

Modeling and Analyzing Spike Timing Dependent Plasticity with Linear Hybrid Automata

Alina Bey Stefan Leue

Technical Report soft-11-03, Chair for Software Engineering,
University of Konstanz
06 May 2011

Abstract

We propose a model for synaptic plasticity according to the Spike Timing Dependent Plasticity (STDP) theory using Linear Hybrid Automata (LHA). We first present a compositional LHA model in which each component corresponds to some process in STDP. We then abstract this model into a monolithic LHA model in order to enable formal analysis using hybrid model checking. We discuss how the availability of an LHA model as well as its formal analysis using the tool PHAVer can support a better understanding of the dynamics of STDP.

1 Introduction

Advances in the area of system biology have led to a large number of complex biophysical and mathematical models of neuronal activities. An overview can be found in [10]. Following a recent trend in algorithmic systems biology [21], we propose to use formal methods in the modeling of neurological systems aspects. The particular method that we use is that of Linear Hybrid Automata (LHA) [1] and symbolic reachability analysis for LHA as implemented in the hybrid model checking tool PHAVer [7]. It is an objective of this paper to illustrate how this type of formal methods usage can help to document system behavior, to explain experimentally observed behavior, and to predict behavior that has not yet been experimentally explored, in the domain of neurological systems biology.

In our analysis we consider the field of biological systems modeling using the example of an existing phenomenological model of synaptic plasticity, in particular Spike Timing Dependent Plasticity (STDP) [17][18]. Synaptic plasticity goes back to the work of D. Hebb. It postulates the ability of the brain to strengthen connections between neurons whenever neighboring

neurons are simultaneously active [12]. Synaptic plasticity is thought to play a key role in associative learning [12]. We present an LHA model for the exploration and documentation of key system properties. First, exploiting the compositionality of LHAs, we define a linear hybrid automaton for each STDP system component, which we then integrate into one hybrid system model. This compositional modeling approach allows for an easy extension of the model as well as a more comprehensible, modular presentation of the system. In a second step, we abstract the model obtained in the first step further and present a compact model of STDP that is suitable for efficient model checking. We use PHAVer as a model checking tool and demonstrate how formal methods can be used to explore salient properties of an STDP model. We illustrate the modeling power of linear hybrid automata in the field of biological system modeling, in particular their ability to describe systems under different point of views: as a transition system based on a control graph, and as a visualization of the sets of reachable states representing system dynamics.

Related Work Related to our work is the modeling of the action potential (AP) of single neurons described in [27]. The authors present an LHA based model of neuronal action potential. The biological system under investigation in that work is hence the excitability of single neural cells. The action potential (AP) itself is caused by a change in the potential in the cell membrane due to different ion currents flowing through it. An action potential is only fired after a certain threshold value has been reached. Otherwise, the AP returns to its resting potential. This behavior is called the bifurcation property of excitable cells, and the authors analyze this behavior of their model using symbolic reachability analysis. In the related work of [26], a model the AP of large groups of neurons represented by cycle-linear hybrid automata (CLHA) is presented. CLHA are a class of LHA that is based on the repeated execution of a single LHA. Apart from nerve cells, other biological systems, such as protein concentration dynamics in nerve cells, have been modeled using hybrid automata [11]. A common feature of these approaches is that they focus on the exact description of the activities in the nerve cell. Instead of giving details of the AP or of cell internal processes, we abstract from those and concentrate on modeling the effects that the spikes of the action potential waves in neighboring neurons have on the synaptic weight change. As a consequence, questions regarding thresholds and bifurcations do not play a central role in our work.

The approach of [13] uses Pathway Logic (PL) to model neural circuits. PL is based on rewriting logic as originally proposed by [19]. The model itself relies on the representation of each neuron as an object with specific properties. The data type of the objects is an algebraic data type consisting of a name, a relation and an operation. System specific interaction between

the objects is realized by local transitions between the states given by rewrite rules. This abstraction enables simulations of the interaction of several thousand neurons. As opposed to our work, PL based model checking cannot deal with hybrid systems properties.

Work described in [11] uses MATLAB and the symbolic quantifier elimination tool QEPCAD to compute reachable state sets for models of other types of biological systems, such as protein concentration dynamics in cells, given as hybrid automata. The algorithm used in this work to compute reachable sets for hybrid automata partitions the state space to obtain an abstract discrete transition system. This enables reachable set computations that are entirely symbolic and there is no numerical instantiation of any system parameters. We apply a linear hybrid automata based approach to build detailed models of the interaction between neighboring neurons, using explicitly given numerical parameters, which allows for a graphical visualization of the numerically given results that we obtain.

Paper Organization We first present an introduction to linear hybrid automata and synaptic plasticity in Section 2. Our LHA model of STDP is given in Section 3. The analysis of the model is described in Section 4, and the results of the analysis are discussed in Section 5.

2 Foundations

2.1 Linear Hybrid Automata

Hybrid automata have been introduced by [1] and provide a formal description of the continuous and discrete components of a hybrid system.

Definition 1 *According to [1], a hybrid automaton is a six tuple consisting of:*

- *A finite set X of real-valued variables. The function $v(x)$ assigns a value to each variable. The valuation is denoted by the set V .*
- *A finite set Loc of vertices in a control graph called locations. A state is a pair (l, v) with $l \in Loc$ and $v \in V$.*
- *The finite set Lab of synchronization labels.*
- *A finite set E of edges in a control graph, the transitions.*
- *A labeling function Act that assigns to each location $l \in Loc$ a set of activities.*
- *A labeling function Inv assigns to each location $l \in Loc$ an invariant $Inv(l) \subseteq V$.*

Definition 2 A hybrid automaton is linear if its invariants and initial states are given by linear predicates over X , flow predicates by convex linear predicates over the set X of variables and jump relations by linear predicates $X \cup X'$.

