



# 1 Visual – Motor Coordination using a Quantum 2 Clustering based Neural Control Scheme

3 NIMIT KUMAR and LAXMIDHAR BEHERA

4 *Department of Electrical Engineering, Indian Institute of Technology, Kanpur, UP 208016,*

5 *India. e-mails: nimit.kumar@ieee.org, lberhera@iitk.ac.in*

6 **Abstract.** Visual – Motor Coordination is a problem considered analogous to the hand-eye  
7 coordination in biological systems. In this work we propose a novel approach to this problem  
8 using Quantum Clustering and an extended Kohonen’s Self-Organizing Feature Map (K-  
9 SOFM). This facilitates the use of the method in varying workspaces by considering the joint  
10 angles of the robot arm. Unlike previous work, where a fixed topology for the input space is  
11 considered, the proposed approach determines a topology as the workspace varies. Quantum  
12 Clustering is a method which constructs a scale-space probability function and uses the  
13 Schrodinger equation and its lowest eigenstate to obtain a potential whose minimum gives the  
14 cluster centers. It transforms the input space into a Hilbert space, where it searches for its  
15 minimum. The motivation of this work is to identify the implicit relationship existing between  
16 the end-effector positions and the joint angles through Quantum Clustering and Neural  
17 Network methods to fine-tune the system to correctly identify the mapping.

18 **Key words.** Hilbert spaces, quantum clustering, scale space clustering, schrodinger equation,  
19 visual – motor coordination

20

## 21 1. Introduction

22 This work introduces a new method based on Quantum Clustering for the visual –  
23 motor coordination of a Robotic Arm. Previous works in this area [4–6, 8, 11] gave a  
24 ‘black-box’ like system with high training time and fixed topological space. Walter and  
25 Schulten [11] extended the Self-Organizing Feature Map Algorithm to give an im-  
26 proved result and added flexibility in the robot parameters. However, the fixed topology  
27 of the feature map and strong dependence on the training parameters could not be  
28 avoided. Behera and Kirubanandan [1] introduced the idea of a hybrid approach, based  
29 on Kohonen’s Self-Organizing Maps with an online training scheme and fixed topol-  
30 ogy. To solve the issue of fixed topology and to involve the implicit relationship between  
31 the input space and the output space, we propose a novel technique using Quantum  
32 Clustering (QC). The QC was originally proposed by Horn and Gottlieb [3] and is used  
33 to determine the necessary topology of the network for the workspace.

## 34 2. Quantum Clustering

35 The QC is a non-parametric clustering technique [7] based on the scale-space clus-  
36 tering algorithm [9]. It uses the Gaussian kernel to generate the Parzen-window

	Journal : <b>NEPL</b> SPS Article No : <b>123R1</b>	Dispatch : <b>9-6-2004</b>	Pages : <b>12</b>
	PIPS No. : <b>5381183</b>	<input type="checkbox"/> LE	<input checked="" type="checkbox"/> TYPESET
	MS Code : <b>NEPL 123R1</b>	<input checked="" type="checkbox"/> CP	<input type="checkbox"/> DISK

37 estimator [2]. Given a set of  $N$  data points,  $\mathbf{x}_1, \dots, \mathbf{x}_N$ , the task is to estimate the  
 38 probability density function  $\Psi(\mathbf{x})$  using the Parzen-window estimator. The estimator  
 39 is constructed by associating with each of the  $N$  data points a Gaussian defined as in  
 40 Equation (1). In this sense it belongs to a more general class of non-parametric  
 41 method – the one using Kernels, here a Gaussian kernel.

$$\Psi(\mathbf{x}) = \sum_{i=1}^N e^{-(\mathbf{x}-\mathbf{x}_i)/2\sigma^2}. \quad (1)$$

43 The maxima of the function  $\Psi(\mathbf{x})$  have been shown to occur at the cluster centers [9],  
 44 where  $1/2\sigma^2$  is the scale of the probability estimator. The function  $\Psi(\mathbf{x})$  is the  
 45 Schroedinger Wave Function and performs a nonlinear transformation of the input  
 46 space into a Hilbert space. It can be seen that as  $\Psi(\mathbf{x})$  is a Gaussian kernel, it is  
 47 positive definite and hence a standard inner product defined in the transformed space  
 48 is also positive definite. That is, by Mercer's theorem [10], the nonlinear transfor-  
 49 mation  $\Psi(\mathbf{x})$  defines a Hilbert space.

