Spatio-Temporal Hybrid Automata for Safe Cyber-Physical Systems: A Medical Case Study*

Ayan Banerjee and Sandeep K.S. Gupta
IMPACT Lab,Arizona State University, Tempe, Az
{abanerj3,sandeep.gupta}@asu.edu

ABSTRACT

Interactions between the computing units and the physical environment in Cyber-Physical Systems (CPSes) are considered to verify safety properties, i.e. ensuring the un-intentional side-effects of cyber-physical interactions are within desired limits. A Linear 1 space dimension Spatio-Temporal Hybrid Automata (L1STHA) is defined to capture the effects of the interactions, in both time and space. Aggregate effects of interactions due to concurrent operations in the computing entities are expressed as a set of inter-dependent partial differential equations associated with dedicated modes of the L1STHA model. A time and space bound L1STHA reachability analysis algorithm is proposed for safety verification, which provides reachable states of the L1STHA with an arbitrary accuracy ϵ. The runtime of the algorithm depends on the requested accuracy. The usage of the L1STHA modeling and analysis is demonstrated for medical CPSes such as infusion pumps.

Categories and Subject Descriptors

F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*Automata*

1. INTRODUCTION

Cyber-Physical Systems (CPSes), where computing units interact with the physical environment for either control of the environment or driving computation are becoming increasingly prevalent in the society, especially in healthcare systems such as infusion pumps [16]. CPSes by definition are safety critical, requiring safety verification before their deployment [26]. As such the need for formal verification to theoretically prove the safety of CPS operations is well recognized [20]. This paper deals with formal modeling and verification of CPS operations.

An important aspect of the CPSes is the seamless and complex interactions among the computing units and the physical environment, referred to as cyber-physical interactions. Such interactions can often be an unwanted or unintentional result of computing operations and may cause hazards to the environment. Examples exist in the medical domain, where heat generated from a pulse oximeter can burn the human skin especially in infants [14] or the chemotherapy drugs may kill normal cells apart from the cancer cells [16]. It is therefore imperative that a CPS maintains the detrimental effects of such interactions within desired limits to ensure safe operations.

Many CPSes, are further distributed in nature, i.e. they comprise of network of medical devices controlled through the wireless channel for example multi-channel infusion pumps. The distributed nodes often perform concurrent operations; thus causing aggregation of the detrimental impact on the environment from multiple nodes. For example, concurrent infusions in multi-channel infusion pump control systems have accumulated drug effects on certain parts of the body depending on the site of infusion from the two channels [16]. Thus, aggregate interactions require proper characterization to ensure the CPSes' safety.

A major aspect in the modeling of such interactions is to capture the continuous dynamics in the environment, which includes the human physiology [4]. These dynamics can be spatio-temporal in nature. For example, the drug concentration in human blood due to the operation of an infusion pump can continuously vary both with time and space [16]. Figure 1 shows the drug concentration for a two-channel infusion pump obtained by simulating the diffusion dynamics suggested in [16]. The glucose concentration depicted by grey solid curves in Figure 1, show spatial variation which changes over time. The figure also shows aggregate effects of interaction at time t = 500s as a result of concurrent infusion. Formal models for CPSes should thus capture four salient features: 1) discrete time behavior of the computing nodes; 2) continuous dynamics of the physical environment; 3) spatio-temporal variation of the continuous physical parameters; and 4) aggregate effects due to concurrent operations of networked computing nodes.

Traditionally, hybrid automata [19, 23] are used for capturing both discrete and continuous behavior of a system. However, current tools to model a hybrid automata such as PHaver or SpaceEx [10], and HybridSAL [27], only consider one dimensional variation of parameters, generally over time, and are hence insufficient for modeling the spatio-temporal variation of physical parameters in CPSes. Safety analysis of hybrid automata generally involve reachability study of the state trajectories to analyze whether the designated sets of unsafe states are reached with progression in time. However, for CPSes reachability study should evaluate state trajectories in both time and space. To this effect, researchers have considered spatial network of hybrid automata to capture the spatial propagation of physical parameters across the human body [6, 11, 22, 28].

Spatial networks of hybrid automata discretize the space as a grid and statically allocate hybrid systems at specific points. Thus, they depend on fixed spatial boundaries to setup the network. However,

*This research was funded in part by NSF grants CNS-0831544, CNS-1231590, and IIS- 1116385. Special thanks to OSEL at FDA for providing infusion pump models.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICCPS'13, April 8-11, 2013, Philadelphia, PA, USA.
Copyright 2013 ACM 978-1-4503-1996-6/13/04. . . $15.00..

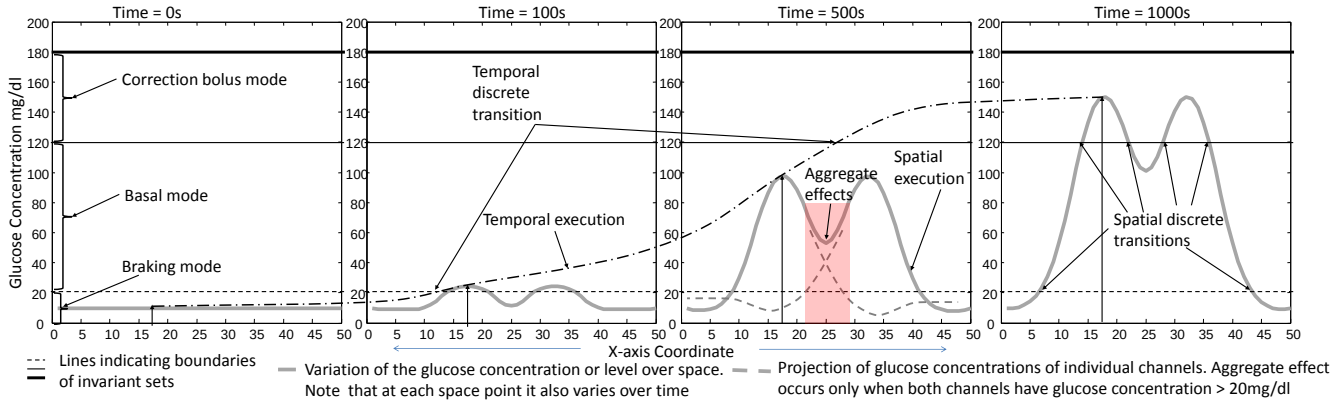

Figure 1: Example execution of the L1STHA model of a multi-channel infusion pump, for 1000 s and over a 50 mm spatial region.

most of the physical dynamics in the human body are *Free Boundary Problems* [9] where the boundary conditions in space causes the spatial extent of interactions to expand. For example, the drug concentration due to infusion at a given time is maximum at the site of infusion and gradually decreases as we move away from the site. A point in space, which has a negligible concentration at time t_1, can have a considerably higher concentration at a later time t_2. Thus, a space point, which was not considered in the analysis since it was outside the boundary, will be ignored by a spatial network of hybrid automata, even if at a later time it has considerable concentration. Thus, they also fail to accurately model the aggregate effects due to concurrent operations in networked computing units.

Modeling aggregate effects in CPSes, requires new dynamic equations in the formal model. For example, the coefficients of the equation governing the temperature rise in the human tissue changes when there are multiple heat sources (sensors) [5, 24, 25]. Thus, a single formal specification is necessary for the network, where aggregate effects can be expressed with new dynamic equations. Hilbertean transforms [8] have also been used to specify CPSes formally. However, to the best of our knowledge, no formal verification of safety has been performed using Hilbertean transform.

Computationally tractable reachability analysis of a hybrid automata is only possible to date for affine dynamics in one dimension [19, 23], specifically due to existence of closed form solutions of the differential equations involved. For the most commonly occurring form of spatio-temporal dynamics, linear second order partial differential equations (PDEs) as observed in Penne's bioheat equation [21] or infusion pump diffusion dynamics [16], there exists no closed form solution even for a single spatial dimension. Hence, traditional method for reachability analysis with zonotopes [12] do not apply. For temporal dynamics without closed form solutions, a time bounded reachability analysis is proposed in [17]. This paper takes a similar approach and proposes *a novel bounded time and space reachability analysis for spatio-temporal hybrid automata, where the interactions are represented as linear second order PDEs with one space dimension.* Specifically the paper makes the following contributions:

1. *Defines* a *Linear 1-space dimension Spatio-Temporal Hybrid Automata (L1STHA)* capturing the aggregate spatio-temporal dynamics of cyber-physical interactions in CPSes;
2. *Develops* a *bounded time and space reachability analysis of L1STHA models* enabling safety verification of CPSes based on their spatio-temporal behavior.
3. *Applies* the *reachability analysis* technique to medical CPSes specifically infusion pumps for safety analysis.