Definition 3 For the composition of hybrid systems let $A_1 = (Loc, X, C, Flow, Inv, Init)$, $A_2 = (Loc, X, C, Flow, Inv, Init)$ denote two hybrid systems. Subsequently the product $A_1 \times A_2$ of A_1 and A_2 is the hybrid system H , such that [7, 1]:

- $Loc_1 \times Loc_2$ (Locations)
- $X = X_1 \cup X_2$ (Variables), $C = C_1 \cup C_2$ (Clocks), $Lab = Lab_1 \cup Lab_2$ (Labels)
- $((l_1, l_2), a_1, \mu(l'_1, l'_2)) \in E$, iff
 1. $(l_1, a_1, \mu_1, l'_1) \in E$ and $((l_1, l_2), a_2, \mu(l'_1, l'_2)) \in E$, iff $(l_2, a_2, \mu_2, l'_2) \in E$
 2. $a_1 = a_2 = a$, or $a_1 \notin Lab_2$ and $a_2 = \tau$ or $a_1 = \tau$, or $a_1 = \tau$ and $a_2 \notin Lab_1$.
 3. $\mu = \mu_1 \cap \mu_2$
- $Flow(l_1, l_2) = Flow_1(l_1) \parallel^{X \cup \dot{X}} \cap Flow_2(l_2) \parallel^{X \cup \dot{X}}$
- $Inv(l_1, l_2) = Inv_1(l_1) \parallel^X \cap Inv_2(l_2) \parallel^X$
- $Init(l_1, l_2) = Init_1(l_1) \parallel^X \cap Init_2(l_2) \parallel^X$

Definition 4 Let σ and σ' denote two states of a hybrid system H . The state σ' is reachable from the state σ , if there is a run of H that starts in σ and ends in σ' [1].

For the computation of the sets of reachable states we use the model checking tool PHAVer which is based on symbolic computations on the sets of reachable states using polyhedra. A particular feature of PHAVer is its use of the Parma Polyhedra Library (PPL) [2] which allows for numerically exact computations on non-convex polyhedra. In PHAVer, each polyhedron represents the set of solutions to a finite system of linear inequalities, called linear constraints [8]. Such a system is given by the invariants and flow predicates of a linear hybrid automaton, as detailed in section 2.1. The execution of a hybrid automaton results in continuous changes of the continuous variables, referred to as flows, as well as discrete state changes, referred to as jumps. This model of computation allows the observation of the possible system behavior over time [1]. The state of a hybrid system contains information regarding the current location as well as the values for the variables at any point in time [1]. We use this hybrid system model to investigate the temporal dynamics of STDP.

2.2 Synaptic Plasticity

The Hebbian rule [12] postulates that connections between neurons can become stronger and more efficient if neighboring neurons are simultaneously active. This property of networks of neurons is called synaptic plasticity and could be verified experimentally [24]. Research on synaptic plasticity is strongly influenced by Hebb's postulate. As a mechanism of learning and memory, the plasticity rule proposed by Hebb states that, if the activity of one neuron drives the activity of another neuron, the connection between the two neurons is strengthened or a new connection is established [12]. [6] verified Hebb's idea: persistent changes of synaptic efficacies are induced by the simultaneous stimulation of presynaptic and postsynaptic neurons. The changes can either mean a depression or a potentiation of synaptic efficacy [24]. The experimental correlates to that theoretical concept are called long-term potentiation (LTP) and long-term depression (LTD). It is widely believed that LTP represents an important mechanism for memory in the brain [5, 14]. All models that are based on Hebbian learning use a very broad definition of the term learning: it refers to all kinds of synaptic changes [10]. The notion of learning that is used here simply refers to changes of the synaptic weight.

2.3 Spike-Timing-Dependent Models (STDP)

Spike-timing-dependent models of synaptic plasticity describe the change of the synaptic weight as a function of the relative timing difference between presynaptic and postsynaptic spike arrival. This interdependency could be proved experimentally [4, 28]. Spike related synaptic activity consists of pre- and postsynaptic spike trains. The model proposed by [20] assumes that each spike train consists of N spikes occurring with constant frequency. The pre- and postsynaptic spikes show a small time difference $\Delta t = t_{post} - t_{pre}$, with t_{pre} and t_{post} denoting the point in time that a presynaptic, respectively postsynaptic, spike occurs.

2.3.1 A Triplet Rule of STDP

Classical models [23, 16] describe the weight change based on the timing difference between pairs of neighboring spikes. But these models do not precisely validate experimental results, e.g., they do not account for frequency effects observed by [25, 22], which reported a stronger potentiation of the synaptic weight at higher frequencies. According to the work of [25], synaptic plasticity can be explained better when considering triplets or quadruplets of spikes. Based on these findings, [20] introduce a model that explains the change of synaptic weight by taking into account the interaction of spike triplets, consisting of one presynaptic and two postsynaptic spikes, or vice versa. The model reproduces the experimental results of [25] and

[22], which show that potentiation processes can be activated by triplets of spikes and depend heavily on the activation timing. At low frequencies, [20] report identical results for pairs and triplets of spikes.

2.3.2 Integration of Neuronal Variables

[20] introduce neuronal variables to model presynaptic neuronal events such as the release of glutamate into the synaptic cleft, or the increase of the calcium concentration through Ca^{2+} channels, which is a postsynaptic event [15]. [20] define variables $r1$ and $r2$ that represent presynaptic events and the variables $o1$ and $o2$ that represent postsynaptic events. They are considered to be abstract variables in the sense that can be related to any kind of biophysical quantities of the above type. For instance, let $r1$ denote the amount of glutamate released into the synaptic cleft and let t_{pre} denote the moment of presynaptic spike arrival. Then the value $r1$ is set to one at the moment at which the presynaptic spike occurs. During the passage of some time span τ , the value of $r1$ decreases to zero. After the next presynaptic spike, $r1$ is reset to one. The following function describes the behavior of the neuronal quantity $r1$:

$$r1(t) = \begin{cases} 1 & \text{if } t = t_{pre} \\ \frac{1}{\tau}t & \text{else} \end{cases}$$

The function can be applied analogously to $o2$, $o1$ and $r2$. The values of $r1$, $r2$, $o1$ and $o2$ are always in the interval $[0, 1]$.