## 50 2.1. IMPLEMENTING QUANTUM CLUSTERING

51 The motivation is to search for a Hamiltonian for which  $\Psi$  is an eigenstate and a  
 52 ground state of the operator  $H$ .

$$H\Psi = \left( -\frac{1}{2\sigma^2} \nabla^2 + V(\mathbf{x}) \right) \Psi = E\Psi. \quad (2)$$

54 Equation (2) is a rescaled form of the Schrodinger Equation in quantum mechanics,  
 55 with  $\Psi$  denoting the eigenstate,  $H$  the Hamiltonian,  $V$  the potential energy and  $E$  is  
 56 the energy eigenvalue. It must be noted that  $\sigma$  is the only parameter in the equation.  
 57  $\sigma$  deduces the correct clustering of the space as it controls the width of the Parzen-  
 58 window [9]. The parameter  $\sigma$  can be controlled in a way such that the technique  
 59 yields the relevant number of clusters.

60 Given  $\Psi$ , one can solve Equation (2) for  $V$ :

$$V(\mathbf{x}) = E + \frac{(1/2\sigma^2) \nabla^2 \Psi}{\Psi}. \quad (3)$$

62 If  $V$  is positive definite, that is  $V \geq 0$ ,  $E$  may be defined as in Equation (4).

$$E = -\min \frac{(1/2\sigma^2) \nabla^2 \Psi}{\Psi}. \quad (4)$$

64 As  $\Psi$  is positive definite, it follows that it has no node and hence  $E$  is the minimal  
 65 eigenvalue of  $V$ . The lowest possible eigenvalue occurs for the harmonic potential in  
 66 which case  $E = d/2$ . This leads to the inequality (5).

$$0 < E \leq \frac{1}{2}d. \quad (5)$$

68 QC, thus works by determining the cluster centers, with the data points assigned to a  
 69 cluster based on the Euclidean distance. In the problem of visual-motor coordination,  
 70 we are interested in locating the cluster centers and hence the use of QC is preferred.

## 71 2.2. WHY USE QUANTUM CLUSTERING?

72 In QC, the cluster centers are obtained while searching for the minima of the po-  
 73 tential functions. In statistical clustering methods, the distribution function is  
 74 maximized. In QC the wave function is interpreted as the probability density func-  
 75 tion itself. However, determining the potential minima is more robust than locating  
 76 the maxima of the distribution or wave function. It has been observed that the  
 77 minima of the potential is more well-behaved than the maxima of the wave function.  
 78 We illustrate clustering using QC method on the historic IRIS dataset. IRIS is a  
 79 three class 4-D problem, with each class denoting a flower type – Virginica, Versicolor  
 80 and Setosa. While Setosa is linearly separable from the other two, the remaining two  
 81 classes are mutually non-separable. In Figure 1 plot the data using the first two  
 82 principal components of the dataset.

83 Figure 2 shows the plot of normalized potential function,  $V/E$ , for the IRIS  
 84 Dataset, with the threshold on deciding the minima as a dotted line. The plot of the  
 85 distribution function is similarly shown in Figure 3. It is evident that the maxima of  
 86 the distribution function is difficult to find and does not occur in all the three regions.  
 87 However, the minima of the potential function can be found in every cluster. Thus  
 88 minima of potential is easily obtained for every cluster than the maxima of the wave  
 89 function.

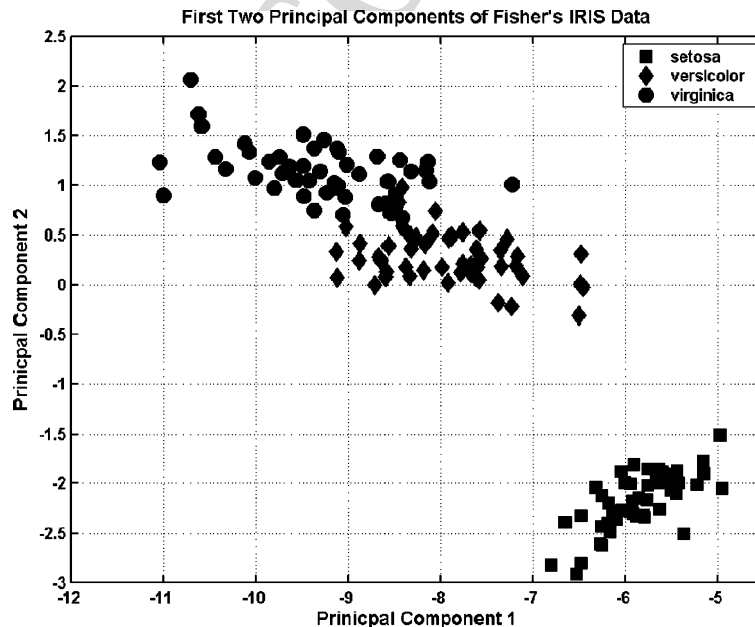


Figure 1. Plot of the first two principal components of the IRIS data.

