2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Reinforcement Learning in Non-Stationary Environments: An Intrinsically Motivated Stress Based Memory Retrieval Performance (SBMRP) Model

Tiong Yew Tang, Simon Egerton and Naoyuki Kubota

Abstract— **Biological systems are said to learn from both intrinsic and extrinsic motivations. Extrinsic motivations, largely based on environmental conditions, have been well explored by Reinforcement Learning (RL) methods. Less explored, and more interesting in our opinion, are the possible intrinsic motivations that may drive a learning agent. In this paper we explore such a possibility. We develop a novel intrinsic motivation model which is based on the well known Yerkes and Dodson stress curve theory and the biological principles associated with stress. We use a stress feedback loop to affect the agent's memory capacity for retrieval. The stress and memory signals are fed into a fuzzy logic system which decides upon the best action for the agent to perform against the current best action policy. Our simulated results show that our model significantly improves upon agent learning performance and stability when objectively compared against existing state-of-the-art RL approaches in non-stationary environments and can effectively deal with significantly larger problem domains.**

I. Introduction

INTRINSIC *motivation* in learning agents was a term first coined by Harlow [1] during 1950. Harlow argued that an intrinsic manipulation drive is needed to explain why mammals persistently solve problems, such as finding routes through mazes, solving puzzles, seemingly without any external stimulation or reward. In recent years the idea of applying intrinsic motivation to RL has gained a lot of interests, especially in situations where the environment conditions are non-stationary, especially were the environment is optimised around emotions, complex interactions, multi-agents, particularly when the agents can transfer knowledge between themselves. In this paper we introduce a novel model which is inspired from biology.

Early RL research began with model-free approaches such as the Q-learner [2]. However model-free approaches are known to suffer from the sample inefficiency issue [3] where the agent requires a large volume of samples to learn a solution policy. RL research has developed considerably from these early years; a current trend is model-based intrinsically motivated models [4]. Fig. 1 shows the differences between a model-free and model-based intrinsically motivated RL approach, the key difference being the internal world memory representation of the agent, or the agents mind, if you will. The internal memory representation of the world is considered by the agent before any action is decided upon or carried out. Information from the agent's external sensor no longer directly affects the agent but is integrated into the agent's internal memory model. What our model does is effect the memory model proposed by Singh et al. [4] with biological stressors. The stress model, based on the Yerkes and Dodson [5] stress curve theory shrinks and contracts the memory model to effectively group actions as either deliberative sets (low stress) or reactive sets (high stress). In high stress situations the agent may ask for help to reduce stress. The help is intended to reduce stress when faced with dynamic situations which prevent the agent from carrying out its goal seeking tasks.

The critic term in model-free approach for Fig. 1 on the left is describing the reward function from external environment. Fig. 1 on the right explains the reward signals for animal. The reward signals term which is specifically used in intrinsic motivated model where rewards signals are triggered from agent's internal (intrinsic) neural signals. This comparison diagram in Fig. 1 had been adapted from Singh et al. [4] with additional stress model elements for our model perspective. The agent's input sensation can be divided into two parts which are concept detection and stress sensation. Agent may experience pain in stressful environment and thus it is treated as environment feedback in our proposed model. Concept detection is represented as a set of detected features about the obstacle object abstract class, class and concept in our model. The reason we have 3 categories of features for our model is because in physical world where image concept detection clarity will always be influenced by its environment noise and agent movements. Furthermore, action output from the learning agent can be normal (non-stressful) or reactive action according to the agent environment's condition. In the centre of the right Fig. 1, the learning agent's internal environment is represented as previous actions in the learning agent's memory and its online neural network machine learning. In our proposed model, the online neural network is used to predict agent's future actions with agent's previous correct actions. In additional to that, the learning agent's internal memory environment is also controlled by the *memory retrieval performance* corresponding to agent's current stress level (based on Lupien et al. [6] discovery) which will be discussed later.

Nature had designed biological stress system within the organism (e.g. Hypothalamic-Pituitary-Adrenal (HPA) Axis [7]) to effectively solve uncertainty in non-stationary en-

Tiong Yew Tang and Simon Egerton are with the School of Information Technology, Monash University Malaysia, Sunway, Malaysia (email: {tang.tiong.yew, simon.egerton}@monash.edu).

Naoyoki Kubota is from Tokyo Metropolitan University, Faculty of System Design, Graduate School of System Design (email: kubota@tmu.ac.jp)

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

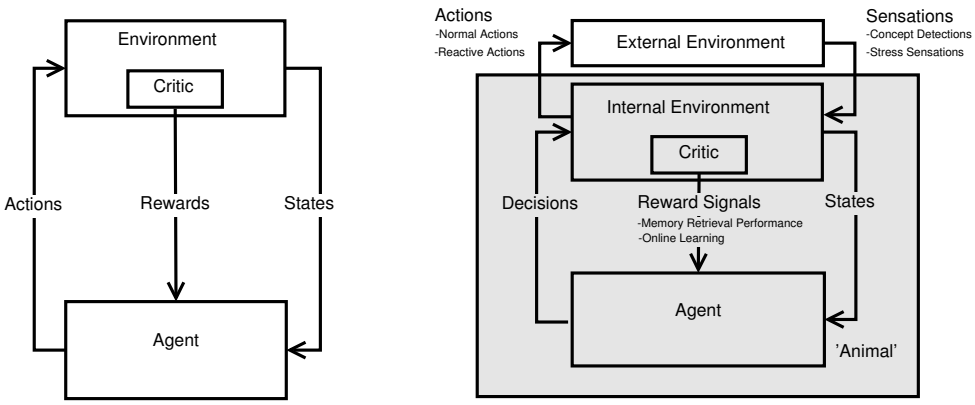

Fig. 1. This figure compares the difference between a model-free RL framework to intrinsically motivated model RL framework adapted from Singh et al. [4].

