

models to control the new trajectory. The idea of using a resonance parameter is based on the working principle of the human mind in recalling only the close or relevant experiences, given a new situation [17],[18],[19].

The proposed method has been implemented using a seven-link industrial manipulator called PA-10, manufactured by Mitsubishi Heavy Industries Ltd. The reference input of joint angle, its velocity, and their output data obtained from the experiments are used for the system identification studies. Promising results have been obtained to prove the ability of the proposed method in capturing nonlinear dynamics of a multi-link manipulator in an effective manner.

II. MODEL CONSTRUCTION AND CONTROLLER DESIGN

In the case of model based manipulator control, the global identification of dynamics with a finite number of RBFs is a formidable task for a complex manipulator such as PA-10. Therefore identifying local nonlinear dynamics using a reduced number of activation functions in the NNs for a fairly complex trajectory in the workspace saves computation time and usage of computer resources. For a complex trajectory, it is best to identify several nonlinear primitive models rather than a set of linear primitive models. Therefore the approach tries to keep a set of pre-identified nonlinear dynamic models in the workspace of the manipulator to fuse them in an effective manner.

The proposed method attempts to identify the dynamics involving only the reference input of the joint angle, its velocity, and their output information, using the RKGNNs trained by an evolutionary algorithm. This needs rearranging of the dynamic equation of a manipulator to form an ODE of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$. Given the dynamics of the manipulator as in equation (1), the state vector \mathbf{x} can be defined by $\mathbf{x} = [\boldsymbol{\theta} \ \dot{\boldsymbol{\theta}}]^T$. Then the ODE given by (2) can be formed using \mathbf{x} and the dynamics defined by equation (1) as

$$\begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{f}_0(\boldsymbol{\theta}) \end{bmatrix}$$

where

$$\mathbf{f}_0(\boldsymbol{\theta}) = \mathbf{M}^{-1}(\boldsymbol{\theta})\{\boldsymbol{\tau} - \mathbf{V}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) - \mathbf{G}(\boldsymbol{\theta}) - \mathbf{F}(\boldsymbol{\theta})\} \quad (1)$$

The right-hand side of the above ODE gives the rate of change of the states given by the left-hand side. Therefore for this family of ODEs, RKGNNs may very accurately identify the right hand side.

For a given trajectory, the input reference angle, its velocity and their output data of all seven joints were experimentally obtained. Then these experimental data were used to train the RKGNNs to identify the dynamics of the manipulator. The weights of the RKGNN and the RBF parameters, $\bar{\mathbf{y}}_l$ and σ_l , were found using the evolutionary algorithm described in references [14] and [15], with the objective function subjected to constraints. The constrained

objective function is given by

$$\begin{aligned} J = & \frac{1}{t_N} \sum_{i=1}^{t_N} \left(\|\boldsymbol{\theta}_d(i) - \boldsymbol{\theta}(i)\| + \|\dot{\boldsymbol{\theta}}_d(i) - \dot{\boldsymbol{\theta}}(i)\| \right) \\ & + \frac{s_t}{2} \sum_{j=1}^{c_N} ([g_j^+(\boldsymbol{\theta})]^2) \end{aligned} \quad (2)$$

where t_N is the total time span of the trajectory, $g_j^+(\boldsymbol{\theta}) = \max\{0, g_j(\boldsymbol{\theta})\}$ is the additional penalty function for the j th constraint satisfaction and s_t is a log penalty parameter such that $s_t = 10 + \log(k + 1)$, k is the number of generations, and c_N is the number of constraints.

One constraint was considered: which is given by

$$g_1(\boldsymbol{\theta}) : -\Gamma(\boldsymbol{\theta}) \leq 0 \quad (3)$$

where $\Gamma(\boldsymbol{\theta})$ is the minimum principle minor of the inertia matrix.