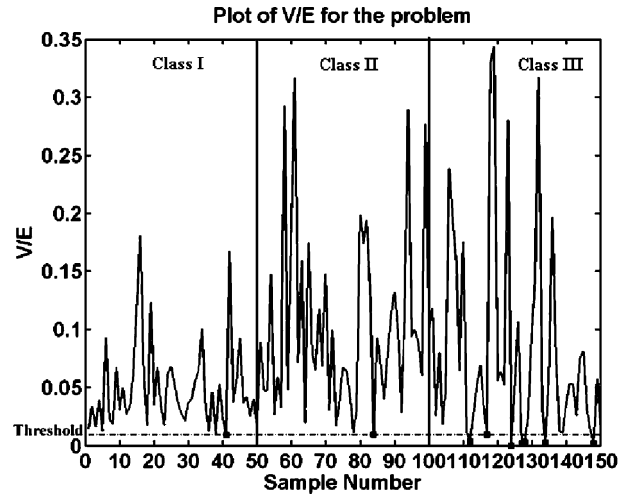


Figure 2. Plot of the potentials at the datapoints of the IRIS data show that all clusters have at least one point as the minima.

### 90 3. Visual-Motor Coordination Control

#### 91 3.1. THE PROBLEM

92 A pair of 2-D 'retinal' coordinates  $u_1, u_2$  are obtained from the two cameras,  
 93 respectively, as shown in Figure 4. The objective is to identify the transformation  
 94  $\Theta(\mathbf{u}_{\text{target}})$  from the retinal coordinates to the joint angles  $\theta$  of the three-joint arm. The  
 95 vector  $\mathbf{u}_{\text{target}}$  is a 4-D vector obtained by grouping the retinal coordinates  $\mathbf{u}_1$  and  $\mathbf{u}_2$ . In

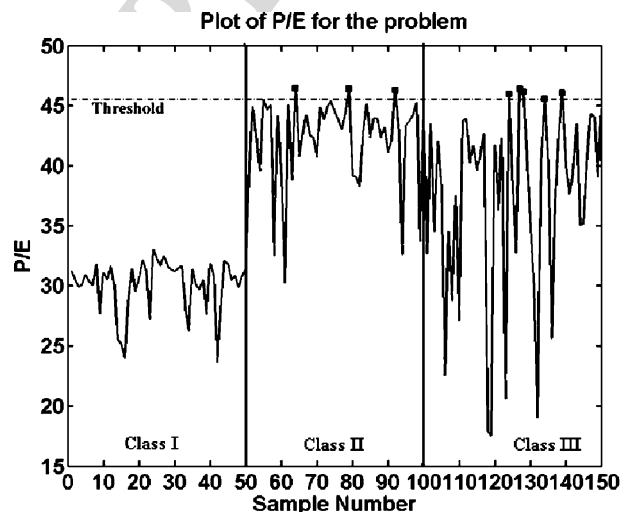


Figure 3. Plot of the wave functions at the datapoints of the IRIS data show that the maxima are not well defined for all the clusters.

196 this work, we consider  $\mathbf{u}_{\text{target}}$  as a 3-D vector by taking into account the actual 3-D  
 197 location of the end-effector in the workspace rather than through the transformed  
 198 camera-plane. That is, we do not consider the camera transformation and work on  
 199 the original end-effector position in the workspace. The first-order Taylor expansion  
 200 of  $\Theta\mathbf{u}_{\text{target}}$  is given by Equation (6).

$$\Theta(\mathbf{u}_{\text{target}}) = \theta_s + \mathbf{A}_s(\mathbf{u}_{\text{target}} - \mathbf{w}_s). \quad (6)$$

