Adaptive Fuzzy Systems in Computational Intelligence

Hamid R. Berenji
Intelligent Inference Systems Corporation
Computational Sciences Division
NASA Ames Research Center
Moffett Field, CA

ADAPTIVE FUZZY SYSTEMS IN COMPUTATIONAL INTELLIGENCE

Hamid R. Berenji
Intelligent Inference Systems Corporation[1]
Computational Sciences Division
Mail Stop 269-2
NASA Ames Research Center
Moffett Field, CA 94035

ABSTRACT

In recent years, the interest in computational intelligence techniques, which currently includes neural networks, fuzzy systems, and evolutionary programming, has grown significantly and a number of their applications have been developed in the government and industry. In future, an essential element in these systems will be fuzzy systems that can learn from experience by using neural networks in refining their performances.

The GARIC architecture, introduced earlier, is an example of a fuzzy reinforcement learning system which has been applied in several control domains such as cart-pole balancing, simulation of the Space Shuttle orbital operations, and tether control. A number of examples from GARIC's applications in these domains will be demonstrated. For more details on the following notes see Refs. 5, 1, 3, 4, 7, 6 and 2.

[1]444 Castro Street, Suite 1014, Mountain View, CA 94041

FUZZY SYSTEMS THAT CAN LEARN

Hamid R. Berenji
Intelligent Inference Systems Corporation
Computational Sciences Division
NASA Ames Research Center

POTENTIALS FOR USE IN NEW MILLENNIUM AND ACCESS TO SPACE PROJECTS

- Demo flights
  - Rapid development of control systems using approximate control rules
  - Automatic refinement of these control systems
- Rendezvous and docking of spacecrafts
- Landing of asteroids
- Rovers to survey planet's surface
- Miniaturization by using smart sensors

EVOLUTION OF FUZZY SYSTEMS

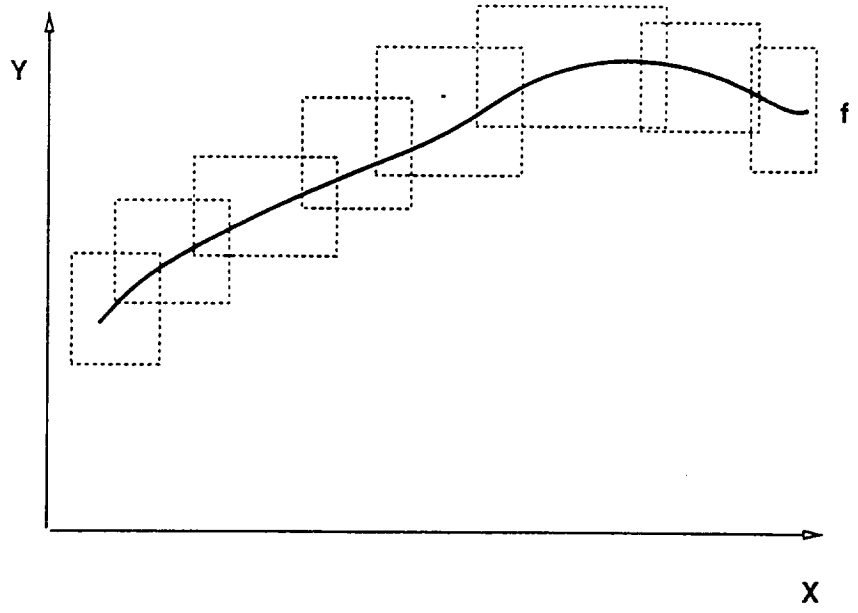
- Stage I:
  - Introducing the fuzzy sets idea
- Stage II:
  - Demonstrating applications
  - Dominated by engineering (control)
- Stage III:
  - Learning fuzzy systems

MOTIVATIONS FOR USING FUZZY LOGIC AND NEURAL NETWORKS IN CONTROL

- Human expert controllers perform well using approximate reasoning
- An analytical model may not always be available.
- If a physical system learns to control itself, then it is **intelligent**.
- Fuzzy logic and neural networks facilitate **interpolation** which removes many of the restrictions of symbolic systems.