In this application, the dynamics were identified for two given trajectories for simplicity and due to time constraints. The two trajectories marked "1" and "2" in Fig. 1 were used for the dynamics identification of the PA-10 manipulator using RKGNNs. Moreover, the new trajectory for which the proposed control concept is tested is shown in solid lines. In Fig. 1, the trajectories are presented in end effector coordinates of position (x, y, z) and orientation (Yaw, Pitch, Roll). It is important to notice from Fig. 1, that all end-effector position and orientation of the two trajectories used for dynamics identification are fairly different specially in position y and pitch orientation. Furthermore, it is clear that the new trajectory used for testing has a fair membership in both the trajectories 1, and 2. The parameters of the RKGNNs were trained using an evolutionary algorithm proposed by the authors in [14]-[15] to identify the dynamics of the PA-10 manipulator around the trajectories 1 and 2.

III. COMPARISON OF A NEW SITUATION WITH EXISTING REPRESENTATIONS

The objective of this process is to assign fuzzy confidences to each stored trajectory for which dynamic models were identified in the form of NNs, with respect to the new objective trajectory $O_{tr}(\boldsymbol{\theta}_d, \dot{\boldsymbol{\theta}}_d)$. This is the first step to assess the relevance of each pre-learnt model to the new trajectory's work-space.

In general humans associate experiences with situations, and the confidence with which a particular experience is applied to a given situation is interpreted in linguistic terms, such as "highly related," "weakly related," *etc.* Therefore, the objective trajectory $O_{tr}(\boldsymbol{\theta}_d, \dot{\boldsymbol{\theta}}_d)$ is taken as input to assign fuzzy confidences to each stored trajectory for which models are identified. The trajectories in this case correspond to the situations and the dynamic models to the specific experiences in human mind. The joint angles and their velocity data are fuzzified using radial basis function fuzzy sets (RBFfs) with the joint angles and their velocity data of each stored trajectory as the center of each

