Generating Safety Guidance for Medical Injection with Three-Compartment Pharmacokinetics Model

Cunxi Yu, Heinz Riener, Francesca Stradolini, Giovanni De Micheli
Integrated Systems Laboratory (LSI)
Ecole polytechnique federale de Lausanne (EPFL)
Lausanne, Switzerland
cunxi.yu@epfl.ch

***Abstract*—Medical cyber-physical systems are a new trend of software controlled physical systems that are increasingly common in medical domains. With rapid developments in medical science and computer technology, safety verification and simulation becomes more challenging. This paper introduces a general model for medical injection systems, which can be used for formal verification, simulation/testing, and computing the *Area Under the Curve* (AUC) metrics, using *Satisfiability Modulo Theories* (SMT) over *Reals*. An algorithm of computing *constrained AUC* for measuring *drug exposure* with relative baseline, is presented based on the *proof of unsatisfiability*. We demonstrate that our model can efficiently solve these problems using the state-of-the-art SMT solver dReal.**

***Index Terms*—Satisfiability Modulo Theories, Medical Cyber-System, Timed System.**

I. Introduction

Medical cyber-physical systems are a new trend of software controlled physical systems that are increasingly common in medical domains. These systems are becoming more and more popular in medical therapy. Medical service is more efficient and convenient for doctors and patients, while they offer the opportunities for medical experts and doctors in studying the treatments and in communicating with patients. Therefore, patients can benefit from the automation of treatment process, which improve therapy effectiveness, lifestyle quality and reducing cost. However, with rapid developments in medical science and computer technology, medical cyber-physical systems are becoming more and more precise and complex, and more real-time reactions of the patient during treatment are sampled and analyzed. Due to the complexity of the systems and a low tolerance for faults in the medical environment, validation and verification of medical cyber-physical systems are crucial [1]. Specifically, the challenges of verifying medical systems mainly come from *timed* and *hybrid* properties. Similarly to all cyber-physical systems, medical systems are mostly about the intersections of computations and physical actions. To create the mathematical models of the entire systems, formal methods that combine both discrete and continuous dynamics are required.

Formal methods have been widely used in checking the properties and reliabilities of hybrid systems, which are modeled using abstract mathematical representations. The main advantage of formal methods comes from the mathematical precision for reasoning the correctness of system models. *Timed Automata* plays a big role in modeling and verifying the timed systems [2]. Particularly for medial systems such as drug administration system, such methods model the behavior of the system using formal representation by employing *Timed Automata extended with Tasks* (TAT). Tools such as UPPAAL [3] and its extension TIMES [4] have been successful in verifying cyber-physical systems in many domains. Model checking based on *satisfiability* theories have also been applied to timed systems [5]. For example, *dReal* [6] is demonstrated to successfully verify biology systems by solving *Satisfiability Modulo Theories* (SMT) problems over the reals with a wide range of nonlinear functions, such as ordinary differential equations (ODEs) [7]. The main advantages of using SMT over reals comparing to real-time model checkers are: **a)** for the systems with frequent changes between different dynamics, SMT performs much faster; **b)** if an unsafe state exists in the system, SMT offers the *proof of unsatisfiability* that provides the information of where and why the unsafe state appears.

To precisely model medical systems, a realistic drug response model has to be used. Various clinical studies show that responsiveness to the treatment with drugs depends on the concentration of the drug in the blood that depends on patients, drug dose, and intake time interval. Pharmacokinetics (PK) [8] is a branch of pharmacology focused on studying the drug disposition in the human body. For many drugs, the concentration in the blood of a patient is highly related to its effectives. Pharmacodynamics (PD) [9] is the study of the biochemical and physiological effects of drugs on the body. Therapeutic Drug Monitoring (TDM) [10] is the approach that unifies the PK-PD knowledge, which shows that drugs with explicit PK-PD relationships and a narrow therapeutic range may be easily under- or overdose. Hence, it is important to develop an approach that generates safety guidance for medical injection using a precise drug response model, such as the drug administration system [11]. In addition, *Area Under the Curve* (AUC), as well as AUC in the baseline measurements (constrained AUC), are commonly used to assess the extent of exposure of a drug [12]. Measuring these two metrics is very important in pharmacokinetics analysis.

The main contributions of this paper are as follows:

- We introduce a general model for medical injection system, which can be applied to both simulation and formal verification, using SMT over Reals. The mathematical three-compartment pharmacokinetics model is used for drug response in the abstract timed model, which is one of the most precise pharmacokinetics models for simulating drug response.