vironment problem. Unknown changes in non-stationary environment will cause stress to organism according to Mason [8]. This observation motivated us to apply biological stress inspired model to non-stationary RL agent problem. This is the main reason we use biological stress inspired model for artificial learning agent because biological stress system consist of similar properties with intrinsic motivated model where both models have internal memory environment representation.

Our proposed approach has three contributions. 1) SBMRP framework had been applied to the non-stationary environment RL with reasonable efficient performance gain when it is compared with other approaches. Please refer to Fig. 7 where proposed SBMRP achieved lowest steps required for each trial. 2) Furthermore, SBMRP framework had maintained its performance stability when it is compared with other approaches. Please refer to Fig. 8 where the SBMRP's 95% confidence interval variations (the coloured area) are lower than other approaches. The SBMRP also maintain its horizontal slope steps per trial stability performance throughout its experiment execution. 3) In addition to that, our proposed method had gradually reduced the action-state complexity. The reduced action-state complexity is refer to the introduction of three action categories where certain actions can only be activated during certain stress conditions (Please refer to section III), therefore the learning agent will only consider less possible actions to be selected during its learning. For example, $normalAction$ category is only consider of 5 possible actions, where else $cognitiveAction$ and $randomAction$ categories are having total of 55 possible actions to be considered.

Scaffolding minds theory was first introduced by Clark [9]. This theory is about complexity reduction in learning environment for learning agent with environment supports. In scaffolding mind perspective, it emphasizes on continuous concept detection for agent's decision making process where it has similar perception-action properties in our proposed approach. Clark had argued that agent's intelligence cannot exist without taking into account of the whole agent's body functionality (not just the agent's brain functions). Therefore agent's total action capabilities (e.g. speaking, movement) are taken into account in RL in our approach. Agent's total executable actions are limited by its physical body stress conditions and environment conditions according to scaffolding mind theory. For example, during a stressful condition the effected agent will be internally triggered its *allostatic process* [10] to enable the agent to execute extraordinary actions such as shut down of immune system, increase heart rate and increase muscle stamina to mitigate threats. These additional stress-enabled actions that is not executable during normal (non-stressful) environment condition. Therefore, the learning agent action-state complexity that is controlled by its stress level and total executable actions are reduced by limiting total actions to be select in different stress conditions. Thus, action-state complexity can be reduced for learning agent.

Yerkes and Dodson [5] are the researchers who first discovered the important relationship between stress arousal levels with cognitive performance (it is also known as the *stress curve*) in the field of biological stress and anxiety research. Since then many researchers continue to further investigate on Yerkes and Dodson stress curve discovery. Lupien et al. [6] investigated the stress curve relationship between stress arousal levels with memory retrieval performance (See Fig. 2). The stress curve relationship is the another important motivation for our proposed SBMRP approach in the perspective of scaffolding minds because the memory retrieval performance of the agent is known by corresponding agent's stress level. Therefore, the level of agent's mind scaffolding (memory processing) intensity will be based on Yerkes and Dodson's stress curve relationship. Referring to Fig. 2, the learning agent during low stressful condition will have minimum memory retrieval performance, middle stressful condition will trigger maximum retrieval performance and high stressful condition will have minimum memory retrieval performance.

II. Literature Review

RL problem is commonly represented in the form of Markov Decision Process (MDP) [12]. In MDP environment,

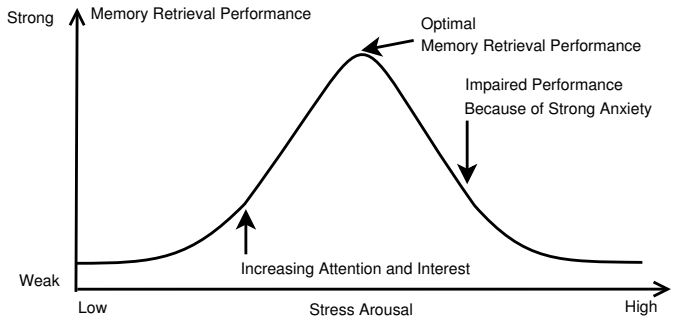

Fig. 2. This is a Hebbian version of Yerkes and Dodson curves [11] that explains the non-linear relationship (*stress curve*) between simulated arousal levels against memory retrieval performance of an organism.

we assume the last observation is representing a summary of history and thus the agent's state can be observes in the last observation. RL methods (e.g. Q-learning [13]) are utilized to obtain an optimal action-selection policy for any given finite MDP. These RL methods will learn an action-value function that will returns the expected utility of executing a given action in a given state and following the optimal policy thereafter. Therefore, if such action-value function is learned, the optimal policy can be determined by selecting the action with the highest value in each state in action-state space.

