\ \mathtt{else}\ \emptyset$ be the set of atomic formulas that $p(t)$ satisfies. And finally, let $T_{C,\{P\}}(t)$ be the set of reachable locations of $T_{C,\{P\}}$ at time t.

There are two main problems against the construction of $T_{C,\{P\}}$. First, reachability domain is unbounded. Even if t only takes integer values, there are still infinitely many time points. Hence, one need to make sure that the signal automaton won't have infinitely many locations. Second, even if the time is bounded, because t is in $\mathbb{R}_{\geq 0}$, finding $f(t)$ and $f(t+\epsilon)$ (for some $\epsilon \in \mathbb{R}_+$) does not necessarily give us any information about $f(t')$ for any $t < t' < t + \epsilon$.

4.1 Bounding the Time

Regarding the unbounded domain problem, we assume that the mapping defined by the reduced model is contracting and hence the function $p(t)$ converges to a *known* invariant distribution p^{inv}. Now if we assume that there is a δ_2 such that $|w \cdot p^{\text{inv}} - b| > \delta_2$, then eventually, for large enough t, $p(t)$ and p^{inv} satisfy exactly the same set of atomic propositions and so $f(t)$ will ultimately become constant. We assume that p^{inv} in known. This assumption is readily verified for large classes of important physical models, such as those with energy balance laws that are dissipative in aggregate. For such classes of model general arguments can be used to derive the prior condition, even in the presence of strong non-linearities, discontinuous dynamics, or other complexities. Furthermore, since p^{inv} is known we can easily check the existence of such δ_2 for a MITL formula ϕ and even compute it in case it does exist.

Figure 2 contains the pseudo code for finding this time bound (we call it T). Note that, if $\|p(t) - p^{\text{inv}}\|_{\text{TV}} < \frac{\delta_2}{\|w\|_{\text{TV}}}$, then we know for all $t' > t$, since $\|p(t') - p^{\text{inv}}\|_{\text{TV}} \leq \|p(t) - p^{\text{inv}}\|_{\text{TV}}$, $p(t')$ has the same label as p^{inv}. Thus, we need a statistical test that checks if $\|p(t) - p^{\text{inv}}\|_{\text{TV}} < \frac{\delta_2}{\|w\|_{\text{TV}}}$ given that we can draw samples from p_t and know p^{inv} exactly. This is exactly what algorithm Close is used for (see Section 2.2).

1: $t \leftarrow 1$
2: **while** $\mathtt{Close}\left(p(t), p^{\text{inv}}, \frac{3\alpha}{4}, \frac{\delta_2}{\|w\|_{\text{TV}}}\right) \neq \mathtt{yes}$ **do**
3: $\quad t \leftarrow t + 1$
4: **return** t

Fig. 2. Part 1 - Finding A Time Bound for Getting Close Enough to Invariant Distribution

4.2 Constructing the Signal

In this section we first show how to construct $T_{C,\{P\}}$. After the construction, we can check the emptiness of $[\![T_{C,\{P\}}]\!]$ intersected with $[\![\phi]\!]$ and $[\![\neg\phi]\!]$ and return yes, no, or unknown, as defined in the beginning of Section 4. In case, unknown is returned as answer, one can repeat the construction using a smaller indifference parameter. At the end of this section we show that under what circumstances the process of repeating the construction with smaller parameters is guaranteed to remove unknown from the set of possible outputs without changing what is guaranteed by our algorithm.

To construct the signal automaton $T_{C,\{P\}}$ one might want to approximate $p(t) = e^{At}p_0$ and then just verify $w \cdot p(t) > b$. However, this would be very expensive for continuous time Markov chains with large number of states. Instead we compute the labels statistically. Notice that we still need to handle the second problem which is no matter how small the time bound T is, there will always be uncountably infinitely many number of time points from 0 to T.