202 The idea is to discretize the workspace  $\mathbf{U}$ , into non-overlapping regions  $\mathbf{F}_r$ , such that  
 203  $\mathbf{w}_r \in \mathbf{F}_r$  is the reference or weight vector,  $\theta_r$  is the zero-order term and  $\mathbf{A}_r$  is the  
 204 Jacobian Matrix, which determines the first order term. Equation (6) gives the local  
 205 approximation of each neuron for the joint angle values of the target point  $\mathbf{u}_{\text{target}}$

### 206 3.2. A JOINT INPUT – OUTPUT SPACE PARTITIONING SCHEME

207 We simulate the workspace using the inverse kinematics relationship between the end-  
 208 effector position in the 3-D world space and the joint angles. This inverse kinematics  
 209 relationship is easily obtained using the geometry of the robot manipulator. A typical  
 210 manipulator is shown in Figure 5. Given a set of end-effector positions  $\mathbf{u}$  and their  
 211 corresponding joint-angles  $\theta$  from the workspace, we consider a 6-D vector  $\mathbf{z}$  by  
 212 grouping  $\mathbf{u}$  and  $\theta$ . The motivation for the proposed approach can be under-stood if we  
 213 observe from Figure 6 that the output space formed by the joint angles is clustered into  
 214 four well-defined regions. Thus, the vector  $\mathbf{z}$  also forms a clustered space. This implicit  
 215 relationship between the workspace and the joint-angle space when considered when  
 216 initializing and training the network, greatly improves the capability of the network  
 217 while decreasing the complexity of the network topology. It also follows from the  
 218 kinematics of the manipulator that not all joint-angle values will be allowed when it is  
 219 expected to work in a given workspace due to dimensional constraints.

220 Using QC, we obtain the cluster centers in the space represented by  $\mathbf{z}$ . The initial  
 221 values for  $\mathbf{w}_r$  and  $\theta_r$  are then extracted back from the 6-D cluster centers. It should be  
 222 noted that though a given joint angle determines a unique end-effector position, it is  
 223 not true otherwise. An end-effector position maybe realized by more than one joint-  
 224 angle values. This implicit information must be accounted for while preparing the

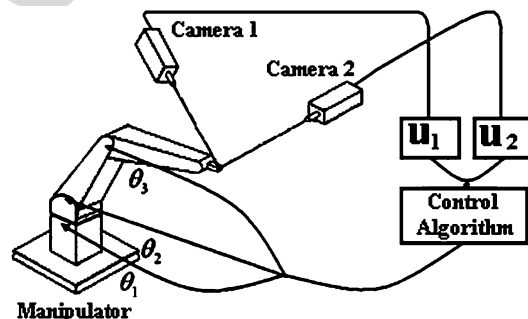


Figure 4. Schematic diagram of the visuo-motor setup.

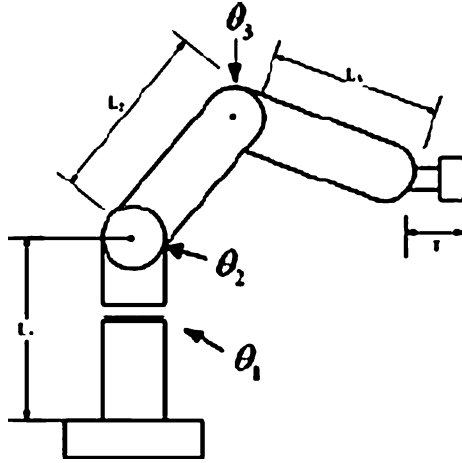


Figure 5. Schematic diagram of the manipulator which can be used to generate the inverse kinematics relationship.

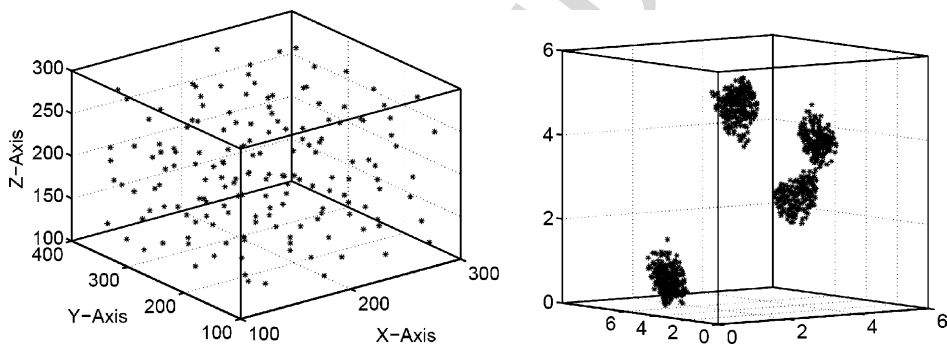


Figure 6. Plot of the reference centers of different neurons and corresponding joint-angles obtained using QC-based method.