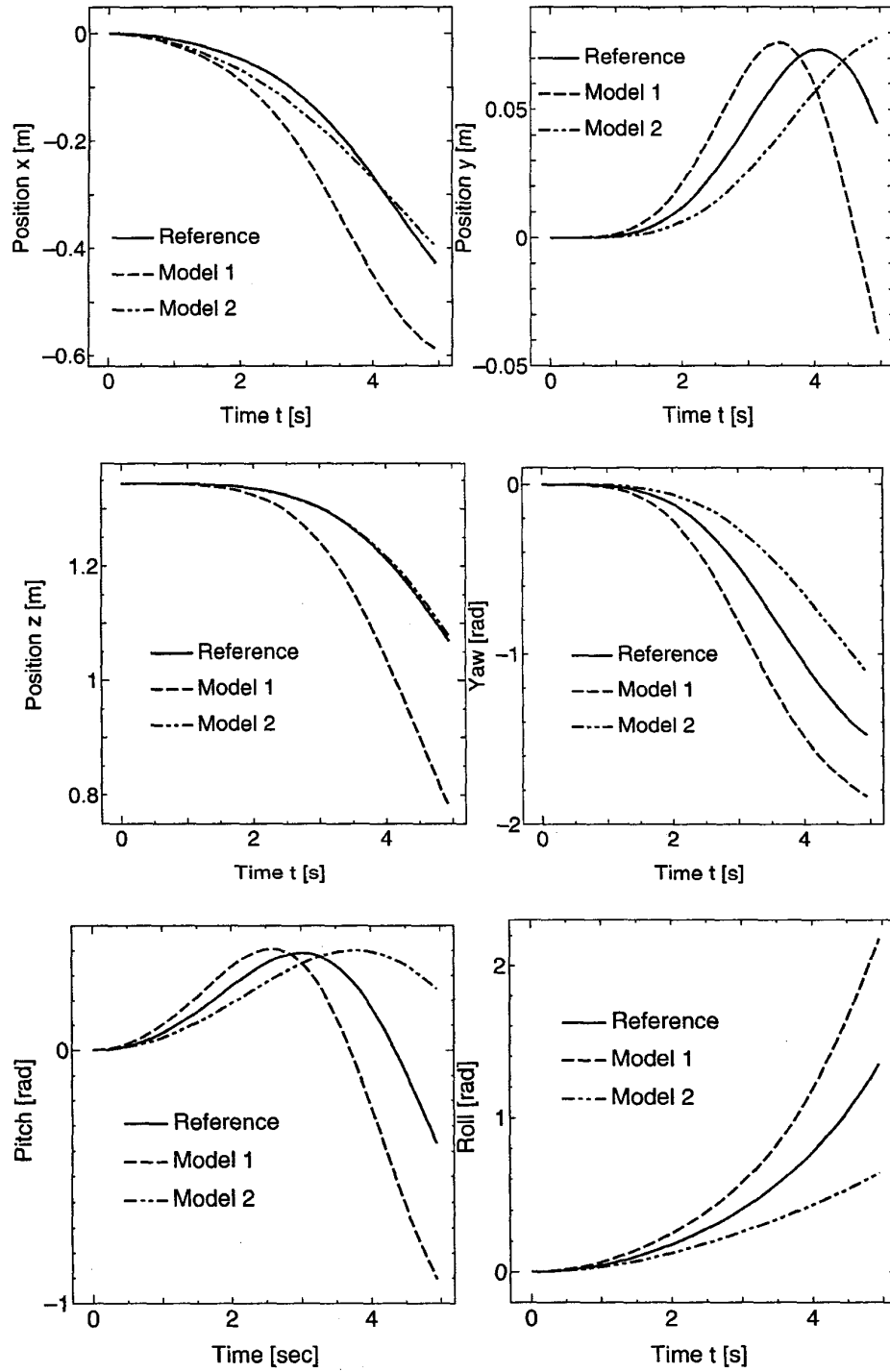


Fig. 1. Comparison of the new trajectory with the ones used for training in end-effector coordinates.

RBFFS. The reason behind using RBFFSs is that it enables direct comparison between the vectors of joint angles and their velocity of the objective trajectory, and those of any of the trajectories for which the dynamics are already

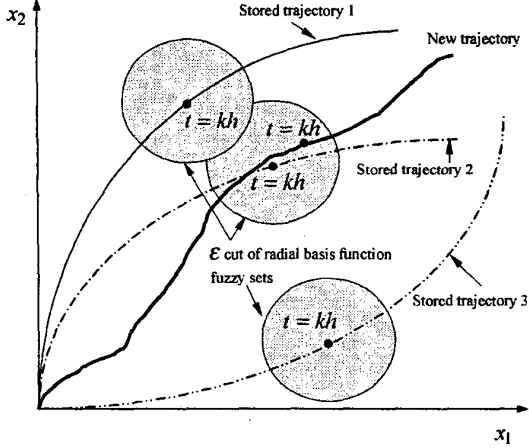


Fig. 2. The basic concept of using the resonance parameter.

identified. This approach is advantageous over taking the burden of comparing each joint angle and its velocity separately. Moreover the usage of RBFFSs has the advantage of reduced computation of confidences.

The fuzzy confidences assigned to O_{tr} by each stored trajectory are given in the vector,

$[\mu_{1,O_{tr}}, \mu_{2,O_{tr}}, \mu_{3,O_{tr}}, \dots, \mu_{p,O_{tr}}]^T$. These confidence values are calculated by

$$\mu_{j,O_{tr}} = \exp\{-[(\mathbf{v} - \mathbf{v}_j)^T(\mathbf{v} - \mathbf{v}_j)/\sigma_j^2]\}, j = 1, 2, \dots, p_m \quad (4)$$

where p_m is the number of trajectories, for which the manipulator dynamics have been identified previously, $\mathbf{v} = [\dot{\theta} \ \dot{\theta}]^T$, and σ_j is the standard deviation of the j th RBFFS. This process works as the sub-conscious part of the proposed controller.

