Defence Science Journal, Vol. 57, No. 2, March 2007, pp. 173-183

SHORT COMMUNICATION

Missile Defence and Interceptor Allocation by LVQ-RBF Multi-agent Hybrid Architecture

S. Thamarai Selvi and R. Malmathanraj

Madras Institute of Technology, Anna University, Chennai-600 044

ABSTRACT

This paper proposes a solution methodology for a missile defence problem using theatre missile defence (TMD) concept. In the missile defence scenario, the concept of TMD is generally used for the optimal allocation of interceptors to counter the attack missiles. The problem is computationally complex due to the presence of enormous state space. The Learning vector quantiser–Radial basis function (LVQ-RBF) multi-agent hybrid neural architecture is used as the learning structure, and Q-learning as the learning method. The LVQ-RBF multi-agent hybrid neural architecture overcomes the complex state space issue using the partitioning and weighted learning approach. The proposed LVQ-RBF multi- agent hybrid architecture improvises the learning performance by the local and global error criterion. The state space is explored with initial coarse partitioning by LVQ neural network. The fine partitioning of the state space is performed using the multi-agent RBF neural network. The discrete reward scheme is used for LVQ-RBF multi-agent hybrid neural architecture. It has a hierarchical architecture which enables quicker convergence without the loss of accuracy. The simulation of the TMD is performed with 500 assets and six priority of assets.

Keywords: Reinforcement learning, LVQ neural network, theatre missile defence, interceptor allocation, Q-learning, RBF neural network, radial basis function, learning vector quantisation

1. INTRODUCTION

The technology era has brought a paradigm shift in the conduct of modern warfare. Missiles equipped with either conventional or nuclear warheads can inflict serious damages to far-flung target locations in a matter of minutes. The defence against such missile attacks is by launching antiballistic missiles to intercept and render ineffective the incoming missiles. The war scenario involves launching multiple missiles with different ranges to hit potential targets. The defending country normally practices the theatre missile defence (TMD) concept, involving allocation of interceptors optimally to counter the incoming missiles. The solution to the interceptor allocation problem, which is analogous to the TMD, is constrained due to state space complexity and the timing complexity. A new solution methodology for the TMD problem is proposed using Learning vector quantiser–Radial basis function (LVQ-RBF) multi-agent hybrid architecture. Q-learning is a reinforcement learning method. The learning scheme, in which the learner learns an optimal behaviour by obtaining repeated rewards/penalties from the environment, is known as reinforcement learning[1].

The learner actively explores the state-action space to acquire gradient information about the

Revised 11 September 2005

reinforcement function by comparing the reinforcement values in that space. At time step t, the learner receives a vector of state input $x(t)$ from the environment $X \in R_n$ where R is the set of real numbers. The learner provides an output $a \in A$ where A is the available set of action space. The critic evaluates the output a at time instant t in the context of input $x(t)$ and sends to the controller an evaluation signal $r(t) \in R$. That signal is termed reinforcement. The reinforcement $r(t)$ is determined by the critic according to an external reinforcement function $r(x(t), a) \in R$. It is assumed that a unique function $a^*(x)$ exists which optimises the reinforcement over the input space. That function $a^*(x)$ is termed the optimal law. Hence, the objective of reinforcement learning is to learn an optimal controller such that for each input $x(t)$, an optimal performance evaluation r is returned. In the Q-learning[1] scheme, the learner builds incrementally a Q function, which attempts to estimate the discounted future rewards for taking actions from any initial state. The Q-learning techniques do not need any database of training pairs as required by the supervised learning. The output of the Q function for state x and action a is denoted by $Q(x, a)$. Recently neural networks have been considered for reinforcement learning by various researchers[2-5]. In this paper the Q learning technique is implemented on LVQ-RBF multi-agent hybrid neural architecture.

The paper further addresses state–space partitioning used to group the homogenous states into separate regions. In addition to state-space partitioning, the paper uses weighted averaging approach to have specialised agents for every local region. This local region-based partitioning results in quicker convergence of temporal error, and thus facilitates learning to a degree. The partitioning of the input space is performed to make learning easier and for better weighted averaging. The implementation of a multi–layer perceptron neural network is explained by Sun[6], *et al*. The performance of the multi-agent decision module is improved using the RBF neural networks in the decision module. The LVQ-RBF multi–agent hybrid architecture approach requires domain-specific knowledge to initialise partitioning. The major advantage of the LVQ-RBF multi-agent hybrid architecture is the ability to learn Q function with both hard partitioning and soft partitioning. The proposed new learning model completes learning by finding an optimal state/action mapping which maximises the sum of future rewards irrespective of the initial state.

2. THEATRE MISSILE DEFENCE (TMD)

The interceptor allocation problem is considered to be a complex dynamic problem which is one of the important module in the TMD. The TMD is the phenomenon of shielding regions with differing priorities from attacker missiles by a class of defence missiles called as interceptors. The allocation of interceptors to overcome the incoming missiles at each and every time instant is a sequential decision problem. The launching of interceptors is constrained by the cost and number of launchers available per attacking wave. The attack may be for several hours. The TMD interceptor problem is a dynamic decision problem as a series of decisions must be made over an extended period of time, with the consequences of a single decision affecting the subsequent decisions.