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(s_t, a_t) \left[R_{t+1} + \gamma \underset{a}{max} Q_t(s_{t+1}, a) - Q_t(s_t, a_t)\right] \quad (1)$$

Let denotes S as the set of states, R as the reward function and A as the set of actions. Then, $Q : S \times A \rightarrow \mathbb{R}$, so that Q is the value set which S and A that are mapped to real value $\mathbb{R}$. Next s is denoted as agent's state and time step is defined as t. The action-state complexity of the RL problem is determined by $S \times A$. Then a is defined as the agent's action, lets $a \in A$ where A are the executable actions for agent and R is reward. Furthermore, α is denoted as agent's learning rates and γ is the discount factor. Then, R_{t+1} is the reward being observed after execute a_t in s_t and where $\alpha_t(s_t, a_t)$ $(0 < \alpha \leq 1)$ is the learning rate that may be the same for all pairs. Thus Q-learning (PlainQ) can be formulated as equation 1.

State Action Reward State Action (SARSA) [13] (Equation 2) is the reduced complexity version of Q-learning where $max\ Q$ selection is omitted to improve computational performance in RL. In theory, SARSA approach will perform its computation tasks faster but faster in solution convergence under certain conditions when it is compared to Q-learning approach [13]. The Q-learning and SARSA are the computational efficient but they are not sample efficient RL approaches. The reason is they are model-free approaches and did not maintain a model during its RL process to predict future action with previous actions in the memory (model). Sample efficiency problem as described by Hester [3] is very important to enable feasibility in physical robot applications where reduced sample training can save huge amount of learning time (thousands of physical robot's actions is slow) in physical world.

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha [R_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)] \quad (2)$$

There is another multi-agent RL method is called Friend-or-Foe Q (FFQ) proposed by Littman [14] (Equation 3). FFQ approach uses the two special Nash equilibria conditions to determine best reward for the learning agents. Yet, these special conditions will cause heavy memory utilization by calculating all possible condition pairs for each agent and obstacle agent during experiment simulation. *Obstacle agent* is an agent that will cause non-stationary effects in experiments. Let's denote π as the policy, a_1, a_2 are the agent pairs to be considered. As stated in equation 3 where the equation consider all possible agent pairs conditions to determine the optimal Q_1, Q_2 values according to the state s.

$$Nash_1(s, Q_1, Q_2) = \underset{\pi \in \Pi(A_1)}{max} \underset{a_2 \in A_2}{min} \sum_{a_1 \in A_1} \pi(a_1) Q_1[s, a_1, a_2] \quad (3)$$

Busoniu *et al.* [15] had proposed Adaptive State Focus Q-learning (ASFQ) for multi-agent RL. ASFQ is constructed based on Q-learner RL algorithm, then it is expanded with multi-agent implementation.

$$Q^i_{t+1}(s^i_t, a^i_t) = (1 - \alpha^i_t) Q^i_t(s^i_t, a^i_t) + \alpha^i_t [r^i_{t+1} + \gamma \underset{a^i \in A^i}{max} Q^i_t(s^i_{t+1}, a^i] \quad (4)$$

Let i as the agent identity index and $S = S^1 \times \ldots \times S^n$ is the complete state is the concatenation of all the agent's current state vectors), the sizes and structure of the i^{th} agents' Q-tables before and after the expansion:

$$Q^i_{before}(s^i, a^i), \quad dim(Q^i_{before}) = |S^i|.|A^i| \quad (5)$$

$$Q^i_{after}(s^1, \ldots, s^n, a^i), \quad dim(Q^i_{after}) = |S^1| \ldots |S^n|.|A^i| \quad (6)$$

where $|.|$ is the cardinality of a set. If $Q^i(s^i, a^i)$ is a good approximation of $Q^i(s^1, \ldots, s^n, a^i)$ then according to Busoniu et al. [15] with equation 4 it is expected to converge quickly and efficiently.

$$Q^i_{after}(\ldots, s^i, \ldots, a^i) = Q^i_{before}(s^i, a^i), \quad where, \quad \forall s^i \in S^i, a^i \in A^i \quad (7)$$

However, if the learning agent state and action dimension increases, then at equation 6 ASFQ suffers from the exponential increase in memory requirements and processing time

which is cause by this concatenation and cardinality of the set $|S^i|.|A^i|$. Thus, ASFQ agent execution of each action step will require exponential time to be processed. In short, ASFQ is unable to performance under high dimensional RL environment when the environment require high dimension of actions to mitigate non-stationary problem. Intrinsically motivated RL proposed by Singh et al. [4] has similar property to our model with intrinsic motivated internal world representation for their agent's model. However, their approaches did not investigate the non-stationary environment learning criteria for RL where learning agents may face during their solution learning. Thus, it is difficult for their approach to adapt to real world setting where non-stationary conditions are common during RL.

III. Stress Based Memory Retrieval Performance (SBMRP) Framework

The proposed SBMRP framework is constructed on top of Q-Learning (Equation 1) RL algorithm. SBMRP is motivated from Herbian version of Yerkes and Dodson stress model [6] which the learning agent's memory retrieval performance will have non-linear relationship to agent's stress arousal level (See Fig. 2). Fuzzy logic system is integrated into this proposal to choose different categories of actions during different stress conditions that affects the learning agent. The fuzzy logic system is needed to determine the *indefinite boundaries* between different action categories for the complex stress inputs experienced by the learning agent. The three proposed action categories which are $normalAction$, $cognitiveAction$ and $randomAction$. The $normalAction$ category is consist of output generated from Q-Learning RL algorithm. Next the $cognitiveAction$ category is the output generated from the proposed SBMRP framework. Finally, $randomAction$ category is the actions that are randomly generated from all possible actions.