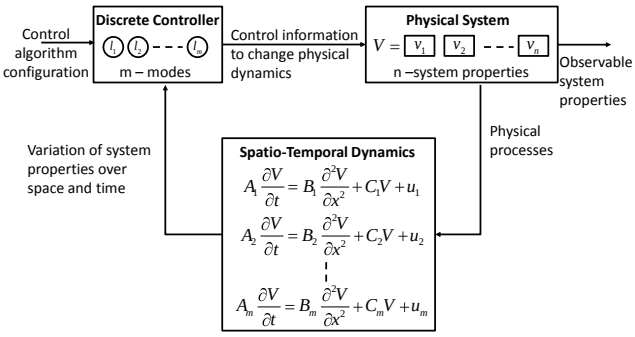

Figure 2: Assumed system model for Cyber-Physical Systems.

In the next section, we discuss some preliminary ideas, an overview of our approach and the infusion pump example used in this paper.

2. PRELIMINARIES AND APPROACH

Our system model of CPSes consists of a computing unit as a controller of the physical environment (Figure 2). The controller takes feedback from the physical environment and makes a decision on the control information, which is transmitted to the actuator. The control algorithm has m modes of operation, $(l_1, l_2, \ldots, l_m)$, each with a different control policy. The physical system can be represented using n continuous variables or physical parameters, which vary according to a second order PDE. In this paper, we restrict these PDEs to one space dimension x.

Each continuous variable can assume a value from the real set at a given time and space point. The vector $\boldsymbol{v}$ formed by the values of each continuous variable is called a *continuous state*. The *continuous state space* is a subset V of $\mathbb{R}^n$ and consists of all possible continuous states. A *state* is a tuple $(l, \boldsymbol{v})$, consisting of a mode l and a continuous state $\boldsymbol{v}$ [13].

In this theory, we assume that the continuous state space is compact. That is each subset J of the continuous state space has a continuous interior denoted by $J^{\boxdot}$, an open set (e.g., $0 < x < 10$), and a boundary denoted as $J^{\square}$ given by boundary conditions (e.g. x = 0, x = 10). The set $J = J^{\square} \cup J^{\boxdot}$ (e.g. $0 \leq x \leq 10$). The entire continuous state space V, is partitioned into a collection of polyhedral subsets $J = \{J_1, J_2, \ldots\}$ such that: a) $\bigcup_{\forall J_i \in J} J_i = V$, and b) $J_i^{\boxdot} \bigcap J_j^{\boxdot} = \emptyset$ for $i \neq j$. Each polyhedron subset is called a *cell* and

is a non-empty set of real numbers. Further, two cells are called *adjacent* if the cells J_i and J_j intersect in an n-dimensional facet. Two cells are *connected* if there is a sequence of adjacent cells between the two.

Hausdorff distance, $d_H(P, Q)$, between two sets P and Q is the maximum distance, computed using l_∞ norm, of any point in P to its corresponding closest point in Q [15].

The main advantage of considering Hausdorff distance is that a δ neighborhood of a continuous state $\boldsymbol{v}$ in $\mathbb{R}^n$, denoted by $B_\delta(\boldsymbol{v})$, computed using l_∞ norm, is a hypercube of side length 2δ. The set of *vertices* of a neighborhood $B_\delta(\boldsymbol{v})$ is denoted by $Vert(B_\delta(\boldsymbol{v}))$. This is illustrated in Figure 4, where δ neighborhood of a continuous state P in two dimension is a square of side length 2δ. This immensely simplifies the computation of convex hull of a set of continuous state required for reachability analysis [2].

A unit outward normal vector $\hat{n}$ from a continuous state $\boldsymbol{v_i}$ on $J_i^{\square}$ is a vector of unit length normal to the tangent on $J_i^{\square}$ at $\boldsymbol{v_i}$ pointing towards complement of J_i, J_i^C. Note that the notion of outward direction encompasses multiple directions in the coordinate space. As shown in Figure 3, both the west at $\boldsymbol{v_i}$ and east at $\boldsymbol{v_j}$ directions are pointing away from the set J_i. The dot product of the temporal or spatial variation of any variable v_i with the unit vector $\hat{n}$ denoted by $\frac{\partial v_i}{\partial t} \bigodot \hat{n}$ or $\frac{\partial^2 v_i}{\partial x^2} \bigodot \hat{n}$, is the amount of variation in the direction from set J_i to J_i^C. This notion is useful for capturing the transition between L1STHA modes.

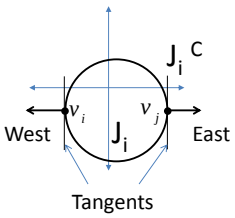

Figure 3: Unit outward normal vector from set J_i at continuous states $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$.

The *image* of any continuous state $\boldsymbol{v}$, over time t and space x, denoted by $D_{t,x}(\boldsymbol{v})$ is the computation of the solution of the linear 1-space dimensional second order PDE, for a time t and space x, with $\boldsymbol{v}$ as the initial condition.

Overview of Approach: The STHA expresses the variation of continuous variables of a CPS according to a control logic specified in the operating modes. In our approach, we first define the L1STHA as discussed in Section 1, and then define its execution logic, which formalizes how the L1STHA operates in space and time. The execution is governed by a linear 2nd order PDE.

The reachability analysis of L1STHA is the method of approximating the states that can occur during the execution of the L1STHA in space and time from a given initial state (defined later), or in other words estimating the *reach set*. This involves computing the image of an initial state using the solution of the PDE (Figure 4). To this effect, we first consider an L1STHA with only a single mode and develop the algorithm to estimate its reach set (Section 5.3).

The reach set of a single mode L1STHA is computed using the following steps (marked in Figure 4): a) we first find a suitable discretization of time and space that guarantees that the error in computing the image of the PDE is within the desired accuracy ϵ, b) we then consider a δ neighborhood of $\boldsymbol{v_0}$ of the initial state $\{l_0, \boldsymbol{v(0)}\}$, $B_\delta(\boldsymbol{v(0)})$, c) the image of the vertices of the neighborhood, $Vert(B_\delta(\boldsymbol{v(0)}))$, is then computed using the PDE for a given time and space, d) γ neighborhood of each point in the image is considered, resulting in a new set of vertices, d) the convex hull [2], of the new set of vertices gives γ approximation of the reach set. As shown later in the formulation γ and δ are related linearly to ϵ. The same algorithm is applicable for estimating the reach set of multi-mode L1STHA (Section 6). The only difference is that in the process of computing the images of vertices the reachability analysis algorithm keeps track of the transitions made by the L1STHA into different modes. For safety analysis, a subset of continuous states is designated as *unsafe*. If the computed reach set intersects the unsafe set, the CPS is deemed unsafe.

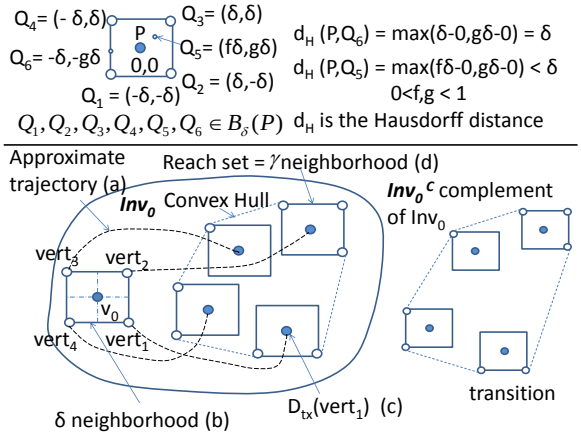

Figure 4: Overview of reach set computation methodology.

Safety in CPSes: Safety of CPSes in the medical domain is defined as the avoidance of unwanted hazards to the human body due to the cyber-physical interactions. Such a definition is generic and is applicable to CPSes in general. Hazards can be of several types as listed in ISO 60601 standard for medical electrical equipments. In this paper, we consider physiological hazards due to drug overdose by infusion pumps.

EXAMPLE 1. **Drug infusion:** *Infusion pumps operate in a close loop with a networked controller to keep the drug concentration in the human blood within recommended limits. The infusion pump has three modes: a) basal, where infusion rate is I_0, b) braking, where infusion rate is a fraction f of I_0, and c) correction bolus, where infusion rate is incremented by I_b. Diffusion dynamics of the drug is spatio-temporal in nature and can be modeled using multi-dimensional PDE Equation 1 [16].*

$$\frac{\partial d}{\partial t} = \nabla(D \nabla d) + \Gamma(d_B(t) - d) - \lambda d, \tag{1}$$

where $d(x,t)$ is the tissue drug concentration at time t and distance x from the infusion site, D is the diffusion coefficient of the blood, Γ is the blood to tissue drug transfer coefficient, and $d_B(t)$ is the prescribed infusion rate at time t, and λ is the drug decay coefficient. A control algorithm in the infusion pump samples Equation 1 and adjusts the infusion levels so as to achieve the desired physiological effects while avoiding hazards such as hyperglycemia.