The defence has a set of missile launchers and an inventory of missiles. The decisions to be made concern the utilisation of interceptors. Given the current attack wave the defence wave must decide on how many interceptors to use against the current attack wave and how many to withhold for use against the future attack waves. The basic elements of the problem are assets of different types, the missiles available to the attacker, and the interceptors available to the defenders. The interceptors are used to destroy the attacker missiles.

Let $P_{D,t}(m,n)$ denote the probability that an asset of type 't', will be destroyed when attacked by 'm' missiles. One assumes that action takes place in discrete time periods, where L_M denotes maximum number of missiles and L_I denotes maximum number of interceptors that can be launched at a period.

The assets are prioritised according to the value V_t and the objective is to maximise the expected value of assets surviving at the end of the battle. The problem of dynamic decision making for the interceptor allocation is formulated as the shortest path problem. The current state has two components

(i, a), where $i=(A_1,A_2,A_3,\ldots,A_n)$ is the first state component. A_1 denote the number of assets with priority 1, and so on.

$a=(m_1,m_2,m_3,\ldots,m_n)$ is the next state component. m_1 denotes the number of incoming attacking missiles towards region of priority 1, and so on.

The neural architecture is trained to map the state-action pair values of the TMD problem into actions. The TMD environment is assumed to be dynamic, with the presentation of new dataset for every learning trial. The frequent changes in the input data set often disrupt the learning process. The past success-directed exploration scheme is modified and adapted to overcome this limitation. The advantage of this exploration scheme is that it selects actions based on the reward accumulation rate of the actions. Thus the LVQ-RBF multi-agent hybrid architecture weight adapts to include the training set using the gradient-descent algorithm and the exploration technique.

3. PROPOSED ARCHITECTURE

The previous work on TMD interceptor allocation by Bertsekas[7], *et al*. utilises the neuro-dynamic programming technique. In this paper, a new architecture to maximise the number of assets used in the simulation has been proposed. Further the categorisation of priority is also increased. In general, for all Markov decision problems the state space increases exponentially as learning progresses. This curse of dimensionality is resolved by including the state-space exploration in the new architecture. The input space for the interceptor allocation problem is explored and selective states that have numerical closeness are grouped together. This partitioning results in separate regions of state space. One state space vector, $i=(ar_1, ar_2, ar_3, \ldots, ar_n)$ and $a=(am_1, am_2, am_3,\ldots,am_n)$ is used as the representative state equation for one region, where ar_1 denotes the representative asset value on first priority region, am_1 denotes the representative attacking missiles value for the first priority region, and so on.

The reinforcement learning can be performed in the following two ways:

- Policy iteration scheme
- Value function iteration scheme

This paper uses Q value function, which is one kind of the value function iteration scheme for learning the interceptor allocation problem.

The value function-based schemes involves hyperdimensional state–space, huge memory and time requirements. To overcome this, the value function is first approximated and then generalisation is used to find the Q values for states that have not been visited. The action selection schemes are useful in selecting the actions with optimal exploration and exploitation of the available knowledge. The proposed architecture (Fig. 1) is implemented on

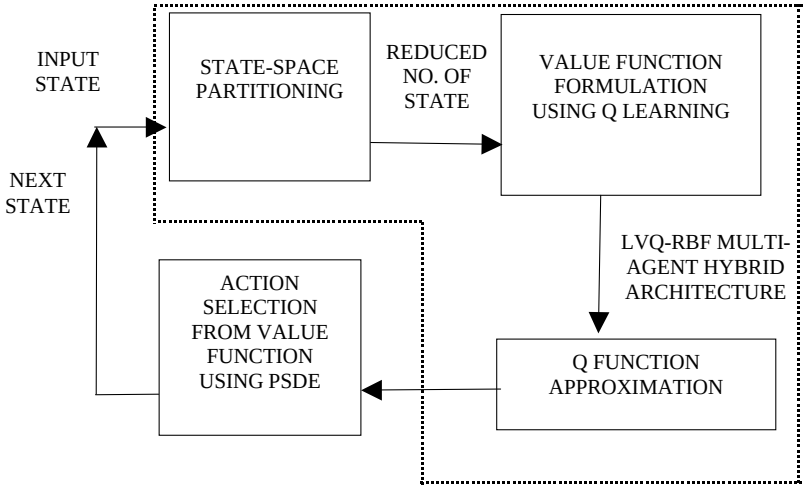

Figure 1. Proposed architecture of missile defence for interceptor allocation.

LVQ-RBF multi-agent hybrid neural architecture and is used to solve the interceptor allocation problem in the TMD domain. It solves the problem using the divide and conquer technique. The TMD problem is solved by dividing the whole state space into separate disjoint regions and is combined according to some online weighted learning scheme. The weighting scheme allows a specific neural network agent to be more compatible to perform learning in one region. The word region refers to the subset of state-space. The partitioning and the weighted learning is implemented using the LVQ-RBF multi-agent hybrid neural architecture and the past success-directed exploration scheme.