2.3.3 Calculating the Synaptic Weight Change

According to [20], the weight change occurs at the moment of synaptic spike arrival, which is denoted by the variables t_{pre} and t_{post} . Let $\hat{r}1$ and $\hat{o}1$ describe the value of $r1$ and $o1$ at time t_{pre} and t_{post} , respectively, and let \hat{weight} denote the synaptic weight right before a spike occurs. A_2^+ and A_2^- are used to represent the values of the negative and positive amplitude of the synaptic spike of the postsynaptic neuron. When considering spike triplets, A_3^+ and A_3^- are used for representing the amplitudes of the second postsynaptic neuron [20]. It is hence easy to model and analyze the change of the synaptic weight for pairs or triplets of spikes. By setting A_3^+ and A_3^- to zero, the model turns into a classical pair based model, [20]. [25] call this the pairing protocol of STDP.

[20] give an equation for the increase of the synaptic weight. At the moment of postsynaptic spike arrival, the new value of the synaptic weight $weight$ is described as follows [20]:

$$weight = \hat{weight} + \hat{r}1 \cdot (A_2^+ + A_3 \cdot \hat{o}2)$$

The decrease of the synaptic weight is given at the moment of presynaptic spike arrival by the following equation [20]:

$$weight = \hat{weight} - \hat{o}1 \cdot (A_2^- + A_3^- \cdot \hat{o}2)$$

The neuronal variables $r1$, $o2$, $o1$ and $o2$ have to be updated after each weight change in order to capture the exact behavior of the neuronal quantities. As described in Section 2.1, the definition of a linear hybrid automaton requires the derivative of the continuous component to be linear. Therefore, we cannot directly represent the multiplicative terms involving A_3^+ and A_3^- in the above equations in an LHA model. In the next section, we first introduce the LHA models for the pairing protocol, and then show how they can be modified to reflect the effect of spike triplets on the change of the synaptic weight.

3 LHA Model for STDP

This section describes our approach to model synaptic plasticity using linear hybrid automata. As explained in section 2.2, the change of synaptic weight depends on the relative timing difference between neuronal spikes. We start with the presentation of LHA models considering pairs of spikes, following the approach of [20].

We define two different hybrid systems for modeling STDP based on pairs of spikes. The first model that we present is very detailed and based on a modular design: the neuronal spiking, the decay of the neuronal quantities and the change in synaptic plasticity are described by separate automata. The composition of these automata yields the hybrid system for STDP. Then we introduce a more abstract LHA model that ignores the exact time course of the neuronal spiking and only triggers the weight change at the moment that a spike occurs. The benefits and drawbacks of both models will be discussed at the end of this section. The values of the parameters used for the hybrid models are adopted from [20]. We use a graphical representation for the automata. It is based on rectangles to denote the states and arrows between the states to represent the transitions. To denote the first derivative of a function, which serves as the continuous component of the hybrid automaton under investigation, we use the Lagrange notation with the prime mark.

3.1 Modular Hybrid System for STDP

Our hybrid system for the model of STDP introduced in Section 2.2 is based on a modular design. We define a hybrid automaton for each key system property. This allows for a potential integration of additional parameters, if necessary, and provides an easily comprehensible system model.

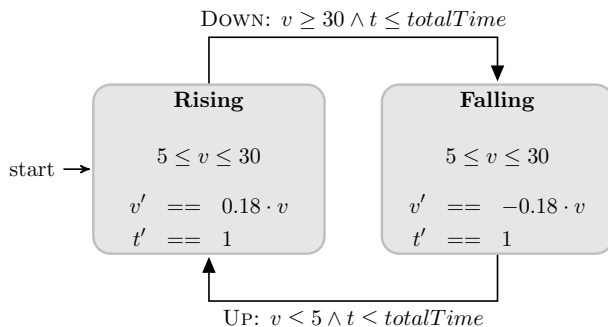


Figure 1: Automaton *preSpiking* for regular neuronal spiking at a frequency of 50 Hz with v denoting the voltage in mV, v' denoting the first derivative of the flow function and the time elapse t . UP and DOWN label the transitions.

3.1.1 Regular Spiking

First, the behavior of the system towards the repeated occurrence of spikes must be defined. We assume that the point in time a spike occurs as well as the number of spikes during a certain period of time constitute the only interesting values in the context of our model. As suggested by [9], the spikes can be assumed to occur at a frequency of 50 Hz with a phase shift of 10 ms between the postsynaptic and the presynaptic spikes. The upper and lower thresholds for the voltage are adapted from [22].

A corresponding hybrid system requires two automata, representing the presynaptic and the postsynaptic spiking, respectively. For the automata *preSpiking* and *postSpiking*, the set of locations $V = \{Rising, Falling\}$, the set of switches $E = \{(Rising, Falling), (Falling, Rising)\}$ and the finite set of variables $X = \{v, t\}$, with v denoting the voltage in mV and t the time in ms, are defined. The invariants are given with $inv(Rising) = 5 \leq v \leq 30 \wedge t \leq totalTime$ and $inv(Falling) = 5 \leq v \leq 30$, the flows are given with $flow(Rising) = v' = 0.18 \cdot v$ and $flow(Falling) = v' = -0.18 \cdot v$. The jump conditions are defined with $jump(Rising) = v \geq 30$ and $jump(Falling) = v \leq 5$. Figure 1 shows the automaton for the neuronal spiking. The arrival of the spikes at the postsynaptic cleft with a time delay of 10 ms is modeled by a second automaton called *postSpiking*. It has the same properties as the automaton *preSpiking* but implements a 10 ms phase shift. The clock variable t guarantees the correct timing: in each state the derivative of t is set to 1. Due to the guard $t \leq totalTime$ the transition can only be executed as long as the time elapse does not exceed the time limit *totalTime*. Figure 2 shows the regular spiking of the presynaptic and postsynaptic neurons as defined by the automata *preSpiking* and *postSpiking*.

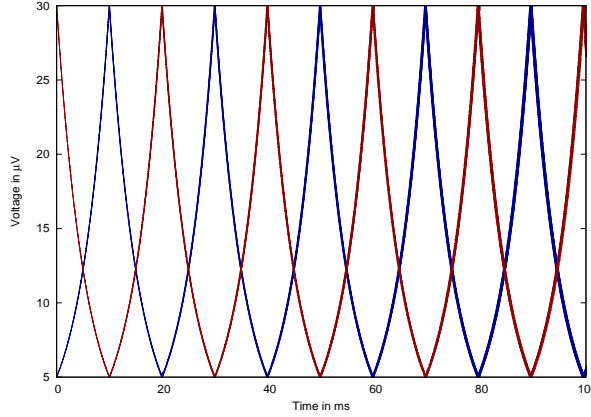


Figure 2: Regular neuronal spiking at a frequency of 50 Hz and a time delay of 10 ms between the presynaptic (red) and postsynaptic (blue) spikes.