3. LINEAR 1-D SPACE SPATIO TEMPORAL HYBRID AUTOMATA

The L1STHA expresses the variation of physical parameters according to a discrete control algorithm of the computing units in the CPS. A L1STHA is defined as follows:

DEFINITION 1. *L1STHA: It is a tuple $\{n, m, \mathbb{L}, Inv, A, B, C, u, Re\}$,*

- *n is the number of continuous variables in the variable set V,*
- *$\mathbb{L}$ is a set of m modes $\{l_1, l_2, \ldots l_m\}$,*
- *$Inv : \mathbb{L} \rightarrow 2^J$ is the invariant set, which maps each mode to a set of cells such that:*

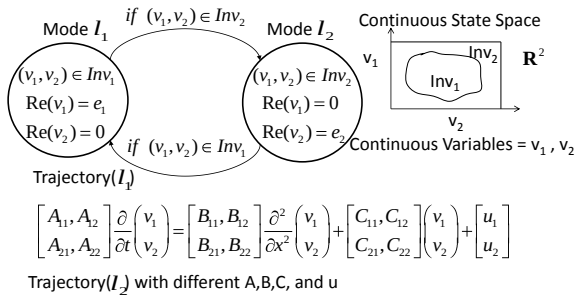

Figure 5: Conceptual illustration of L1STHA with two modes l_1 and l_2 and two continuous variables v_1 and v_2.

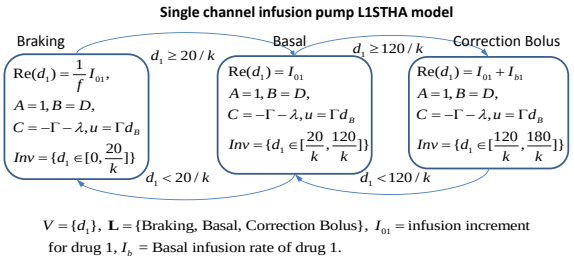

Figure 6: The L1STHA model of single channel infusion pump.

  - *for each $l \in \mathbb{L}$ the cells in $Inv(l)$ are all connected,*
  - *for any two different modes $\{l_i, l_j\} \in \mathbb{L}$, $Inv^{\square}(l_i) \bigcap Inv^{\square}(l_j) = \emptyset$,*
  - $\bigcup_{\forall i \in \{1 \ldots m\}} Inv(l_i) = V$,
- $A : \mathbb{L} \rightarrow \mathbb{R}^{n \times n}$, *maps a mode to an $n \times n$ real valued matrix,*
- $B : \mathbb{L} \rightarrow \mathbb{R}^{n \times n}$, *maps a mode to an $n \times n$ real valued matrix,*
- $C : \mathbb{L} \rightarrow \mathbb{R}^{n \times n}$, *maps a mode to an $n \times n$ real valued matrix,*
- $u : \mathbb{L} \rightarrow \mathbb{R}^n$, *maps a mode to an $n \times 1$ real valued vector,*
- $Re : \mathbb{L} \times V \rightarrow \mathbb{R}^n$ *is a reset function that sets initial conditions of the variables in V at each mode $l_i \in \mathbb{L}$.*

Associated with the definition of L1STHA is the definition of its trajectory that relates the variables A,B,C, and u.

Definition 2. *Trajectory: The trajectory of an L1STHA with n continuous variables for time $t \in \mathbb{R}$ and within a region $s \subset \mathbb{R}$ at a mode $l_i \in \mathbb{L}$ is defined as the 1-D space spatio-temporal mapping $\eta : [0, T] \times S \rightarrow \mathbb{R}^n$ such that:*

- *$\eta(t, x)$ for $t \in [0, T]$ and $x \in S$ follows the PDE:*

$$A(l_i)\frac{\partial \eta(t,x)}{\partial t} = B(l_i)\frac{\partial^2 \eta(t,x)}{\partial x^2} + C(l_i)\eta(t,x) + u(l_i), \qquad (2)$$

  where $A(l_i)$, $B(l_i)$, $C(l_i)$, and $u(l_i)$ are for a mode l_i;
- *and $\eta(t, x) \in Inv_i \; \forall t \in [0, T]$ and $x \in S$.*

The duration of the trajectory at a given spatial coordinate x is denoted by $\eta|_x.dur$ and its spatial range at a given time t is denoted by $\eta|^t.range$. A trajectory ends if at any space or time point the continuous variables cross invariant set boundaries. Figure 5 shows a conceptual view of a L1STHA. It has two modes l_1 and l_2. Each mode is associated with an invariant set, a subset in the real space $\mathbb{R}^2$. The invariants are used to determine transitions between modes (discussed later). Each mode has a reset function (assigning of constant values (e_1,e_2)), which reflects the effect of control operation on the continuous variables "whenever" or "wherever" the L1STHA first enters a mode. In each mode the trajectory is a linear 1-space dimensional PDE, with different values of A,B,C, and u. L1STHA is applied in Example 1 as follows:

Example 2. **Infusion Pump:** *The L1STHA model for the infusion pump is shown in Figure 6. The L1STHA model has three modes in the set $\mathbb{L}$ - a) correction bolus mode, b) braking mode, and c) basal infusion mode. The L1STHA is initially at the basal infusion mode l_0. The continuous variable for the L1STHA model of infusion pump is the blood glucose concentration in the blood, which varies over space and time. The L1STHA is in: a) the basal infusion mode if the blood glucose concentration is within 20 mg/dl and 120 mg/dl with infusion rate I_0, b) the correction bolus mode l_1, if the blood glucose concentration is greater than 120 mg/dl with infusion rate ($I_b + I_0$) and c) the braking mode l_2, if the blood glucose concentration is below 20 mg/dl with infusion rate I_0/f. Thus, the Inv set consists of the mappings $\{Inv(l_0) = ([20/k, 120/k])\}$, $\{Inv(l_1) = ([120/k, \infty])\}$, and $\{Inv(l_2) = ([0, 20/k])\}$. Here k is a constant factor that converts blood glucose concentration to insulin concentration. Such linear relationship is suggested by the Bergman Minimal Model [1]. It can be seen that the invariant mapping Inv satisfies the conditions in Definition 1. In each mode, the reset function Re represents the decision of the control algorithm to increase or decrease the infusion rate whenever the L1STHA enters the mode. The A,B,C, and u values can be derived from the diffusion Equation 1 and the trajectory Definition 2 as $A = 1$, $B = D$, $C = -\Gamma - \lambda$, and $u = \Gamma d_b(t)$.*

With this definition of L1STHA we can also characterize aggregate effects as shown in the following example.

Example 3. **Multi-channel Infusion with aggregate effects:** *Infusion pumps used in chemotherapy [16], often have multiple channels of infusion leading to aggregate effects of drugs. If we consider a region of the body at a fairly large distance from the site of infusion of a drug, the concentration decreases to negligible amounts (below a low threshold) at a given time according to Equation 1. However, over time the concentration at that region may increase to such an extent that it cannot be ignored. In case of a two channel infusion this phenomenon can happen for both the drugs at a given region. In that case the effective concentration of drug is a non-trivial combination of the dynamics of the individual drugs. Specifically, the drug concentration also follows Equation 1 but with modified parameters and conditions. To capture this aggregate effects, we first consider L1STHA models similar to Example 2, for the individual drugs with concentration d_1 and d_2. The L1STHA of the mutli-channel pump has a mode set which is the Cartesian product of the mode sets of the individual L1STHA models. If we consider that the low threshold is 20/k mg/dl, then aggregate effects can only occur when $(d_1, d_2) \in ([20, \infty], [20, \infty])$ i.e., when the individual L1STHA models are either in basal or correction bolus modes. Hence of the nine possible modes that can occur due to the Cartesian product only four are aggregate effect modes (Figure 7). The aggregate effect can be modeled by introducing a new variable d_3 to all the modes. In modes without aggregate effects $d_3 = 0$, while in modes with aggregate effects d_3 follows a new PDE with new parameters D_3, Γ_3, λ_3, and d_{B3} as suggested by [16]. The transition to the aggregate effect modes can only occur if both $d_1 > 20/k$ and $d_2 > 20/k$. The equation expressing the aggregate effect has to be specified to the L1STHA, its execution will deter-*

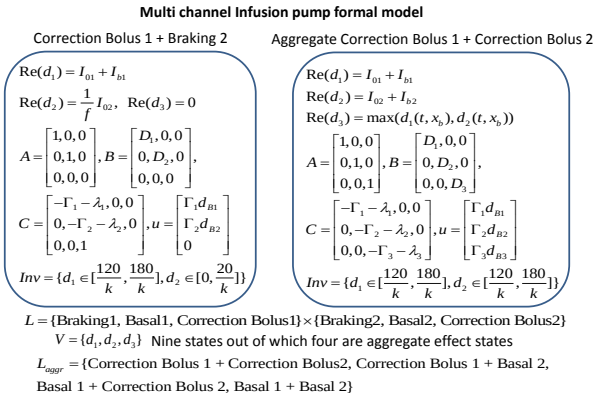

Figure 7: The normal and aggregate effect modes in the L1STHA model of multi-channel infusion pumps.

mine when and where the aggregate effect occurs and with what intensity. Note that this condition is spatio-temporal in nature and unlike spatial networks of hybrid automata, impose no restriction on the space or time at which aggregate effects can occur.