3.1 LVQ-RBF Multi-agent Hybrid Architecture

3.1.1 State-Space Exploration

Let the current state have two components (*i*, *a*). The input state-space available for the problem results in a combinatorial explosion of states. The state-space available to model this problem is as high as (s_6^{600}). The exact modelling of the environment is obtained by reducing the redundant information about the input state-space. The state-space is explored in two phases to suppress the redundancy in the state information. The presence of enormous state input is commonly referred to as curse of dimensionality[8,9]. This causes the computational and timing complexity. The initial partitioning of state space into regions is performed by using learning vector quantisation (LVQ) neural network. In the LVQ-based partitioning, the target vector is presented by exploiting the domain knowledge of available (*i*,*a*) combination. The LVQ network utilises the supervised learning scheme and offers accurate classification. Figure 2 shows the proposed state-space exploration and Fig. 3 shows the LVQ-based state-space exploration. The complete architecture involves LVQ-based initial architecture and function approximation by multiple radial basis function (RBF) agents, as shown in Fig. 4.

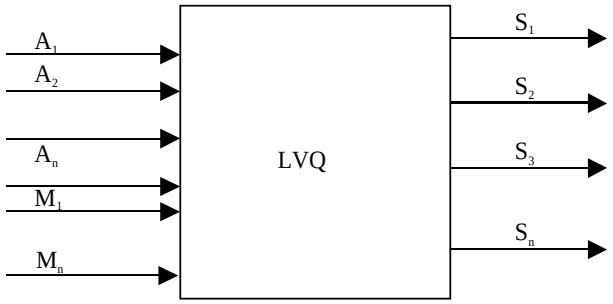

Figure 3. Learning vector quantisation-based state-space exploration.

4. ONLINE PARTITIONING

In the online partitioning approach, the multiple agents along with weighted combination approach is used. In this online optimisation scheme, the same learning rule is used for both the partitioning scheme and the online learning of individual agents. The output is generated in a winner take all passion with every region handled by the best learning agent. This technique is implemented using three radial basis function neural networks. The current state is applied to all the neural networks and the Q values are separately calculated. In this approach, besides the internal weights of individual neural networks there are separate gating weights connected to every neural network. The Q values required for action selection in the value function-based reinforcement learning scheme is obtained by the weighted combination of Q values from individual neural networks. In the weighted averaging approach

$$a(x) = [\Sigma_k \, w_k * \, a_k(x)]/\Sigma_k \, w_k \qquad (1)$$

where k = 1,2,3…n denotes the number of agents utilised for solving sequential decision problem. X is an input, k denotes an agent [$k \in (1,n)$], $a_k(x)$ is the output of agent k, w_k is the weight of agent k, and $a(x)$ is the combined output. In this problem, the Q value function is approximated by radial basis function (RBF) neural networks. The weighting approach of w_k' s reduces the error, where w_k's are

Figure 2. Proposed state-space exploration.