Algorithm 1 stressDetection

1) **Function** $stressDetection(agent, concept)$
2) $\delta = 0$;
3) //Only obstacle agent will cause stress
4) **If Not** Empty Space Detected **Then**
5) $delta = (randi([1, significant])/factor)\times$
6) $RestaurantWorldStressTable(concept, 1)$;
7) **Else**
8) $agent.h_{gc} = agent.h_{gc}\times\gamma$ //Stress discount rate;
9) **End If**
10) $agent.h_{gc} = agent.h_{gc} + \delta$ //Include delta stress;
11) **If** $agent.h_{gc} > 1$ **Then** //Set $agent.h_{gc}$ upper limit
12) $agent.h_{gc} = 1$;
13) **End If**
14) **If** $agent.h_{gc} < 0$ **Then** //Set $agent.h_{gc}$ lower limit
15) $agent.h_{gc} = 0$;
16) **End If**
17) **Return** $agent.h_{gc}$; //Return updated stress level value
18) **End Function**

In addition to that, we also adopted the scaffolding minds perspective [9] to our proposed SBMRP framework where the environment interactions and its feedback provide the cues for learning agent's next action. The learning agent stores its newly detected concepts into its memory and based on the memory to construct a solution to its current situation (refer to function $stressMemoryProcessing$). Next, the intrinsically motivated RL property is located in the proposed agent's memory processing model (also refer to function $stressMemoryProcessing$) where agent will execute its corresponding actions within the memory retrieval boundary determined by its current stress conditions. The learning agent have to persistently executes a batch of actions selected from the agent's memory until all the actions in the batch had been executed and replaced with a new batch of actions (also refer to Fig. 5).

Let's denote h_{gc} as the current agent's stress arousal level (h is stands for hormone and gc is for *Glucocorticoid* stress hormone). Let's denote gamma γ as the stress level discount rate. Then, we define $randi$ as the function to assign random integer in the range with the parameter range. We also define delta δ as the stress level changes causes by the obstacle agent. The δ stress intensity assignment is according to the obstacle's class on $RestaurantWorldStressTable$. SBMRP model begins with $stressDetection$ function (See algorithm $stressDetection$). The $stressDetection$ function main feature is to assign stress changes to the learning agent's h_{gc} level according to different obstacle agent's class.

Then based on the learning agent's two inputs criteria, SBMRP framework will activates the fuzzy logic system to choose an action category from the three possible action categories according to the fuzzy logic system output. The reason to have three action categories are due to when multiple stress input criteria are introduced to the learning agent and it will trigger allostatic process [10] in the learning agent. Then, our proposed model will have captured the learning agent behaviors which are changes in total possible executable actions (e.g. biological changes of behaviors such as faster metabolism, extra stamina can enable running, shouting and etc). Therefore, the three action groups will represent different actions that are *only executable* individually by the learning agent during different stress conditions determined by the output from the fuzzy logic system.

The highest cognitive activities (maximum memory retrieval performance) are suitable for learning agent during medium stress condition according to Yerkes and Dodson [5]. This stress condition is caused by task assignment human class obstacle agent. For example, a manager instructs a learning agent to perform an unknown working task at a robot restaurant world. The learning agent try to randomly assembles a set of actions (within the bounded memory in scaffolding mind perspective) to construct a solution for the medium stress condition safely. The reason why during medium stressful condition the learning agent can safely explore different actions set solution for the unknown working task problem is because the learning agent still can

tolerate certain amount of mistake of its actions during the medium stressful condition.

A. Fuzzy Logic System

There is a fuzzy logic system integrated into SBMRP framework. The fuzzy logic system consists of 2 inputs ($stressLevel$ and $frustrationRate$) and 1 output ($actionCategory$) variables. The $stressLevel$ is referring to h_{gc} learning agent current stress level. The $frustrationRate$ is the failure of response rate detected from learning agent when it is interacting with obstacle agent. In addition to that, it also maintained a 9 fuzzy logic rules set which are predefined for fuzzy inference system in each experiment setting. Each of the fuzzy logic rules is having the same weight importance. The Fig. 4 is a 3D surface output presentation of the proposed SBMRP framework's fuzzy logic system. Next the Fig. 3 is the membership function settings of the proposed SBMRP framework's fuzzy logic system. The defuzzification method selected for the experiment setting is mean of maximum.