- The model is demonstrated that it can efficiently and precisely simulate the medical injection process. The model can formally prove(disprove) if the expected drug concentration-time objectives are *reachable* with given injection actions, and return *sat* (*unsat*). This is done by checking bounded *δ-Satisfiability*[13]. The proof of *unsatisfiability* is generated if it returns *unsat*, which indicates the unreachable state(s) and the corresponding time location(s).
- The proposed model computes *AUC* and *constrained AUC* simultaneously during the verification or simulation process. For computing constrained AUC, we introduce an algorithm based on *proof of unsatisfiability* of SMT over reals.

II. Problem Formulation

A timed system is defined with the finite set of continuous clocks $\mathbb{T}$ and a set of constraints over the clocks. Mostly, the constraints are represented as conjunctions, disjunctions, and negations of expressions over the clocks. Each transition in such system is labeled by a constraint over the state or clock values, namely *guard*, which indicates the condition to trigger the transition. Each state is constrained by an *invariant*, which restricts the possible values of the clocks for being in the state, which can then enforce a transition to be taken. The following notations are used for problem formulation.

Let a timed system be a tuple $\mathcal{A}$=$\langle\mathbb{S}$, $\mathbb{T}$, Inv, $\mathbb{E}$, $\mathcal{ACT}$, $init\rangle$.

- *$\mathbb{T}$ is a finite set of clocks. $t_i \in \mathbb{T}$, and $t_i \in R^+$, $i \in$ [0,n].*
- *$\mathbb{S}$ is a finite set of states. $s_i \in \mathbb{S}$ is the state at i^{th} time.*
- *Inv is the associated invariant for each state.*
- *$\mathbb{E}$ is a finite set of transitions, where e_i is a tuple $\langle s_i$, s_j, g, act, $\mathbb{T}_{i\rightarrow j}$ $\rangle$, $e_i \in \mathbb{E}$. The state changes from s_i to s_j over a set of clocks $\mathbb{T}_{i\rightarrow j}$. g is the guard of transition e_i, and act is the action of e_i.*
- *$\mathcal{ACT}$ a finite set of actions the system made.*
- *init is the initial values of all the parameters for encoding the system.*

In this work, the state $\mathbb{S}$ are the concentrations of different compartments. The clock set $\mathbb{T}$ = [0, t_n]. The action set $\mathcal{ACT}$ are the inputs that triggers the transitions. Simulation and AUC calculation can be achieved with the same formulation of formal verification.

Problem 1: The medical therapy objectives O in concentration-time format and the injection actions $\mathcal{ACT}$, are provided by the doctor or electronic drug system. Let the upper bound clock be t_n and the initial states be s_0, and O={(c_{i_0}, t_{i_0}), (c_{i_1}, t_{i_1}), ..., (c_{i_j}, t_{i_j})}. Each element of O is a pair of concentration[1] and time, $\forall$ t_{i_j} $\le$ t_n. The verification goal is checking if all the objectives in O can be reached by system $\mathcal{A}$ with given actions $\mathcal{ACT}$. This can be done by checking the following: $\forall t_i \in \mathbb{T}$, checking if $O \subset$ ($\mathbb{S}$, $\mathbb{T}$) according transitions $\mathbb{E}$; if O is a subset of ($\mathbb{S}$, $\mathbb{T}$), the system is *safe*; otherwise, the system is *unsafe*.

Simulation and AUC calculation can be achieved by replacing O. For simulating the system, O=($\emptyset$, $t \ge t_n$), i.e., asking if the system can reach the clock of t_n without any constrains, which is **always safe**. According to the definition of AUC [14], the AUC of the concentration C equals to AUC=$\int_0^{t_n} \frac{d[C]}{dt}dt$.

Problem 2: Let the upper bound clock be t_n and the initial states be s_0. Given the system $\mathcal{A}$, the injection actions $\mathcal{ACT}$, and a concentration lower bound l, we define AUC_{under} to be the area above the concentration curve but below the bound l. Similarly, we can define AUC_{over} if an upper bound concentration limit is given. Such concentration bounds can be provided by the doctors for medical therapy or by the medicine researchers for drug analysis. AUC_{under} and AUC_{over} are the two types of *constrained AUC* considered in this paper. One example of AUC_{under} is shown in Figure 1 with a lower bound limit l=0.002, where t_x is the first clock when the concentration is lower than l and t_y is the first clock when the concentrations are higher than l. Note that l could be time continuous function, or a relative drug exposure baseline [12]. Then, AUC_{under} can be computed as

$$AUC_{under} = \int_{t_x}^{t_y} l\ dt - \int_{t_x}^{t_y} \frac{d[C]}{dt}dt \qquad (1)$$

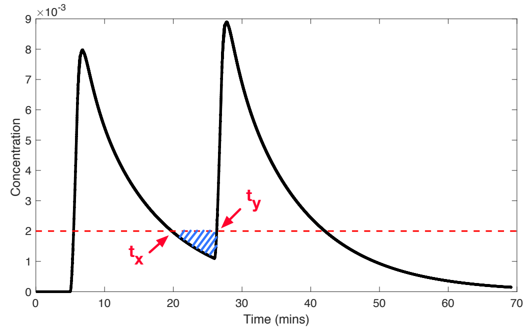

Fig. 1: Example of constrained AUC.