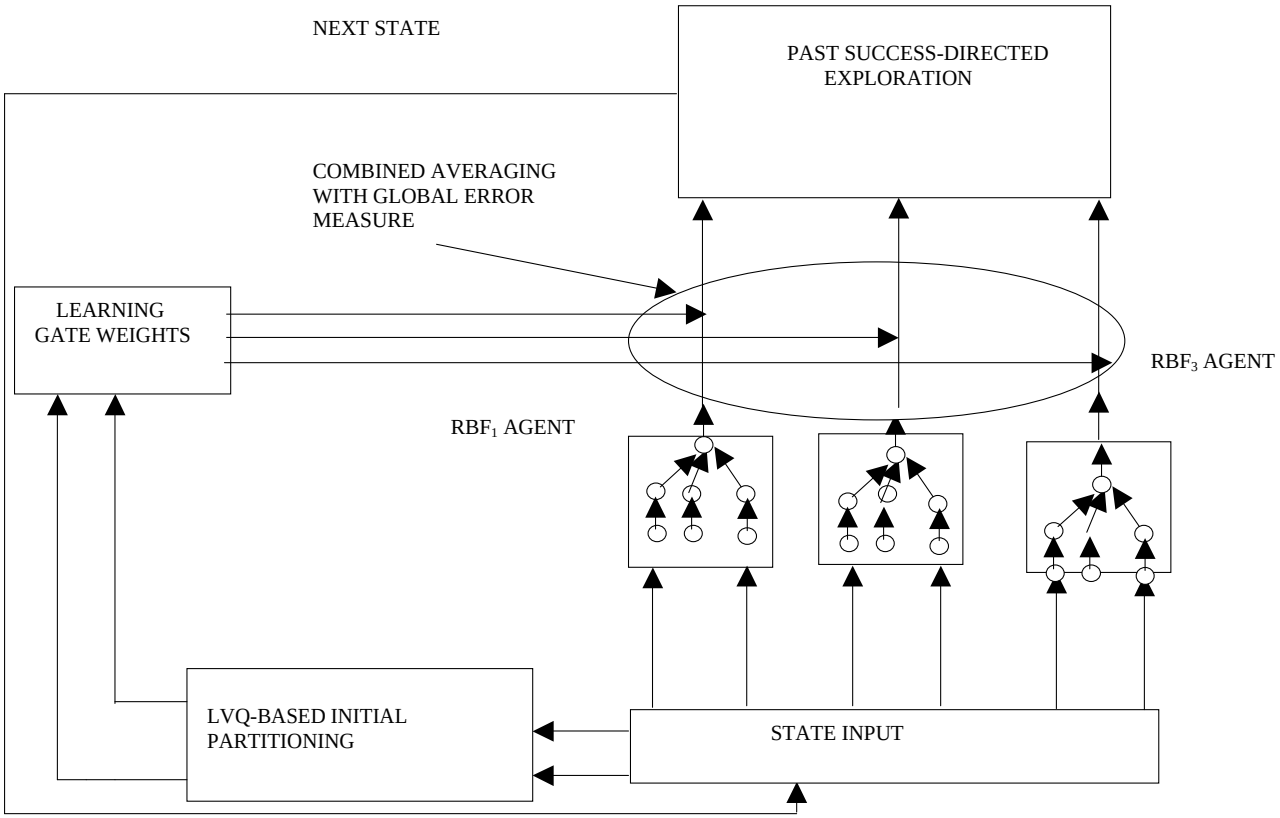

Figure 4. Learning vector quantisation-radial basis function multi-agent hybrid architecture.

weights used for every agents output subject to constraints

$$w_k \geq 1;\ \Sigma w_k = 1 \qquad (2)$$

To minimise the gating weight error, the gradient descent algorithm is used,

$$\text{error}=\Sigma_x\ \text{error}(x)=[\Sigma_x\ y(x)\text{-}w_k a_k(x)]\text{^}2 \qquad (3)$$

The error-based approach ensures that the combined outcome is better than an individual agent (on an average), they ensure that the combination weights are optimal in the sense of minimising the weighted errors. The combination of output agents and the weights is combined along with the RBF network learning algorithm and the gradient descent algorithm.

The word region refers to the subset of inputs. The whole input range is divided into a set of homogenous subsets by using LVQ and online partitioning technique. The error measure obtained is termed as overall Bellman residual and is used for learning the gateweights and the region weights.

5. RBF-BASED FUNCTION APPROXIMATION

In complex problems, the multi agent concept can provide better results. Also, the larger state space effects the function approximators to learn the Q function. The online soft partitioning can be carried out based on a temporal difference and gradient descent criterion. The RBF network allows to learn partitioning together with the learning of Q function. The RBF network constitutes the way of partitioning of the input space. These functions have the highest activation at their centres and gradually taper off until having zero activation at a certain distance away from their respective centres. These model the phenomenon as soft partitioning of the input region. Further the temporal error is minimised by using the global error criterion.

At each iteration of the training, the focus on some instances creates a form of region in the input space; those instances that have the highest weights are the centres of the region while instances having lower weights constitute boundaries of the

region. Q learning is a reinforcement learning method where the learner builds incrementally a Q function, which attempts to estimate the discounted future rewards for taking actions from given states. The output of the Q function for state x and action a is denoted by $Q(x,a)$.

When action a has been chosen and applied, the system moves to a new state, y, and a reinforcement signal, r, is received. $Q(x,a)$ is updated by

$$Q(x,a) = (1-\alpha)\ Q(x,a) + \alpha\{r+\gamma V(y)\} \quad (4)$$

where $V(y)$ is the value of the state y, defined by

$$V(y)= \max_{b\varepsilon A(y)}\ Q(y,b) \quad (5)$$

where $A(y)$ is the possible action set, α is the learning rate, and γ is a discount factor.

The Q value function is approximated by multiple RBF agents. The Q value used for action selection is obtained using the weighted combination of multiple RBF agents. In Q learning, updating can be done online, without explicitly using probability estimates. Its done based on actual state transitions; that is online simulation is performed. The updating is also incremental, necessitated by the fact that only the information about the current state transition is used. That is

$$Q(x,a) = Q(x,a) + \alpha\{g(x_{t+1})\}+ \gamma^*(\max_{b\varepsilon A(y)} Q(y,b) - Q(x,a))$$

where α is the learning rate and a the action is selected by past success-directed exploration scheme.

6. ACTION SELECTION FROM VALUE FUNCTION

6.1 Exploration/Exploitation Trade-off

A key feature of this LVQ-RBF multi-agent hybrid neural network learning is the scheme of exploration/exploitation procedure. In the initial stages of learning to select the actions, the learning agent must have some other technique than by selecting the action with maximum value. In general, the learning agent must accumulate as much reward as possible, hence, should select actions that procure more reward. However some actions with small immediate reward may yield even more reward in the long run, but to find these out, the learner has to choose these though these look less promising. Therefore, the choice of actions can have a significant effect on the behaviour of learning.

The exploration/exploitation[11] procedure can be categorised into directed and undirected[12]. The undirected techniques behave randomly to choose the actions for every rule. The Boltzmann distribution[13] comes under this category. Directed exploration methods are biased by some features of the learning process. The counter-based techniques are best example of this category. In the Q learning as well as fuzzy Q learning procedures, initially Q values have less significant values, hence the action selection/ exploration technique should incorporate some other measure.