The analysis of L1STHA models requires solution to the differential equations, which depend on initial and boundary conditions and govern mode transitions.

4. L1STHA EXECUTION MODEL

The definition of L1STHA is not complete and useful unless we define its execution and discrete transition. Figure 1 shows a simulation of the multi-channel infusion pump L1STHA following the execution model to be discussed in this section. The figure shows three modes of operation of the infusion pump: *basal*, *correction* and *braking*. The plots show the trajectory, solution of Equation 1, with respect to space at given times (0s, 100s, 500s, and 1000s). It is to be noted that we define the L1STHA execution model with a *time bound* T and a *space bound* S.

Definition 3. *Initial State: The initial state of a L1STHA at a given spatial coordinate $x \in S$, is $\{l_0, \boldsymbol{v_0}\}$ such that continuous state $\boldsymbol{v_{0,x}} \in Inv_0$ and $\eta(0, x) = \boldsymbol{v_{0,x}}$.*

As shown in the graph with time = 0s in Figure 1, initial state at the space point x = 15mm is the value of the glucose concentration, which is 10 mg/dl.

Definition 4. *Initial Configuration: The initial configuration of a L1STHA is the function $\eta(0, .) : x \to \mathbb{R}^n$, and $x \in S$.*

The initial configuration is the spatial variation of the glucose concentration at time t = 0s as shown by the thick gray line in Fig. 1.

Definition 5. *Mode Boundary: The mode boundary for any mode $l \in \mathbb{L}$ at a time t is the boundary of a spatial region s_l such that $\eta(t, x) \in Inv_l$, $\forall x \in s_l$.*

As time progresses the L1STHA is at different modes at different spatial regions. If we consider the time t = 500s, then the spatial regions from 0mm to 9 mm and 42mm to 50mm are in braking mode, while 9mm to 42mm is in basal infusion mode. Hence, each mode has a spatial boundary at a given time called mode boundary while s_l for a mode l is the spatial region in which the L1STHA is in mode l. Note that the mode boundary shifts as time progresses.

Definition 6. *Boundary State: The boundary state of a mode l_0 at a given time $t \in [0, T]$, is a continuous state $\boldsymbol{v_{t,x_0}} \in Inv_0$ such that $\eta(t, x_0) = \boldsymbol{v_{t,x_0}}$, where x_0 is in the mode boundary of l_0.*

The value of the glucose concentration at the mode boundary at a given time is the boundary state at that time. For t = 500s the boundary state of the aggregate basal mode is with glucose concentration 78.4 mg/dl.

Definition 7. *Temporal Execution: A temporal execution α_t of a L1STHA at a given spatial coordinate $x \in S$ from an initial state $(l_0, \boldsymbol{v_{0,x}}) \in \mathbb{L} \times \mathbb{R}^n$, is a concatenation of trajectories at x, $\alpha_t = \eta_0|_x \eta_1|_x \ldots$ where:*

- $\eta_0(0, x) = \boldsymbol{v_{0,x}}$,
- $\eta_k(0, x) = Re(\eta_{k-1}(\eta_{k-1}|_x.dur, x))$, *and*
- $\alpha_t.dur = \sum_i \eta_i|_x.dur$,

where η_k is the trajectory defined at a mode $l_k \in \mathbb{L}$, $Re(.)$ is the reset function (Definition 1), and $\alpha_t.dur$ represents the duration of the temporal trajectory. $\eta_k|_x$ is the trajectory at a given space point x for a mode l_k.

A temporal execution is the variation of a continuous variable at a given space coordinate over time, as shown by the chain-dot lines in Figure 1. During the execution there can be transitions to different modes with changes in the dynamic equations. The first condition in the definition states that the execution starts from an initial state, the second condition shows the concatenation, where $\eta_{k-1}|_x.dur$ denotes the time that the trajectory crossed the boundary of the invariant set for mode l_{k-1} at space x.

Definition 8. *Spatial Execution: A spatial execution α_s of a L1STHA at a given time $t \in [0, T]$ from a boundary state $(l_0, \boldsymbol{v_{t,x_0}}) \subset \mathbb{L} \times \mathbb{R}^n$ is a concatenation of trajectories at time t, $\alpha_s = \eta_0|^t \eta_1|^t \ldots$ where:*

- $\eta_0(t, x_0) = \boldsymbol{v_{t,x_0}}$,
- $\eta_k(t, x_0) = Re(\eta_{k-1}(t, \eta_{k-1}|^t.range))$, *and*
- $\alpha_s.range = \bigcup_i \eta_i|^t.range$,

where η_k is the trajectory defined at a mode $l_k \in \mathbb{L}$, and $\alpha_s.range$ represents the range of the spatial execution. $\eta_k|^t$ is the trajectory at a given time t for a mode l_k.

A spatial execution is the variation of the continuous variables over space at a given time as shown by the thick gray lines in the four different graphs. At different times the spatial execution changes (Figure 1).

Definition 9. *Temporal Discrete Transition: A temporal discrete transition at a given space point x from mode l_i to l_j occurs at a continuous state $\boldsymbol{v_{t',x}}$ at a time t', whenever $\boldsymbol{v_{t',x}} \in Inv_i^{\boxdot} \bigcap Inv_j^{\boxdot}$, where $\boldsymbol{v_{t',x}} = \lim_{t \to t'} \boldsymbol{v_{t,x}}$ and $\boldsymbol{v_{t,x}} \in Inv_i^{\boxdot}$ for $t \in [t' - \tau, t']$: $\tau > 0$.*

Temporal transition at a given space coordinate x occurs at a time t' if as the time approaches t', the continuous state approaches the boundary of the invariant set of a mode l_i. The L1STHA transits to the state l_j if the invariant sets are connected and the continuous state approaches the intersection of the boundaries of the invariant sets of l_i and l_j. An example is shown in Figure 1, where the glucose concentration crosses invariant set boundaries from basal to correction bolus mode as time progresses (chain and dot line).

Definition 10. *Spatial Discrete Transition: A spatial discrete transition at a give time t from the mode l_i to l_j occurs at a continuous state $\boldsymbol{v}_{t,x'}$ at a spatial coordinate x', wherever $\boldsymbol{v}_{t,x'} \in Inv_i^{\square} \bigcap Inv_j^{\square}$, where $\boldsymbol{v}_{t,x'} = \lim_{x \to x'} \boldsymbol{v}_{t,x'}$ and $\boldsymbol{v}_{t,x'} \in Inv_i^{\square}$ for $x \in [x - s, x]$ for some $s > 0$.*

Spatial transition at a given time occurs similar to a temporal transition when the spatial execution crosses invariant set boundaries as shown in the leftmost graphs in Figure 1.

It is to be noted that in this theory we consider every transition to be deterministic and transversal. A deterministic transition means that at any time or space point the L1STHA from a given mode can only transit to a unique mode. This also guarantees that at a fixed time and space point the L1STHA is at an unique mode. Further, a transversal transition is assumed. This means that if there is a transition from Inv_0 to Inv_1 at state $\boldsymbol{v}_{t,x}$, then the dot product of the time or space differential of the continuous variables with the outward normal unit vector on $Inv_0^{\square}$ at $\boldsymbol{v}_{t,x}$ is greater than zero [7]. A mathematically rigorous definition is given in Appendix [3]. This assumption prevents zeno behavior [18], where in a very small amount of time there are infinite transitions. A L1STHA, where every transition is deterministic and transversal is called a Deterministic Transversal L1STHA (DTL1STHA).

4.1 Defining the ϵ reach set of a DTL1STHA

The *reach set* of DTL1STHA from a given initial configuration V_0 is the set of continuous states reached by a spatial or temporal execution originating from the continuous state of any initial state in the configuration V_0 at some time t and some spatial coordinate x. If we bound the time and space of the analysis to T and S, then space and time bound reach set is denoted by $R_{TS}(V_0)$. The reachability analysis of DTL1STHA involves determining the ϵ approximation of this reach set, which is defined as follows.

Definition 11. *ϵ reach set of DTL1STHA: Given an $\epsilon > 0$ a set of continuous states M of a DTL1STHA starting from an initial configuration $V_0 \subset \mathbb{R}^n$ over a time duration of T and within a spatial region S, is called an ϵ reach set if:*

- *$R_{TS}(V_0) \subseteq M$, and $d_H(R_{TS}(V_0), M) \leq \epsilon$.*

Thus, in simpler terms for any continuous state $\boldsymbol{v}$ in the $R_{TS}(V_0)$, there exists a point in M such that it is greater than $\boldsymbol{v}$ by at most ϵ. As we will discuss, the reachability analysis algorithm computes an over approximation of the ϵ reach set for a given $\delta > 0$ neighborhood of the initial configuration V_0. To this effect, it is required to solve the PDE as discussed in the next section.

