Synchronization of Dynamic Networks for Knowledge Representation and Higher-Level Fusion

Neil A. Bomberger, Allen M. Waxman, and Felipe M. Pait
Fusion Technology and Systems Division
ALPHATECH, Inc.
Burlington, MA, USA
{bomberger, waxman}@alphatech.com

Abstract - *This paper presents a novel approach to higher-level (2+) information fusion and knowledge representation using semantic networks composed of coupled spiking neuron nodes. Networks of spiking neurons have been shown to exhibit synchronization, in which sub-assemblies of nodes become phase locked to one another. This phase locking reflects the tendency of biological neural systems to produce synchronized neural assemblies, which have been hypothesized to be involved in feature binding. The approach in this paper embeds spiking neurons in a semantic network, in which a synchronized sub-assembly of nodes represents a hypothesis about a situation. Likewise, multiple synchronized assemblies that are out-of-phase with one another represent multiple hypotheses. The initial network is hand-coded, but additional semantic relationships can be established by associative learning mechanisms. Our approach will be demonstrated on a simulated scenario involving the tracking of suspected criminal vehicles between meeting places in an urban environment.*

Keywords: Information fusion, Fusion 2+, situation assessment, spiking neural networks, semantic knowledge representation.

1 Introduction

The goal of higher-level information fusion (L2/L2+ Fusion) is to process data and combine it with existing knowledge in order to attain situation awareness [1, 12, 15]. Attempts to achieve L2+ fusion have been made using a variety of techniques, including rules-based reasoning, logic-based methods, Bayesian networks, fuzzy logic, and neural networks (see [4] for a review). However, although there are effective statistical and learning methods for lower-level fusion (Object-level Refinement), there is little consensus on effective methods for higher-level fusion.

This paper proposes a new approach to the problem of higher-level fusion based on semantic networks composed of spiking neurons. The dynamics of the spiking networks lead to transient synchronization of node activity, which can be viewed as temporal binding of these nodes in short-term memory (STM). Pair-wise associative learning between synchronized network nodes increases the weight between these nodes, and acts as a form of long-term memory (LTM).

The semantic networks are organized as modular knowledge structures, with different domains of knowledge programmed in separate knowledge modules. This allows a domain of knowledge to be separately learned or programmed by an expert as a module, which can then be connected to other knowledge modules.

Network synchronization could also allow *concepts* to be established in LTM as groups of nodes that are temporarily bound through distributed synchronized oscillations. These concepts can be learned in a higher level of the knowledge hierarchy. The synchronization of the semantic network nodes acts as a presentation mechanism which simultaneously activates all nodes that belong to a concept, and thus allows them to be learned together. This will not be addressed in this paper, but is currently under investigation.

The work presented in this paper on spiking semantic networks is part of a larger program sponsored by the AFOSR. The goal of this program is to investigate novel approaches to higher-level fusion for situation assessment, in the context of a simulated scenario involving vehicles belonging to suspected members of a criminal gang driving around an urban environment, with individual vehicles gathering at various meeting places.

This program requires the fusion of multisensor data (EO, HSI, GMTI, HRR), construction of a 3-D site model of an urban environment, simulation of moving vehicles and associated radar GMTI detections, tracking simulated vehicles as they move, using high-resolution range (HRR) radar profiles to allow *learning-while-tracking* [7], and embedding the site model, imagery, moving targets, and tracks in a 3-D viewer. An architecture that brings together this multisensor data with knowledge in the context of a geospatial environment was previously described [14]. Our existing neural-based image fusion and mining analyst tools are used to extract features from multisensor imagery (Fig. 1). We have developed a *Target Motion Generator* in Java to simulate both individual vehicle movement and urban background traffic on road networks (Fig. 2). A 3-D site model of a generic urban environment has been constructed which includes HyMap hyperspectral imagery, high-resolution EO imagery, and 3-D building models and DEM extracted from the stereo EO imagery. (EO and HyMap imagery were kindly provided by Kodak Commercial & Government Systems.) We have also developed 3-D viewer software that allows the simulated vehicle movements and vehicle tracks to be embedded in the site model and viewed interactively in 3-D (Fig. 3).

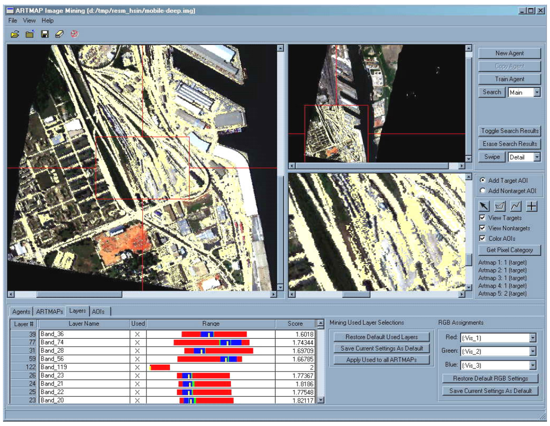

Fig. 1. Mining for roads in HyMap data, using ALPHATECH's Neural Fusion image analyst tools.