The general idea is to partition $[0, T)$ into a finite set of intervals with *small enough* widths and then determine all the labels in each interval by only finding labels at a single time of that interval. An interval Δ is small enough iff changes of $w \cdot p(t)$ during that interval is bounded by δ_1, which is an input parameter to our algorithm, and like the other parameters, smaller values of δ_1 makes the algorithm more precise but requires more number of samples. Formally, we have $|w \cdot \dot{p}(t)| = |wAe^{At}p_0| \leq \|w\|_{\text{TV}} \times \|A\|_{\text{TV}} \times e^{T\|A\|_{\text{TV}}} \times \|p_0\|_{\text{TV}}$. Denoting the right hand side by h, let Δ be any number such that $h\Delta < \frac{\delta_1}{3}$. For any time $t \in [0, T]$ and $t' \in [t - \Delta, t + \Delta] \cap [0, T]$ we have the followings:

1. if $w \cdot p(t) - b > \frac{\delta_1}{3}$ then $w \cdot p(t') - b > 0$
2. if $w \cdot p(t) - b < -\frac{\delta_1}{3}$ then $w \cdot p(t') - b < 0$
3. if $|w \cdot p(t) - b| \leq \frac{2\delta_1}{3}$ then $|w \cdot p(t') - b| \leq \delta_1$

We use $\mathcal{A}_1$, introduced in Section 2.2, to decide about label of $T_{C,\{P\}}(t)$. More, precisely, let

$$res_1 = \mathcal{A}_1^{\delta_1/3}(p(t), w, b + \frac{\delta_1}{3}, \alpha', \gamma')$$
$$res_2 = \mathcal{A}_1^{\delta_1/3}(p(t), w, b - \frac{\delta_1}{3}, \alpha', \gamma')$$

If $res_1 = \mathtt{yes}$ then for any time t' in the interval $w \cdot p(t') > b$ with bounded error α' and we set $T_{C,\{P\}}(t) = \{P\}$. Otherwise, if $res_2 = \mathtt{no}$ then for any time t' in the interval $w \cdot p(t') < b$ with bounded error α' and we set $T_{C,\{P\}}(t) = \{\emptyset\}$. Otherwise, for any time t' in the interval $|w \cdot p(t) - b| \leq \frac{2\delta_1}{3}$ with bounded error $\max(2\alpha', \gamma')$.

$h \leftarrow \|w\|_{\text{TV}} \times \|A\|_{\text{TV}} \times e^{T\|A\|_{\text{TV}}} \times \|p_0\|_{\text{TV}}$
$\Delta \leftarrow \frac{\delta_1}{3h}$
$n \leftarrow |\texttt{AP}| \left\lceil \frac{T}{\Delta} \right\rceil$
$T_{C,\{P\}} \leftarrow$ an empty automaton
$\texttt{X} \leftarrow \{t\}$
$q_{\mathsf{last}} \leftarrow \perp$
for $i \leftarrow 0$ to $\left\lfloor \frac{T}{2\Delta} \right\rfloor$ **do**
  $\alpha' \leftarrow \min(\frac{\alpha}{4n}, \frac{\gamma}{2n})$
  $\gamma' \leftarrow \frac{\gamma}{n}$
  $res_1 \leftarrow \mathcal{A}_1^{\delta_1/3}\Big(p((2i+1)\Delta), w, b + \frac{\delta_1}{3}, \alpha', \gamma'\Big)$
  $res_2 \leftarrow \mathcal{A}_1^{\delta_1/3}\Big(p((2i+1)\Delta), w, b - \frac{\delta_1}{3}, \alpha', \gamma'\Big)$
  add a new location q to $\texttt{Q}$
  if $res_1 = \texttt{yes}$ **then** $\texttt{L}(q) \leftarrow \{P\}$
  else if $res_2 = \texttt{no}$ **then** $\texttt{L}(q) \leftarrow \emptyset$
  else $\texttt{L}(q) \leftarrow \texttt{unknown}$
  $\texttt{I}(q) \leftarrow 2i\Delta \le t < 2(i+1)\Delta$
  if $q_{\mathsf{last}} \neq \perp$ **then** $\texttt{E} \leftarrow \texttt{E} \cup \{(q_{last}, q, \emptyset)\}$
  else $\texttt{Q}^{\mathsf{init}} \leftarrow \{q\}$
  $q_{\mathsf{last}} = q$
add a new location q to $\texttt{Q}$
$\texttt{I}(q) \leftarrow true$
$\texttt{Q}^{\texttt{final}} \leftarrow \{q\}$
$\texttt{E} \leftarrow \texttt{E} \cup \{(q_{last}, q, \emptyset), (q, q, \emptyset)\}$
if $w \cdot p^{\text{inv}} > b$ **then** $\texttt{L}(q) \leftarrow \{P\}$
else $\texttt{L}(q) \leftarrow \emptyset$
$T_{C,\{P\}} \leftarrow$ replace any location in $\texttt{Q}$ labeled `unknown` with two locations q labeled $\{P\}$ and q' labeled $\emptyset$. Also, duplicate edges from/to q and q' accordingly.
Add $(q, q', \emptyset)$ and $(q', q, \emptyset)$ to $\texttt{E}$ for every split locations in the previous step.
return $T_{C,\{P\}}$