IV. CONFIDENCE BASED MODEL SELECTION AND EXECUTION

The sub-conscious process of assigning confidences to the situations in which existing experiences were acquired, leads to generating resonance signals to the memory to retrieve the specific experiences relevant to the given situation. The important fact is that at last, only a few experiences that are highly related to the new situation are reminded for detailed consideration, from among a vast base of experiences gathered during one's whole life. Here in this method, a resonance parameter ϵ is used to mimic this human cognitive process of experience mining. Therefore the resonance parameter ϵ screens the appropriate models to be used to control the new trajectory $O_{tr}(\theta_d, \dot{\theta}_d)$ such that model j of trajectory j is selected if $\mu_{j,O_{tr}} > \epsilon$. This approach guarantees stability by selecting the nonlinear models only in the neighborhood of the objective trajectory. Moreover it reduces computation time by neglecting ineffective dynamic models. The designer should select a value for ϵ after considering the range of state space covered by each local nonlinear model, computation time

for each model, accuracy needed, etc. In this case $\epsilon = 0.3$. The basic idea of using the resonance parameter is depicted in Fig. 2 for a two dimensional case, where x_1, x_2 are the characteristic variables of the trajectory, t is time, i is the sampling step, h is sampling time width. It can be seen that at $t = ih$, only the ϵ cut of the RBFFS of stored trajectory 2 encircles the new trajectory. Therefore in this case the dynamic model identified for the stored trajectory 2 is taken to control the new trajectory. In another case more than one ϵ cuts of the RBFFSs can encircle the given trajectory at any given time. For the PA-10 case, the vector of the characteristic variables of the trajectories consists of 14 variables for the joint angles and their velocity.

The selected models are used to generate joint velocity commands by using each nonlinear model as given by

$$\ddot{\theta}_{cj} = \hat{M}_j^{-1}(\theta) \left[\tau_{dj} + \{K_p e + K_v \dot{e} + K_I \int e\} + \hat{V}_j(\theta, \dot{\theta}) + \hat{G}_j(\theta) \right] \quad (5)$$

where $\ddot{\theta}_{cj}$ is the manipulated joint acceleration; \hat{M}_j^{-1} is the estimated inertia matrix; τ_{dj} is the torque command estimated by the torque network; K_p , K_v ; and K_I are the proportional; derivative and integral gains respectively; e is the vector of joint angle errors; $\hat{V}_j(\theta, \dot{\theta})$ is the estimated velocity force matrix; and $\hat{G}_j(\theta)$ is the estimated gravitational force matrix. In this case, j denotes the j th model trained using RKGNNs.

V. FUZZY EVALUATION OF FINAL JOINT VELOCITY COMMAND

A very complex cognitive process in human mind is the fusion of isolated experiences to work in a given environment. It is hard to shrink this idea down to a comprehensive mathematical model. Yet the advent of fuzzy linguistic reasoning opens a fair basis to establish a simple mathematical process to fuse isolated mathematical models. This fusion process is carried out using the fuzzy confidences already assigned to each dynamic model.

First the selected models were used to generate joint space acceleration commands to control the given trajectory. Then the joint space acceleration commands assigned to the given trajectory by each model is taken and a defuzzified final control command is calculated by

$$\dot{\theta}_c = \int \left(\frac{\sum_{j=1}^q \mu_{j,O_{tr}} \ddot{\theta}_{cj}}{\sum_{j=1}^q \mu_{j,O_{tr}}} + \Omega(e, \dot{e}) \right) dt \quad (6)$$

where $\dot{\theta}_c$ is the vector of final manipulated joint velocity commands, q is the number of models selected for control the objective trajectory $O_{tr}(\theta_d, \dot{\theta}_d)$, and $\Omega(e, \dot{e})$ is the correction factor to compensate for the unmodeled dynamics and the uncertain friction forces. In this case it is calculated as

$$\Omega(e, \dot{e}) = K_p^0 e + K_v^0 \dot{e} \quad (7)$$

where K_p^0 and K_v^0 are proportional and derivative gains, respectively. In this application,

$$\begin{aligned}
\mathbf{K}_p &= 2\mathbf{I}_{7 \times 7}, \\
\mathbf{K}_v &= 7.5\mathbf{I}_{7 \times 7}, \\
\mathbf{K}_i &= 0.001\mathbf{I}_{7 \times 7}, \\
\mathbf{K}_p^0 &= 2\mathbf{I}_{7 \times 7}, \\
\mathbf{K}_v^0 &= 2\mathbf{I}_{7 \times 7}.
\end{aligned}$$