Based on Eq. 1, the problem is to find the clocks t_x and t_y. We introduce an algorithm that obtains such clocks based on a *proof of unsatisfiability* in Section IV.

III. Background

A. Three-Compartment Model

Mathematic models of a human body are created to study physiologic or pharmacologic kinetic characteristics. The compartment model can simulate the biologic processes involved in the kinetic behavior of a drug after it has been introduced into the body, leading to a better understanding of its pharmacodynamic effects []. Mostly, one compartment model is not sufficient to represent the pharmacokinetics of a drug. A two- and three-compartment model have wider applicabilities. In this work, we use three-compartment to represent

[1] c_{i_j} is a comparison function, e.g., c_{i_0} $\le$ 0.01 or c_{i_0}==0.01.

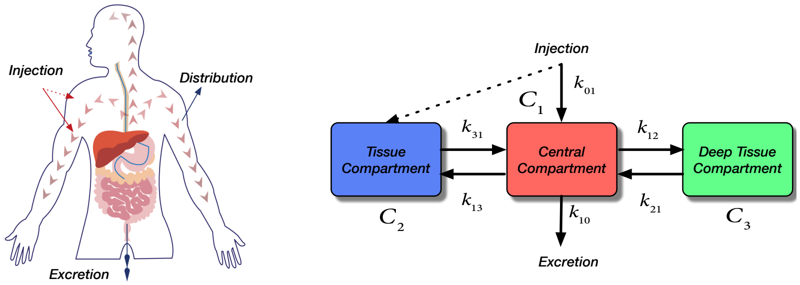

Fig. 2: Three-Compartment pharmacokinetics model. Injection could be taken in either central compartment C_1, such as blood injection, or tissue compartment C_2, such as muscle injection.

the pharmacokinetics of a drug, specifically using Michaelis-Menten elimination model [15][16]. The abstract model is shown in Figure 2, including central compartment, tissue, and deep tissue compartment sub-models. The three-compartment represents a drug that is distributed most rapidly to a highly perfused central compartment such as blood and brain. This is also the compartment which takes the injection. The drug is distributed less rapidly to the tissue compartment such as muscle, and very slowly to the deep tissue compartment, containing such poorly perfused tissue as bone and fat. The deep tissue compartment may also represent tightly bound drug in the tissues.

After the injection, it is first distributed to the central compartment C_1. There is then redistribution to tissue compartment C_2 with good perfusion, with further redistribution to the poorly perfused deep tissue C_3. The rates of infusion, $k_{12}, k_{21}, k_{13}, k_{31}$, depend on the rate of transfer between the various theoretical compartments of the body. Elimination (drug clearance) only happens at the central compartment, with rate k_{10}.

The model is described using *ordinary differential equations* (ODEs) [15]. In general, there are two dynamic models of three-compartment model for modeling an injection system, i.e., *distribution* model and *injection* model. The distribution dynamic model represents the distribution and dilution of the injection, as shown in Eq. 2 C_1, C_2, and C_3 are the concentration of the central compartment, tissue compartment and deep tissue compartment, over time t. The central compartment concentration C_1 depends on the rate of excretion (-$k_{10}C_1$) and the rates of distributing to the other two compartments (-$k_{12}C_1$-$k_{13}C_1$), and the other two compartments are only related to C_1.

$$\begin{aligned} \frac{d[C_1]}{dt} &= -(k_{10}+k_{12}+k_{13}) \cdot C_1 + k_{21} \cdot C_2 + k_{31} \cdot C_3 \\ \frac{d[C_2]}{dt} &= -k_{21} \cdot C_2 + k_{12} \cdot C_1 \\ \frac{d[C_3]}{dt} &= -k_{31} \cdot C_3 + k_{13} \cdot C_1 \end{aligned} \tag{2}$$