Fig. 3. Part 2 - Constructing the Signal for Atomic Proposition P

In the last case, we (1) set $T_{C,\{P\}}(t) = \{q, q'\}$, (2) set $\texttt{L}(q) = \{P\}$ and $\texttt{L}(q') = \emptyset$, (3) let automaton non-deterministically enters q or q', and (4) let automaton non-deterministically switches between $\{P\}$ and $\emptyset$ for arbitrary number of times, while their common invariant permit. Figure 3 shows the pseudo code for constructing a signal automaton.

$T_{C,\{P\}}$ has only one clock variable that is never reset. We initially find interval width Δ using the argument we talked about right before presenting this pseudo code. n is the number of partitions multiplied by the number of atomic propositions. We construct $|\texttt{AP}|$ signal automata, therefore we make sure that the probability of making error in each $T_{C,\{P\}}$ is bounded by $\frac{\alpha}{4|\texttt{AP}|}$ and $\frac{\gamma}{|\texttt{AP}|}$. As a result, the total error will add up to $\frac{\alpha}{4}$ and γ, and when we add this error with the error bound in Figure 2 the total error will be bounded by α and γ. For each interval we use $\mathcal{A}_1$ to find the label at the center of that interval. And use the result of $\mathcal{A}_1$ to label the corresponding location. After labels of all the sub-intervals are determined, we add one *final* location q to the automaton and label it by $\{P\}$ if the invariant distribution satisfies P, and label it by $\emptyset$ otherwise. q is the only final location and have a trivial self loop to enable infinite runs with infinite locations from the final set of the automaton. If a location is labeled by `unknown`, it is very likely that $|w \cdot p(t) - b| < \delta_1$ for all t in the corresponding interval. In this case we replace that location with two locations q and q', and label them with $\{P\}$ and $\emptyset$. Old edges are duplicated to go to/from both new locations and there will be two edges from q to q' and vice a versa. Adding the edges between q and q' means that in this particular interval we don't know how $f(t)$ behaves. Therefore, we non-deterministically go to one of q or q' and before leaving them we can have arbitrary number of switches between q and q'. This is the reason why $[\![T_{C,\{P\}}]\!]$ may not be a singleton, but it always (with high probability) is a super set of $[\![C, \phi]\!]$.

It is obvious that the algorithm always terminates. It provides the following guarantees about its output $res = \mathcal{A}^{\delta_1,\delta_2}(C, p_0, \phi, \alpha, \gamma)$:

$$\mathbb{P}[res = \texttt{no} \mid C \vDash \phi] \le \alpha \tag{19a}$$
$$\mathbb{P}[res = \texttt{yes} \mid C \nvDash \phi] \le \alpha \tag{19b}$$

It states that the probability of having false positive or negative is at most α. It is true, since when the algorithm returns `no` (`yes`) while the correct answer is `yes` (`no`), it means that the algorithm made at least one mistake. The probability of finding wrong T is at most $\frac{3\alpha}{4}$. Assuming T is computed correctly, the probability of determining a signal incorrectly for an atomic proposition is bounded by $\frac{\alpha}{4|AP|}$. Because the answer is not `unknown`, it means mistakes in returning `unknown` had not effect in the output. So the error bound for the final result is bounded by α. Obviously, we need to have some guarantee regarding the output `unknown`. Otherwise, the algorithm can always return `unknown` while satisfying all of its promises. Before that we need a few more definitions.