3.1.2 Modeling the neuronal variables $r1$ and $o1$

As explained in Section 2.2, the variables $r1$ and $o1$ represent not precisely specified neuronal quantities. The value of $r1$ is set to one at the moment of the presynaptic spike arrival and the value of $o1$ is set to one at the moment of postsynaptic spike arrival. According to [20], the values of the neuronal variables $r1$ and $o1$ decrease to zero during a time interval of 16.8 ms ($r1$) and 33.7 ms ($o1$). Figure 3 shows the automata for the neuronal variables $r1$ and $o1$. The invariants and guards for the single states $r1$ and $o1$ are given with $inv(r1)=r1 \geq 0$, $inv(o1)=o1 \geq 0$ and $jump(r1)=r1 \leq 0$ and $jump(o1)=o1 \leq 0$. The flow functions are defined with $flow(r1)=r1'=-1/16.8$ and $flow(o1)=o1'=-1/33.7$.

3.1.3 Modelling the Change of Synaptic Weight

The weight function suggested by [20] is used for the automaton that models the change of the synaptic weight. The finite control graph $G = (V, E)$ is defined as follows: $V = \{Decrease, noChange, Increase\}$ and $E = \{(noChange, Increase), (noChange, Decrease), (Decrease, noChange), (Increase, noChange)\}$. We define the finite set of variables $X=\{x, w\}$ with x denoting the elapsed time and w denoting the weight change. $R1$ and $o1$ are input variables from the automata $r1Aut$ and $o1Aut$. The invariants are given with $inv(Increase)=x \leq 1$, $inv(Decrease)=x \leq 1$ and $inv(noChange)=True$. The flow conditions are defined with $flow(noChange)=w'=0$, $flow(Increase)=w'=0.0046 \cdot r1$ and $flow(Decrease)=w'=-0.003 \cdot o1$. The guards are given with $(Increase, noChange) = x \geq 1$ and $Decrease, noChange = x \geq 1$. Figure 4 shows the graphical representation of the automaton for the weight change with w' de-

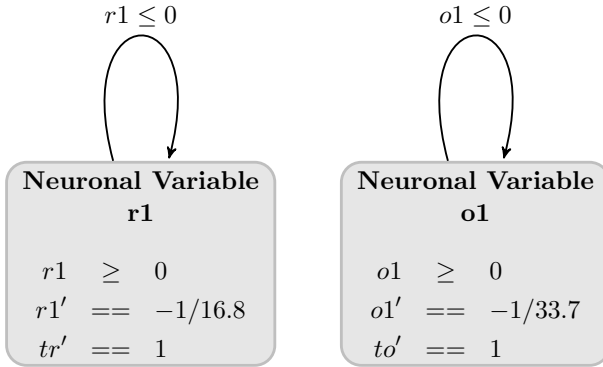


Figure 3: Automata for the neuronal variables $r1$ and $o1$, with tr and to denoting the time elapse and $r1$ and $o1$ representing arbitrary neuronal quantities.

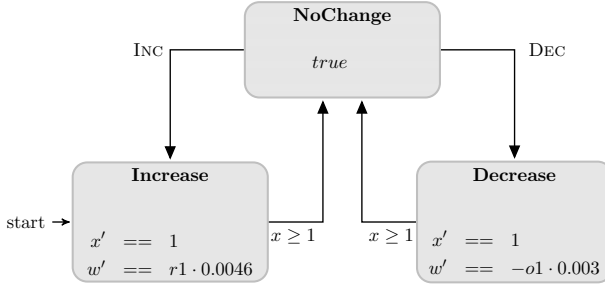


Figure 4: Automaton for the weight change, with w' denoting the first derivative of the weight function, x denoting the time elapse and synchronization labels INC, DEC.

noting the synaptic weight. We first consider the pairing protocol: whenever one of the automata for the presynaptic and postsynaptic spiking triggers a change from the state *rising* to the state *falling*, that is labelled by the synchronization labels *Inc* and *Dec*, the weight automaton executes a transition from the state *noChange* to the state *Increase* (*Decrease*).

3.1.4 Composition of the automata

For calculating the weight change with respect to the timing difference between the postsynaptic and presynaptic spikes, the defined automata have to be executed in parallel. The parallel composition of the automata can be constructed using a product operation as described in Definition 3. Let $A_1 = (Loc, X, C, Flow, Inv, Init)$, $A_2 = (Loc, X, C, Flow, Inv, Init)$ denote two hybrid systems, then the product $A_1 \times A_2$ of A_1 and A_2 is the hybrid system H . The hybrid system H for the STDP model consists of the cross

product of the automata, such that:

$$H = preSpiking \times postSpiking \times o1Aut \times r1Aut \times weight$$

Identical labels are used for switches that should be executed in parallel. For example, whenever the presynaptic automaton makes the transition from the state *rising* to the state *falling*, the automaton for the weight change has to execute a transition from the state *noChange*, which is the waiting state, to its successor state *decrease*. In order to prevent the automaton for the weight change from staying for an infinite time in one of the states calculating the weight change, it has to switch back to its waiting state after one time unit. In addition, the decay of the neuronal variables *r1* and *o1* influences the weight change. The decay is modelled by the automata *r1Aut* and *o1Aut*. Since these quantities are used to compute the increase and the decrease of the weight change, the variables *r1* and *o1* are jointly used by the automata *weightAutomaton*, *r1Aut* and *o1Aut*.