FUZZY SYSTEMS AS UNIVERSAL APPROXIMATORS

- Fuzzy rules as patches for approximating a function.
- A fuzzy system can approximate any real continuous function to any degree of accuracy



LEARNING METHODS

Supervised learning
- Reinforcement learning
Unsupervised learning

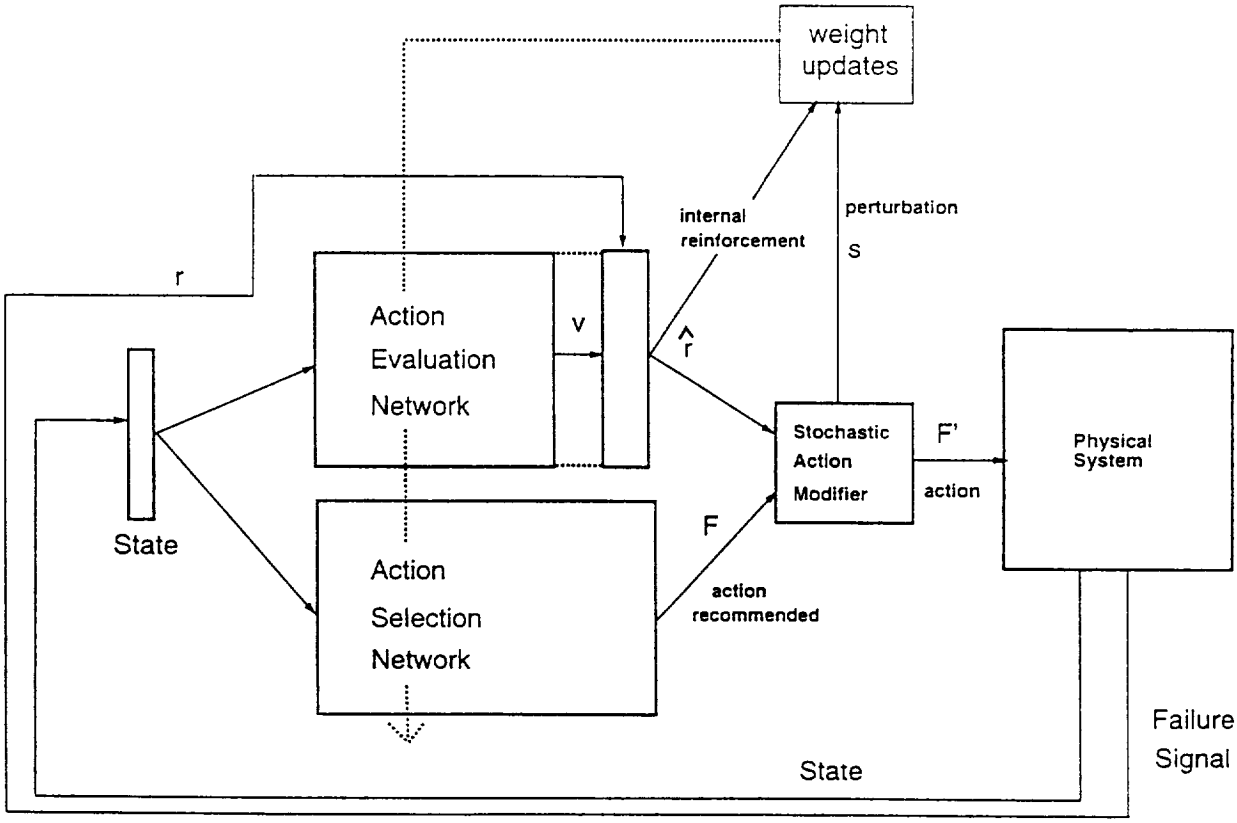
- In supervised learning, a teacher provides the desired control objective at each time step to the learning system.
- In reinforcement learning, the teacher's response is not as direct, immediate, and informative as in supervised learning and it serves more to evaluate the state of the system.
- The presence of a teacher or a supervisor to provide the correct control response is not assumed in unsupervised learning.