The second dynamic is required to model the concentrations of the three compartments when an injection is taken. The difference compared to the distribution dynamic is in the ODE of C_1, shown in Eq. 3 R_{inject} is the rate of drug injection which is a constant number. The amount of drug injected $\int_t^{t+\Delta t} R_{inject} dt = R_{inject} \cdot \Delta t$. Note that distribution (dilution) of the body naturally processes all the time. By adding R_{inject} in the first ODE of Eq. 3, this model successfully describes the injection process with distribution. In one injection monitoring system, there could be more than one dynamic models if the injection rate can be adjusted.

$$\begin{aligned} \frac{d[C_1]}{dt} &= -(k_{10}+k_{12}+k_{13}) \cdot C_1 + k_{21} \cdot C_2 + k_{31} \cdot C_3 + R_{inject} \\ \frac{d[C_2]}{dt} &= -k_{21} \cdot C_2 + k_{12} \cdot C_1 \\ \frac{d[C_3]}{dt} &= -k_{31} \cdot C_3 + k_{13} \cdot C_1 \end{aligned} \tag{3}$$

B. Satisfiability Modulo Theory (SMT)

The Satisfiability Modulo Theories (SMT) problem is a decision problem for logical formulas with respect to first-order logic. In other words, SMT departs from treating the problem in a strictly Boolean domain and integrates different well-defined theories (Boolean variable, bit vectors, integer/floating arithmetic, reals, etc.) into a DPLL-style SAT decision procedure [5]. Some of the most effective SMT solvers that are developed for specific problems. For example, Boolector [17] is the most efficient SMT solver in solving bit-level decision problem; Z3 [18] and CVC [19] have been widely used in verifying software. SMT formulas over the real numbers can encode a wide range of problems, particularly in modeling hybrid systems. dReal [6] is the state-of-the-art SMT solver over reals that can model the verification problem of hybrid system.

IV. Modeling

This section introduces the modeling of the injection systems, the verification problem and the algorithm of calculating the constrained AUC, using the non-linear SMT solver dReal. First, a set of global definitions has to be claimed. According to Eq. 2 and Eq. 3, these include the definition of static variables and dynamic variables. The static variables include distribution, absorption and excretion rate k_{ij}, e.g., using syntax ″#define k_{10} 0.4;″. The concentration of each compartment C_i and the clock *time* are defined as dynamic variables with a bound, e.g., using syntax ″[0, 60] time;″.

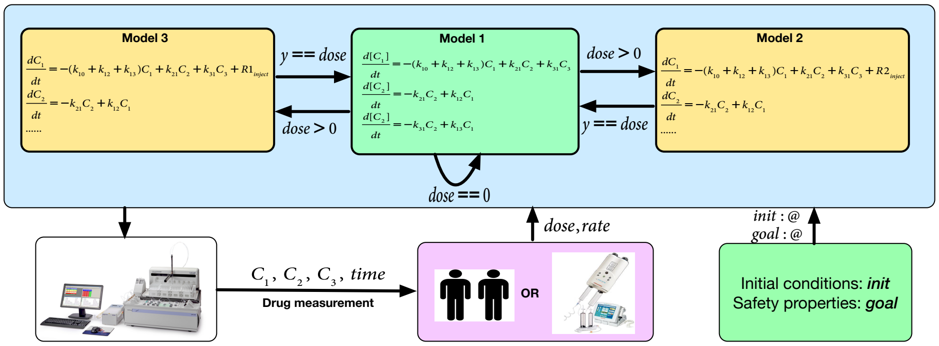
Model 3
$\frac{dC_1}{dt} = -(k_{10}+k_{12}+k_{13})C_1 + k_{21}C_2 + k_{31}C_3 + R1_{inject}$
$\frac{dC_2}{dt} = -k_{21}C_2 + k_{12}C_1$
......
$y == dose$
$dose > 0$
Model 1
$\frac{d[C_1]}{dt} = -(k_{10}+k_{12}+k_{13})C_1 + k_{21}C_2 + k_{31}C_3$
$\frac{d[C_2]}{dt} = -k_{21}C_2 + k_{12}C_1$
$\frac{d[C_2]}{dt} = -k_{31}C_2 + k_{13}C_1$
$dose == 0$
$dose > 0$
$y == dose$
Model 2
$\frac{dC_1}{dt} = -(k_{10}+k_{12}+k_{13})C_1 + k_{21}C_2 + k_{31}C_3 + R2_{inject}$
$\frac{dC_2}{dt} = -k_{21}C_2 + k_{12}C_1$
......
$C_1, C_2, C_3, time$
Drug measurement
OR
$dose, rate$
$init : @$
$goal : @$
Initial conditions: *init*
Safety properties: *goal*

Fig. 3: Generic modeling of injection system with two injection dynamics.