Definition 11. A time $t \in \mathbb{R}_{\ge 0}$ is *ϵ-critical* for CTMC C and atomic formula P iff $|w \cdot p(t) - b| < \epsilon$. We denote the set of ϵ-critical times of CTMC C and an atomic formula P by $\texttt{crtl}^{\epsilon}(C, P)$. Moreover, for any $T \in \mathbb{R}_+$ we define $\texttt{crtl}^{T,\epsilon}(C, P)$ to be the set $\{t \in \texttt{crtl}^{\epsilon}(C, P) \mid t < T\}$.

Definition 12. For a CTMC C, atomic formula P, time bound $T \in \mathbb{R}_+$, and $\epsilon \in \mathbb{R}_+$, we define $G^{C,T,\epsilon}(P)$ to be the set of functions that are obtained by slightly perturbing $w \cdot p(t)$ at critical times of C and P. Formally, function g is in $G^{C,T,\epsilon}(P)$ iff it satisfies the following two conditions for every time $t \in [0, T)$:

(1) $t \in \texttt{crtl}^{T,\epsilon}(C, P) \Rightarrow |g(t) - b| < \epsilon$,
(2) $t \notin \texttt{crtl}^{T,\epsilon}(C, P) \Rightarrow g(t) = w \cdot p(t)$

Intuitively, a ϵ-perturbed function is ϵ-close to b when $w \cdot p(t)$ is ϵ-close to b. Otherwise, it is exactly $w \cdot p(t)$. Every element of $G^{C,T,\epsilon}(P)$ induces a perturbed signal. Let $F^{C,T,\epsilon}(P)$ be the set of such signals. If for every $P \in \texttt{AP}_\phi$ we pick a signal from $F^{C,T,\epsilon}(P)$, the result set induces a perturbed signal of ϕ on C. We denote the set of all such signals by $[\![C, \phi]\!]^{\epsilon}$. We call C *ϵ-robust* on ϕ iff either all of those signals satisfy ϕ or none of them satisfy ϕ. Formally, C is ϵ-robust on ϕ iff for all $f_1, f_2 \in [\![C, \phi]\!]^{\epsilon}$ we have $f_1 \vDash \phi$ iff $f_2 \vDash \phi$. We say C is *robust* on ϕ iff C is ϵ-robust on

ϕ for some $\epsilon \in \mathbb{R}_+$. Regarding the `unknown` output, our algorithm guarantees the following condition:

$$\mathbb{P}[res = \texttt{unknown} \mid C \text{ is } \delta_1\text{-robust on } \phi] \leq \alpha + \gamma \quad (20)$$

This is true because, the probability of computing at least one incorrect label is bounded by $\alpha + \gamma$. If the algorithm does not make a mistake in its labeling phase and still returns `unknown`, we know that C is not δ_1-robust on ϕ.

$\mathcal{A}$ takes δ_1 as one of its parameters. The problem with this parameter is that one may not know in advance a correct value δ_1. Large values causes the algorithm to return `unknown`, and small values make the algorithm runs slow. In order to solve this problem one can start with a large value for δ_1 and decrease it when the algorithm returns `unknown` for that δ_1.

Theorem 8. For any CTMC C and MITL formula ϕ, if C is robust on ϕ, iteratively reducing δ_1 in our algorithm guarantees that it will eventually return an answer which is not `unknown` while satisfying conditions 19a and 19b.

5. CONCLUSION

In this work, we studied the statistical verification of MITL formulas on dynamical systems. Viewing the dynamical system as a continuous-time Markov process, we employed the Mori-Zwanzig method to reduce the original system to a CTMC. Accordingly, the original MITL formulas reduce to MITL formulas on the CTMC. In addition, we proposed a statistical verification algorithm for checking MITL formulas on the CTMC. We showed that verifying the MITL formulas on the reduced system guarantees the correctness of the corresponding MITL formulas on the original system.