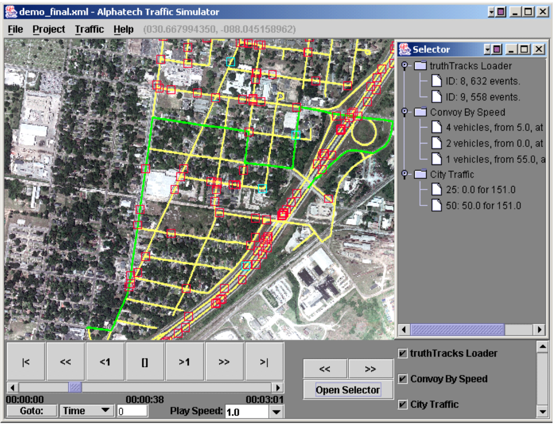

Fig. 2. Vehicle scripting and traffic simulator, showing underlying imagery, road network, and generated traffic (red squares) using ALPHATECH's *Target Motion Generator*.

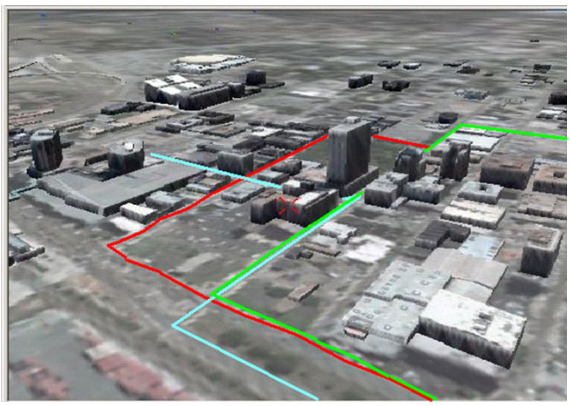

Fig. 3. 3-D site model constructed of downtown area of an urban environment, with tracks from vehicles embedded in 3-D viewer.

2 Model Dynamics and Phenomena

In our knowledge hierarchy, the semantic layer is composed of multiple modules of spiking neural networks. Each node in the network consists of a spiking neuron and has a semantic meaning, such as an entity, category, or attribute. This use of spiking networks for semantic representation is based on the theory that observed synchronous oscillations in biological neural systems play a role in binding [10, 11, 13]. Shastri [8, 9] is one of the few investigators to attempt to apply synchronized spiking networks to problems of semantic knowledge representation, but his work has focused on creating complicated inferential reasoning networks that posit synchrony as a mechanism, without actually simulating the spiking dynamics of the networks.

This section describes the model of spiking neurons, spiking and synchronization phenomena produced by different types of connectivity, and spike timing-based Hebbian associative learning.

2.1 Integrate-and-Fire Model

For our simulation of spiking neuron dynamics, we adopt the integrate-and-fire (IaF) model of Horn and Opher [6]. Their model equations are devised to be a simplified two-variable version of the equations of neurodynamics derived experimentally for spiking neurons, such as the Hodgkin-Huxley equations [5]. The Horn and Opher equations use two variables with relevant meanings (membrane voltage and refractory dynamics) and allow fast simulation while approximating the important behavior of spiking neurons.

This IaF model obeys the following equations:

$$\dot{v}_i = -kv_i + \alpha + cm_i v_i + m_i (I_i + \sum w_{ij} f_j) \tag{1}$$

$$\dot{m}_i = -m_i + H(m_i - v_i) \tag{2}$$

$$f_i \propto -\frac{dm_i}{dt} H\left(\frac{dm_i}{dt}\right) \tag{3}$$

where v_i is the sub-threshold electrical potential variable, m_i is the refractory dynamics variable, and f_i is the firing variable for neuron i. Likewise Eq. (1) specifies the membrane potential dynamics, Eq. (2) the refractory dynamics, and Eq. (3) the firing dynamics of neuron i. I_i is the input to the neuron, and $w_{ij} f_j$ is the weighted input from neuron j to neuron i. $H(x)$ is the Heaviside step function, defined as $H(x)=0$ for $x<0$, and $H(x)=1$ for $x \geq 0$.

Horn and Opher [6] have applied their IaF networks to different types of clustering and segmentation, including image segmentation. In this paper we apply these networks to the problem of semantic information processing.

The IaF model equations have been implemented in the *Simulink* graphical programming environment built around *Matlab* (*http://www.mathworks.com*) to form an IaF neuron block. This block is vectorized to produce simulations with multiple IaF neurons. The IaF neurons are connected by an LTI system block, which allows weight and delay arrays to be defined in order to specify the connectivity. This permits convenient array-based configuration of network connectivity, and produces simulations that are orders of magnitude faster than can be produced by drawing individual connections between neuron blocks in Simulink.

2.2 Spiking and Synchronization Phenomena

Our goal is to use network synchronization to represent entities that are semantically related, and out-of-phase elements to represent entities that are not related. The first step to that end is to identify various spiking and synchronization phenomena of these networks.

In [6], Horn and Opher demonstrate different forms of network synchronization behavior produced by three types of homogeneous connectivity: excitatory connections without delay, inhibitory connections without delay, and inhibitory connections with delay. Fig. 4 shows the results of our simulations of these cases for 150 fully-connected neurons, which agree with the results of Horn and Opher. Note that without delay, excitatory connections lead to synchronization, while inhibitory connections lead to non-synchronization. Delay added to inhibitory connections eventually leads to synchronization which is more tightly locked than with excitatory connections/no delay, but the synchronization takes longer to be established.