According to equation (6), the calculation of the vector of final manipulated joint velocity commands involves two operations. The major contribution comes from the fuzzy fusion of the joint space acceleration commands given independently by the stored models. This is performed by first calculating the vectors of independent joint space acceleration commands calculated based on the stored models given by equation (5), and then calculating a vector of final joint space acceleration commands using the fuzzy confidences assigned to the given trajectory by the stored trajectories. Note that out of p number of stored models, only a q number of models that are deemed to be closely related to the given trajectory, are selected at a given time. This screening is performed using a cognitive resonance parameter.

The second operation in equation (6) to calculate the vector of final manipulated joint acceleration commands is given in equation (7). This is intended to compensate for the unmodeled dynamics and the uncertain friction forces. Of course one could argue that the integral control part shown in equation (5) compensates the unmodeled dynamics. Yet in this case, RBFNNs are adopted to identify the dynamics of a manipulator. Suppose a situation arises where the input to the NNs is far away from the region covered in the input space by the RBFs. In this case, there can be a possibility that the activation of the RBFs is weak. Therefore, in such a situation, no matter how effective the integral controller is, multiplication by the RBFNN based estimated inverse of the inertia matrix as shown in equation (5) to calculate the joint space acceleration command, would result in weak control performance. The second operation given by equation (7) strategically covers this kind of situations.

Next consider the tuning of gains in equations (5) and (7). Obtaining a set of theoretically optimum gains in equations (5) and (7) could be very much easier if the dynamic model was exactly the same as the actual dynamics of the manipulator. The fact that this is not the case in estimated dynamics, the mismatch between actual and modelled dynamics may cause servo errors to be excited. Therefore, care should be taken in selecting the values for $\mathbf{K}_p \sim \mathbf{K}_v^0$ in equations (6) and (7). Therefore, in this case the gain tuning was done using an experimental trial and error method to ensure reliable performance. For simplicity, the gain vectors of proportional, derivative and integral control parts were designed to have the same gain value for all the joints. Of course selecting special gain values for each joint could be better than selecting the same value for all joints, yet it is not considered worthwhile here, because gain tuning is not a major purpose of this study.

VI. EXPERIMENTAL RESULTS ON PA-10 MANIPULATOR

The experiments were carried out using an actual PA-10 manipulator manufactured by Mitsubishi Heavy Industries

Ltd. A simple visual C++ program was written as a console application to access the ports for data communication between the servo controller and the computer. Furthermore, the main advantage of the proposed method here, is that even if the sampling time width is changed, the NNs do not have to be trained again because the sampling time width is external to the function NNs in the RKGNNs. One other approach to reduce the sampling period is to set the resonance parameter in the proposed method so that the number of models selected at any given time out of the stored models is fixed to be a small number, resulting in less calculations of a PC based controller.