REFERENCES

Alur, R. and Dill, D.L. (1994). A theory of timed automata. *Theor. Comput. Sci.*, 126(2), 183–235.

Alur, R., Feder, T., and Henzinger, T.A. (1996). The benefits of relaxing punctuality. *J. ACM*, 43(1), 116–146.

Batu, T., Fortnow, L., Rubinfeld, R., Smith, W.D., and White, P. (2013). Testing closeness of discrete distributions. *J. ACM*, 60(1), 4:1–4:25.

Beck, C., Lall, S., Liang, T., and West, M. (2009). Model reduction, optimal prediction, and the mori-zwanzig representation of markov chains. In *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009*, 3282–3287.

Chorin, A.J., Hald, O.H., and Kupferman, R. (2000). Optimal prediction and the mori-zwanzig representation of irreversible processes. *Proceedings of the National Academy of Sciences*, 97(7), 2968–2973.

Chow, Y.S. and Robbins, H. (1965). On the asymptotic theory of fixed-width sequential confidence intervals for the mean. *The Annals of Mathematical Statistics*, 36(2), 457–462.

Clarke, E.M., Emerson, E.A., and Sistla, A.P. (1986). Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, 8(2), 244–263.

Gillespie, D. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4), 403–434.

Kloetzer, M. and Belta, C. (2008). A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1), 287–297.

Liu, B., Hagiescu, A., Palaniappan, S.K., Chattopadhyay, B., Cui, Z., Wong, W.F., and Thiagarajan, P.S. (2012). Approximate probabilistic analysis of biopathway dynamics. *Bioinformatics*, 28(11), 1508–1516.

Liu, B., Hsu, D., and Thiagarajan, P.S. (2011). Probabilistic approximations of ODEs based bio-pathway dynamics. *Theoretical Computer Science*, 412(21), 2188–2206.

Manna, Z. and Pnueli, A. (1992). *The Temporal Logic of Reactive and Concurrent Systems*. Springer-Verlag New York, Inc., New York, NY, USA.

Ouaknine, J. and Worrell, J. (2008). Some recent results in metric temporal logic. In *Proceedings of the 6th International Conference on Formal Modeling and Analysis of Timed Systems*, FORMATS '08, 1–13. Springer-Verlag, Berlin, Heidelberg.

Sen, K., Viswanathan, M., and Agha, G. (2005). On statistical model checking of stochastic systems. In K. Etessami and S.K. Rajamani (eds.), *Computer Aided Verification*, number 3576 in Lecture Notes in Computer Science, 266–280. Springer Berlin Heidelberg.

Soudjani, S.E.Z. and Abate, A. (2013). Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. *SIAM Journal on Applied Dynamical Systems*, 12(2), 921–956.

Tabuada, P. and Pappas, G. (2006). Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 51(12), 1862–1877.

Wald, A. (1945). Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2), pp. 117–186.

Wang, Y., Roohi, N., West, M., Viswanathan, M., and Dullerud, G. (2015). Statistical verification of nonlinear systems using set oriented methods. HSCC '15. Seattle, WA, USA.

Wongpiromsarn, T., Topcu, U., and Murray, R.M. (2010). Receding horizon control for temporal logic specifications. In *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '10, 101–110. ACM, New York, NY, USA.

Younes, H.L.S. (2006). Error control for probabilistic model checking. In *Verification, Model Checking, and Abstract Interpretation, 7th International Conference, VMCAI 2006, Charleston, SC, USA, January 8-10, 2006, Proceedings*, 142–156.

Younes, H.L.S. and Simmons, R.G. (2006). Statistical probabilistic model checking with a focus on time-bounded properties. *Information and Computation*, 204(9), 1368–1409.

Zuliani, P. (2014). Statistical model checking for biological applications. *International Journal on Software Tools for Technology Transfer*, 1–10.

Zuliani, P., Platzer, A., and Clarke, E.M. (2013). Bayesian statistical model checking with application to stateflow/simulink verification. *Formal Methods in System Design*, 43(2), 338–367.