1) **If** ($stressLevel$ is low) **and** ($frustrationRate$ is low) **then** ($actionCategory$ is normalAction)
2) **If** ($stressLevel$ is low) **and** ($frustrationRate$ is middle) **then** ($actionCategory$ is normalAction)
3) **If** ($stressLevel$ is low) **and** ($frustrationRate$ is high) **then** ($actionCategory$ is cognitiveAction)
4) **If** ($stressLevel$ is middle) **and** ($frustrationRate$ is low) **then** ($actionCategory$ is normalAction)
5) **If** ($stressLevel$ is middle) **and** ($frustrationRate$ is middle) **then** ($actionCategory$ is cognitiveAction)
6) **If** ($stressLevel$ is middle) **and** ($frustrationRate$ is high) **then** ($actionCategory$ is cognitiveAction)
7) **If** ($stressLevel$ is high) **and** ($frustrationRate$ is low) **then** ($actionCategory$ is cognitiveAction)
8) **If** ($stressLevel$ is high) **and** ($frustrationRate$ is middle) **then** ($actionCategory$ is randomAction)
9) **If** ($stressLevel$ is high) **and** ($frustrationRate$ is high) **then** ($actionCategory$ is randomAction)

Next, $stressActionSelection$ function is an online learning function with feed forward Artificial Neural Network (ANN). Let's denotes σ and mu as the sigma and mean are for the Gaussian function $gaussmf$. The ANN is denoted as $agent.net$. The inputs for the online learning function are detected features when the learning agent encounters the obstacle agent within nearest 8 grid areas. If the learning agent executed the correct action for the obstacle agent within nearest 8 grid areas, then the obstacle agent will provides positive feedback to the learning agent else negative feedback. The $stressMemoryProcessing$ is a function to process agent's memory according to learning agent current stress level h_{gc}. The learning agent's memory retrieval performance is determined by the output of $gaussmf$.

$$mrp = gaussmf(agent.h_{gc} \times 10, [\sigma\mu]) \times 7 \quad (8)$$

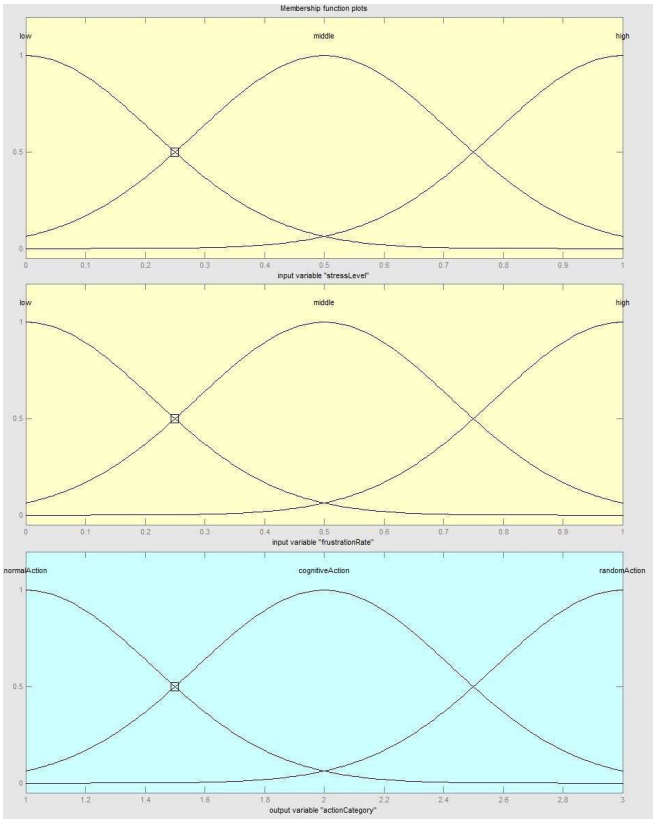

Fig. 3. This figure illustrates the membership functions and its graph settings of $stressLevel$ (first graph), $frustrationRate$ (second graph) and $actionCategory$ (third graph).

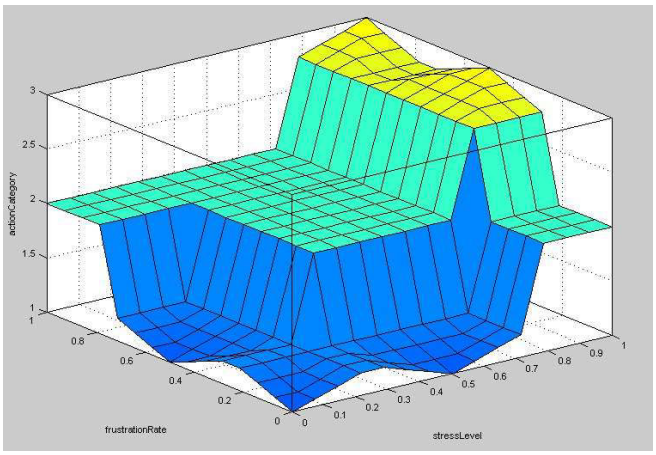

Fig. 4. This figure illustrates the relationship between $stressLevel$, $frustrationRate$ and $actionCategory$ in 3D surface presentation.

Then, we define mrp as the memory retrieval performance (measured in agent's action count). The Gaussian function $gaussmf$ is used to compute the learning agent's mrp corresponding to agent stress level $agent.h_{gc}$. Equation 8 shows how mrp was derived with the learning agent's current stress level $agent.h_{gc}$. We define $agent.memory$ as the learning agent's memory matrix which stores the previous actions in the agent's memory. The maximum mrp is set to 7 because Miller had identified that humans had 7 maximum concepts storage for short term memory [16].