4.2 Solving the PDE

For the ease of representation let us consider that there is only one continuous variable i.e., n = 1. In that case, the matrices A, B, and C and the vector u will reduce to singleton a, b, c, and u. In order to compute the reachable states of the DTL1STHA, we need to have the capability to solve the PDE 2. The initial conditions for solving this PDE is given by - $\eta(0, x) = I_b \forall x \in S$, where I_b is a constant. It is to be noted that we will solve this PDE as a free boundary problem. This assumption comes from the observations in various medical device examples. In Example 2, as time progresses the effect of insulin spreads further away from the site of infusion. At any given position the insulin concentration increases with time. Thus, there exists no space point where the insulin concentration stays steady. Hence, we consider this problem as a free boundary problem, where the boundary at which the minimum insulin concentration is observed moves away from the site of infusion with some velocity. However, at the site of the infusion the flow rate of drug is constant. This condition is given by - $-b\frac{\partial \eta(t,x)}{\partial x} = I_0$, where I_0 is a constant.

Given these initial and boundary conditions we solve the free boundary PDE. We first take a Laplace transform of the PDE on the time domain. This results in an ordinary 2nd order differential equation, which can be solved using standard methods. Inverse Laplace transform is then applied to the solved equation to obtain the final solution of the PDE. The inverse Laplace transform does not have a closed form solution. At best we could represent it using error functions as shown in Equation 3.

$$\begin{aligned}\eta(t,x) &= I_b + \frac{I_0}{(\sqrt{4bc})}[e^{-\sqrt{\frac{c}{b}}x} erfc(\frac{x}{\sqrt{4\frac{b}{a}t}} - \sqrt{\frac{c}{a}t}) \\ &\quad -e^{\sqrt{\frac{c}{b}}x} erfc(\frac{x}{\sqrt{4\frac{b}{a}t}} + \sqrt{\frac{c}{a}t})]\end{aligned} \tag{3}$$

An elaborate solution with matrices is provided in Appendix [3]. In this theory, we compute the value of $\eta(t, x)$ from Equation 3 using error function tables. With this assumption let us determine the ϵ reach set for a single mode DTL1STHA.

5. TIME AND SPACE BOUNDED ϵ REACH SET OF A SINGLE MODE DTL1STHA

We want to find out the ϵ reach set of a single mode DTL1STHA from an initial configuration V_0 by sampling the time with interval h_x at a space point x and the space with grid size h_t at a given time. To bound the error of discretization to ϵ, we need to find h_t and h_x such that from any sampled state $\eta(t, x)$ the trajectory remains within the ϵ neighborhood, $B_\epsilon(\eta(t, x))$, before the next sampled state is considered.

To ensure this the following two conditions must satisfy,

$$\max_{\tau \in [0,h_x]} (\eta(t + \tau, x) - \eta(t, x)) < \epsilon, \tag{4}$$

$$\max_{s \in [0,h_t]} (\eta(t, x + s) - \eta(t, x)) < \epsilon \tag{5}$$

we take the maximum entry if η is a vector. We first consider the time differential of η and use Equation 2 to obtain -

$$\max_{\tau \in [t,t+h_x]} (\frac{\partial \eta(\tau,x)}{\partial \tau}) \leq \frac{b}{a} \max_{\tau \in [t,t+h_x]} (\frac{\partial^2 \eta(\tau,x)}{\partial x^2}) + (\frac{c}{a}) \max_{\tau \in [t,t+h_x]} (\eta(\tau, x)) + (\frac{u}{a}) \tag{6}$$

We need to express $\max_{\tau \in [t,t+h_x]}(\frac{\partial^2 \eta(\tau,x)}{\partial x^2})$ and the term $\max_{\tau \in [t,t+h_x]}(\eta(\tau, x))$ in terms of the mode invariants so that they can have fixed values during a trajectory evaluation at any space or time. The absolute max of the term $\max_{\tau \in [t,t+h_x]}(\eta(\tau, x))$ in a given mode is the maximum value that η can take while staying in the mode. This means that $\max_{\tau \in [t,t+h_x]}(\eta(\tau, x)) = \eta^{max} = \max_{t \in T, x \in S} Inv_0$.

To find the maximum of the double differentiation we express it using the standard Taylor series expansion as follows:

$$\frac{\partial^2 \eta(t,x)}{\partial x^2} = \frac{2\eta(x) - \eta(x + h_t) - \eta(x - h_t)}{h_t^2} \tag{7}$$

The maximum value of this term is $\frac{2(\eta^{max} - \eta^{min})}{h_t^2}$, where $\eta^{min} = \min_{t \in T, x \in S} Inv_0$. Similarly from Equation 5 we can derive the maximum value of the spatial variation. Substituting Equations 7 and 6 in Equations 4 and 5 we get Equations 8 and 9, which can be simultaneously solved to get h_x and h_t (full derivation shown in Appendix [3]).

$$\frac{b}{a}\frac{2(\eta^{max} - \eta^{min})}{h_t^2} h_x + \frac{c}{a}\eta^{max} h_x + \frac{u}{a} h_x = \epsilon/2 \tag{8}$$

$$\frac{\frac{a}{b}(\eta^{max}-\eta^{min})h_t^2}{2h_x} - \frac{c}{b}\eta^{min}h_t^2/2 - \frac{u}{b}h_t^2/2 = \epsilon/2 \quad (9)$$

Note that we select h_t and h_x such that the trajectory between sampled states is within $\epsilon/2$ neighborhood, which is a stricter assumption than required. This will help us to ensure that the approximate reach set computed from a δ neighborhood of the initial configuration is within an ϵ boundary of the actual reach set. We now define a methodology to compute the ϵ reach set from an initial configuration V_0.

Lemma 5.1. *A bounded $\epsilon > 0$ reach set $R_{TS}(V_0, \epsilon)$ of a one mode DTL1STHA from an initial configuration $V_0 \subset Inv_0$ from time $[0, T]$ and within space S can be determined as -*

$$R_{TS}(V_0, \epsilon) = \bigcup_{k=0}^{\beta-1} \bigcup_{j=-(\kappa-1)}^{\kappa-1} B_\epsilon(\eta(kh_x, (x_0 + jh_t))), \quad (10)$$

where $\beta = T/h_x$, $\kappa = S/(2h_t)$, x_0 is the origin of the coordinate system, and h_x and h_t are derived from Equations 8 and 9. The set has the following properties:

i) $\lim_{\epsilon \to 0} R_{TS}(V_0, \epsilon) = R_{TS}(V_0)$ *and*

ii) It contains the $\epsilon/2$ neighborhood of $R_{TS}(V_0)$ i.e,

$$\bigcup_{z \in R_{TS}(V_0)} B_{\epsilon/2}(z) \subseteq R_{TS}(V_0, \epsilon) \quad . \quad (11)$$

The lemma states that to compute the ϵ reach set, we first consider a discrete space point $x_0 + jh_t$ at a discrete time kh_x. The trajectory for the single mode DTL1STHA is then computed at this space point and time step. It is then over-approximated by constructing a hypercube of ϵ side length with the state $\eta(kh_x, (x_0 + jh_t))$ at its center. This over approximation is done for each time and space point, up to the maximum time T and the space bound S. The ϵ reach set is then the union of all the resulting hypercubes. A reach set computed in such a method has two properties: a) as ϵ tends to zero the set approaches the actual reach set from the initial configuration V_0, and b) the set encompasses the $\epsilon/2$ neighborhood of the actual reach set. Proofs of these claims are provided in Appendix [3].

5.1 ϵ reach set for a set of initial configurations

We now extend Lemma 5.1 to consider a set of initial configurations which are within a δ neighborhood of the initial configuration V_0, $B_\delta(V_0)$. We first show that there exists a $\delta > 0$ such that the reach set starting from the set $B_\delta(V_0)$ is contained in the ϵ reach set computed using Lemma 5.1.

Lemma 5.2. *Given an $\epsilon > 0$, an initial configuration V_0 of a DTL1STHA, a time interval [0,T] and a spatial region S, there exists a $\delta > 0$ such that,*

$$R_{TS}(B_\delta(V_0)) \subseteq R_{TS}(V_0, \epsilon), \quad (12)$$

where $R_{TS}(B_\delta(V_0))$ is the reachable set starting from the set $B_\delta(V_0)$ up to a time T and space boundary S. In particular we will show that, $\delta = \epsilon/2H$ for $H = (1 + \frac{2}{\sqrt{4bc}})$.

Lemma proof is in Appendix [3]. Next we extend Lemma 5.2 to consider γ approximation of the reachable states starting from a δ neighborhood of the initial configuration, $R_{TS}(B_\delta(V_0), \gamma)$. We can find a γ such that $R_{TS}(B_\delta(V_0), \gamma)$ is also within $R_{TS}(V_0, \epsilon)$.

