



**HAL**  
open science

# Multichannel Recurrent Kernel Machines for Robot Episodic-Semantic Map Building

Wei Hong Chin, Chu Kiong Loo, Stefan Wermter

► **To cite this version:**

Wei Hong Chin, Chu Kiong Loo, Stefan Wermter. Multichannel Recurrent Kernel Machines for Robot Episodic-Semantic Map Building. 1st SMILES (Sensorimotor Interaction, Language and Embodiment of Symbols) workshop, ICDL 2020, Nov 2020, Valparaiso, Chile. hal-02982754

**HAL Id: hal-02982754**

**<https://inria.hal.science/hal-02982754>**

Submitted on 28 Oct 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Multichannel Recurrent Kernel Machines for Robot Episodic-Semantic Map Building

Wei Hong Chin\*, Chu Kiong Loo<sup>†</sup> and Stefan Wermter<sup>‡</sup>

\*Graduate School of Systems Design

Tokyo Metropolitan University, Tokyo, Japan 191-0065

Email: chinweihong@ieee.org

<sup>†</sup>Faculty of Computer Science & Information Technology

University of Malaya, Kuala Lumpur, Malaysia 50603

Email: ckloo.um@um.edu.my

<sup>‡</sup>Department of Informatics

University of Hamburg

Hamburg, Germany 22527

Email: wermter@informatik.uni-hamburg.de

**Abstract**—For humans to understand the world around them, multimodal integration is essential because it enhances perceptual precision and reduces ambiguity. Computational models replicating such human ability may contribute to the practical use of robots in daily human living environments. In this paper, we propose Multichannel Recurrent Kernel Machines (MC-RKM) for continuously generating a topological semantic map from multiple sensors. The proposed method consists of two hierarchical memory layers: i) Episodic Memory and ii) Semantic Memory layer. Each layer consists of one or more than one Infinite Echo State Network with a different learning task. The Episodic Memory layer incrementally clusters incoming sensory data as nodes and learns fine-grained spatiotemporal relationships of them. The Episodic Memory layer learning is in an unsupervised manner. The Semantic Memory layer utilizes task-relevant cues to adjust the level of architectural flexibility and generate a topological semantic map that contains more compact episodic representations. The generated topological semantic map represents the memory of the robot in which it is used for robot path planning and navigation.

## I. INTRODUCTION

With the exponential growth of autonomous robotics [2], intelligent robots are gradually evolving from the conventional organized industrial production world to the unstructured, diverse and mutual work-space of co-existence, cooperation and integration with human beings. Although many robot control algorithms with greater robustness and stability have been developed in the last decade [1], but they cannot adequately model the unstructured world, unpredictable irregular events are likely to occur [3], [4].

A generally recognized approach is recently proposed to provide a robot with a longer range of autonomy and a safer human-robot collaboration system by combining a robot with multiple sensors for sensing the interactive environment and achieving self-monitoring of external disturbances by evaluating multimodal sensory signals in real time [5].

Echo State Networks can be considered as large, randomly recurring neural networks with a single sequential, trained readout layer. The network computes a wide range of non-

linear, spatial-temporal mappings of input data. The reservoir can be seen as a spatial-temporal kernel in which the mapping of a high-dimensional space is explicitly computed. [6] proposed a Recurrent Kernel Machines (RKM) that extend the idea of Echo State Networks to infinite sized recurrent neural networks. The proposed method regarded as recursive kernels.

Consider a recurrent network (RNN) with internal weights  $W$ , input weights  $V$ , and internal state  $s$ . When receiving input  $x_t$  at time  $t$ , the RNN output as follows:

$$y_t = h(Vx_t + Ws_t) \quad (1)$$

where  $h$  is a combination of the activation function (such as the hyperbolic tangent) and the projection function. The basic principle behind a recursive method is that Equation (1) can be written as follows:

$$h(Ws_t + Vx_t) = h\left([W|V] \begin{bmatrix} s_t \\ x_t \end{bmatrix}\right) \quad (2)$$

and it is a function of the concatenation of the input with the previous internal state. The same logic can also be extended to kernel functions that the basis function inputs are a concatenation of the current input and the previous recursive mapping,

$$\phi(x_t, \phi(x_{t-1}, \phi(\dots))) = \phi([x_t | \phi(x_{t-1} | \phi(\dots))]). \quad (3)$$