For the purpose of comparison, experiments were carried out with the conventional method widely adopted in the industry for applications such as arc welding and laser cutting [20]. In industrial mechatronic servo applications, servo systems are connected to each axis of the manipulator and they are controlled independently. The servo controllers control the angle and the angular velocity of the motors, not the position and orientation of the end-effector. In the conventional method, the nonlinear dynamic effects of the composite manipulator system is ignored and the dynamics of the mechatronic servo systems are assumed to be those of the servo controllers and the motors without any concern of the mechanisms. Therefore the conventional method considers the robot manipulator as a completely decoupled system and the control performance mainly depends on the servo controllers for each motor. This type of controllers is commonly known as semi-closed-loop controllers. To implement the conventional method to control the manipulator for a given desired trajectory, the position and orientation data should be transformed to the joint angles and angular velocities. Then the joint space data are given as a teaching signal to the servo controller as input, without any modification to compensate for the dynamics of the manipulator. Therefore, the control performance deteriorates in high-speed servo applications due to the dynamics of the manipulator becoming prominent at elevated speeds. This drawback has forced industrial mechatronic servo controllers using the conventional method to adopt a slow working speed that has a direct adverse impact on the production efficiency. In the case of redundant manipulators where the effect of the dynamics of the manipulator mechanism is significant especially at high speeds, the conventional method is expected to give very poor results. Therefore effective compensation of dynamics is the solution given in the proposed method. The accuracy of the dynamics identified using the RKGNNs is important to validate the effectiveness of the proposed method. As stated earlier, the proposed method attempts to improve the accuracy and the information processing capacity of the RBFNNs by several methods. One is the structuring of the NNs to clearly represent the structure of the dynamics of robotic manipulators. This effective structuring of the NN results in higher accuracy and higher information processing capacity. The most important feature here is the symmetry and positive definiteness of the inertia matrix. To ensure that the inertia matrix evolves to be a positive def-

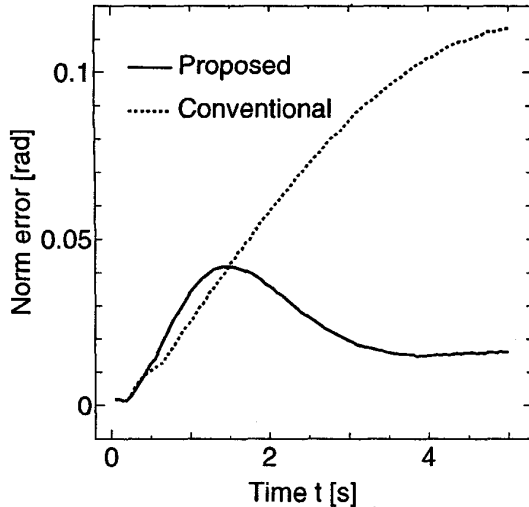


Fig. 3. Comparison of tracking error.

inite matrix, a special constrained optimization technique was adopted where the log penalty parameter s_t given in equation (2). In addition, s_t was scaled based on the importance of the constraints. Therefore the log penalty parameter is large enough to exert pressure on the individuals of the evolutionary algorithm to survive only if the resultant inertia matrix is positive definite. The angular space norm error is shown in Fig. 3, calculated by $\|\theta_{rd} - \theta_r\|$, where, θ_{rd} is the r th reference joint angle, and θ_r is the r th followed angle by the manipulator. It can be clearly seen in Fig. 3 that the performance of the proposed method is superior to that of the conventional method. Note that in the conventional method, the error increases in the latter part as the angular velocities of the manipulator joints increase. This is exactly what is expected in the conventional method due to the drawback of poor performance at elevated speeds, resulting from poor compensation of mechanical dynamics as explained above. The conventional method seems very simple though, imposes constraints on the production speed, reducing the working efficiency. The proposed method has been able to reduce the error as the joint angular speeds increase, due to better compensation of dynamics. Therefore the proposed method provides a promising solution to the current industrial problem of increasing production efficiency with a good product quality due to good control performance.

REFERENCES

- [1] K. A. Thoroughman and R. Shadmehr, "Electromyographic Correlates of Learning an Internal Model of Reaching Movements," *The Journal of Neuroscience*, vol. 19, no. 19, pp. 8573–8588, 1999.
- [2] R. Shadmehr and K. A. Thoroughman, "Learning and Memory Formation of Arm Movements," in: J. M. Winters and P. E. Crago (eds), *Biomechanics and Neural Control of Posture and Movement*, Soringer-Verlag, New York, pp. 347–353, 2000.
- [3] K. A. Thoroughman and R. Shadmehr, "Learning of Action Through Adaptive Combination of Motor Primitives," *Nature*, vol. 407, no. 12, pp. 742–747, 2000.