Lemma 5.3. *Given an $\epsilon > 0$, an initial configuration V_0 of a single mode DTL1STHA, a time interval T, and a spatial region S, there exists a $\delta > 0$ and a $\gamma > 0$ such that, $R_{TS}(B_\delta(V_0), \gamma) \subseteq R_{TS}(V_0, \epsilon)$. In particular we show that, $R_{TS}(V_0) \subseteq R_{TS}(B_{\epsilon/(4H)}(V_0), \epsilon/4) \subseteq R_{TS}(V_0, \epsilon)$.*

Lemma proof in Appendix [3]. Thus, if we compute the ϵ reach set following Lemma 5.1, then we also over approximate the reach set from a $\epsilon/(4H)$ neighborhood of the initial configuration V_0 by $\epsilon/4$. In our reachability analysis algorithm we will thus use Lemma 5.1 to compute the reach set and we will get a $\gamma = \epsilon/4$ approximation of the reach set from a $\delta = \epsilon/(4H)$ neighborhood of initial configuration V_0.

5.2 Determining exit condition from an invariant set

Since we are over approximating the reach set at any time and space point by discretization, there can be a chance to ignore the exit point of a trajectory from an invariant set. We show that, for a given $\epsilon > 0$, we can always find a $\delta > 0$, h_x and h_t such that no such exit conditions are ignored.

Lemma 5.4. *Given a DTL1STHA, an initial state $\{l_0, \boldsymbol{v_0}\}$, if the trajectory $\eta(t, x)$ exits the invariant set Inv_0 at time $t_1 < T$ and space point $x_1 \in S$, then for all small enough $\delta > 0$ there exists some h_x and h_t such that, $B_\delta(\eta(jh_x, kh_t))$ is outside Inv_0 for some j and k.*

Proof: Let us consider that $\boldsymbol{v_{t_1,x_1}}$ lies at the boundary $Inv_0^{\square}$ of DTL1STHA. Since the solution of the PDE in Equation 2 is continuous over space and time, there exists a boundary $B_r(\boldsymbol{v_{t_1,x_1}})$, $0 < r < \epsilon$ such that for every point in the neighborhood the deterministic transverse temporal or spatial transition condition holds (Definitions 9 and 10). This means that for all states $\boldsymbol{v} \in B_r(\boldsymbol{v_{t_1,x_1}}) \cap Inv_0^{\square}$, $\dot{\boldsymbol{v}} \bigodot \hat{n} > 0$ and $\frac{\partial^2 \boldsymbol{v}}{\partial x^2} \bigodot \hat{n} > 0$, i.e., the dot product of the rate of change of the state over time or space with the unit normal vector $\hat{n}$ at $\boldsymbol{v}$ in the direction outward of Inv_0 is greater than 0. This indicates that there is a tendency of the state to exit Inv_0 at any subsequent space or time increment. To capture this transition, we need to find a h_x and h_t such that a state in $B_r(\boldsymbol{v_{t_1,x_1}})$ is captured in the reach set computation. Let us consider h_x and h_t from Equations 8 and 9 by replacing the right hand side by $\epsilon/4G$, where $G = \epsilon/r > 1$. From Lemma 5.3 we know that the reach set computation method in Lemma 5.1 will result in the $\epsilon/(8G) = r/8$ approximation of the reach set from a $\epsilon/(8GH) = r/(8H)$ neighborhood of $\boldsymbol{v_{t_1,x_1}}$. Hence, Lemma 5.1 will find a point in $B_r(\boldsymbol{v_{t_1,x_1}})$ and at some $t = jh_x$ and $x = kh_t$, $B_\delta(\boldsymbol{v_{t,x}})$ exits Inv_0.

Given this guarantee that there exists an h_t and h_x such that all transitions are captured by the reach set computation technique of Lemma 5.1 let us device an algorithm for computing the reach set for a single mode DTL1STHA.

5.3 Algorithm for ϵ reach set of a single mode DTL1STHA

The inputs to the algorithm are: i) a single mode DTL1STHA $D\Sigma$, ii) an initial configuration V_0, iii) an invariant set Inv, iii) the parameter ϵ, iv) a parameter α, v) the time bound T, and vi) the space bound S. The algorithm outputs the set of reachable states, the time and space at which DTL1STHA exits the invariant set Inv and γ and δ parameters of Lemma 5.3.

Algorithm 1 first considers a δ boundary of the initial configuration V_0 and computes the reachable states that are either within Inv or are in Inv^C, where Inv^C is the complement set of Inv,. Note that if a set of reachable state intersects with both Inv and Inv^C then it means that at the given space and time DTL1STHA is in two different modes which is not feasible. However, since the computed reachable states are an ϵ approximation such infeasible cases may

occur during execution of the algorithm. But from Lemma 5.4 we know that there exists a δ and a γ for which such infeasible cases do not occur. Hence, if such an infeasible case occurs the algorithm updates the values of γ and δ such that the reach set is either within Inv or Inv^C.

The algorithm increments time in steps of h_x. For time $t = 0$, the algorithm starts with $B_\delta(V_0)$ and computes its image for different space points starting from x = 0, to x = S in steps of h_t. After each computation of image the algorithm checks whether the image is within Inv. If it is not within Inv there can be two cases: a) it is within Inv^C, in that case the algorithm terminates and the reachable set is the image of $B_\delta(V_0)$ at the current time and space location, and b) it is not within Inv^C, in that case the algorithm returns an empty set as the reachable state, updates the parameters γ and δ by multiplying the previous values with α and restarts the computation at t = 0 and x = 0. If the image is within Inv for all $x \in S$, the algorithm then increments t by h_x. It then computes the image of $B_\delta(V_0)$ at x = 0, and t = t + h_x. The computation of images with respect to spatial dimension is again repeated till $t < T$. The pseudocode of Algorithm 1 is available in Appendix [3].

Algorithm 1 (R_{set},t,x) = CalcReach(DΣ,Inv,V_0,T,S,ϵ,α)

```
 1: while reach set is empty do
 2:     Compute the h_x and h_t from Equations 8 and 9;
 3:     while t ≤ T do
 4:         Start from x = 0; Set P_c = V_0;
 5:         Set the image of P_C, D_{t,x}(P_c) = P_c;
 6:         while x ≤ S do
 7:             Increment space by h_t
 8:             D_{t,x}(P_c) = Compute the image at x from D_{t,x-h_t}(P_c);
 9:             if (Current image is neither in Inv nor Inv^C) then
10:                 return null set, restart computation with γ = αγ
                      and δ = αδ;
11:             else if (Current image in Inv^C or in Inv) then
12:                 R_set = R_set ⋃ D_{t,x}(P_c);
13:                 return (R_set, D_{t,x}(P_c), t, x, γ, δ);
14:             end if
15:         end while
16:         Increment time by h_x
17:         Repeat the image computation technique and check for
              invariant crossing (lines 9 - 14) for the next time;
18:     end while
19: end while
```

Image Computation: Let us consider that the current reach set is $D_{t,x}(P_c)$ (Algorithm 1) starting from a δ neighborhood of P_c, $B_\delta(P_c)$, computed as discussed in Section 2. The set $D_{t+h_x,x}(P_c)$ or $D_{t,x+h_t}(P_c)$ is computed from the set $D_{tx}(P_c)$ by exploiting the polyhedral structure of $B_\delta(P_c)$. The algorithm first considers the set of vertices $Vert(t, x)$ of the image $D_{tx}(P_c)$. Then for each vertex $vert(t, x) \in Vert(t, x)$, $vert(t + h_x, x)$ or $vert(t, x + h_t)$ is computed using the solution of the dynamic Equation 3. Hence the set $Vert(t + h_x, x)$ or $Vert(t, x + h_t)$ can be computed from $Vert(t, x)$. The image $D_{t+h_x,x}(P_c)$ or $D_{t,x+h_t}(P_c)$ can then be computed from the convex hull of the set of vertices in $Vert(t + h_x, x)$ or $Vert(t, x + h_t)$, respectively. To obtain a γ approximation of the reachable states we have to take γ boundaries of the convex hull. Instead we can take γ boundaries of the vertices in $Vert(t + h_x, x)$ or $Vert(t, x + h_t)$ and then take the convex hull of the vertices of the boundaries. Lemma 5.5 shows that the two approaches produce identical results.

LEMMA 5.5. *If $\mathbb{H}$ is the convex hull of the set $\bigcup_{v \in V} Vert(B_\gamma(\boldsymbol{v}))$ then it is the closed γ neighborhood of the convex hull of the set V.*

The proof of Lemma 5.5 is given in Appendix [3].

We will now show that Algorithm 1 terminates in a finite number of steps and gives the $R_{TS}(B_\delta(V_0), \gamma)$ as output.