Using this structure as a reference, [6] have shown that recursive variants of kernels with  $k(x, x') = f(\|x - x'\|^2)$  and  $k(x, x') = f(x \cdot x')$  form can be derived. The recursive-SE kernel, for example, has the form

$$\kappa_t^{SE}(x, x') = \exp\left(-\frac{\|x_t - x'_t\|^2}{2l^2}\right) \exp\left(\frac{\kappa_t^{SE}(x, x') - 1}{\sigma_p^2}\right). \quad (4)$$

By taking the properties of RKM, we proposed a computational model termed as Multi-channel Recurrent Kernel

Machines (MC-RKM) for continuously generating topological map. The proposed method consists of two hierarchical memory layers which is the Episodic Memory layer and Semantic Memory layer. Following sections explain the details of the MC-RKM.

## II. PROPOSED METHOD

The proposed method generates new nodes and topology connections according to novel consecutive sensory input in each layer. The Episodic Memory layer (EML) consists of one Infinite Echo State Network [6], [7] that clusters incoming input vectors as nodes in an unsupervised manner. Besides, EML further learns the nodes' activation patterns for spatiotemporal encoding. Next, the Semantic Memory layer (SML) receives neural activation trajectories from each network of EML and task-relevant signals (labels) from users to update the network and generate more compact representations of episodic experience with semantic meaning. Thus, the EML and SML mitigate catastrophic forgetting through continually generated nodes if the network encounters novel inputs and updated nodes' weights if the received input is similar to previously learned knowledge. Also, inactive or outlier nodes will be eliminated for maintaining the storage capacity.

In robot navigation, EML acts as a novelty detector where each node in the network represents a group of similar input vectors and generates new nodes if the incoming input vectors do not fit into the network. Besides, EML encodes the robot's movement sequence, and each node in the network stores the robot's position for localization.

In SML, the network encodes a set of nodes in EML to define the location of the explored world. Each node in SML represents an area of the world and the robot uses information to adjust its moving behaviors, such as wall following, obstacle-avoidance or fast-speed movement. The overall architecture is shown in Figure 1.

### A. Episodic Memory Layer (EML)

The EML has an Infinite Echo State Network. The network dynamically expands or shrinks in response to input vectors. New nodes will be generated to represent incoming inputs and links will be generated to connect nodes.

The episodic layer of memory acquires different sensory data. Thus, the episodic layer of memory encodes multiple input data. In the previous work, the authors only concatenate all outputs of episodic networks as inputs to semantic networks. However, this will cause high-dimensional data to be more dominant than low-dimensional data when selecting the winning node. Thus, we introduce a multichannel architecture to evaluate the winning node of the episodic network and modify the activation equation for the node. The notations of the MC-RKM is tabulated in Table I.

Initially, the network produces 2 recurrent nodes based on the sensory input obtained. Each neuron of the layer consists of the weight vector  $w_j$ . For subsequent learning, the network determines the node that best fits the current sensory  $x(t)$  using equation 5-7.

TABLE I  
THE MC-RKM NOTATIONS

Notation	Definition
$c$	Number of channel
$T_j(t)$	Activation value of node $j$ at $t$
$\kappa(t)$	Recursive kernel at $t$
$w_b(t-1)$	Best matching node weights at $t-1$
$\alpha^c \in [0, 1]$	Significance factors
$r_j$	Regularity counter of node $j$
$\gamma_j$	Contributing factor of node $j$
$\tau_j, \lambda$	Decay factors for regularity counter
$\sigma_i^2, \sigma^2$	Kernel width
$\rho$	Learning threshold
$P_{(m,n)}$	Temporal connection between node $m$ and $n$
$V$	Associative matrix for labeling
$b$	Index of best matching node

$$b = \arg \min(T_j(t)), \quad (5)$$

$$T_j(t) = \sum_{c=1}^C \alpha^c \cdot \kappa^c(t), \quad (6)$$

$$\kappa^c(t) = \exp \left[ -\frac{\|x^c(t) - w_j^c\|^2}{2\sigma_i^2} \right] \exp \left[ \frac{\kappa^c(t-1) - 1}{\sigma^2} \right]. \quad (7)$$

Equation 7 creates the Infinite Echo State which is identical to Eq. 4.