A. Modeling Dynamics in SMT

The main part is the dynamic model, which is the three-compartment model of the medical injection system. A complete dynamic model must include all the elements retried in the tuple of the timed automata $\mathcal{A}$ (Section II). We first introduce the SMT model of the *distribution* dynamic shown in Eq. 4.

To define a dynamic model, we first declare the label of the model with a numerical value m (line 1). The transition between different dynamics is described using the pointer m of the model. Second, the *invariant* for the states are defined (lines 2 and 3), which is a conjunction of logic formulas which must always hold in a model. For the *distribution* dynamic, the invariant define that the concentrations of all the compartments $C_i \geq 0$, and there is no absorption (*dose=0*). The continuous dynamics of a model by providing a set of ODEs of the distribution dynamic are included in $flow$, where $d/dt[C_1]$ represents $\frac{d[C_1]}{dt}$, t is the global variable $time$. The first formulas of $jump$ is interpreted as $guard$, i.e., a logic formula specifying a condition to make a transition. Note that this allows a transition but does not force it. The second argument of jump denotes the target model m of the triggered transition, and applies to the dynamic variables in the logic formulas. In this conjunction, C_i' represent the dynamic variable $\dot{C}_i$.

```
1  $mode\ 1;$
2    $invt:$
3       $(and(C_1 \geq 0)(C_2 \geq 0)(C_3 \geq 0)(dose \leq 0)(dose \geq 0));$
4    $flow:$
5       $d/dt[C_1] = -(k_{10} + k_{12} + k_{13})C_1 + k_{21}C_2 + k_{31}C_3;$
6       $d/dt[C_2] = k_{12}C_1 - k_{21}C_2;$                                    (4)
7       $d/dt[C_3] = k_{13}C_1 - k_{31}C_3;$
8       $d/dt[x] = 1;$
9    $jump:$
10       $(guard_{model1})$
           $==> @2(and(C_1' = C_1)(C_2' = C_2)(C_3' = C_3)(x' = x));$
```

An extra dynamic variable $\dot{x}$ is introduced in all the dynamic model (line 8) to represent the clock. $\dot{x}$=$\dot{time}$ with $\frac{dx}{dt}$=1. This is because dReal doesn't support $time$ to be used in $guard$. Note that $guard$ is specifically constructed according to the hybrid system. For example, if the injection will be taken when the concentration of tissue compartment is equal to c (e.g., using electronic pump), guard=(and $(C_2 \leq c)$). If the injection is taken periodically every t_p, to model two absorptions, guard=(and$(\mathsf{x} \leq t_p)(\mathsf{x} \geq t_p)(\mathsf{x} \leq 2t_p)(\mathsf{x} \geq 2t_p)$). In both cases, $jump$ will point to the *injection* dynamic(s). Our SMT model is very flexible to model a hybrid system with both feedback control and human operators.

The difference between the ODEs of injection and distribution dynamics is C_1. However, the SMT model has to be changed. The invariant condition should be replaced with $dose > 0$ in the previous model. Multiple modifications need to be done in $flow$. The differential equation of C_1 is replaced by line 5 in Eq. 5. An extra dynamic variable y defined with $d/dt[y] = R_{inject}$ is used for constructing $guard$, where y is calculating the amount of drug injected (line 8). Once the transition is made, we have to reset y in case there are multiple doses in the hybrid system. For the systems that have various injection rates $R_{injection}$, we need to create separate injection model for each of them.

```
1  $mode\ 2;$
2    $invt:$
3       $(and(C_1 \geq 0)(C_2 \geq 0)(C_3 \geq 0)(dose > 0));$
4    $flow:$
5       $d/dt[C_1] = -(k_{10} + k_{12} + k_{13})C_1 + k_{21}C_2 + k_{31}C_3 + R_{inject};$
...
8       $d/dt[y] = R_{inject};$
10    $jump:$
11       $(and(y \geq dose)(y \leq dose))$
           $==> @3(and(C_1' = C_1)...(y' = 0));$
                                                                        (5)
```

Finally, the initial states of the first model and the safety goal of the hybrid system have to defined. If the hybrid system is initialized with model 1, $init$ should start with @1 with all variables set to 0. $goal$ shares the same syntactic structure of $init$. The safety properties can be constructed using the conjunctions of formulas. For example, line 3 in Eq. 6 is checking if C_2 is in [0.005, 0.01] during time [10, 15]; line 5 checks if $C_1 \leq 0.1$ is always safe over all the clocks.