A more interesting case is when the connectivity is not homogeneous. Of particular interest for our work is whether there exist patterns of connectivity for which multiple out-of-phase groups of neurons are produced, with synchronization between neurons within a group. This behavior allows a semantic network to be produced in which nodes that are semantically related are synchronized, while unrelated nodes spike out-of-phase.

Fig. 5a shows a connectivity pattern which produces this behavior. All connections have zero delay, with excitatory connections between neurons in a group, and inhibitory connections between neurons in different groups. The spiking behavior for this configuration is displayed in Fig. 5b. As these figures show, the neurons begin spiking with random phase, but as time progresses the neurons in each excitatorily-connected group become phase-locked to one another. Each of these phase-locked groups occupies its own slot in the phase space, that is, each group is out-of-phase with the other groups.

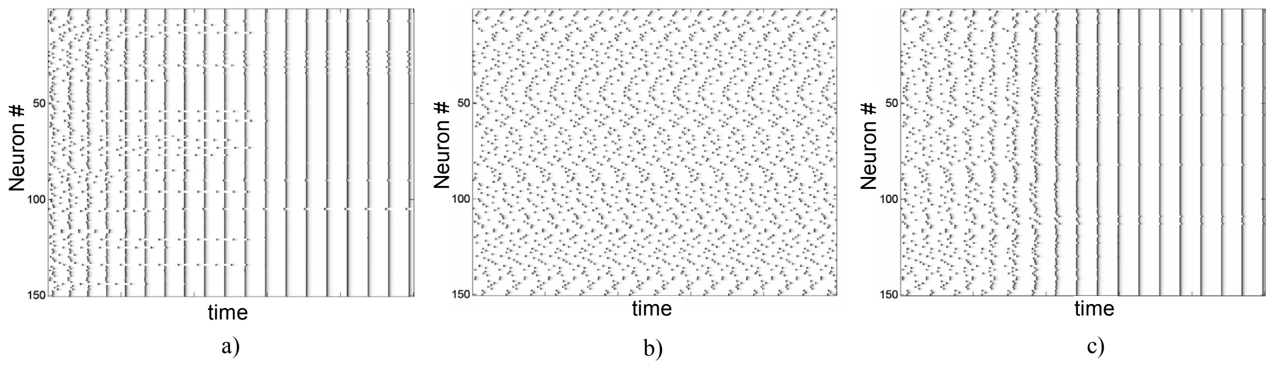

Fig. 4. Synchronization phenomena for 150 neurons with different connectivity types. Each row represents the activity profile of one of the 150 neurons, and each dark point represents a spike. a) Excitatory connections, no delay. b) Inhibitory connections, no delay. c) Inhibitory connections, with delay.

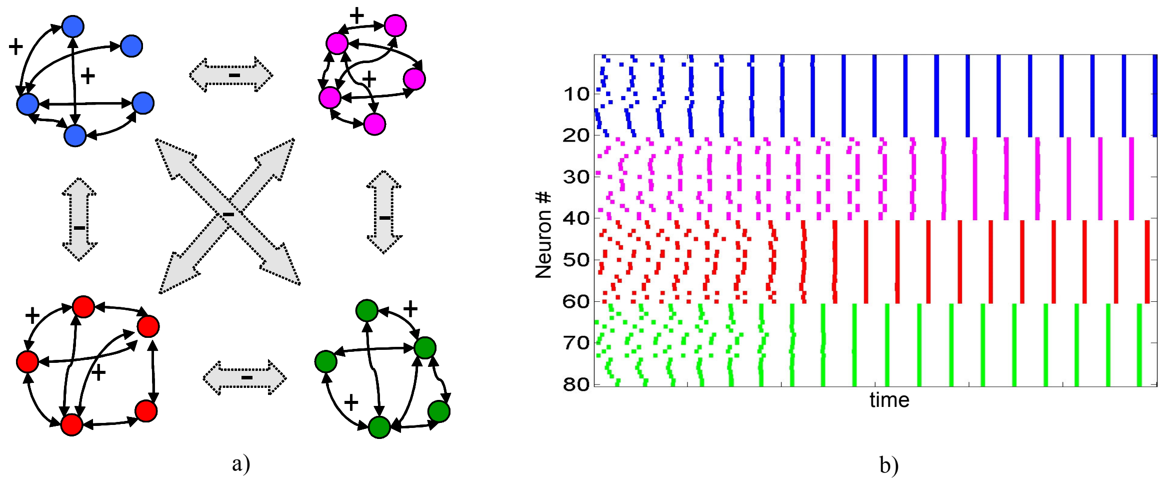

Fig. 5. Connectivity pattern leading to multiple out-of-phase synchronized sub-groups. a) Schematic diagram of connectivity. Each sub-group represents 20 fully-connected neurons with excitatory connections (no delay). Each neuron has an inhibitory connection (no delay) to every other neuron that is not in its sub-group. b) Activity of neurons across time. Neurons begin with random phase, then as time progresses each sub-group becomes phase-locked within itself, and out-of-phase with every other sub-group.

2.3 Associative Learning

Associative learning is often referred to as Hebbian learning, due to Hebb's postulate that if one neuron repeatedly plays a role in causing another neuron to fire, a chemical process occurs that effectively strengthens the synaptic connection between them [3]. Mathematically, this can be formulated in terms of spike rate or spike timing. For spike rate-based associative learning, if two neurons are connected and are simultaneously highly active, the weight on the connection between them increases. Spike timing-based associative learning only occurs if there is persistent simultaneous spiking, within some time window, between two neurons. In this way, learning only occurs if neurons are synchronized; if they are not synchronized, they can be simultaneously active and no associative learning will occur. This allows multiple groups of neurons to be active simultaneously without confusion between them.