Next, the activation value of the best matching node  $J$  is computed as follows:

$$a_b(t) = \exp(-T_b) \quad (8)$$

if the activation value  $a_b(t)$  is less than a preset threshold  $a_T$ . A new node  $N$  is added to the network with the new weights as below:

$$w_N = 0.5 \cdot (x(t) + w_b) \quad (9)$$

A new link is created to connect the winner node  $b$  and the second best matching node. If the  $a_b(t)$  is larger than  $a_T$ , it means that the winner node  $b$  can represent the input  $x(t)$ . Thus, the winner node  $b$  and its neighbor nodes  $n$  are updated according to input  $x(t)$  as follows:

$$w_{j(\text{new})} = \gamma_j \cdot r_j \cdot (x(t) - w_{j(\text{old})}) \quad (10)$$

If there is no connection between the best matching node  $a_b(t)$  and the second best matching node, a new connection will be established to connect it. Each edge has an age counter that increases by one on each iteration. The age of the link between the best matching node and the second best matching node is set back to zero. Connections with an age greater than the preset threshold will be excluded, and nodes without connections and their habituation counter greater than the preset value will be removed from the network.

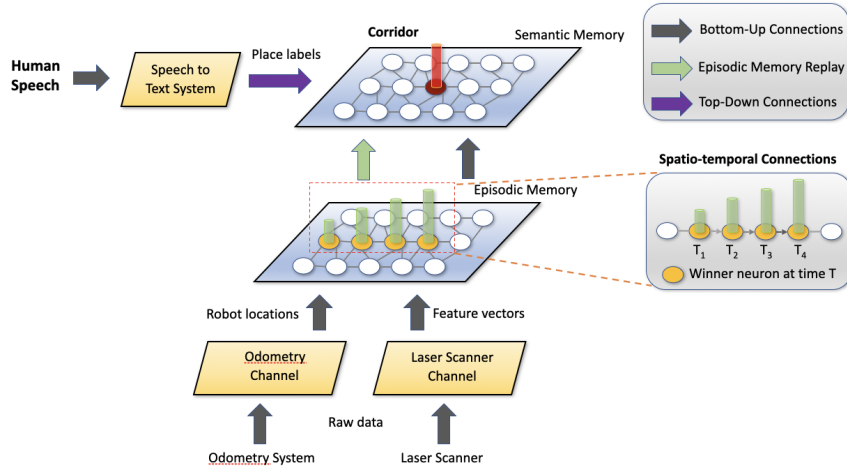


Fig. 1. The Multichannel Recurrent Kernel Machines overall architecture

In addition, each episodic node contains a regularity counter  $r_j \in [0, 1]$  indicating its fire intensity over time. The newly created episodic node is valued at  $r_j = 1$ . In each iteration, the regularity value of the best matching node and its neighbor nodes decreases by using the following equation:

$$\Delta r_j = \tau_j \cdot \lambda \cdot (1 - r_j) - \tau_j \quad (11)$$

Thus, the significance of the regularity of the node can be correlated with the significance or importance of the information stored in the node. Nodes that have been frequently triggered in response to learning inputs will usually have a lower regularity value, given in the regularity equation 11. Isolated nodes would be disconnected from its network if the link reaches the threshold. In the robot navigation mission, the topological network grows along the robot's path, nodes created from the start of the journey will be removed from its network. Thus, we have incorporated a new form of node removal [8] with the following equation:

$$v = \mu(H) + \sigma(H) \quad (12)$$

where  $H$  is a vector representation of the regularity of all the nodes in the network,  $\mu$  is the mean function, and  $\sigma$  is the standard deviation. Nodes with regularity values above the threshold will be deleted.

A new episodic node can only be connected to the network if  $b_J(t) < \rho_b$  and  $r_J < \rho_r$ . If the threshold value of activation and regularity is satisfied, the episodic nodes will be updated using equation 10.