Algorithm 2 stressActionSelection

Function $stressActionSelection$($agent$, $action$)
  If agent received feedback **Then**
    For each $agent$
      //Prepare detected features for ANN training;
    End For
  End If
  For each $action$
    If agent $action$ had trained with features **Then**
      //Activate $agent.net$ and assign output probability;
    End If
  End For
  If sample reached batch total **And** is feedback **Then**
    For each $action$
      If $action$ is triggered for training **Then**
        //Train $agent.net$ with detected features;
        //Reset the sample batch;
      End If
    End For
  End If
  If no activated $action$ **Or** is first call **Then**
    //Assign random $action$;
  Else
    //Return maximum output probability $action$;
  End If
End Function

The $agent.memory$ capacity will expand or contract according to agent's current stress level $agent.h_{gc}$. The $agent.action\_set$ is denoted as the batch of actions that represented as the temporary memory (short term memory) for agent's actions that needed to be executed in sequence (Refer to red color action numbers in Fig. 5). After all the old actions in the previous batch is completed, then the new batch of actions will be selected to be assigned to $agent.action\_set$. The total length of $agent.action\_set$ is depending on the mrp with maximum of 7 actions. Please refer to algorithm $stressMemoryProcessing$ and Fig. 5 for further understanding.

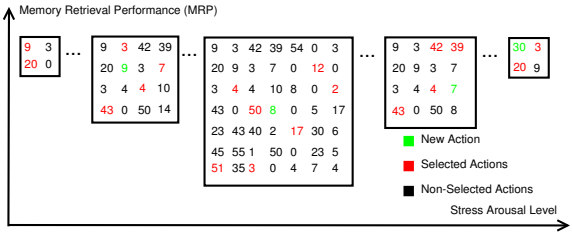

Fig. 5. The diagram is the visual illustration of $stressMemoryProcessing$ function. The green number in the box represents the newly assigned action to the agent's $agent.memory$. The red color number in the box represents the selected actions for batch action execution. Lastly the black color number in the box is the temporary memory storage of previous executed actions. The agent's stress arousal levels $agent.h_{gc}$ will have influence to agent's memory retrieval performance mrp.

Algorithm 3 stressMemoryProcessing

Function $stressMemoryProcessing(agent, action)$
  $\sigma = 2$; //Define mrp standard deviation
  $\mu = 5$; //Define mrp mean
  $mrp = gaussmf(agent.h_{gc} \times 10, [\sigma\ \mu]) \times 7$;
  If $mrp < 1$ **Then**
    $mrp = 1$; //Set lower limit for mrp
  End If
  If $agent.action\_set$ still have non-selected actions **Then**
    //Select the first action from $action\_set$
    //Remove the first action from $action\_set$
  Else
    //Randomly choose row and column $coordinates$
    //Assign $action$ to $agent.memory$ at selected random $coordinates$
    //Randomly select uniqe set of $|mrp|$ total actions from $agent.memory$ for $agent.action\_set$
  End If
End Function

IV. Experiment Settings

This simulation environment was adopted from Busoniu's Multi Agent RL (MARL) Matlab toolbox [17]. We constructed a virtual robot restaurant environment (See Fig. 6) for performance comparisons with implemented state of the art RL approaches in MARL toolbox. The experiments are executed on Intel 3770 i7 CPU at 3.4GHz, 16GB of RAM PC with Windows 7 64 bits operating system.

The experiments settings are divided into two parts. The reduced and the full trial experiment settings. The purpose of reduced trial setting experiment is to visualize immediate results available for the heavy processing RL approaches such as ASFQ and FFQ. The full trial setting experiment is used to visualize the overall experiment's performance stability. This simulation environment's parameter for reduced trial settings were set to 30 trials and maximum $5,000$ steps per trial. This simulation environment's parameter for full trial

settings were set to 100 trials and maximum $100,000$ steps per trial. Both reduced and full trial settings accumulate total of 5 sample experiments to compute the 95% confidence interval. The robot restaurant environment RL parameters were configured as follows: learning rates $\alpha = 0.2$, discount factor $\gamma = 0.95$, eligibility trace decay rate $\lambda = 0.5$ and exploration probability $\epsilon = 0.333$. Robot restaurant environment is a 10×10 virtual grid world with no wall obstacle.

If the nearer learning agent is detected then the obstacle agent will pursuit the nearer learning agent except for neutral obstacle agent class. For all the obstacle agents except neutral class, these obstacle agents have the authority to retain the learning agents as their original location until the obstacle agent is provided the correct consecutive actions from the learning agent. This retain effect is only applicable to learning agent in the nearest 8 grid areas to obstacle agent at robot restaurant world. The hostile human and hostile animal class will need to have 4 consecutive correct actions from learning agent in order to let learning agent set free from retention. Then, the task assignment human class require 8 consecutive correct actions to enable learning agent to be set free from retention. The reason for task assignment human class that needed more consecutive correct actions is because in real world work instructions from task assignment human to the learning agent is more detailed and required precise consecutive actions and intrinsic motivation to get the task done (e.g. What foods to bring). On the other hand, hostile obstacle agents class requires less precise consecutive actions to mitigate because in actual distress call for help does not need to be precise to be effective.

An online ANN machine learning algorithm with feed forward network had been assigned to our proposed SBMRP model (See algorithm $stressMemoryProcessing$). The online ANN machine learning will be trained in backpropagation algorithm. The parameters are set with 55 hidden nodes, learning epochs of 10, each input batch of 5 inputs with 32 binary features. The 32 input binary features is consist of 2 binary features for abstract class, 5 binary features for class and 15 binary features for concept. The table I shows the abstract classes, classes and concepts for obstacle agent in robot restaurant world and it will be represented in input features for SBMRP online learning. An online ANN machine learning algorithm with feed forward network had been assigned to our proposed SBMRP model (See algorithm $stressMemoryProcessing$). The parameters are set with 55 hidden nodes, learning epochs of 10, each input batch of 5 inputs with 32 binary features. The 32 input binary features is consist of 2 binary features for abstract class, 5 binary features for class and 15 binary features for concept. The table I shows the abstract classes, classes and concepts for obstacle agent in robot restaurant world and it will be represented in input features for SBMRP online learning.