The networks in this paper use a spike-based associative learning mechanism, which is specified by

$$\frac{d}{dt} w_{ij}(t) = \alpha \cdot f_i(t) f_j(t) - \beta \cdot f_j^{\,2}(t), \qquad (4)$$

where $w_{ij}(t)$ is the weight of the connection from pre-synaptic neuron j to post-synaptic neuron i, $f_i(t)$ is the firing activity of post-synaptic neuron i, $f_j(t)$ is the firing activity of pre-synaptic neuron j (from Eq. (3)), α is the learning rate parameter, and β is the decay rate parameter. The first term on the right side leads to an increased value of w_{ij} when neuron i and neuron j spike simultaneously, that is, when $f_i(t)$ and $f_j(t)$ have high values. The second term results in decrease of w_{ij} when there is a pre-synaptic spike without a post-synaptic spike ($\alpha > \beta$). This results in weight decay during periods when pre-synaptic firing does not lead to post-synaptic firing. That is, the semantic item j is not associated with item i. The values of w_{ij} are bounded by the requirement that values stay in the range [0, 3].

Eq. (4) is a simplification of a more general formulation of spike-based associative learning [2], and lacks an explicit temporal window to control the degree of synchrony required for learning. A temporal window is implicit in Eq. (4) in the shape of the spike profiles specified by Eq. (3), in that a spike profile has some width and thus two spikes are not required to be precisely simultaneous in order for them to temporally overlap and associative learning to occur. Eq. (4) is computationally simple and captures the behavior we are interested in, but it may be useful in the future to use a more general formulation in which the temporal window can be explicitly controlled.

The transient synchronization of the spiking neurons acts as a form of short-term memory (STM). The synchronization represents an aspect or hypothesis about the current situation, but then dissipates after the conditions responsible are no longer present. The weight increase between synchronized neurons due to associative learning acts as one form of long-term memory (LTM), in that these weights persist after the conditions that led to the synchronization are gone. These weights express a pair-wise association learned or programmed between semantic items.

3 Example Knowledge Networks

3.1 Semantic Knowledge Representation

In the semantic layer of our knowledge networks, knowledge is distributed and is represented by connections between nodes. Each node represents a semantic item, an entity, category, or attribute, and the connections between nodes represent the relatedness of the corresponding items. The goal for the semantic layer is to achieve synchronous activity in which synchronized sub-groups represent hypotheses about a situation.

The semantic knowledge that can be represented includes features that define a category (a specific instance of a category can be learned as a collection of feature values) and relationships between entities. Fig. 6 illustrates a knowledge tree demonstrating category relationships that can be represented in this type of semantic network.

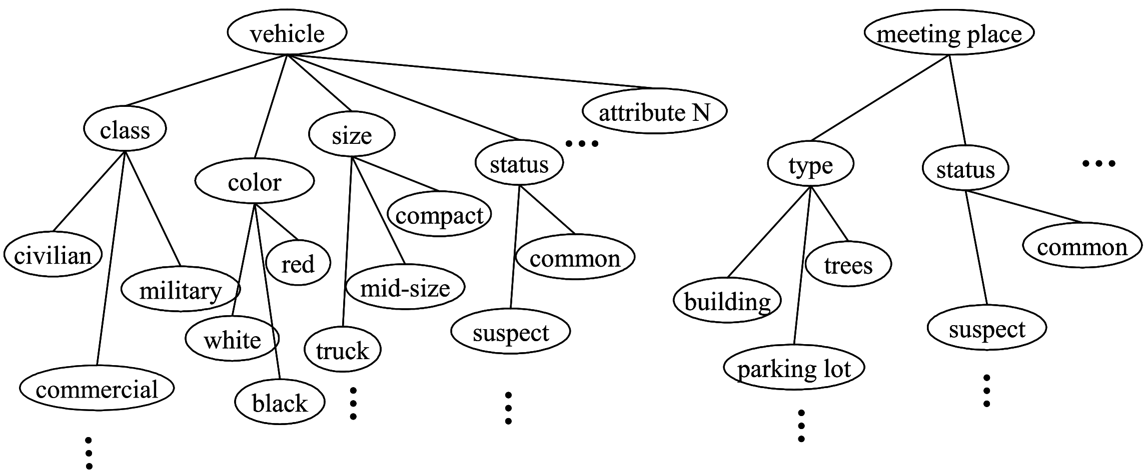

Fig. 6. Example knowledge trees implemented in semantic networks about *vehicles* and *meeting places*.

Large excitatory weights on connections represent a high degree of relatedness, while small weights represent a small degree of relatedness. Inhibitory weights correspond to instances of the same type that are not known to be related. For example, two nodes representing specific instances of vehicles have an inhibitory connection between them unless they are otherwise related (e.g. by being simultaneously located at the same meeting place). If both of these nodes are active, this causes the groups that each participates in to be out-of-phase with one another, unless the groups are related by another connection. Currently, all connections in the semantic network are implemented with zero delay.