In the EML, a series of events forms an episode that stores unique past occurrences and episodes that are connected to each other. We introduce temporal connections to learn recurrent node activation patterns in the network.

Temporal links encode the series of nodes that have been triggered during the learning stage. For each learning iteration, a temporal connection between two sequentially triggered

nodes will be increased by 1. Specifically, when the best matching node  $b$  triggered at time  $t$  and then  $t-1$ , the temporal relation between them is reinforced as follows:

$$P_{(b(t), b(t-1))}^{\text{new}} = P_{(b(t), b(t-1))}^{\text{old}} + 1 \quad (13)$$

In this way, for each recurrent node  $m$ , the next node  $g$  can be retrieved from the encoded time series by choosing the largest value of  $P$  as below:

$$g = \arg \max P_{(m, n)} \quad (14)$$

where  $n$  are the neighbors of  $m$ . As a consequence, the recurrent node activation sequence can be restored without needing any input data.

### B. Semantic Memory Layer (SML)

The semantic memory layer is hierarchically connected to the episodic memory layer. It consists of an Infinite Echo State Network that receives bottom-up inputs from an episodic memory layer, top-down inputs such as labels or tags for the creation of representations that comprise semantic information on a wider timescale. Semantic information can be extracted by supplying signals from the top-down signals.

The mechanism of neural activity in the semantic memory layer is analogous to an episodic memory layer with an additional requirement for creating a new node. In this layer, node learning occurs when the network correctly predicts the class label of the labeled input sequence from the episodic memory layer via the learning process. A new node will be generated if the predicted network class label is wrong. This is therefore the additional factor that modulates the rate of update of the nodes. In addition, each semantic node stores information over time sequences higher than episodic nodes due to the hierarchical learning of input data.

Thus, the semantic network determines the winning node based on the best matching node (BMN) of episodic network as follows:

$$b_s = \arg \min(T_j^{\text{sm}}(t)), \quad (15)$$

$$T_j^{\text{sm}}(t) = \exp \left[ -\frac{\|x(t) - w_j\|^2}{2\sigma_i^2} \right] \exp \left[ \frac{T_j^{\text{sm}}(t-1) - 1}{\sigma^2} \right]. \quad (16)$$

With these equations, the chosen node is either supposed to be the right semantic node for the particular sequence of episodic inputs, or it is more prominent than other semantic nodes or both.

The semantic memory layer receives the input neural data from the episodic memory layer, i.e. the BMNs of the episodic memory layer with respect to  $x(t)$ . The BMNs in this layer are determined using equations 15-16. The input is from bottom-up neural episodic weights, so  $x(t)$  is replaced by  $w_b^{\text{em}}$  for node learning. The labeling approach is similar to episodic memory layer where each node in semantic memory layer is assigned to a label that obtains from  $x(t)$  using equation 18 and 19.

As a result, a new semantic node is generated only if the BMN  $b$  does not meet three requirements: 1)  $a_b^{\text{sm}}(t) < \rho_a$ ; 2)  $r_b^{\text{sm}} < \rho_r$ ; 3) BMN's label  $\zeta_b^{\text{sm}}$  is not identical with the data input's label  $\zeta$  (equation 20). Notice that this label matching condition in the semantic memory layer is not taken into account if the data input is not labeled. If the winner of the semantic node  $b$  predicts the label  $\zeta_b$  same with the class label  $\zeta$  of input  $x(t)$ , the node learning process is triggered by additional learning factor  $\psi = 0.001$ . Therefore, Equation 10 will then become:

$$w_{j(\text{new})}^{\text{sm},c} = \psi \cdot \gamma_j \cdot r_j \cdot (w_b^{\text{em},c} - w_{j(\text{old})}^{\text{sm},c}) \quad (17)$$

As such, the semantic memory layer learns more compact representations of data input labels. Data labels regulate layer stability and plasticity where new semantic nodes are created only if the network is unable to predict the correct data input class label. In addition, the network learning rate of bottom-up measurements decreases if the class prediction is accurate.