In this paper, the learning starts from scratch i.e., q values are initialised to zero, it should use maximum exploration initially. As it learns more the exploration shall be minimized. Previous works on fuzzy Q learning used either Boltzmann distribution or pseudo stochastic exploration techniques[14]. In this paper the past success-directed exploration technique[15] has been modified, for RBF Q learning. This technique biases exploration by the amount and rate of success of the learner. The learner exploits more either if it acquires reward at an increasing rate or if the learner stops receiving reward due to change in environment.

The average discounted reward reflects both the amount and frequency of received immediate rewards, and is defined by

$$\mu^r_t = {}^t\Sigma_{k=1} v^{(t-k+1)} r_t / {}^t\Sigma_{k=1}\ v^{(t-k+1)} \quad (6)$$

where v belongs to [0,1] is the discount rate and r_t is the reward received at time t. The discount factor determines how past rewards are viewed: The further back in time, the less effect these have on the average reward. Past success-directed exploration combines with ξ Greedy algorithm to have a value of ξ as follows

$$\xi t = 0.8\ \exp(-\alpha\mu^r_t)^3 + 0.1 \quad (7)$$

where α is the learning rate.

7. LEARNING ALGORITHM FOR GATING WEIGHTS

The LVQ-RBF multi-agent hybrid neural network concept is used to allow one best neural network agent to learn the Q value function of one region. The gate weights are used to provide the priority for the neural network towards a region. There are two approaches to the learning of gate weights, viz., (a) local error-based approach, and (b) global error-based approach.

In the local error-based approach, gradient descent is performed on $[y(x) - \Sigma_k w_k a_k(x)]^2$. In the global error–based approach, the gradient descent is performed on $\Sigma_k w_k [y(x) - a_k(x)]^2$.

The error measure is used in the combination of the gradient descent and temporal difference learning to enable the neural network learns the complex Q value function. The combined errors used for learning of multiple agent is defined as follows

$$\text{error}(x_t,u)=\Sigma_k w_k(x_t)[d_k(x_t,u)\text{^}2] \quad (8)$$

Another way to calculate the squared error is the overall Bellman residual

$$\text{error}(x_t,u)=[d^0(x_t,u)\text{^}2], \quad (9)$$

where

$$d^0(x_t,u) = \gamma\Sigma_k[(w_k(x_{t+1})/\Sigma_k(w_k(x_t)] * \max_{b\varepsilon A(y)} Q(y,b)+g(x_t+1) - \Sigma_k[(w_k(x_{t+1})/\Sigma_k(w_k(x_t)] * Q(x,a)$$

Using the incremental steepest descent, after experiencing the state action pair (x_t,u), the updating is performed as follows

$$\text{del}w_k(x_t) = \text{alpha}*[(\Sigma_{j\sim=k} w_j(x_t) (d_j(x_t,u)\text{^}2) - d_k(x_t,u)\text{^}2)]/\Sigma_j w_j(x_t)\text{^}2) \quad (10)$$

The error correction rule for the network is given as follows:

$$\text{del}w^k_{net}(x_t) = \text{beta}* d_k(x_t,u) * [w_k(x_t)/\Sigma_j w_j(x_t)] * \delta Q(x,a)/\delta w^k_{net} \quad (11)$$

where alpha and beta are the learning rates. When the gating weights $w_k(x_t)$ are generated by a gating network (based on inputs x_t), the del $w_k(x_t)$ may be viewed as the amount of updating applied to the corresponding gating weight.

As a whole, the technique can be summarised as follows:

- Apply the current state to all the agents
- Calculate the Q values of all agents
- Combine the Q values from multiple agents to have a common Q value with weighted approach for the current state action pair
- Perform the exploration/exploitation technique to select current action
- Change to the next state
- Calculate the Q values
- Based on the maximum Q value select an action
- Perform temporal error correction for network and gating weights using gradient descent algorithm.

7.1 State-space Transition and Reward Function

The simulation has been performed with both continuous-valued and the discrete-valued rewards. The discrete-valued (0 and 1) reward function is obtained from the environment. The reward accumulation rate proportionally increases as according to the assets saved per iteration. For the state-space transition two assumptions are used:

- The destroying efficiency of the Interceptor to overcome the incoming missile is complete.
- The maximum capability of an attacking missile is that it can destroy only one asset of any priority.

8. RESULTS AND DISCUSSION

A number of experiments was carried out with the design goal to maximise the reward values, assets saved, and action space exploration. The simulation scheme involves six different priority