THEOREM 5.6. *Given the input ($D\Sigma$, Inv, V_0, T, S, ϵ, α), Algorithm 1 terminates in a finite number of steps and outputs the set $R_{TS}(B_\delta(V_0), \gamma)$.*

Proof: The algorithm terminates whenever the current time is greater than T and the current space point is outside S. Since the time and space points increment in fixed steps the algorithm terminates in a finite number of steps for these conditions. If the algorithm terminates at time T and at space limit S, then from Lemma 5.3, there exists a δ_1 and a γ_1 such that $R_{TS}(B_\delta(V_0), \gamma)$ is a subset of R_{set}, since the reach set is computed following Lemma 5.1.

However, if the ϵ reach set intersects both Inv and Inv^C then the algorithm restarts computation with reduced δ and γ. This has the potential of running into infinite loops with δ and γ reducing to arbitrarily small value.

If there exists a time $t_1 < T$ or a space point $x_1 \in S$ such that, the trajectory moves to Inv^C then from Lemma 5.4 there exists a $\delta_1 > 0$ such that $B_{\delta_1}(\eta(jh_x, kh_t)) \subset Inv^C$ for some j and k. Hence the algorithm will stop at $t_a = jh_x$ and at space location $x_a = kh_t$. For this t_a and x_a, from Lemma 5.3 there exists a γ_2 and δ_2 such that $R_{t_a x_a}(B_{\delta_2}(V_0), \gamma_2)$ is a subset of R_{set} returned by Algorithm 1.

Now if $t_f = min(t_a, T)$ and $x_f = min(x_a, S)$, and $\delta = min(\delta_1, \delta_2)$ and $\gamma = min(\gamma_1, \gamma_2)$, the algorithm does terminate and outputs the ϵ reach set.

6. ALGORITHM FOR ϵ REACH SET OF MULTIPLE MODE DTL1STHA

Lemma 5.4, the resulting Algorithm 1, along with the assumption that the transitions are all deterministic and transversal can be used to augment Algorithm 1 to use it for multi-mode DTL1STHA. Using Algorithm 1 an updated reachable set can be obtained on transition, which can intersect multiple invariant sets of different modes. However, a DTL1STHA can only transit to one mode. We prove subsequently, that by proper adjustment of δ and γ parameters it is possible to obtain an ϵ reach set using Algorithm 1 such that the reach set intersects only a single invariant set on transition.

In this regard, we need to find the time t and the spatial location x at which a transition takes place and the value of the trajectory $\eta(t, x)$ at the transition point. In the next lemma we show that it is possible to approximate the time of transition and the value of the trajectory.

LEMMA 6.1. *Given that at $t < T$ and $x < S$ the trajectory $\eta(t, x) \in Inv_0^{\boxminus}$ satisfies deterministic transversal transition conditions, there exists a $\delta > 0$ such that $B_{2\delta}(\eta(t, x)) \subset Inv_0 \cup Inv_1$ and also there exists Δ_t and Δ_x such that:*

1. *$\eta(\tau, x) \in Inv_1^{\boxminus}$ for $\tau \in [t, t + \Delta_t]$ and $x \in [x, x + \Delta_x]$*
2. *$\bigcup_{\eta \in F} R_{[t,t+\Delta_t],[x,x+\Delta_x]}(\eta) \subset Inv_1^{\boxminus}$, where, $F = B_\delta(\eta(t, x)) \cap Inv_0^{\boxminus} \cap Inv_1^{\boxminus}$.*

Lemma 6.1 gives a methodology to over-approximate $\eta(t, x)$ if it satisfies the deterministic transversal transition condition. It takes a similar approach as Lemma 5.4 to estimate the state at the time t and space x of transition. It approximates the state $\eta(t, x)$ by intersecting δ neighborhood of $\eta(t, x)$ with the Inv_0 and Inv_1. The lemma further states that if the reach set from $B_\delta(\eta(t, x)) \cap Inv_0^{\boxminus} \cap Inv_1^{\boxminus}$ is computed for an appropriate Δ_t time interval and Δ_x space interval then the reach set lies within Inv_1. The proof of the lemma is given in Appendix [3].

Given an appropriate $\delta > 0$ from Lemma 6.1, an appropriate δ_0 neighborhood of V_0 can also be found as guaranteed by Lemma 5.4, which ensures that the ϵ reach set computed following Lemma 5.1 lies within Inv_1.

LEMMA 6.2. *If the reachable states during a deterministic transverse transition at time t_1 and space location x_1 is within a δ neighborhood of $\eta(t_1, x_1)$, there exists a δ_0 such that $D_{t_1,x_1}(B_{\delta_0}(V_0)) \subset B_\delta(\eta(t_1, x_1))$ and $D_{t_1,x_1}(B_{\delta_0}(V_0)) \cap Inv_0 \cap Inv_1$ is an over-approximation of $\eta(t_1, x_1)$, where $D_{t_1,x_1}(B_{\delta_0}(V_0))$ is the reach set computed by Algorithm 1.*

Lemma 6.2 states that if for $\delta > 0$ a transition is detected by computing the reach set using Lemma 5.1, as in Algorithm 1, a corresponding δ_0 boundary of the initial configuration can be found. Lemma 6.2 can then be used in arithmetic induction to prove that if the reach set for any transition k is computed following Lemma 6.1 then a δ_0 can always be found such that Algorithm 1 can capture it. The associated lemma and its proof is given in Appendix [3]. This allows us to use Algorithm 1 with some changes for computing reach set of multi-mode DTL1STHA. The changes include a $Transition$ function that can output the new mode, and the new initial configuration to be used for each transition. Further, if the $Transition$ algorithm fails to find the new mode, then Lemma 6.2 allows it to reduce the value of δ_0 and restart the overall computation with an increased assurance of finding the new mode in the next iteration.

The reachability analysis algorithm for multi-mode DTL1STHA takes as input: a) DTL1STHA $D\Sigma$, b) initial mode l_0, c) the initial configuration V_0, d) the time bound T, e) the space bound S, f) the approximation parameter ϵ, and g) the parameter α used to adjust δ and γ. It outputs the reach set R_{set}. The algorithm has the following simple steps:

a) From an initial state it starts computing the reach set following Algorithm 1. On computation of reach set the algorithm checks whether it is within the invariant set of the current mode.

b) Whenever a reach set computed is not within the invariant set, the $Transition$ algorithm is called. $Transition$ algorithm checks whether the reach set is within a unique Inv_i. If it is within an unique Inv_i the algorithm returns the reach set, the new mode, and the new initial configuration for the mode. If not then the algorithm recalculates δ and γ and returns an empty reach set.

c) If $Transition$ returns a valid transition then the algorithm resumes its computation from the new initial configuration and the new mode and the current time and space point.

d) Else it restarts the computation at time t = 0 and space x = 0 with the new values of δ and γ. The full pseudocode for the $Transition$ algorithm is available in Appendix [3].

The reachability algorithm for multi-mode DTL1STHA is given in Appendix [3]. The multi-mode reachability analysis algorithm has the same exit condition as Algorithm 1. In Theorem 5.6 we have already proved that Algorithm 1 terminates in finite time and outputs ϵ reach set. Hence, Theorem 5.6 is also a proof that the multi-mode reachability analysis algorithm terminates in finite time. The number of times the multi-mode reachability analysis algorithm executes the $Transition$ algorithm is $O(\frac{T}{h_x}\frac{S}{h_t} log_\alpha(\delta))$, since there are two loops for time and space and each can be reiterated at the max $log_\alpha(\delta)$ times. This is because the algorithm starts with a high α and progresses by dividing α by a factor d.

7. INFUSION PUMP SAFETY VERIFICATION

The infusion pump DTL1STHA models are as described in Examples 2 and 3. In this paper, we only show the reachability analysis of the multi-channel infusion case.

To perform the safety analysis of the infusion pump using the reachability analysis algorithm, we considered five control parameters of the pump: 1) *control input delay*, the time taken to transmit the control input from the controller to the infusion pump, 2) *set point*, the drug concentration that is required to be maintained (d_B), 3) *basal value*, the infusion rate when the pump is in basal state I_b, 4) *sampling interval*, the time interval at which the controller samples the spatio-temporal model, and 5) *infusion increment*, the maximum amount by which the controller can increment the infusion rate I_0. A setting of the infusion pump is a tuple with numeric values assigned to the above five parameters. The reachability analysis as discussed in Section 6 was performed on every setting of the model. The safety threshold of glucose concentration was set at 180 mg/dl. Whenever the reach set in any setting has a state with glucose concentration greater than 180mg/dl it is considered unsafe. The safe settings of the infusion pump determined using the above-mentioned method is shown in Figure 8. Figure 8 shows the 2-D projection of the 5-D graph generated from the analysis procedure. The control input delay was varied from 10 to 100 seconds, the infusion increment step from 100 to 1000 ug/min, the set point from 100 to 1000 ug/l, the basal value from 600 to 1000 ug/min, and sampling interval from 5 to 25 s. In each of the figure showing any two parameters, the unshaded regions are the unsafe settings, where the glucose concentration exceeded 180 mg/dl. The shaded regions are safe for the assumed values of the control parameters. The graphs are reflective of the control action of the infusion pump. For example, in the left most graph, for low control input delay and low infusion increments we see that the infusion pump is unsafe. This is due to the fact that many infusion increments are made at a high frequency leading to unstable behavior of the pump. We also observe that pump settings with higher control input delays and lower infusion increments are safer. However, if the increments are too high, that also leads to unsafe configurations. Note that we have several islands in the graphs, which are indicative of some optimal control input delay (delay not too high) and infusion increment (increment not too low) pair that leads to safe infusion pumps. The graphs also show hypoglycemia cases with glucose level lower than 40 mg/dl if a configuration fall within the region bounded by dashed lines. A configuration is safe from both hyper and hypo-glycemia if it lies in the region bounded by solid line and does not fall within the region bounded by dashed lines.