### C. Data Labeling

During the learning process, a class label of  $l$  from the input data can be sequentially assigned to each recurrent node. The  $l$  label is obtained from the  $L$  class label. For this labeling method, the frequency of each individual label in the network is stored in the  $V(j, l)$  associative matrix. This means that each recurrent node  $j$  contains a distribution counter that stores the frequency of a given sample label that has been assigned to it. When a new node  $N$  is generated and the label  $\zeta$  associated with the input data  $x(t)$  is given, the matrix  $V$  is increased by one row and initialized on the basis of  $V(N, \zeta) = 1$  and  $V(N, l) = 0$ . When an existing BMN  $b$  is selected for updating, the  $V$  matrix is updated as follows:

$$V(b, \zeta)_{(\text{new})} = V(b, \zeta)_{(\text{old})} + \varphi^+ \quad (18)$$

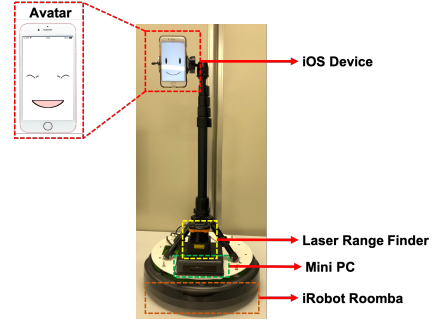


Fig. 2. Robot attached with various sensors

$$V(b, l)_{(\text{new})} = V(b, l)_{(\text{old})} + \varphi^- \quad (19)$$

Notice that  $\varphi^+$  must always be smaller than  $\varphi^-$  and the label  $\zeta$  is within the  $L$  class label. If the data label  $\zeta$  does not exist in  $L$ , a new column in  $V$  is inserted and initialized to  $V(b, \zeta) = 1$  and  $V(b, l) = 0$ . If there is no label associated with the given input data, the matrix  $V$  will not be updated. Thus, the winning label  $\zeta_j$  for a node  $j$  can be calculated as follows:

$$\zeta_j = \text{label}(j) \equiv \arg \max V(j, l) \quad (20)$$

where  $l$  is label in class label  $L$ .

The advantage of this labeling method [9] is that there is no need to predefine the number of Class Labels in advance. This is important when dealing with learning activities as its number of classes is unspecified.

### III. EXPERIMENTAL SETUP AND RESULTS

We validate our proposed method using a iRobot mobile robot that attached with a Hokuyo Laser scanner, and Intel i5 processor mini-computer. The laser scanner signal was sampled at 10 Hz. Since the robot has to traverse the environment autonomously, we developed a Fuzzy motion movement behavior that allows the robot for obstacle avoiding and wall-following. The moving speed of the robot varies from 0.05m/s to 0.5m/s. To generate a topological map, the EML receives data from the laser scanner and odometry while the SML receives EML outputs. For the task of classification and prediction, we first determine the best matching node in the SML. The predicted label can be retrieved from the best matching node of the SML.

The experiments were conducted in the 7th floor of university corridor, study area, and rest area that connecting with each other. The grid map of the experimental place as shown in Figure 3(a). We conduct the experiment in such environmental conditions is to validate our proposed method is able to work in natural environment with moderate changing of environmental conditions. The parameters setting of the MC-RKM for the experiments is tabulated in Table II.

We commanded the robot to traverse the experimental place starting from the study area and travel to the rest area

TABLE II  
PARAMETER SETTINGS

Parameter	Value
$\alpha^1, \alpha^2$	0.5
$\rho_a$	0.75
$\rho_r$	0.1
$\tau_j$	0.5
$\lambda$	1.05
$\sigma_i^2, \sigma^2$	1.1, 1.4
$\gamma_b$	0.2
$\gamma_n$	0.001
$r_e$	0.001

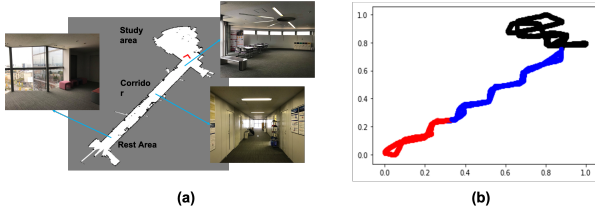


Fig. 3. (a) Experimental environment (b) Robot initial navigation path

through the corridor then back to the start point again. During the traverse, MC-RKM continually learns incoming sensory information and generates the semantic map for representing the environment. For space labeling, we developed a speech-to-text iOS application to obtain the space labels for labeling the semantic map. As shown in Figure 3, the semantic map is segmented into 3 regions, the study area is colored in black and the corridor is colored in blue and rest area colored in red respectively.