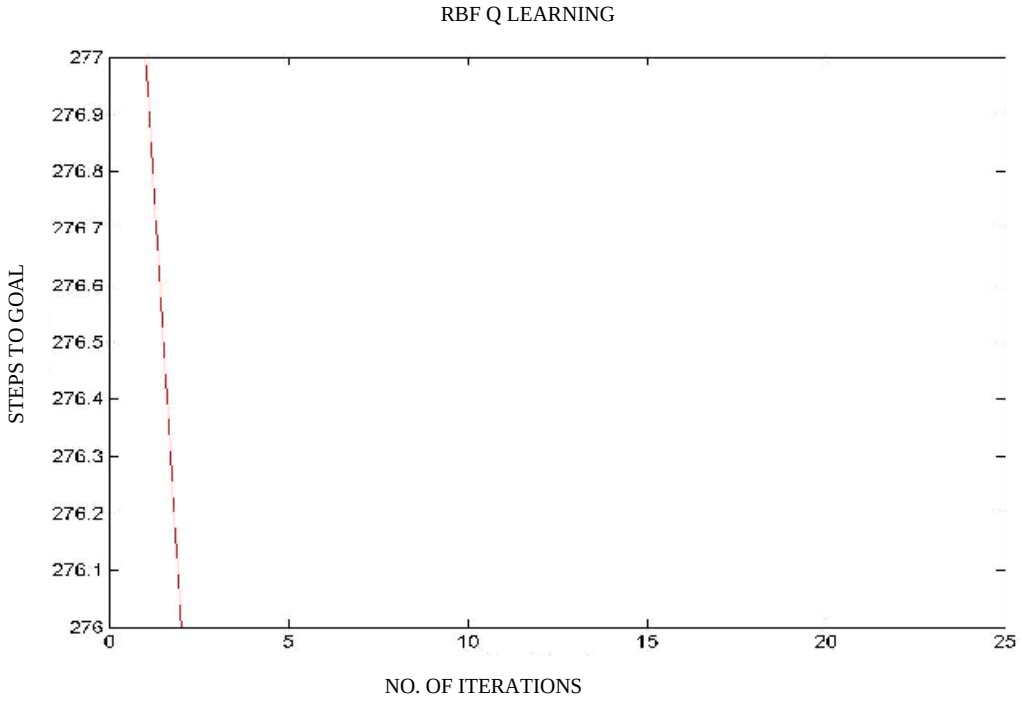

Figure 5. Plot to show the ability of the decision module to withstand 276 attack waves.

regions and each priority region consists of atleast 500 assets. The input state equation used is i=[a_1 a_2 a_3 a_4 a_5 a_6, m_1 m_2 m_3 m_4 m_5 m_6], where a_1– a_6 denote the number of assets present in the ith priority, and m_1–m_6 denote the number of incoming missiles towards the ith priority.

A number of experiments were carried out with the following design goals to learn the complex Q value function by the appropriate agents with an ability to withstand 276 attack waves (Fig.5).

The past success-directed exploration scheme was used to maximise the survivability of assets. The LVQ-RBF hybrid neural network-based decision module was trained to allot interceptors to defend varying priority regions. The attacking option used in the simulation was denoted by

attack=[12 12 12 12 12 12;
5 5 5 5 5 5;
15 15 15 15 15 15;
10 10 10 10 10 10;
19 19 19 19 19 19;
4 4 4 4 4 4;
3 3 3 3 3 3];

The subscript a_{ij} denotes the number of attacking waves used to attack the assets of j^{th} priority. The output actions used in the simulation is

waa=[10 10 10 10 10 10;
5 5 5 5 5 5;
15 15 15 15 15 15;
16 16 16 16 16 16];

Figure 6 shows the ability of the past success-directed exploration scheme to explore better performance in the state-spaces. The exploration scheme uses the random action selection strategy during the initial stages of learning. The value of the parameter v used in the simulation was 0.9. The plot to show the learning trial is shown in Fig. 7. The step to goal was minimised as the number of iterations increased. The first row denotes the number of incoming attacking missiles and the next row denotes the number of interceptors allotted to overcome the incoming attack waves.

The learning of the reinforcement function is depicted clearly in the case of LVQ-RBF hybrid neural network-based decision module. The RBF neural network used in the architecture contains 10 centres. The Euclidean distance measure was

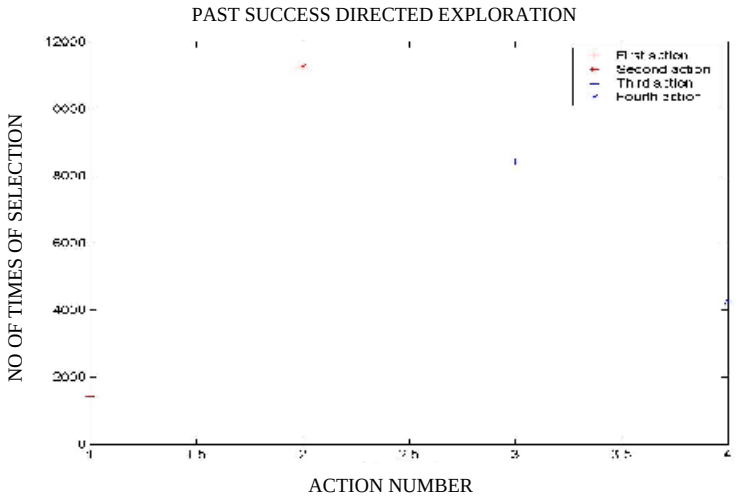

Figure 6. Plot to show the number of times every action selected by past success directed exploration.

used to calculate the distance between the centers and the input data. This difference is used in the reinforcement learning scheme for the error criterion. The value of alpha used in the simulation was 0.4. The simulation was performed with different initial states for RBF Q learning and one of the learning trial is shown in Fig. 7. The region weights used the gradient descent learning scheme for updating the weights and the sample output is shown in Fig. 8. The reward values plot shown in Fig. 9 depicts the accumulation of rewards by the LVQ-RBF hybrid neural network. A total of 500 iteration was performed and the model plan to allot number of interceptors is shown in Fig. 10.