3.2 Knowledge Hierarchy

As described in the previous section, the *semantic layer* represents collections of features that specify a general category, as well as relationships between entities (specific instances of categories defined in the *concept layer*). Likewise, nodes in the *concept layer* are defined as collections of specific feature values, that is, specific entities/attributes in the semantic layer (Fig. 7). The concept layer nodes are defined at various levels of resolution: objects, events (which can include objects, location, action, and time), and groups of events.

Once specific concepts are defined as a collection of feature values, these concepts are reused as nodes in both the *semantic* and *concept* layers. This creates a knowledge hierarchy, in which concepts are defined as collections of items from the *semantic layer*, and once a concept is defined, it can be used to define additional relationships in the *semantic layer*, as well as additional concepts/events in the *concept layer*. For example, if two vehicles are assigned nodes in the *concept layer*, the semantic relationships of those vehicles with other entities can be defined, and they can also be used to define events involving those vehicles. The construction of a knowledge hierarchy can continue indefinitely in this way, with concepts defined in lower levels used to define semantic relationships and new concepts in upper levels of the hierarchy.

In this paper, we concentrate on the *semantic layer*, and use object nodes for semantic representation that we assume have already been defined in the *concept layer*. Future work will show how new concepts can be learned.

3.3 Knowledge Modules

The knowledge encoded in the semantic and concept layers is organized into sub-domains of knowledge, and these sub-domains are programmed as separate knowledge modules. For example, in Fig. 7 the knowledge trees can be separated into a sub-domain representing *vehicles* and a sub-domain representing *meeting places*.

The benefit of separation of knowledge into modules is that it allows the semantic relationships in these sub-domains to be learned or programmed by a sub-domain expert or knowledge engineer. Then, associations are learned or programmed between these knowledge modules. The next section demonstrates a simple example of connecting separate knowledge modules and learning associations between them.

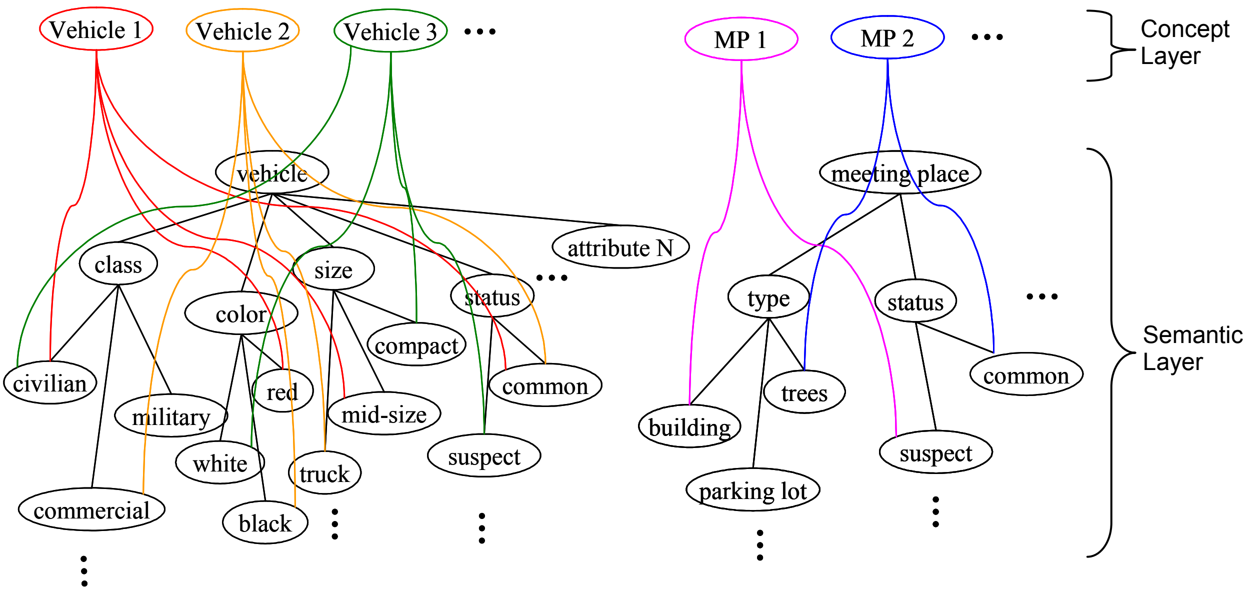

Fig. 7. *Concept layer* added to *semantic layer* from Fig. 6. Concepts are defined in the concept layer as collections of entities/attributes from the semantic layer

4 Computing with Dynamic Information Networks for Situation Assessment

4.1 Simulink Implementation

A simplified version of the knowledge network in Fig. 7 is implemented as three different knowledge module blocks in *Simulink*, as shown in Figure 8: the *Vehicles*, *Meeting Places*, and *Attributes* modules. Each block contains semantic spiking nodes and an internal connectivity weight matrix, with a subset of the nodes used for input/output to the other blocks. The connections between knowledge module blocks are realized as connectivity blocks which contain arrays of weights, with vector-valued inputs and outputs. There is such a connectivity block from the *Vehicles* module to the *Meeting Places* module, and vice versa. There is also a connectivity block from the *Vehicles* block to the *Attributes* block, which encodes the attributes of specific vehicles in the *Vehicles* block. This connectivity block is different from the other two, in that the weights in its connectivity matrix can be modified through associative learning. Notice that this block receives input from the *Attributes* block as well as from the *Vehicles* block, giving it access to post-synaptic as well as pre-synaptic spiking activity.