After the first traverse, with the semantic information, the robot navigated the environment with different moving behaviors according to the region. For example, the robot is switched to obstacle avoidance mode in the study area where the place populated with moving people and furniture. When it traversed to the corridor, the moving behavior is then switched to the wall following mode and fast-speed mode since the corridor is a straight path.

We repeated the experiment for ten times and the quality of the generated episodic and semantic map was measured by Total Quantization Error (TQE) and localization rate. TQE measures the similarity between the received sensory information and the weights of the nodes of the EML and EML. Figure 4 shows the TQE of the network for each traverse. Next, the localization rate is measured by computing the euclidean distance between the winner node's encoded location and robot current location that obtained from the SLAM algorithm. The robot is localized successfully if the euclidean radius is within a threshold value ( $0.1m$ ) and the winner node's encoded label same with the ground-truth label. The localization result is shown in Figure 5. Figure 6 and 7 shows the final episodic map and semantic map generated by the MC-RKM.

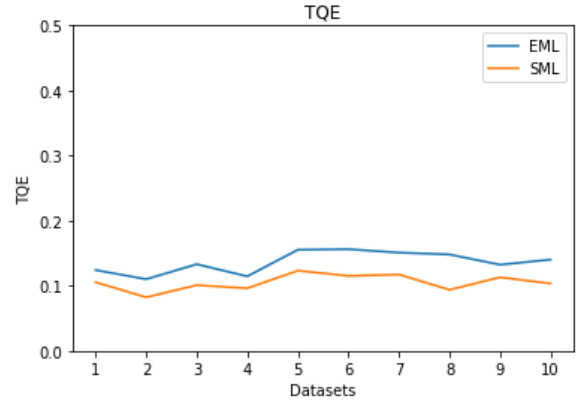


Fig. 4. The Total Quantization Error for each traverse

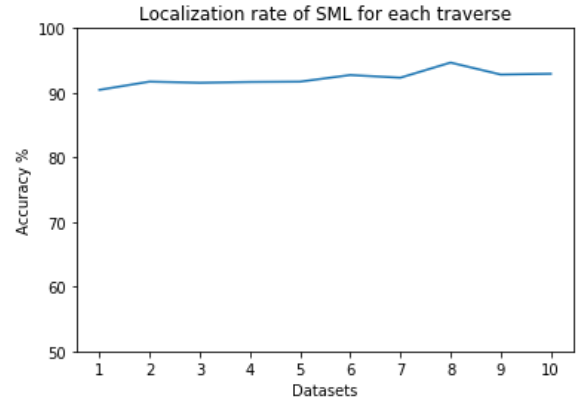


Fig. 5. The localization rate of for each traverse

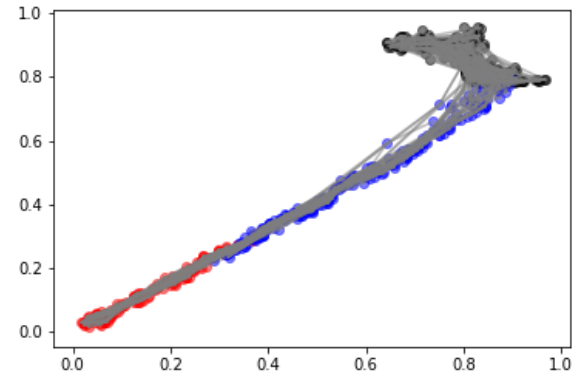


Fig. 6. Episodic memory map generated by the MC-RKM

#### IV. DISCUSSION

The MC-RKM was validated in actual robot experiments. Results have shown that the proposed system is capable of producing topological maps as the robot travels the environment. However, the values of certain parameters of the MC-RKM have been chosen empirically. Further studies are required to determine the optimum value of the parameters. Furthermore, the effectiveness of the topological map produced is not