B. System modeling

The general system modeling of the injection systems is shown in Figure 3. The drug response model is built with one distribution dynamic and two injection dynamics since there exist two injection rates. The proposed SMT model can formulate any control units (the injection control) if the decisions are made based on time and the concentrations. This is done by modifying $guard$ for each dynamic model formula. For example, assume that there are two injections with amount d_1 and d_2 at t=5 and t=20 over $time$=[0, 60], using the injection rates $R1$ and $R2$, respectively. The transitions are $model\ 1 \rightarrow model\ 2 \rightarrow model\ 1 \rightarrow model\ 3 \rightarrow model\ 1$. The time condition should trigger the transitions between model 1, and models 2 and 3. $guard$ of model 1 should describe $x\ ==\ 5$ OR $x\ ==\ 20$. However, SMT over the reals only supports conjunction of formulas. Hence, we need to model OR using inversion and AND, such that (x=5) $\vee$ (x=20) $\rightarrow$ $\overline{(x \neq 5) \wedge (x \neq 20)}$. The SMT formula is

$$jump : (not(and(x < 5)(x > 5))\ (and(x < 20)(x > 20)))$$

Similarly, the control decisions made based on concentration, or the combination of concentration and time, can be modeled using the same approach.

$$\begin{array}{ll} 1 & init : @1(and(C_1 = 0)(C_2 = 0)(C_2 = 0)(x = 0)); \\ 2 & goal : @1(and \\ 3 & \qquad (and(C_2 \geq 0.01)(C_2 \leq 0.005)(x \geq 10)(x \leq 15)) \\ 4 & \qquad (and(C_3 \geq 0.003)(C_3 \leq 0.001)(x \geq 20)(x \leq 25)) \\ 5 & \qquad (and(C_1 \leq 0.1)) \\ 6 & \qquad (and(...))) \end{array} \tag{6}$$

C. Area Under Curve (AUC) and Constrained AUC

To compute AUC of the three concentrations, we just need to add three dynamic variables and differential equations in all the models. AUC_i are the dynamic variables of AUC of i^{th} concentration. According to the definition of AUC, the derivative of AUC is the concentration function, which can be simply represented using Eq. 7.

$$d/dt[AUC_1] = C_1;\ d/dt[AUC_2] = C_2;\ d/dt[AUC_3] = C_3; \tag{7}$$

As mentioned in Section II Figure 1, to compute constrained AUC, t_x and t_y that indicates the bounded clocks of the error regions are required. Once bounded clocks are available, constrained AUC can be computed by adding the Eq. 1 into the hybrid model. Note that multiple error regions may exist. Hence, we introduce an algorithm that iteratively collects the bounded clocks by using the *proof of unsatisfiability* generated by dReal (with option –proof), shown in Algorithm 1.

The algorithm takes the SMT formula of the system ψ, the error bound l as inputs and generates the bounded clocks of the error regions. The algorithm includes two global $goal$ formulas for ψ, g and g' indicating *safe* and *unsafe*. First, the algorithm checks if there exists an error state by checking if ψ is always safe with error bound l (line 4). If there is no error state over $time$, the algorithm will be terminated. If there exit error states, the algorithm will start collecting the bounded clocks (lines 7-16). In each iteration, it first extracts the starting clock of the error region, t_x^i, by extracting the smallest clock in the proof. Note that the proof can be generated iff the problem is unsat. Hence, $goal$ of ψ is complemented. The proof of unsat includes the starting clock of error region for g', which is the ending clock of the error region for g. Once i^{th} iteration is done, $goal$ is reset to g, and the initial time t_{init} is set to t_y^i such that $time$=$[t_y^i, t_n]$ in the next iteration. This makes sure that the next iteration skip all the previous collector regions. The bounded clocks (t_x^i, t_y^i) be returned if there is no more error region (lines 7 and 17).

Algorithm 1 Constrained AUC

```
Input: Hybrid system formula $\psi$ in SMT
Input: Error bound $l$, $time$ = [0, $t_n$]
Output: Bounded clocks of unsafe region with error bound $l$.
 1: $g$: $l$ is infeasible (safe); $g'$: $l$ is feasible (unsafe);
 2: $goal$=$g$;
 3: $t_{init} = 0$; $i$=0;
 4: if $\forall\ t_i \in time$, ($\psi \hookrightarrow goal$) is always $SAT$ then
 5:     $t_x^i$ = $t_y^i$ = $null$;
 6: end if
 7: while $\exists\ t_i \in time$, ($\psi \hookrightarrow goal$) is $UNSAT$ do
 8:     Extract $t_x^i$ from the proof of unsat;
 9:     $t_{init} = 0$; $goal$=$g'$;
10:     if $\exists\ t_i \in time$, ($\psi \hookrightarrow goal$) is $UNSAT$ then
11:         Extract $t_y^i$ from the proof of unsat;
12:     else
13:         $t_y^i$=$t_n$;
14:     end if
15:     $i$++; $goal$=$g$; $t_init$=$t_y^i$;
16: end while
return $t_x^i$ and $t_y^i$, $\forall i$;
```