125 network. This forms one of the motivation for resorting to the proposed clustering  
 126 method. It has been observed that such a selection needs a small fine-tuning of the  
 127 cluster centers and hence has a very fast learning time. In addition, by an adaptive  
 128 selection of clusters from the workspace, we have reduced the number of neurons  
 129 significantly.

### 130 3.3. QUANTUM CLUSTERING VS KOHONEN'S SELF-ORGANIZING MAPS FOR VISUAL 131 MOTOR COORDINATION

132 In the Kohonen's Self-organizing Feature Map based method, a fixed topology for  
 133 the space is used by creating a 3-D lattice of neurons. These neurons then organize  
 134 themselves homogeneously so that the workspace is well spanned and Equation (6)  
 135 can be used to identify the joint-angles for a given end-effector position. This has two  
 136 major disadvantages:

- 137 – the number of neurons have to be pre-decided and fixed, thus allowing little  
 138 flexibility;  
 139 – the topology of the workspace is fixed once the lattice parameters are specified.

140 A fixed topology of the workspace is not necessary because that constraints the  
 141 neurons to occur uniformly. The receptive field dimensions of all neurons in the  
 142 visual system is not same. The proposed method has the same efficiency using almost  
 143 half the neurons used in the K-SOM based approach and a flexible topology.  
 144 Figure 6 shows the distribution of points in a workspace and the corresponding  
 145 points in the joint-angle space.

#### 146 4. Visual-Motor Coordination using Quantum Clustering based Neural 147 Control Scheme

##### 148 4.1. THE NEURAL LEARNING ALGORITHM

149 The Learning Algorithm adopted in the present work is motivated by the extended  
 150 Kohonen's Self-Organizing Feature Maps introduced by Walter and Schulten in  
 151 [11]. Given an end-effector position target  $\mathbf{u}_{\text{target}}$ , a winner neuron  $\mu$  is selected, based  
 152 on the Euclidean distance metric in the workspace. The neuron whose reference  
 153 vector is closest to the target is declared winner. The arm is given a coarse movement,  
 154  $\theta_0^{\text{out}}$  by the initial output of the network determined by Equation (7) resulting in the  
 155 end-effector to move to a position  $\mathbf{v}_0$ . This is followed by a fine movement deter-  
 156 mined by  $\theta_1^{\text{out}}$ , using Equation (8), giving a correcting movement to the arm and the  
 157 end-effector reaching the final position of  $\mathbf{v}_1$ . It has been shown that a series of  
 158 similar corrective actions will converge to the correct end-effector position. However,  
 159 we use only one corrective fine movement and achieve a considerable amount of  
 160 accuracy.

161 The collective averaged output is evaluated using Equation (7).

$$\theta_0^{\text{out}} = s^{-1} \sum_k h_k^{\text{mix}} (\theta_k + \mathbf{A}_k (\mathbf{u}_{\text{target}} - \mathbf{w}_k)), \quad (7)$$

163 where  $s = \sum_k h_k^{\text{mix}}$  and  $h_k^{\text{mix}} = \exp(-\|\mu - k\|/2\sigma_{\text{mix}}^2)$ , with  $\mu$  representing the winner  
 164 neuron such that  $\mathbf{w}_\mu$  is closest to the vector  $\mathbf{u}_{\text{target}}$ .  $\theta_0^{\text{out}}$  gives a coarse movement to the  
 165 arm such that the end-effector reaches a position  $\mathbf{v}_0$ . The correcting fine movement is  
 166 evaluated using Equation (8) resulting in a final movement of the end-effector to  $\mathbf{v}_1$ .

$$\theta_1^{\text{out}} = \theta_0^{\text{out}} + s^{-1} \sum_k h_k^{\text{mix}} \mathbf{A}_k (\mathbf{u}_{\text{target}} - \mathbf{v}_0). \quad (8)$$