- If supervised learning can be used in control (e.g., when the input-output training data is available), it has been shown that it is more efficient than reinforcement learning.
- Many control problems require selecting control actions whose consequences emerge over uncertain periods for which input-output training data are not readily available. In such domains, reinforcement learning techniques are more appropriate than supervised learning.

REINFORCEMENT LEARNING

- Assumes no supervisor to critically judge the chosen control action at each time step.
- The learning system is told indirectly about the effect of its chosen control action.
- Previous works: Samuel's checkers-playing program, Michie and Chambers BOXES system, Barto, Sutton, and Anderson's AHC algorithm.
- Reinforcement learning has its roots in studies of animal learning, and *learning automata* research in control engineering.
- Construct an internal evaluator or a *critic* capable of evaluating the dynamic system's performance.

GENERALIZED APPROXIMATE REASONING-BASED INTELLIGENT CONTROL (GARIC)
(H. Berenji and P. Khedhar)



THE GARIC ARCHITECTURE

- The Action Selection Network maps a state vector into a recommended action F, using fuzzy inference.
- The Action Evaluation Network maps a state vector and a failure signal into a scalar score which indicates state goodness. This is also used to produce internal reinforcement $\hat{r}$.
- The Stochastic Action Modifier uses both F and $\hat{r}$ to produce an action F' which is applied to the plant.

- **Layer 1:** the input layer, consisting of the real-valued input variables.
- **Layer 2:** nodes represent possible values of linguistic variables in Layer 1.
- **Layer 3:** conjunction of all the antecedent conditions in a rule using *softmin* operation.
- **Layer 4:** a node corresponds to a consequent label with an output.
- **Layer 5:** nodes as output action variables where the inputs come from Layer 3 and Layer 4.

THE ACTION EVALUATION NETWORK (Cont.)

- The output unit of the evaluation network:

$$\upsilon[t,t+1] = \sum_{i=1}^{n} b_i[t]\, x_i[t+1] + \sum_{i=1}^{h} c_i[t]\, y_i[t,t+1]$$

where υ is the prediction of reinforcement.

- Evaluation of the recommended action:

$$\hat{r}[t+1] = \begin{cases} 0 & \text{start;} \\ r[t+1] - \upsilon[t,t] & \text{failure;} \\ r[t+1] + \gamma\,\upsilon[t,t+1] - \upsilon[t,t] & \text{else} \end{cases}$$

where $0 \leq \gamma \leq$ is the *discount rate*.

THE ACTION EVALUATION NETWORK (Cont.)

- The input is the state of the plant and the output is an evaluation of the state (a score), denoted by υ.

- The υ-value is suitably discounted and combined with the external failure signal to produce internal reinforcement $\hat{r}$.

- The output of the units in the hidden layer is:

$$y_i\left[t,t+1\right] = g\left(\sum_{j=1}^{n} a_{ij}\left[t\right] x_j\left[t+1\right]\right)$$