4.2 Scenario: Meetings Among Vehicles

This approach for knowledge representation and associative learning, using synchronization of coupled spiking networks, is demonstrated with a scenario that involves a series of meetings between vehicles belonging to suspected members of a criminal gang. Initially, one vehicle is suspected of belonging to a criminal. This is represented by a strong excitatory connection to the *Suspect* node in the *Attributes* module from the suspected vehicle in the *Vehicles* module. This connection leads to the *Suspect* node firing in synchrony with the suspected vehicle node. Non-suspected vehicles initially have a weak excitatory connection to the *Suspect* node, which indicates that these vehicles are not suspects. However, the connection is still present, which means a stronger connection can be learned through associative learning.

In this scenario, the initial suspect vehicle meets with several non-suspect vehicles, and then one of these vehicles travels to another meeting place to meet with another vehicle. The goal is to use the mechanisms of spiking synchronization and Hebbian associative learning to associate the *Suspect* node with the vehicles that meet with a suspected vehicle. This emergent knowledge is then distributed among the nodes and connections of the concept and semantic layers.

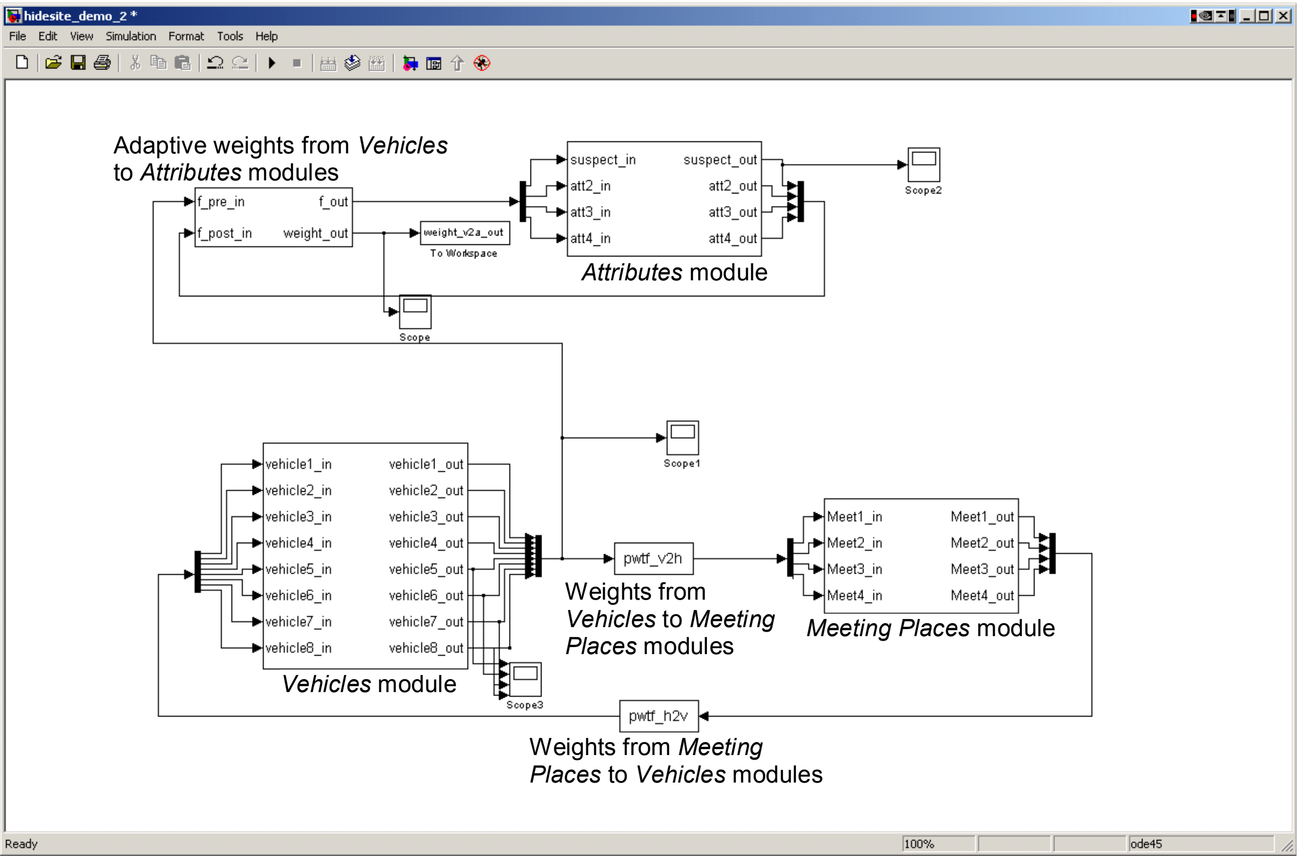

Figure 8. Simulink implementation of knowledge network in Fig. 7 as three modules (*Vehicles, Meeting* Places, and *Attributes)* joined by connectivity blocks that contain weight arrays.

4.3 Example

In this example, the *Vehicles* block contains 8 vehicles, *Vehicle-1* to *Vehicle-8*. A single vehicle, *Vehicle-7*, starts out as a suspect vehicle (represented by a large weight from *Vehicle-7* node to *Suspect* node in the connectivity block from *Vehicles* to *Attributes*), while the other 7 vehicles are non-suspect vehicles. It is assumed that the identity of the vehicles can be maintained as they are tracked from radar GMTI/HRR sensing using our *learn-while-tracking* approach [7].