168 The learning scheme used in the present work can be grouped as under:

$$\Delta \mathbf{v} = \mathbf{v}_1 - \mathbf{v}_0, \quad (9)$$

$$\Delta \theta^{\text{out}} = \theta_1^{\text{out}} - \theta_0^{\text{out}}, \quad (10)$$

$$\Delta\theta_k = \theta_0^{\text{out}} - \theta_k^{\text{out}} - \mathbf{A}_k(\mathbf{v}_0 - \mathbf{w}_k), \quad (11)$$

$$\Delta\mathbf{A}_\mu = \|\Delta\mathbf{v}\|^{-2}(\Delta\theta^{\text{out}} - \mathbf{A}_\mu \cdot \Delta\mathbf{v})\Delta\mathbf{v}^T, \quad (12)$$

$$\mathbf{w}_k \leftarrow \mathbf{w}_k + \epsilon h'_{\mu k}(\mathbf{u}_{\text{target}} - \mathbf{w}_k), \quad (13)$$

$$\theta_k \leftarrow \theta_k + \epsilon' h'_{\mu k} \Delta\theta_k, \quad (14)$$

$$\mathbf{A}_k \leftarrow \mathbf{A}_k + \epsilon' h'_{\mu k} \Delta\mathbf{A}_k. \quad (15)$$

176 The functions  $h_{\mu k}$  and  $h'_{\mu k}$  are defined as

$$h_{\mu k} = \exp\left(-\frac{\|\mu - k\|}{2\sigma^2}\right), \quad (16)$$

$$h'_{\mu k} = \exp\left(-\frac{\|\mu - k\|}{2\sigma'^2}\right), \quad (17)$$

179 with  $\mu$  and  $k$  representing the 3-D indices associated with the corresponding neuron.  
 180 The parameters  $\epsilon, \epsilon', \sigma, \sigma'$  and  $\sigma_{\text{mix}}$  vary during the training time depending on the  
 181 current iteration and can be expressed using the general expression as below

$$\eta = \eta_{\text{initial}} \left( \frac{\eta_{\text{final}}}{\eta_{\text{initial}}} \right)^{(t/t_{\text{max}})}. \quad (18)$$

183 In Equation (18),  $\eta \in \{\epsilon, \epsilon', \sigma, \sigma', \sigma_{\text{mix}}\}$ ,  $t$  is the current iteration and  $t_{\text{max}}$  is the total  
 184 number of iterations to be performed by the network.

#### 185 4.2. INDEXING SCHEME FOR THE ADAPTIVE TOPOLOGY OF EXTENDED KSOM

186 In this work, we use an adaptive scheme for determining the topology of the work-  
 187 space using QC of the joint space formed by grouping the end-effector positions and  
 188 the corresponding joint-angles. After the clustering step, a set of points,  $\mathbf{w}_1, \dots, \mathbf{w}_N$ , in  
 189 the workspace are obtained, which are the potential cluster centers and hence denote  
 190 the reference or weight vector for the neurons. The index of each neuron is then  
 191 obtained using a normalized and scaled version of the corresponding weight vector.  
 192 The normalization is carried out by dividing each dimension by the corresponding  
 193 linear dimension of the workspace. It should be noted that in the proposed approach,  
 194 the index values are real values and not necessarily integers, unlike the method in [6]  
 195 and [11]. The motivation for the proposed indexing scheme is that it represents both  
 196 the actual and the relative location of neurons in the workspace.

### 197 5. Implementation and Results

#### 198 5.1. PARAMETERS AND INITIALIZATION

199 The workspace is defined by a region  $200 \text{ mm} \times 300 \text{ mm} \times 200 \text{ mm}$ . The arm lengths  
 200 are all equal to 254 mm and the manipulator has a rigid wrist portion having a