8. CONCLUSIONS

In this paper, we propose a novel linear 1-space dimensional spatio-temporal hybrid automata that can capture the spatio-temporal aggregate cyber-physical interactions. We also proposed a time and space bounded reachability analysis technique for DTL1STHA to estimate the reach set with arbitrary estimation error ϵ. The complexity of the algorithm depends on the parameter ϵ and is higher for lower ϵ. We used the DTL1STHA modeling and reachability analysis on infusion pump control systems and showed the safe and unsafe conditions of the pump operation. In future we will extend the reachability analysis algorithm to multi dimensional space.

9. REFERENCES

[1] K. A. Aalborg, K. E. Andersen, and M. Hjbjerre. A Bayesian Approach to Bergman's Minimal Model. In *in: C.M. Bishop, B.J. Frey (Eds.), Proceedings of the Ninth International Workshop on Artificial Intelligence,*, 2003.

[2] H. Alt, J. Blőmer, and H. Wagener. Approximation of convex polygons. In M. Paterson, editor, *Automata, Languages and Programming*, volume 443 of *Lecture Notes in Computer Science*, pages 703–716. Springer Berlin / Heidelberg, 1990.

[3] A. Banerjee and S. K. S. Gupta. Appendix for ICCPS 2013. http://impact.asu.edu/BanerjeeICCPSApp.pdf.

[4] A. Banerjee and S. K. S. Gupta. Your mobility can be injurious to your health: Analyzing pervasive health

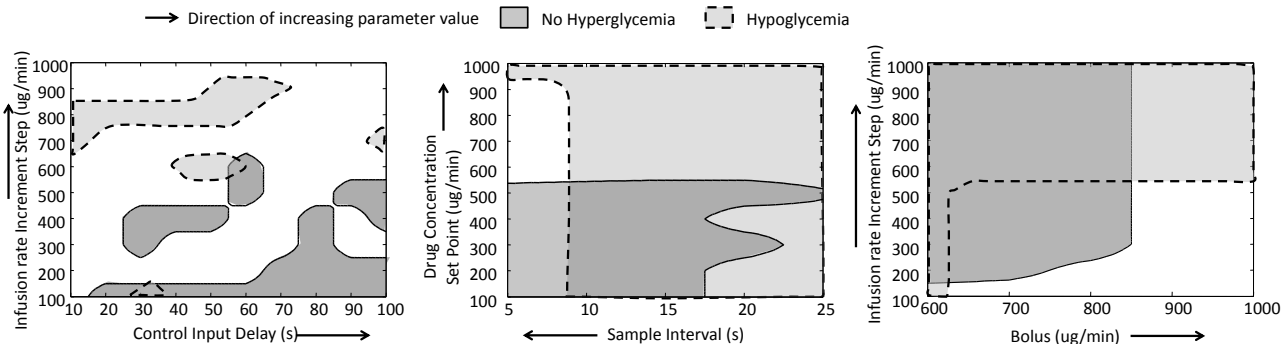

Figure 8: Safe and unsafe initial configurations of the infusion pump.

monitoring systems under dynamic context changes. In *IEEE International Conference on Pervasive Computing and Communications (PerCom),*, pages 39 –47, march 2012.

[5] A. Banerjee, S. Kandula, T. Mukherjee, and S. K. S. Gupta. Band-aide: A tool for cyber-physical oriented analysis and design of body area networks and devices. *ACM Trans. Embed. Comput. Syst.*, 11(S2):49:1–49:29, aug 2012.

[6] E. Bartocci, F. Corradini, M. R. Di Berardini, E. Entcheva, R. Grosu, and S. A. Smolka. Spatial Networks of Hybrid I/O Automata for Modeling Excitable Tissue. *Electronic Notes in Theoretical Computer Science (ENTCS'08)*, 194(3):51–67.

[7] D. Bleecker and G. Csordas. *Basic Partial Differential Equations*. Chapman and Hall, 1995.

[8] M. C. Bujorianu. An integrated specification logic for cyber-physical systems. In *ICECCS 2009*, pages 291–300.

[9] C. Cortazar, M. Elgueta, and J. D. Rossi. A nonlocal diffusion equation whose solutions develop a free boundary. *Annales Henri Poincare*, 6:269–281, 2005.

[10] G. Frehse. Phaver: Algorithmic verification of hybrid systems past hytech. In *HSCC*, pages 258–273, 2005.

[11] R. Ghosh and C. Tomlin. A query-based technique for interpreting reachable sets for hybrid automaton models of protein feedback signaling. In *Proceedings of the American Control Conference*, pages 4417–4422, June 2005.

[12] A. Girard and C. Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Proceedings of the 11th international workshop on Hybrid Systems: Computation and Control*, HSCC '08, pages 215–228, Berlin, Heidelberg, 2008. Springer-Verlag.

[13] S. Graf and W. Zhang, editors. *Automated Technology for Verification and Analysis, 4th International Symposium, ATVA 2006, Beijing, China, October 23-26, 2006*, volume 4218 of *Lecture Notes in Computer Science*. Springer, 2006.

[14] D. G. M. Greenhalgh, M. B. R. Lawless, B. B. Chew, W. A. Crone, M. E. Fein, and T. L. M. Palmieri. Temperature threshold for burn injury: An oximeter safety study. *Journal of Burn Care and Rehabilitation*, 25(5):411–415, 2004.

[15] N. Grégoire and M. Bouillot. Hausdorff distance between convex polygons. http://cgm.cs.mcgill.ca/ godfried/teaching/cg-projects/98/normand/main.html.

[16] T. L. Jackson and H. M. Byrne. A mathematical model to study the effects of drug resistance and vasculature on the response of solid tumors to chemotherapy. *Mathematical Biosciences*, 164(1):17 – 38, 2000.

[17] K.-D. Kim, S. Mitra, and P. R. Kumar. Bounded ϵ-reach set computation of a class of deterministic and transversal linear hybrid automata. *CoRR*, abs/1205.3426, 2012.

[18] A. Lamperski and A. Ames. On the existence of zeno behavior in hybrid systems with non-isolated zeno equilibria. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 2776 –2781, dec. 2008.

[19] J. Lee, S. Bohacek, J. Hespanha, and K. Obraczka. Modeling communication networks with hybrid systems. *Networking, IEEE/ACM Transactions on*, 15(3):630–643, June 2007.

[20] NSF. Program solicitation. http://www.nsf.gov/pubs/2011/nsf11516/nsf11516.htm.

[21] H. H. Pennes. Analysis of tissue and arterial blood temperature in the resting human forearm. In *Journal of Applied Physiology*, volume 1.1, pages 93–122, 1948.

[22] A. Schafer and M. John. Conceptional Modeling and Analysis of Spatio-Temporal Processes in Biomolecular Systems. In S. Link and M. Kirchberg, editors, *Sixth Asia-Pacific Conference on Conceptual Modelling (APCCM 2009)*, volume 96 of *CRPIT*, pages 39–48, Wellington, New Zealand, 2009.

[23] C. Sinem, E. Mustafa, and K. T. John. Lifetime analysis of a sensor network with hybrid automata modelling. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 98–104, New York, NY, USA.

[24] Q. Tang, N. Tummala, S. K. S. Gupta, and L. Schwiebert. Communication scheduling to minimize thermal effects of implanted biosensor networks in homogeneous tissue. *Biomedical Engineering, IEEE Transactions on*, 52(7):1285–1294, July 2005.

[25] Q. Tang, N. Tummala, S. K. S. Gupta, and L. Schwiebert. TARA: thermal-aware routing algorithm for implanted sensor networks. *Lecture Notes in Computer Science*, 3560:206–217, 2005.

[26] The Networking Information Technology Research and Development Program. Different definition of cyber physical systems.

[27] A. Tiwari. Hybridsal relational abstracter. In P. Madhusudan and S. Seshia, editors, *Computer Aided Verification*, volume 7358 of *Lecture Notes in Computer Science*, pages 725–731. Springer Berlin Heidelberg, 2012.

[28] Q. Yong, W. Ying-Jie, and J. Li-Min. Hybrid cellular automata model for railway transportation system and its implementation on GIS. In *Intelligent Vehicles Symposium*, pages 543 – 546, June 2003.