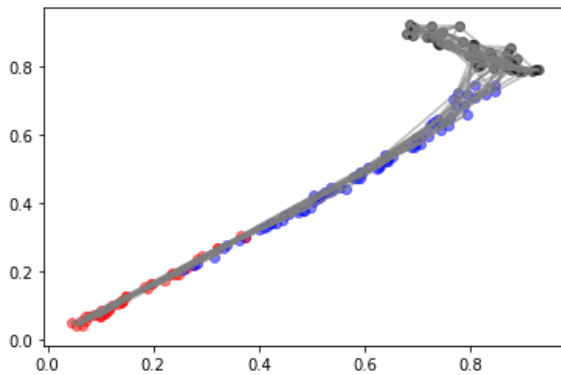


Fig. 7. Semantic memory map generated by the MC-RKM

evaluated in the paper as we concentrate on the development of the model.

## V. CONCLUSION

In this paper, we propose Multichannel Recurrent Kernel Machines called MC-RKM to continuously generate a topological episodic-semantic map from multiple sensors. Experimental results showed that MC-RKM was able to produce a topological map while the robot was exploring the environment. For future work, we aim to implement optimization algorithms to determine the optimal value for the parameters. In addition, we intend to carry out a path planning system to verify the usefulness of the topological map. Finally, we will conduct experiments in more complex and unstructured environments to further verify the feasibility of the proposed approach.

## ACKNOWLEDGMENT

This research was supported by the Georg Forster Research Fellowship for Experienced Researchers from Alexander von Humboldt-Stiftung/Foundation, and the ONRG International Fund (IF017-2018) and University of Malaya and International Collaboration Fund for project Developmental Cognitive Robot with Continual Lifelong Learning (IF0318M1006) from MESTECC, Malaysia.

## REFERENCES

- [1] W. H. Chin, Y. Toda, N. Kubota, C. K. Loo and M. Seera, "Episodic Memory Multimodal Learning for Robot Sensorimotor Map Building and Navigation," in *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 2, pp. 210-220, June 2019, doi: 10.1109/TCDS.2018.2875309.
- [2] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier and P. Fraise, "Collaborative manufacturing with physical human-robot interaction", *Robot. Comput. Integr. Manuf.*, vol. 40, pp. 1-13, Aug. 2016.
- [3] D. Park, H. Kim, Y. Hoshi, Z. Erickson, A. Kapusta and C. C. Kemp, "A multimodal execution monitor with anomaly classification for robot-assisted feeding", *Proc. IEEE Int. Conf. Robots Syst. (IROS)*, pp. 5406-5413, Sep. 2017.
- [4] W. H. Chin, N. Kubota, C. K. Loo, Z. Ju and H. Liu, "Spatial Map Learning with Self-Organizing Adaptive Recurrent Incremental Network," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-6, doi: 10.1109/IJCNN.2019.8851919.
- [5] D. Kappler et al., "Data-driven online decision making for autonomous manipulation", *Proc. Robot. Sci. Syst. (RSS)*, pp. 1503-1510, Jul. 2015.
- [6] M. Hermans and B. Schrauwen, "Recurrent Kernel Machines: Computing with Infinite Echo State Networks," *Neural computation*, 24, 104-33, 2011, 10.1162/NECO\_a\_00200.
- [7] H. Soh and Y. Demiris, "Spatio-Temporal Learning With the Online Finite and Infinite Echo-State Gaussian Processes," *IEEE transactions on neural networks and learning systems*, 26, 2014, 10.1109/TNNLS.2014.2316291.
- [8] W. S. Liew, C. Kiong Loo, V. Gryshchuk, C. Weber and S. Wermter, "Effect of Pruning on Catastrophic Forgetting in Growing Dual Memory Networks," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-8, doi: 10.1109/IJCNN.2019.8851865.
- [9] I. Parisi German, J. Tani, C. Weber and S. Wermter, "Lifelong Learning of Spatiotemporal Representations With Dual-Memory Recurrent Self-Organization," *Frontiers in Neurobotics*, vol. 12, pp. 78, 2018.