When a vehicle enters a meeting place location, the weight value from the node representing the vehicle to the node representing the meeting place is temporarily increased, and vice versa. When the vehicle leaves the meeting place, these weight values are returned to zero. This weight change represents a spatial *focus of attention*, indirectly placing excitatory connections between all vehicles that are at a meeting place at the same time.

The sequence of vehicle motion is as follows:

1. *t=200*: *Vehicle-7* and *Vehicle-5* enter *Meeting-Place-1*
2. *t=900:* vehicles leave *Meeting-Place-1*
3. *t=1200: Vehicle-7* and *Vehicle-8* enter *Meeting-Place-2*
4. *t=1800:* vehicles leave *Meeting-Place-2*
5. *t=2100: Vehicle-5* and *Vehicle-2* enter *Meeting-Place-3*
6. *t=2700:* vehicles leave *Meeting-Place-3*

The temporary excitatory connections between the vehicles and meeting place nodes cause these nodes to become transiently synchronized (Fig. 9a). Note that the vehicle nodes become synchronized when their corresponding vehicles are in a meeting place together, according to the scenario listed above. If a vehicle is associated with the *Suspect* node in the *Attributes* block, nodes of other vehicles that are also at the meeting place will become synchronized during this time with the *Suspect* node. This synchronization leads to associative learning between a vehicle node and the *Suspect* node if the synchronization is maintained for a long enough period of time (Fig. 9b). Note that *Vehicle-2* learns an association to *Suspect* from being at a meeting place with *Vehicle-5*, which had earlier learned an association to *Suspect* from being at a meeting place with *Vehicle-7*, the initial suspect vehicle.

Fig. 10 shows the weights from the *Vehicles* nodes to the *Suspect* node at different times. *Vehicle-7* starts with a high weight to *Suspect*, and *Vehicle-5* and *Vehicle-8* become associated with *Suspect* through synchronization with *Vehicle-7.* Then, *Vehicle-2* learns an association with *Suspect* through synchronization with *Vehicle-5.*

When the vehicles leave the meeting place, the weights between the vehicle nodes and meeting place node return to their original low values, the synchronization dissipates, and the vehicle nodes once more become out-of-phase due to the inhibitory connections between them within the *Vehicles* block. However, some of these vehicles are now considered *suspect* by association.

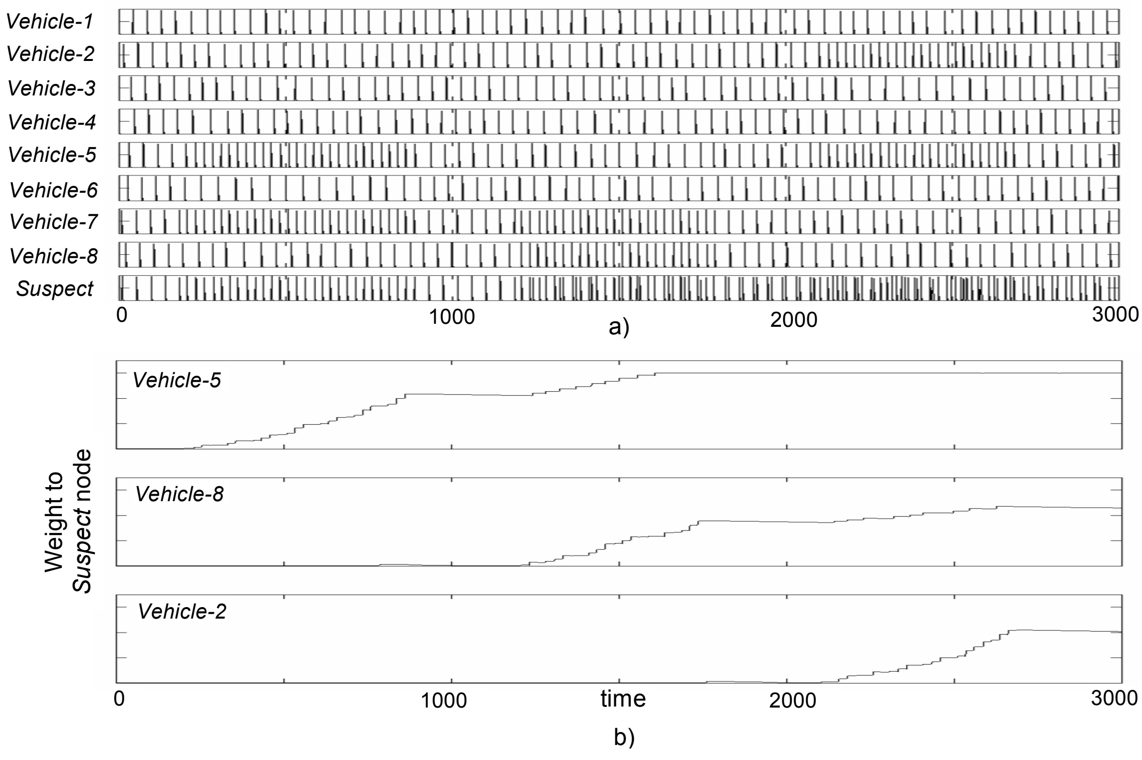

Fig. 9. **a)** Spiking activity of *Vehicle* nodes and *Suspect* node. *Vehicle* nodes become synchronized when vehicles enter a meeting place together according to the scenario above, e.g. *Vehicle-5* and *Vehicle-7* are synchronized from *t=200 to t=900.* **b)** Weight from nodes *Vehicle-5*, *Vehicle-8*, and *Vehicle-2* to *Suspect* node. Note that the weight from a *Vehicle* node to the *Suspect* node increases when it becomes synchronized to a *Vehicle* node that already has a high weight to the *Suspect* node, due to associative learning.