V. Simulation Results

The experiment results are divided into two parts, the reduced trial experiment settings (Fig. 7) and full trial experiment settings (Fig. 8).

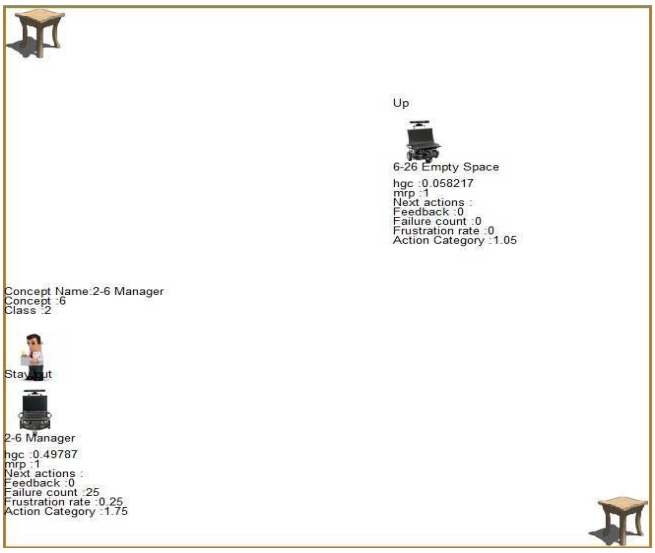

Fig. 6. This figure illustrates the simulated robot restaurant environment.

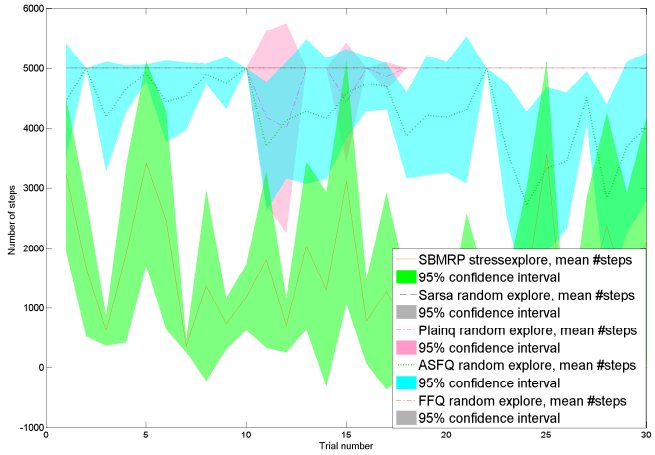

Fig. 7. The 30 trials and $5,000$ maximum steps experiment results for SBMRP, SARSA, PlainQ, ASFQ and FFQ model simulation comparison for robot restaurant world. The line indicates the mean number of steps for the 5 sample experiment executions. The green, grey, red, blue and grey area are the 95% confidence interval for the SBMRP, SARSA, PlainQ, ASFQ and FFQ simulation results. The confidence interval grey area is zero for both SARSA and FFQ (due to $5,000$ steps per trial limitation).

VI. Result Analysis

The experiment observation for the reduced trial setting (Fig. 7) clearly indicates SBMRP is the overall best performance solution step convergence approach (the least step needed for each trial). Therefore, other approaches which are not suitable for non-stationary (stress-conditioned) environment because of the additional stress mitigation actions that will cause solution step convergence performance to depreciate. Next refer to Fig. 8, SBMRP had achieves overall performance stability after the 100 trials experiment execution with minimum variations on confidence interval (See Fig. 8) and its maintenance in horizontal slope steps per trial

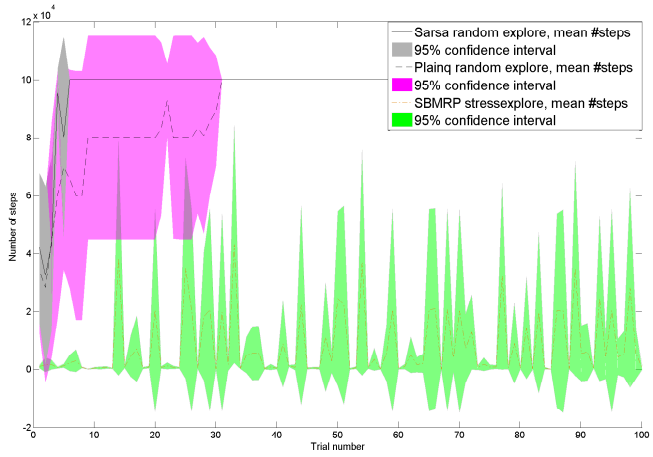

Fig. 8. The 100 trials and 100,000 maximum steps experiments results for SARSA, PlainQ and SBMRP model simulation comparison for robot restaurant world. The line indicates the mean number of steps for the 5 sample experiment executions. The grey, red and green areas are the 95% confidence interval for the SARSA, PlainQ and SBMRP simulation results.

performance. PlainQ approach performs poorly in term of stability because of its high variations in 95% confidence interval. Although the SARSA's 95% confidence interval variations did not shows much different with SBMRP (due to the maximum cap of 100,000 steps per trial), however SARSA's (same behavior as PlainQ) number of steps per trial increases over the trials (positive slope). On the other hand, SBMRP number of steps per trial maintained its horizontal slope throughout the trials which is much more stable compared to SARSA. The PlainQ and SARSA positive slope phenomena occurred may due to their limited adaptability in non-stationary environment.