3.2 Abstracted LHA Model for STDP

The state explosion problem, which is inherent to all model checking procedures, results in a potential computational inefficiency of reachability queries. This requires models that are as small and abstract as possible. For improved analysis, we therefore build an abstracted linear hybrid automata model of STDP without parallel composition. It works with fewer variables and avoids the modelling of the regular spiking. Under the assumption, that the neuronal spiking has a fixed frequency with no variation in amplitude, one can omit the automata *preSpiking* and *postSpiking* which model the regular spiking. Each spike leads to a change (decrease or increase) of the synaptic weight. Therefore it is sufficient to integrate the timing into the automaton in order to model the change of synaptic weight. If we assume a spike frequency of 50 Hz there is going to be an increase in synaptic weight at $t_{post} = 10, 30, 50, 70, \dots$ and an increase at $t_{pre} = 20, 40, 60, 80, \dots$. The resting phase after the presynaptic and postsynaptic spike is modelled by the locations *WaitPreInc* and *WaitPostInc*. The automata *r1Aut* and *o1Aut* are also integrated into the weight automaton by adding a self loop to each of the states. Additionally, the clock *x* acts as a guard to guarantee the occurrence of transitions at time t_{pre} and t_{post} . The clock *t* represents the overall time elapse. Figure 5 shows the automaton for the optimized model. It is worth mentioning that we aim to give examples for the modeling power of hybrid systems. Providing evidence that the modular model from Section 3.1 and the abstract model presented in this section are equivalent in any formal sense goes beyond the scope of this paper.

Our abstracted model for STDP can also be used to represent the triplet model of STDP. We used the following idea in order to account for the multiplicative operations on variables needed for the triplet model. Whenever

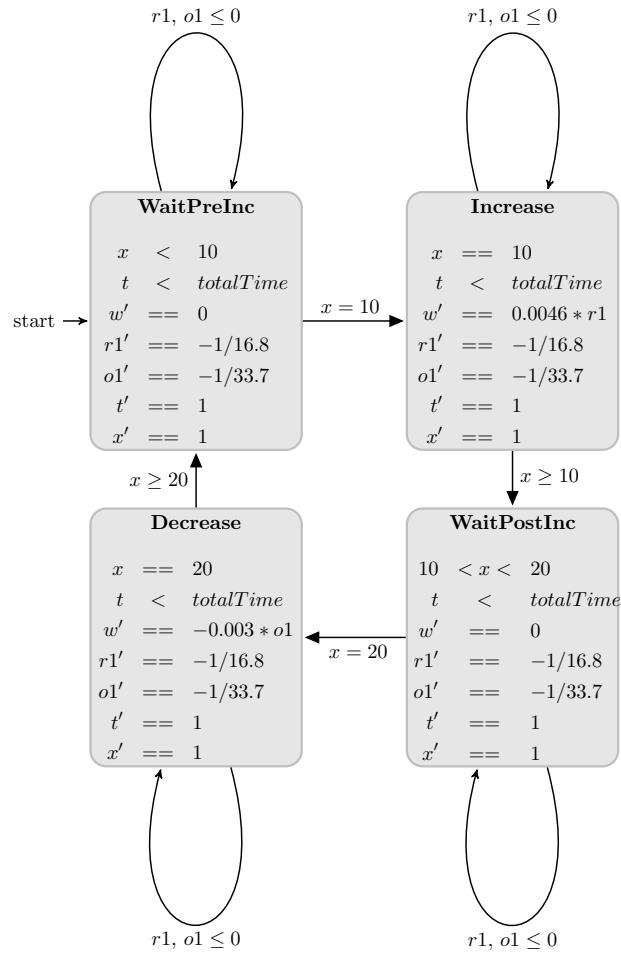


Figure 5: Automaton for the compact model with synaptic weight w , variables $r1$ and $o1$ and the clocks x and t .

during a run of the LHA, the multiplication of $r1 \cdot o2 \cdot A_3^-$ and $o1 \cdot r2 \cdot A_3^+$ is required, we stop the state space exploration and write to disc the set of reachable states computed up to that point. Additionally, we store the values of all variables in a parameter file. Then, we use an external program for the multiplication and store the result in the parameter file. We read in the parameter file and continue the computation of the set of reachable states. Formally, this corresponds to a partitioning of the state space. Since we store a snapshot of the state space and the values of the variables after each step, we preserve all reachability properties by this partitioning procedure.

4 Analysis

The potential of using hybrid systems model checking, such as implemented in the PHAVer tool, lies in being able to symbolically compute and explore the state space of the STDP models that we developed. The state space of an LHA model represents, amongst others, information regarding the temporal dynamics of system parameters, i.e., information which parameters can attain which value after a certain passage of time. Since PHAVer uses a symbolic representation of the state space, it is then possible to a) graphically represent the temporal valuation of system parameters, and b) to answer queries whether certain parameter values or combinations of values correspond to a possible evolution of the system. We call the latter queries reachability queries. We illustrate in this section how both types of analyses can be used to obtain a better understanding of the model, as well as to predict behavior that has not yet been observed experimentally. For reasons of computational efficiency, we only use the abstracted LHA model during our analysis. All reachability analysis described in this Section was performed using PHAVer.

4.1 Reachability Analysis

PHAVer computes the set of reachable states by using a piecewise linear approximation of the continuous components of the hybrid system under investigation. The reachability analysis for the STDP model takes into account the change of synaptic weight over time and allows us to produce a graphical representation of these changes using suitable plotting tools. We use those graphical representations for the documentation of the system behavior, and to confirm the experimental results of [20] as being consistent with our model. In particular, we were able to verify the experimental results of [20] using the following analysis steps.

The authors of [20] verify their model, on which ours is based, with respect to the pairing experiments described in [25]. By setting A_3^+ and A_3^- to zero, their model turns into a classical pair-based model. [20] report a weight change of $\Delta w \approx 0.09$ in Figure 2C and Figure 2D of their paper.

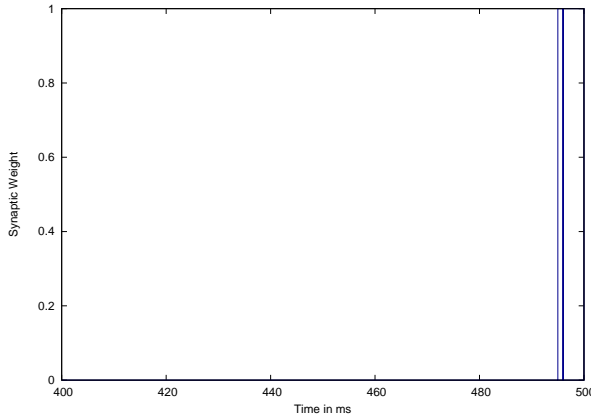


Figure 6: Visualization of the forbidden states as a projection to time (x-axis) and synaptic weight(y-axis).