9. CONCLUSION

An efficient solution for the TMD problem involving the sequential allocation of defence resources over a period of time sequences has been proposed. The proposed system facilitates learning better due to the partitioning of state space and the adaptation

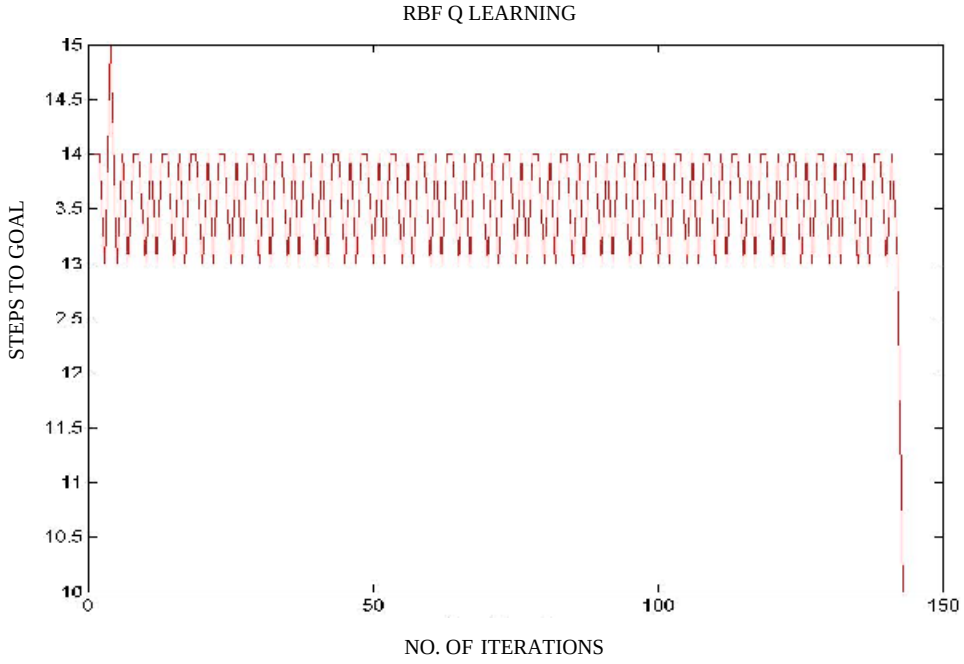

Figure 7. Plot to show the learning trial.

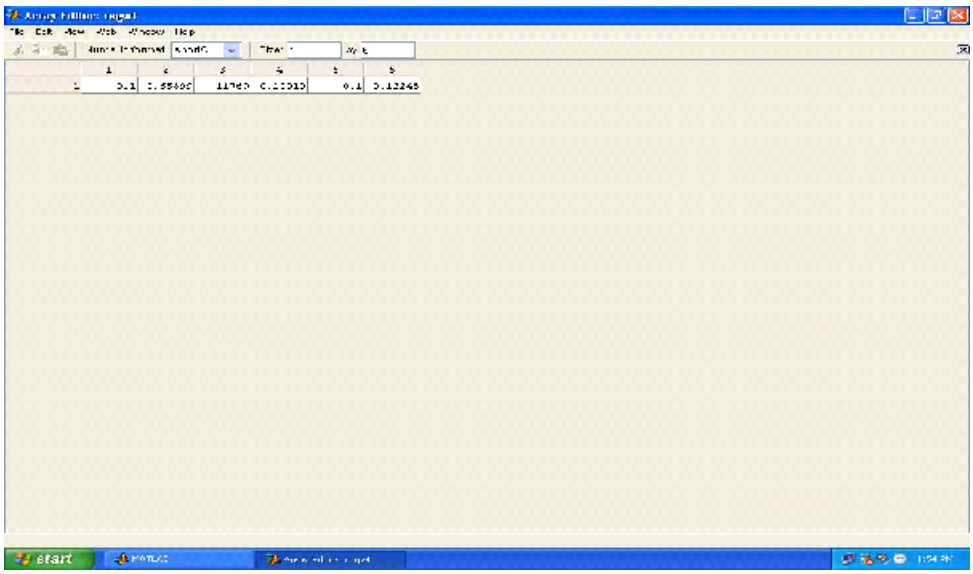

Figure 8. Plot to show the learning of region weights.

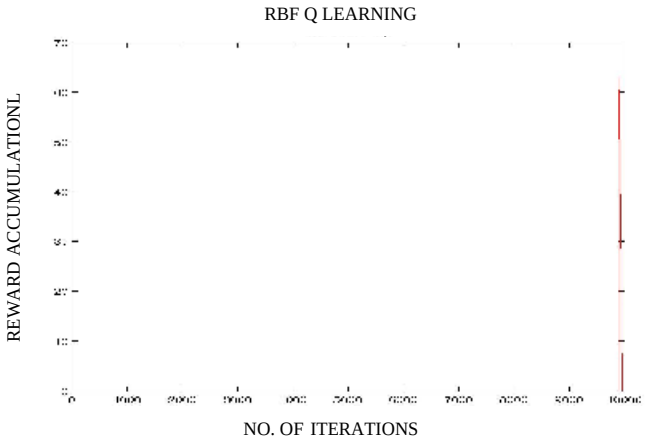

Figure 9. Reward accumulation values with discrete reward scheme.