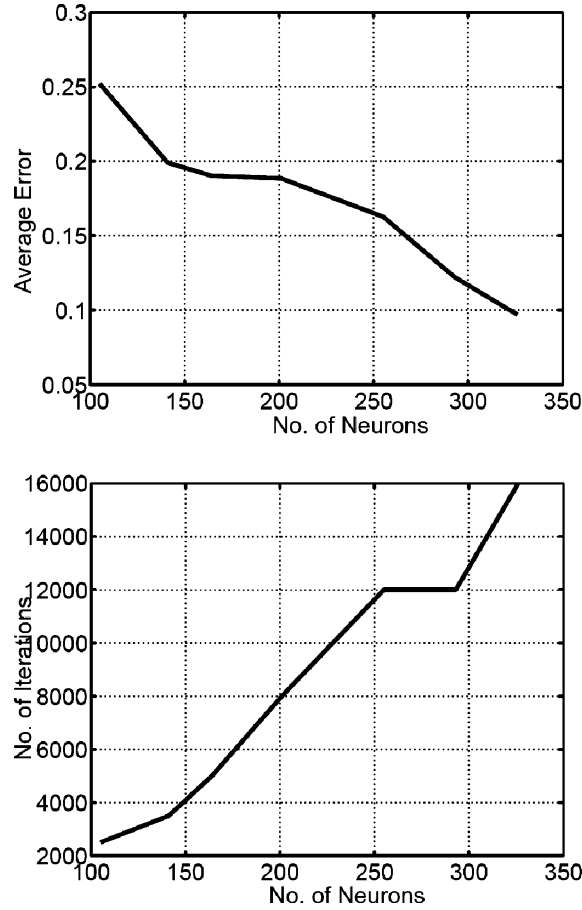


Figure 7. Plot of the average error and no. of iterations required as the number of neurons varies.

Table I. Performance (in terms of the mean normed error) of extended K-SOM based method when 5000 iterations were used to train the system.

SOM topology	No. of neurons	200 Random points	Circle	Sphere
12×7×4	336	0.1438	0.0722	0.0711
8×7×5	280	0.1551	0.0750	0.0715
6×6×5	180	0.1931	0.1162	0.1133
6×5×5	150	0.2122	0.1001	0.0924

201 length equal to 50 mm as used in [1]. The initialization of parameters was similar to  
 202 that in [1]. The values were chosen as  $\epsilon_i = 1.0$ ,  $\epsilon_f = 0.05$ ,  $\epsilon'_i = 0.9$ ,  $\epsilon'_f = 0.9$ ,  $\sigma_i = 2.5$ ,  
 203  $\sigma_f = 0.01$ ,  $\sigma'_i = \sigma_{\text{mix}_i} = 2.5$ ,  $\sigma'_f = \sigma_{\text{mix}_f} = 0.01$ . The subscript i and f denote the initial  
 204 and final values of the corresponding parameters, respectively.

Table II. Performance (in terms of the mean normed error) of QC based method when 5000 iterations are used.

$q = 1/2\sigma^2$	No. of neurons	200 Random points	Circle	Sphere
0.14	255	0.1481	0.0803	0.0968
0.125	181	0.1804	0.10	0.11
0.12	164	0.1814	0.09	0.13
0.11	141	0.2291	0.1088	0.1334

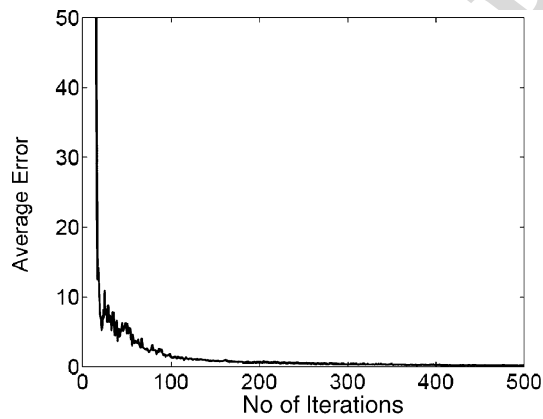


Figure 8. Plot of the average error with the number of iterations for 164 neurons.

## 205 5.2. PERFORMANCE AND RESULTS

206 The topology of the workspace is determined by the number of clusters obtained.  
 207 Unlike the Kohonen-SOM based approach where the workspace topology in the  
 208 form of a 3-D map has to be pre-specified after a heuristic approach mostly based on  
 209 hi-and-trial, the QC-based approach has only one free parameters in form of the  
 210 cluster width parameter  $q = 1/2\sigma^2$ , which controls the number of clusters generated.  
 211 Each cluster denotes the receptive field of a neuron.

212 The proposed algorithm is tested on 200 Random Points in the workspace, a circle  
 213 in a 2-D plane and a Sphere. The error measured is reported using the Mean Nor-  
 214 malized Error (MNE). MNE for a set of dataset with  $N$  points in the workspace is  
 215 defined by Equation (19).

$$\text{MNE} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{u}_{\text{target}} - \mathbf{u}_{\text{predicted}}\|. \quad (19)$$

217 We begin by evaluating the number of neurons which are necessary for repre-  
 218 senting the workspace appropriately. We observe that larger number of neurons give