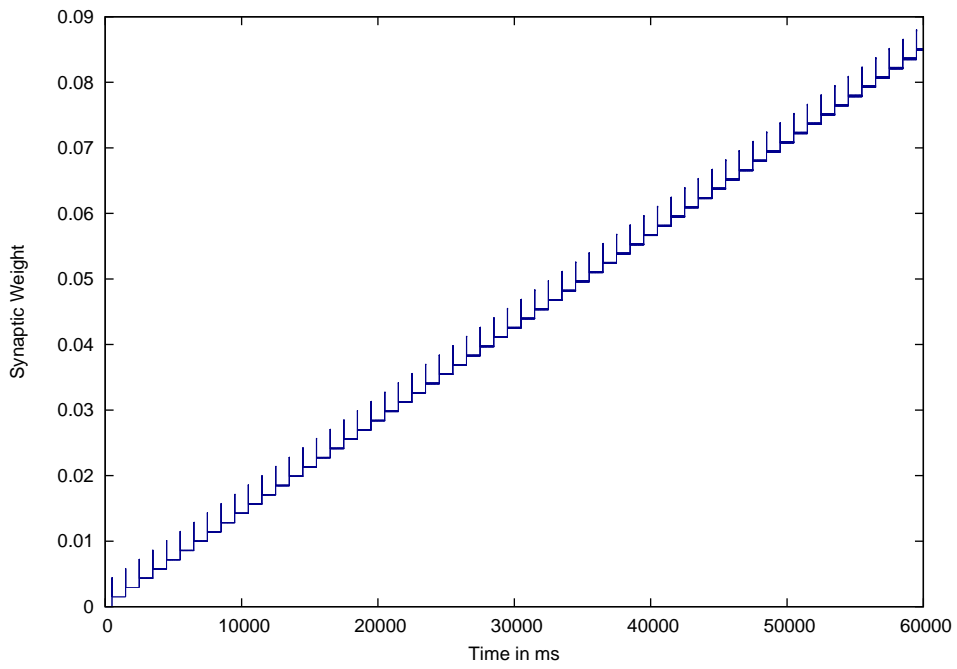
The results correspond to the weight change of $\Delta w = 0.0858$ computed by our model, c.f. Figure 7. In particular, figure 7 (a) shows the set of reachable states for our STDP model as a projection to time and synaptic weight when applying the pairing protocol with a parameter set based on the hippocampal experiments by [25].

By applying the triplet rule, [20] reproduce the results of the triplet experiments by [25]. When considering spike triplets, our STDP model yields a weight change of $\Delta w = 0.1967$ when the spike triplet consists of one presynaptic and two postsynaptic spikes. This corresponds to the results of Figure 4D in [20]. Figure 7 (b) shows the set of reachable states computed for our STDP model as a projection to time and synaptic weight for the triplet model.

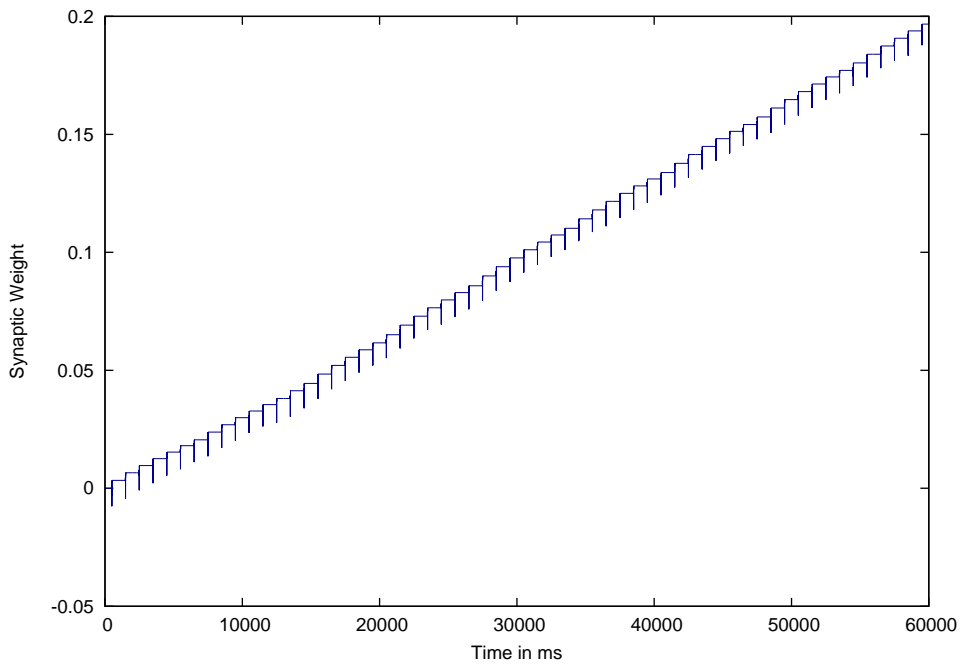
4.2 Safety Verification

PHAVer has the capability to verify safety properties. A safety property is a property stating that something bad does never happen [3], e.g., an undesirable system configuration or value. Such a property can be verified by defining a set of bad or forbidden states R_{bad} . If the intersection of the set of reachable states R with the set of bad states R_{bad} is empty, this will give the formal proof that these bad states are not reachable. For example, let us assume that we wish to prove that the synaptic weight w after 496 ms does not exceed a value of 1. In the PHAVer input language this can be defined as a reachability query, characterizing bad states, as follows:

$$S_{bad} = weight.\{t \geq 496 \& w \geq 1\}$$

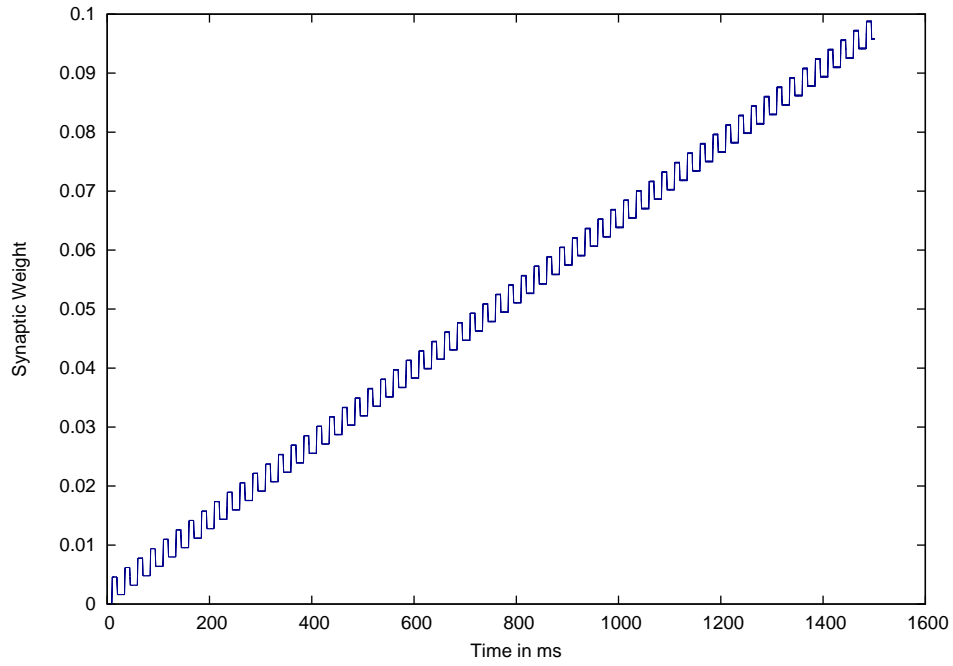


(a) Results of our pair-based model with parameter set based on [25].

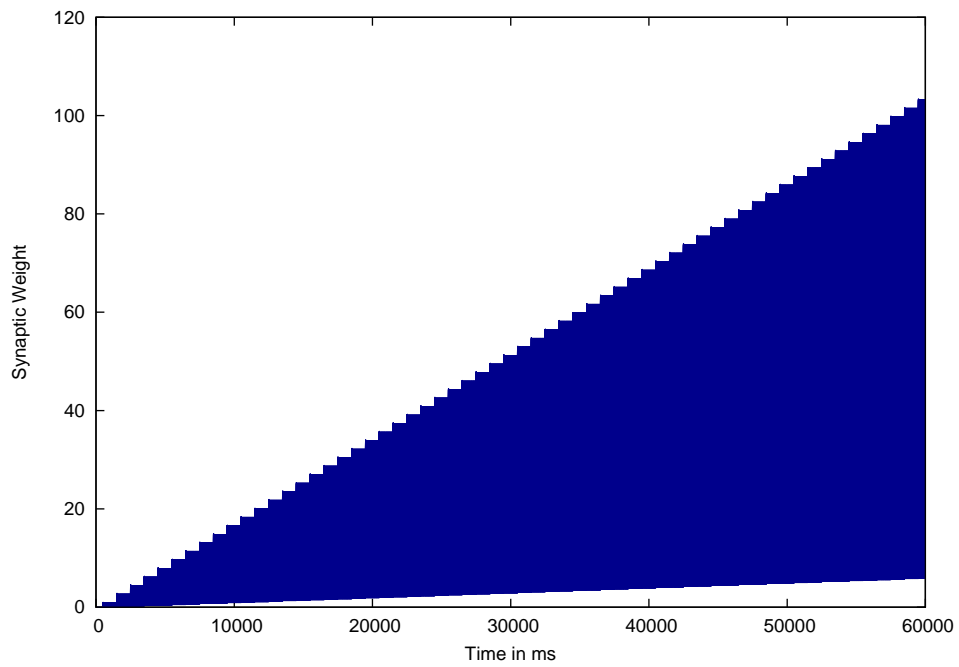


(b) Results of our triplet model with parameter set based on [25].

Figure 7: Set of reachable states as a projection to time (x-axis) and synaptic weight (y-axis).



(a) Changing the frequency from 1 Hz to 40 Hz in our pair-based model.



(b) Spike happens at any time during an interval of 6 ms in our pair-based model.

Figure 8: Effects of changing parameters on the computation of the sets of reachable states for the hybrid system as a projection to time (x-axis) and synaptic weight (y-axis).

As a result, PHAVer returns the intersection of the set of bad states with the set of reachable states, which is the empty set. When obtaining the empty set as a result, we have a formal proof that the weight w does not exceed the value of one after the first spike at $t = 496$ ms. PHAVer plots the set of the defined forbidden states as shown in Figure 4.2.

As a second example, let us postulate that w after 1000 ms is still zero. In the PHAVer input language, the converse of this condition, representing a set of forbidden system configurations, can be expressed as

$$S_{bad} = weight.\{t \leq 1000 \& w \geq 0\}.$$

It is easy to see that the weight is increased during the first 1000 ms and that hence the forbidden system configuration is reachable. As a result, PHAVer returns the intersection of the set of forbidden states with the set of reachable states. If it is not the empty set, the states forming the intersection are given and serve as an example of the violation of the property.

Reachable Forbidden States:

$$weight.\{increase \dots t \leq 496 \& weight \geq 0 \& \dots\};$$

Notice that the PHAVer output is shortened for better readability, irrelevant information is abbreviated with \dots .

4.3 Prediction of system behavior

Our approach allows for the prediction of potential system behavior for some assumed input or parameter values. In particular, the effects of parameter changes can be computed by the model checker and read out of the graphical representation of the reachability analysis. We consider the effect of varying three quantities in our model for the pairing protocol.

4.3.1 Varying the values of the variables $r1$ and $o1$

We set the time span for the decay of the neuronal variables $r1$ and $o1$ to 50 ms for both variables, a value that we have chosen rather arbitrarily. One can observe an increase of the synaptic weight. We obtain $\Delta w = 0.0981$ due to the larger time span that $r1$ needs to reach zero.