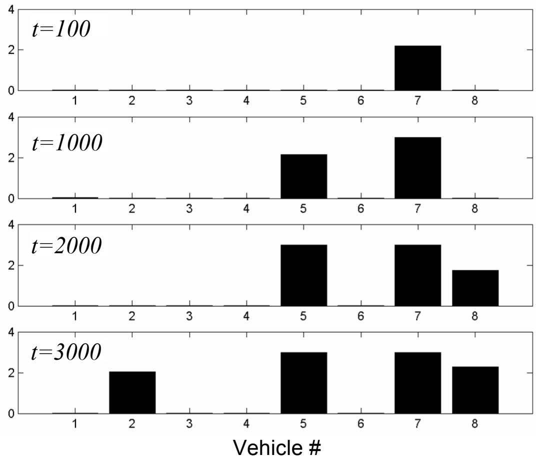

Fig. 10. Weights from *Vehicles* nodes to *Suspect* node, at different times during simulation.

5 Conclusions

In this paper, we have described a novel approach to semantic knowledge representation with networks of spiking neurons. We have demonstrated how these networks can use transient synchronization of semantic item nodes to represent events, and also use this synchronization as a mechanism to drive associative learning between the nodes involved in the event. We have used simple examples to demonstrate these phenomena, but a question for future research is how well these networks can scale to more complex examples.

Our future research will expand the current implementation to more complex semantic knowledge networks, address hierarchical concept learning (as described in Sec. 3.2), and associate learned concepts with outcomes based on prior experience.

Acknowledgements

The material in this paper is based upon work supported by the AFOSR under Contract No. F49620-03-C-0022.

We'd like to acknowledge Rich Ivey and Paul Ilardi for their work on the *learn-while-tracking* and 3-D multisensor visualization software, Nathan Sheldon for his work in developing the traffic simulator for scripting vehicles, and Justin Cutietta and Simon Waxman for their work in constructing the 3-D site model of an urban environment.

References

[1] Mica R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors Journal*, 37 (1): 32-64, 1995.

[2] Wulfram Gerstner and Werner Kistler. Chapter 10: Hebbian models. In *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, pp. 351-385, Cambridge University Press, New York, 2002.

[3] Donald O. Hebb. *The Organization of Behavior*. John Wiley & Sons, New York, 1949.

[4] Michael L. Hinman. Some computational approaches for situation assessment and impact assessment. In *5th International Conference on Information Fusion*, Annapolis, MD, 2002.

[5] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, 117 (4): 500-44, 1952.

[6] David Horn and Irit Opher. Collective excitation phenomena and their applications. In W. Maass and C. M. Bishop, editors, *Pulsed Neural Networks*, pp. 297-320, MIT Press, Cambridge, MA, 1999.

[7] Rich T. Ivey, Allen M. Waxman, Dave A. Fay, and Dan P. Martin. Learn-while-tracking, feature discovery and fusion of high-resolution radar range profiles. In *6th International Conference on Information Fusion*, Cairns, Australia, 2003.

[8] L. Shastri. Advances in *Shruti*: A neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony. *Applied Intelligence*, 11: 79-108, 1999.

[9] L. Shastri and V. Ajjanagadde. From simple associations to systematic reasoning: A Connectionist encoding of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, 16: 417-494, 1993.

[10] Wolf Singer. Synchronization of neuronal responses as a putative binding mechanism. In M. A. Arbib, editors, *The Handbook of Brain Theory and Neural Networks*, 1st ed., pp. 960-964, MIT Press, Cambridge, MA, 1995.

[11] Wolf Singer. Synchronization, Binding, and Expectancy. In M. A. Arbib, editors, *The Handbook of Brain Theory and Neural Networks*, 2nd ed., pp. 1136-1143, MIT Press, Cambridge, MA, 2003.

[12] A. Steinberg, C. Bowman, and F. White. Revisions to the JDL Data Fusion Model. In *Proc. of the SPIE Sensor Fusion: Architectures, Algorithms, and Applications III*, pp. 430-441, 1999.

[13] Christoph von der Malsburg. The what and why of binding: The modeler's perspective. *Neuron*, 24: 95-104, 1999.

[14] Allen M. Waxman, David A. Fay, Brad J. Rhodes, Tim S. McKenna, Neil A. Bomberger, and Val K. Bykoski. Information fusion for image analysis: geospatial foundations for higher-level fusion. In *5th International Conference on Information Fusion*, Annapolis, MD, 2002.

[15] Frank White. Data Fusion Lexicon. Joint Directors of Laboratories, Technical Panel for C^3, Data Fusion Sub-Panel, Naval Ocean Systems Center, San Diego, 1987.