V. Experimental Results

The experimental results are conducted on MacOS with 2.3 GHz Intel Core i7 x4 with 16 GB memory. We solve the hybrid SMT formulas using dReal[6] in the single-thread [6]. Algorithm 1 is implemented in C++ using dReal as a blackbox that generates the proof of unsat. We demonstrate our approach using the example used for illustrating the injection system modeling in Section IV. The system has a time bound [0, 60] hours and has two injections triggered by $time$=5 and $time$=20. To show the complete results of all the states up to $time$=60, the $goal$ is set as goal: @1 ($x \geq 60$);.

The results are included in Figure 4. The x-axis represent the time. Left-hand y-axis represents the concentrations C_1, C_2, and C_3. Right-hand y-axis represents the AUC of each concentration. All the results are time continuous with interval 0.005 second defined the precision of the SMT solver (with option –precision). The runtime of generating all the results in Figure 4 is less than 15 seconds. If a set of concentration-time objectives O are provided by the users, the $goal$ has to be modified using Eq. 6. Mostly, checking the satisfiability of O takes less CPU time than simulating over all the clocks. This is because the SMT solving process will be terminated as soon as an unsafe state s_{unsafe} is detected. For example, if O includes $C_2 \leq 2e^{-4}$ for clocks in [20, 25], the solver will return *unsat* and terminate at the first clock that $C_2 > 2e^{-4}$.

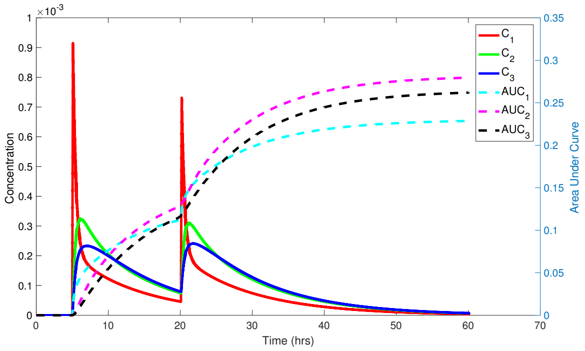

Fig. 4: Continuous results of the concentrations and area under curves (AUCs) generated by dReal up to $time$ = 60 hours.

We show the result of computing the constrained AUC of C_2 using the same system, shown in Figure 5. The runtime overhead of Algorithm 1 compared to the original SMT formulas varies on the error bound function l. If the given exposure baseline (error bound l) is a linear function, such as $l = 1e^{-4}$, Algorithm 1 computes the constrained AUC with almost no runtime overhead. A non-linear function l could significantly increase the runtime complexity, which mainly comes from the SMT solver dReal. As shown in Figure 5, there are two error regions indicated by the bounded clocks $t_x^{1,2}$ and $t_y^{1,2}$. We can see that $AUC2(C_2 \leq 1e^{-4})$ is a time continuous function, and its value increases iff the clocks are in [t_x^1,t_y^1] and [t_x^2,t_y^2].

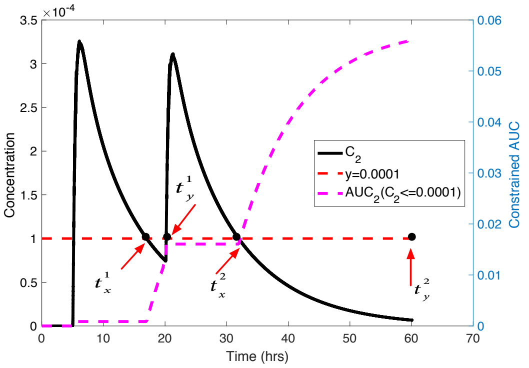

Fig. 5: Constrained AUC: AUC_{C_2} with error bound $l = 1e^{-4}$.

VI. Conclusion

This paper presents an efficient formal model that can solve the formal verification, simulation, and measurements of medical injection systems using Satisfiability Modulo Theories over Reals. We demonstrate that the proposed model can be used to model an injection system with actions performed by electronic injection system or human. The experimental results show the capabilities of our model in verification, simulation, and measuring the drug *Area Under the Curve* (AUC) and constrained AUC metrics. Using the state-of-the-art SMT solver dReal, our model produces high precision results over a wide clock range with only a few seconds.