4.3.2 Changing the frequency

Experiments by [25] and [22] indicate that a change in frequency has an effect on the change of the synaptic weight. The pairing protocol version of the model suggested by [20], however, fails to reproduce this frequency dependency. According to [20], a variation of the frequency only leads to a negligible change of the synaptic weight when pairs of spikes are considered. Our LHA based model of the pairing protocol yields results that are

identical to the findings of [20]. Figure 8 (a) shows the effect of changing the frequency to 40 Hz when considering a set of 60 spikes. We also observe only a small change of the synaptic weight when increasing the frequency of the neuronal spiking. When applying the parameter set of the visual cortex data by [22] and increasing the frequency to 50 Hz, our model yields a weight change $\Delta w = 0.9$, which is close to the experimental results of [22] for that frequency. [20] report the related experimental data using their model in [20], Table 1.

4.3.3 Modelling parameter ranges or time windows

Additionally, the model can be used to express uncertainties in the behavior of the observed system. Let us assume that the point in time at which a presynaptic spike occurs is not exactly known, but can happen some time during an interval of 6 ms. The value of the synaptic weight is therefore determined by a range, depending on the exact timing of the spike. The graphical representation of the set of reachable states in figure 8 (b) takes into account the value for the synaptic weight for all possible points in time at which the spike may occur. The synaptic weight can take any value in the colored area, depending on the timing of the spikes.

5 Conclusion

We have presented two LHA models for the STDP learning rule by [20]. The first model is composed of a number of independent components, each of which is representing an important process contributing to the complex dynamics of the STDP model. The main contribution of this model is that it allows us to document knowledge about system behavior in a modular and easily comprehensible and modifiable fashion. We then presented a compact abstraction of this model, which we subjected to formal analysis using the PHAVer model checker. This analysis allowed us to gain insight in the temporal dynamics of the STDP model, and also to predict behavior when changing the underlying model parameters. We then described a procedure to deal with the multiplication of variable values, which allowed us to handle the triplet spike model in addition to the pairing protocol. Using our models we were able to reproduce the results of [20] by using parameters sets based on the experiments of [25]. In future work we are interested in extending our model to more complex models of STDP and to extend the modeling and analysis to more complex networks of neurons.

Acknowledgements We wish to thank Jamil Ahmad for his comments on an earlier version of this work.

References

- [1] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, pages 209–229, London, UK, 1993. Springer-Verlag.
- [2] R. Bagnara, E. Ricci, E. Zaffanella, and P. M. Hill. Possibly not closed convex polyhedra and the parma polyhedra library. In *Proceedings of the 9th International Symposium on Static Analysis, SAS '02*, pages 213–229, London, UK, 2002. Springer-Verlag.
- [3] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT Press, Cambridge, Massachusetts, 2007.
- [4] G. Bi and M. Poo. Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.*, 18(24):10464–10472, 1998.
- [5] T. Bliss and S. Laroche. ZAP and ZIP, a story to forget. *Science*, 313:1058–1059, 2006.
- [6] T. Bliss and T. Lomo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of Physiology*, 232(2):331–356, 1973.
- [7] G. Frehse. PHAVer: Algorithmic verification of hybrid systems past HyTech. *Int. J. Softw. Tools Technol. Transf.*, 10:263–279, May 2008.
- [8] K. Fukuda. Frequently asked questions in polyhedral computation. Technical report, Swiss Federal Institute of Technology, Lausanne and Zurich, Switzerland, 2004.
- [9] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383(6595):76–78, September 1996.
- [10] W. Gerstner and W. M. Kistler. *Spiking Neuron Models*. Cambridge University Press, first edition, August 2002.
- [11] R. Ghosh and C. Tomlin. Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-notch protein signalling. *Systems Biology*, 1:170–183, 2004.
- [12] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Wiley, New York, first edition, June 1949.

- [13] S. M. Iyengar, C. Talcott, R. Mozzachiodi, E. Cataldo, and D. A. Baxter. Executable symbolic models of neural processes. In *Network tools and applications in biology NETTAB07*, 2007.
- [14] M. S. J. Whitlock, A. Heyen and M. Bear. Learning induces long-term potentiation in the hippocampus. *Science*, 313:1093–1097, 2006.
- [15] U. Karmarkar and D. Buonomano. A model of spike-timing-dependent plasticity: one or two coincidence detectors. *Journal of Neurophysiology*, 88:507–513, 2002.
- [16] W. M. Kistler and J. L. V. Hemmen. Modeling synaptic plasticity in conjunction with the timing of pre- and postsynaptic action potentials. *Neural Comput.*, 12:385–405, February 2000.
- [17] W. Levy and O. Steward. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience*, 8(4):791–797, 1983.
- [18] H. Markram, J. Lbke, M. Frotscher, and B. Sakmann. Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs. *Science*, 275(5297):213–215, 1997.
- [19] N. Marti-Oliet and J. Meseguer. Rewriting logic as a logical and semantic framework. In D. Gabbay, editor, *Handbook of Philosophical Logic*. Kluwer Academic Publishers, 1997.
- [20] J. Pfister and W. Gerstner. Triplets of spikes in a model of spike timing-dependent plasticity. *J. Neurosci.*, 26(38):9673–9682, 2006.
- [21] C. Priami. Algorithmic systems biology. *Commun. ACM*, 52:80–88, May 2009.
- [22] P. Sjöström, G. Gina G Turrigiano, and S. Nelson. Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron*, 32(6):1149–1164, 2001.
- [23] S. Song, K. D. Miller, and L. F. Abbott. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, 3(9):919–926, September 2000.
- [24] G. Stent. A physiological mechanism for Hebb’s postulate of learning. *Proceedings of the National Academy of Sciences of the United States of America*, 70(4):997–1001, 1973.
- [25] H. Wang, R. Gerkin, D. Nauen, and G. Bi. Coactivation and timing-dependent integration of synaptic potentiation and depression. *Nature Neuroscience*, 8(2):187–193, 2005.

- [26] P. Ye, E. Entcheva, S. Smolka, and R. Grosu. Modelling excitable cells using cycle-linear hybrid automata. *Systems Biology, IET*, 2(1):24–32, January 2008.
- [27] P. Ye, E. Entcheva, S. Smolka, and R. Grosu. Symbolic analysis of the neuron action potential. In *Bioinformatics and Biomedical Engineering, 2008 (ICBBE 2008)*, pages 836–839, May 2008.
- [28] L. Zhang, W. H. H. Tao, C. Holt, and M. Poo. A critical window for cooperation and competition among developing retinotectal synapses. *Nature*, 395:37–44, 1998.