VII. Conclusion and Future Directions

We have proposed a novel intrinsically motivated reenforcement learning approach which is based upon biological principles of stress following the Yerkes Dodson biological stress model. We have shown our model performs favourably against four state-of-the-art models in non-stationary environments, leading to faster optimal policy convergence and policy stability once convergence has been achieved. Furthermore, our model can deal with larger problem domains. Our future work will explore adapting our model for enable distributed learning and knowledge transfer between multiple agents.

TABLE I

The $RestaurantWorldStressTable$ representation for the obstacle agent in simulated experiment environment.

Stress	Abstract Class	Class	Concept
3	Human	Hostile Human	Robber
3	Human	Hostile Human	Punk
3	Human	Hostile Human	Aggressive Boy
3	Human	Hostile Human	Gangster
3	Human	Hostile Human	Aggressive Man
2	Human	Task Assign Human	Manager
2	Human	Task Assign Human	Technician
2	Human	Task Assign Human	Owner
2	Human	Task Assign Human	Customer
2	Human	Task Assign Human	Waiter
1	Human	Neutral Human	Visitor
1	Human	Neutral Human	Cleaner
1	Human	Neutral Human	Boy
1	Human	Neutral Human	Girl
1	Human	Neutral Human	Policeman
3	Animal	Hostile Animal	Wild Dog
3	Animal	Hostile Animal	Aggressive Dog
3	Animal	Hostile Animal	Wild Cat
3	Animal	Hostile Animal	Aggressive Cat
3	Animal	Hostile Animal	Rat
1	Animal	Neutral Animal	Puppy Dog
1	Animal	Neutral Animal	Shih Tzu Dog
1	Animal	Neutral Animal	Chiwawa Dog
1	Animal	Neutral Animal	Persian Cat
1	Animal	Neutral Animal	Siamese Cat

References

[1] H. F. Harlow, "Learning and satiation of response in intrinsically motivated complex puzzle performance by monkeys." *Journal of Comparative and Physiological Psychology*, vol. 43, no. 4, p. 289, 1950.

[2] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[3] T. Hester and P. Stone, "Texplore: real-time sample-efficient reinforcement learning for robots," *Machine Learning*, vol. 90, no. 3, pp. 385–429, 2013.

[4] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, "Intrinsically motivated reinforcement learning: An evolutionary perspective," *Autonomous Mental Development, IEEE Transactions on*, vol. 2, no. 2, pp. 70–82, 2010.

[5] R. M. Yerkes and J. D. Dodson, "The relation of strength of stimulus to rapidity of habit-formation," *Journal of comparative neurology and psychology*, vol. 18, no. 5, pp. 459–482, 2004.

[6] S. Lupien, F. Maheu, M. Tu, A. Fiocco, and T. Schramek, "The effects of stress and stress hormones on human cognition: Implications for the field of brain and cognition," *Brain and Cognition*, vol. 65, no. 3, pp. 209 – 237, 2007.

[7] C. M. Pariante and S. L. Lightman, "The hpa axis in major depression: classical theories and new developments," *Trends in neurosciences*, vol. 31, no. 9, pp. 464–468, 2008.

[8] J. W. MASON, "A review of psychoendocrine research on the sympathetic-adrenal medullary system," *Psychosomatic medicine*, vol. 30, no. 5, pp. 631–653, 1968.

[9] A. Clark and M. A. Boden, *Being there: putting brain, body, and world together again*. MIT Press Cambridge, MA, 1997.

[10] B. S. McEwen and J. C. Wingfield, "The concept of allostasis in biology and biomedicine," *Hormones and behavior*, vol. 43, no. 1, pp. 2–15, 2003.

[11] D. M. Diamond, A. M. Campbell, C. R. Park, J. Halonen, and P. R. Zoladz, "The temporal dynamics model of emotional memory processing: a synthesis on the neurobiological basis of stress-induced amnesia, flashbulb and traumatic memories, and the yerkes-dodson law," *Neural plasticity*, vol. 2007, 2007.

[12] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[13] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998, vol. 1, no. 1.

[14] M. L. Littman, "Friend-or-foe q-learning in general-sum games," in *ICML*, vol. 1, 2001, pp. 322–328.

[15] L. Busoniu, B. De Schutter, and R. Babuska, "Multiagent reinforcement learning with adaptive state focus." in *BNAIC*, 2005, pp. 35–42.

[16] G. A. Miller, "The magical number seven, plus or minus two: some limits on our capacity for processing information." *Psychological review*, vol. 63, no. 2, p. 81, 1956.

[17] L. Busoniu, R. Babuska, and B. Schutter, "Multi-agent reinforcement learning: An overview," in *Innovations in Multi-Agent Systems and Applications - 1*, ser. Studies in Computational Intelligence, D. Srinivasan and L. Jain, Eds. Springer Berlin Heidelberg, 2010, vol. 310, pp. 183–221.