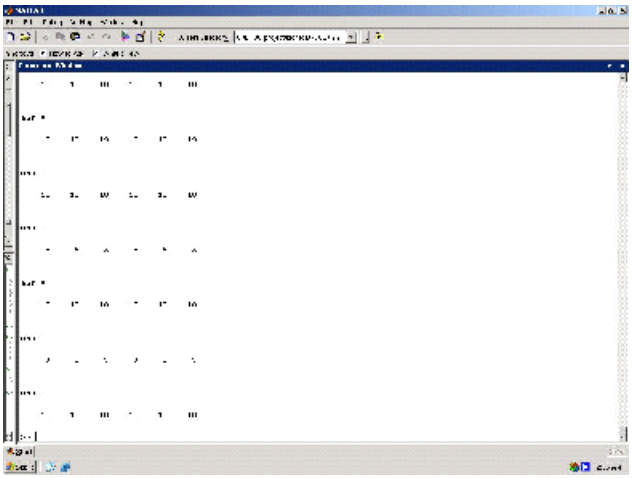

Figure 10. Simulation plan to launch interceptors.

of past success-directed exploration technique to have maximum exploration over action space. The initial partitioning of state space is performed using LVQ neural network. The multiple agent concept was used along with Q value function to learn from experience. Further, the RBF network was used for Q function approximation. The plan to launch interceptors was used to frame decisions for defending regions of varying priority.

ACKNOWLEDGEMENT

The authors express their sincere thanks to Defence Research and Development Organisation, New Delhi from which the final assistance for the research project titled "Missile Defence and Interceptor allocation by RBF neural networks" was obtained. This paper is based on the research conducted by us for this project.

REFERENCES

1. Kaelbling, L.P.; Littman, M.L. & Moore, A.W. Reinforcement learning: A survey. *J. Artifi. Intelli. Res.*, 1996, **4**, 237-85.

2. Song, Kai-Tai & Sun, Wen-Yu. Robot control optimization using reinforcement learning. *J. Intelli. Robotic Syst.*, 1998, **21**, 221-38.

3. Werbos, P.J. Generalisation of back propagation with application to a recurrent gas market model. *Neural Networks*, 1988, 339-56.

4. Lee, Ivan S.K. & Lau, Henry Y.K. Adaptive state space partitioning for reinforcement learning. *Engg. Appli. Arti. Intell.*, 2004, **17**, 577-88.

5. Gullapalli, V.; Franklin, J.A. & Benbrahim, H. Acquiring robot skills via reinforcement learning. *IEEE Contr. Syst.*, 1994, **14**(1), 13-24.

6. Sun, R. & Peterson, T. Multi-agent reinforcement learning: Weighting and partitioning. *Neural Networks*, February 1999, 12(4-5), 127-53.

7. Bertsekas, Dimitri P.; Homer, M.L.; Logan, D.A.; Patek, S.D & Sandell, N.R. Missile defence and interceptor allocation by neuro dynamic programming. *IEEE Trans. Syst. Man Cybernetics*, 2000, **30**(1), 42-51.

8. Gad; I.B.; Morris, Genela & Bergman, Hagai. Information processing, dimensionality reduction and reinforcement learning in the basal ganglia. *Prog. Neuro Biology*, 2003, **71**, 439-73.

9. Bellman, R. Adaptive control processes: A guided tour. Princeton University Press, 1961.

10. Bishop, C.M. Neural networks for pattern recognition. Oxford University Press, 1995.

11. Wyatt, Jereme. Exploration and inference in learning from reinforcement. University of EdinBurgh, 1997. (PhD Dissertation).

12. Glorennec, P.Y. Fuzzy Q learning and dynamical fuzzy Q learning. *In* Proceedings of the 3rd IEEE International Conference on Fuzzy Systems, Orlando, June 1994.

13. Jouffe, L. Fuzzy inference system learning by reinforcement methods. Tech Report INSA–96081, 1996.

14. Thrun. Efficient exploration in reinforcement learning. Tech Report CS-92-102, Carnegie Melon University.

15. Patrascu, R. & Stacey, D. Adaptive exploration in reinforcement learning. *In* International Joint Conference on Neural Networks, July 1999, **4.** pp. 2276-281.

16. Russell, Stuart & Norvig, Peter. Artificial intelligence: A modern approach. Prentice Hall of India, 2003.

17. Iwata, K. & Sakai, H. A new criterion using information gain for action selection strategy in reinforcement learning. *IEEE Trans. Neural Networks*, 2004, **15**(4).

18. Horn, H.; Dror, G. & Quenet, B. Dynamic proximity of spatio temporal sequences. *IEEE Trans. Neural Networks*, 2004, **15**(5).