- [4] N. V. Dounskaia, S. P. Swinnen, and C. B. Walter, "A Principle of Control of Rapid Multijoint Movements," in: J. M. Winters and P. E. Crago (eds), *Biomechanics and Neural Control of Posture and Movement*, Soringer-Verlag, New York, pp. 390–404, 2000.
- [5] S. Giszter, M. Davies, and W. Kargo, "Augmenting Postural Primitives in Spinal Cord: Dynamic Force-Field Structures Used in Trajectory Generation," in: J. M. Winters and P. E. Crago (eds), *Biomechanics and Neural Control of Posture and Movement*, Soringer-Verlag, New York, pp. 334–346, 2000.
- [6] F. A. Mussa-Ivaldi, S. F. Giszter, and E. Bizzi, "Linear Combinations of Primitives in Vertebrate Motor Control," *Proc. Natl. Acad. Sci., USA*, vol. 91, pp. 7534–7538, 1994.
- [7] R. J. Dolan, G. R. Flink, E. Rolls, M. Booth, A. Holmes, R. S. J. Frackowiak, and K. J. Friston, "How the Brain Learns to see Objects and Faces in an Impoverished Context," *Nature*, vol. 389, no. 9, pp. 596–599, 1997.
- [8] R. Palm and D. Driankov, *Model Based Fuzzy Control*, Springer-Verlag, Berlin, 1997.
- [9] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and its Applications to Modeling and Control," *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, 1985.
- [10] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338–353, 1965.
- [11] D. M. Wolpert and M. Kawato "Multiple Paired Forward and Inverse Models for Motor Control," *Neural Networks*, vol. 11, pp. 1317 – 1329, 1998.
- [12] K. Kiguchi and T. Fukuda, "Fuzzy Selection of Fuzzy-Neuro Robot Force Controllers in an Unknown Environment," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1182–1187, 1999.
- [13] M. Ravindranathan and R. Leitch, "Model Switching in Intelligent Control Systems," *Artificial Intelligence in Engineering*, vol. 13, no. 2, pp. 175–187, 1999.
- [14] T. Nanayakkara, K. Watanabe and K. Izumi, "Evolving in Dynamic Environments Through Adaptive Chaotic Mutation," in *Proc. of Fourth International Symposium on Artificial Life and Robotics*, vol. 2, pp. 520–523, 1999.
- [15] T. Nanayakkara, K. Watanabe, K. Kiguchi and K. Izumi, "Adaptive Optimization in a Class of Dynamic Environments Using an Evolutionary Approach," Submitted to the *Evolutionary Computation Journal*. MIT press.
- [16] T. Nanayakkara, K. Watanabe, K. Kiguchi and K. Izumi, "Evolutionary Learning of a Fuzzy behavior Based Controller for a Non-Holonomic Mobile Robot in a Class of Dynamic Environments." Accepted to be published in the *International Journal of Intelligent and Robotic Systems*.
- [17] D. E. Shulz, R. Sosnik, V. Ego, S. Haidarliu, and E. Ahissar, "A Neuronal Analogue of State-Dependent Learning," *Nature*, vol. 403, pp. 549–553, 2000.
- [18] T. J. Sejnowsky, "Synapses Get Smarter," *Nature*, vol. 382, pp. 759–760, 1996.
- [19] H. Markram and M. Tsodyks, "Redistribution of Synaptic Efficacy Between Neurocortical Pyramidal Neurons," *Nature*, vol. 382, pp. 807–810, 1996.
- [20] S. Goto, M. Nakamura and N. Kyura, "Accurate Contour Control of Mechatronic Servo Systems Using Gaussian Networks," *IEEE Trans. on Industrial Electronics*, vol. 43, no. 4, pp. 469–476, 1996.