ACKNOWLEDGMENTS: This project is funded by ERC-2014-AdG 669354 grant.

References

[1] S. A. Seshia, S. Hu, W. Li, and Q. Zhu, "Design automation of cyber-physical systems: Challenges, advances, and opportunities," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 9, pp. 1421–1434, 2017.

[2] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical computer science*, vol. 126, no. 2, pp. 183–235, 1994.

[3] K. G. Larsen, P. Pettersson, and W. Yi, "Uppaal in a nutshell," *International journal on software tools for technology transfer*, vol. 1, no. 1-2, pp. 134–152, 1997.

[4] T. Amnell, E. Fersman, L. Mokrushin, P. Pettersson, and W. Yi, "Times ba tool for modelling and implementation of embedded systems," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2002, pp. 460–464.

[5] A. Biere, M. Heule, and H. van Maaren, *Handbook of satisfiability*. IOS press, 2009, vol. 185.

[6] S. Gao, S. Kong, and E. M. Clarke, "dreal: An smt solver for nonlinear theories over the reals," in *International Conference on Automated Deduction*. Springer, 2013, pp. 208–214.

[7] B. Liu, S. Kong, S. Gao, and E. Clarke, "Parameter identification using delta-decisions for biological hybrid systems," CMU SCS Technical Report, CMU-CS-13-136, Tech. Rep., 2014.

[8] L. Shargel, B. Andrew, and S. Wu-Pong, *Applied biopharmaceutics & pharmacokinetics*. McGraw-Hill Medical Publishing Division, 2015.

[9] S. L. Shafer and J. R. Varvel, "Pharmacokinetics, pharmacodynamics, and rational opioid selection," *Anesthesiology*, vol. 74, no. 1, pp. 53–63, 1991.

[10] M. Rybak, B. Lomaestro, J. C. Rotschafer, R. Moellering, W. Craig, M. Billeter, J. R. Dalovisio, and D. P. Levine, "Therapeutic monitoring of vancomycin in adult patients: a consensus review of the american society of health-system pharmacists, the infectious diseases society of america, and the society of infectious diseases pharmacists," *American Journal of Health-System Pharmacy*, vol. 66, no. 1, pp. 82–98, 2009.

[11] B. Donato, F. Stradolini, A. Tuoheti, F. Angiolini, D. Demarchi, G. De Micheli, and S. Carrara, "Raspberry pi driven flow-injection system for electrochemical continuous monitoring platforms," in *IEEE BioCAS*. IEEE, 2017.

[12] J. D. Scheff, R. R. Almon, D. C. DuBois, W. J. Jusko, and I. P. Androulakis, "Assessment of pharmacologic area under the curve when baselines are variable," *Pharmaceutical research*, vol. 28, no. 5, pp. 1081–1089, 2011.

[13] S. Gao, J. Avigad, and E. M. Clarke, "δ-complete decision procedures for satisfiability over the reals," in *International Joint Conference on Automated Reasoning*. Springer, 2012, pp. 286–300.

[14] M. J. Pencina, R. B. D'Agostino, and R. S. Vasan, "Evaluating the added predictive ability of a new marker: from area under the roc curve to reclassification and beyond," *Statistics in medicine*, vol. 27, no. 2, pp. 157–172, 2008.

[15] B. P. English, W. Min, A. M. Van Oijen, K. T. Lee, G. Luo, H. Sun, B. J. Cherayil, S. Kou, and X. S. Xie, "Ever-fluctuating single enzyme molecules: Michaelis-menten equation revisited," *Nature chemical biology*, vol. 2, no. 2, pp. 87–94, 2006.

[16] J. Nyberg, C. Bazzoli, K. Ogungbenro, A. Aliev, S. Leonov, S. Duffull, A. C. Hooker, and F. Mentré, "Methods and software tools for design evaluation in population pharmacokinetics–pharmacodynamics studies," *British journal of clinical pharmacology*, vol. 79, no. 1, pp. 6–17, 2015.

[17] A. Niemetz, M. Preiner, and A. Biere, "Boolector 2.0," *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 9, 2015.

[18] L. De Moura and N. Bjørner, "Z3: An efficient smt solver," in *Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2008, pp. 337–340.

[19] C. Barrett, C. L. Conway, M. Deters, L. Hadarean, D. Jovanović, T. King, A. Reynolds, and C. Tinelli, "CVC4," in *Computer aided verification (CAV)*. Springer, 2011, pp. 171–177.