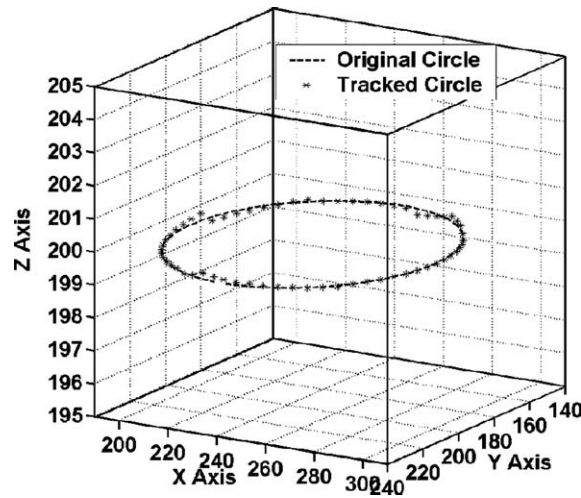


Figure 9. Plot of the tracked circle using 164 neurons.

219 better results, but also require more training. This is evident from Figure 7. Figure 7  
 220 shows that increasing the number of neurons shows a decline in the MNE. However,  
 221 for faster training, it is important to have only a sufficient number of neurons.

222 Our simulations lead us to the conclusion that the adaptive topology based scheme  
 223 proposed in this work needs only 164 neurons for the workspace in use. This number  
 224 can be compared with the work in [1], where  $12 \times 7 \times 4 (= 336)$  neurons are used. The  
 225 result for three situations using different SOM topology and also with different  
 226 cluster widths is demonstrated in Tables I and II. They show that the QC based  
 227 method achieves better results with fewer neuron units, mainly because of the fact  
 228 that, the topology of the workspace is not fixed but is determined through a joint  
 229 clustering process.

230 The error vs. iterations graph is shown in Figure 8 for an adaptive scheme of 164  
 231 neurons. Figure 9 shows a circle being tracked in the workspace with an average  
 232 error of 0.09 mm. These results are comparable to that obtained using KSOM-based  
 233 method where a much higher number of neurons were used.

## 234 6. Conclusion

235 In this work, a new method for the visual-motor coordination Control using an  
 236 adaptive scheme and a flexible topology for the workspace is proposed. The novelty  
 237 of the method lies in a joint input–output partitioning scheme. This leads to fewer  
 238 neurons being used with the receptive field of every neuron having a non-uniform  
 239 distribution in space. The motor control is carried out through a collection of  
 240 neurons, all in a neighborhood, firing and determining the precise movement, instead  
 241 of one neuron acting independently. QC used in this work, determines the local

242 maxima of the potential and identifies them as the cluster centers. It should be noted  
 243 that the method still adopts a classical computation methodology.

#### 244 **References**

- 245 1. Behera, L. and Kirubanandan, N.: A hybrid neural control scheme for visual-motor  
 246 coordination. *IEEE Control Systems Magazine*, **19**(1999), 34–41.  
 247 2. Duda, R. Hart, P. and Stork, D.: *Pattern Classification*. John Wiley & Sons, 2001.  
 248 3. Horn, D.: Clustering via hilbert space. *Physica A*, **302**(2001), 70–79.  
 249 4. Kuperstein, M.: Adaptive visual – motor coordination in multijoint robots using parallel  
 250 architectures. In *Proceedings of the IEEE International Automation and Robotics*, pp.  
 251 1595–1602, 1990.  
 252 5. Kuperstein, M.: Neural model of adaptive hand-eye coordination for single postures.  
 253 *Science*, **239**(1998), 1308–1311.  
 254 6. Martinetz, T. Ritter, H. and Schulten, K.: Three dimensional neural network for learning  
 255 visuomotor coordination of a robot arm. *IEEE Transactions on Neural Networks*, **1**(1990),  
 256 131–136.  
 257 7. Ripley, B.: *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.  
 258 8. Ritter, H. Martinetz, T. and Schulten, K.: Topology conserving maps for learning  
 259 visuomotor coordination. *Neural Networks*, **2**(1998), 159–168.  
 260 9. Roberts, S.: Non-parametric unsupervised cluster analysis. *Pattern Recognition*, **30**(1997),  
 261 261–272.  
 262 10. Vapnik, V.: *Statistical Learning Theory*. John Wiley & Sons, 1998.  
 263 11. Walter, J. and Schulten, K.: Implementation of self-organizing neural networks for visuo-  
 264 motor control of an industrial robot. *IEEE Transactions on Neural Networks*, **4**(1993),  
 265 86–95.  
 266