where

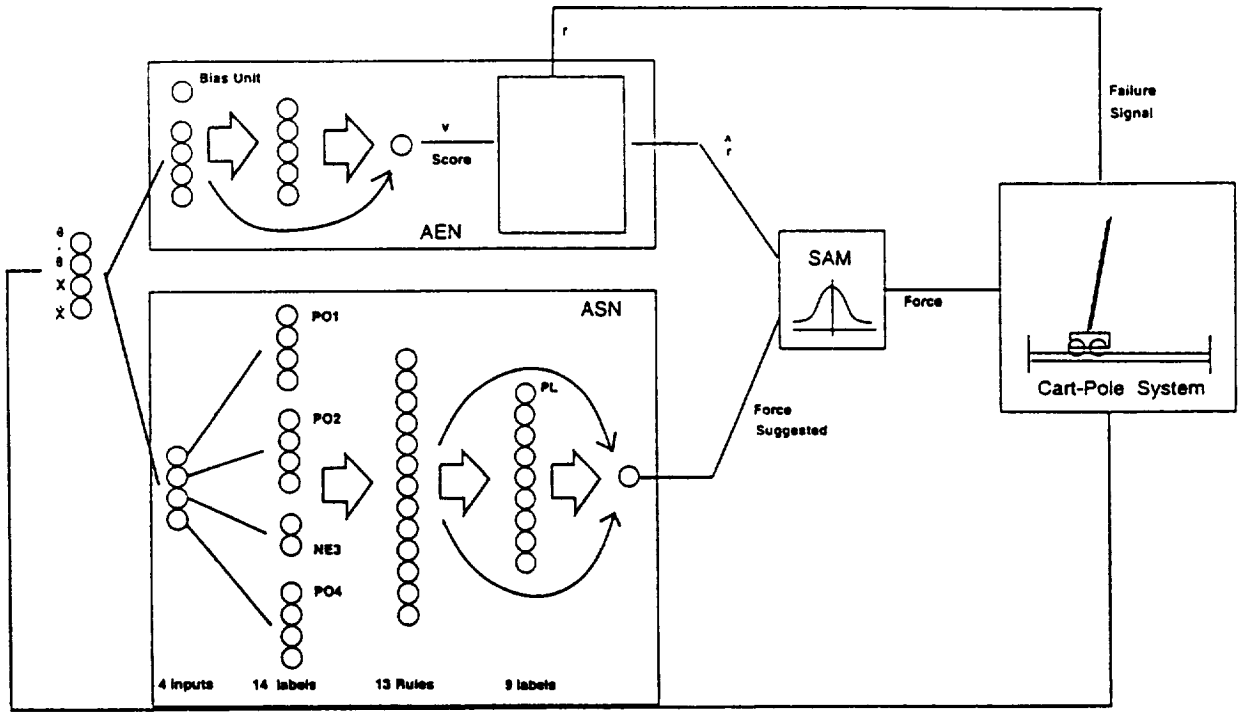
$$g(s) = \frac{1}{1+e^{-s}}$$

and t and $t+1$ are successive time steps.

THE ACTION EVALUATION NETWORK

- The AEN plays the role of an adaptive critic element and constantly predicts reinforcements associated with different input states.
- The only information received by the AEN is the state of the physical system in terms of its state variables and whether or not a failure has occurred.
- The AEN is a standard two-layer feed forward net with sigmoids everywhere except in the output layer.

GARIC APPLIED TO CART-POLE BALANCING



- We use the following learning rule

$$\Delta p = \eta \frac{\partial \upsilon}{\partial p} = \eta \frac{\partial \upsilon}{\partial F} \frac{\partial F}{\partial p}$$

- We assume that $\partial \upsilon / \partial F$ can be computed by the instantaneous difference ratio

$$\frac{\partial \upsilon}{\partial F} \approx \frac{d\upsilon}{dF} \approx \frac{\upsilon(t) - \upsilon(t-1)}{F(t) - F(t-1)}$$

STOCHASTIC ACTION MODIFIER

- Uses $\hat{r}$ from the previous time step and the action F recommended by the ASN to generate a Gaussian random action F' with mean F and standard deviation $\sigma\left(\hat{r}\left(t-1\right)\right)$. $\sigma(\ )$ is a non-negative and monotone decreasing function such as $\exp\left(-\hat{r}\right)$. F' is applied to the plant.

- Stochastic perturbation leads to a better exploration of state space and better generalization ability:

$$s(t) = \frac{F'(t) - F(t)}{\sigma\left(\hat{r}\left(t-1\right)\right)}$$

- The magnitude of the deviation $\left|F' - F\right|$ is large when $\hat{r}$ is low, and small when the internal reinforcement is high.

RULE STRENGTH CALCULATION USING SOFTMIN OPERATOR

- Using the softmin, the *strength* of Rule 1 is:

$$w_1 = \frac{\mu_{A_1}(x_0)e^{-k\mu_{A_1}(x_o)} + \mu_{B_1}(y_0)e^{-k\mu_{B_1}(y_o)}}{e^{-k\mu_{A_1}(x_o)} + e^{-k\mu_{B_1}(y_o)}}$$

- Similarly, we can find w_2 for Rule 2.

- The control output of Rule 1:

$$z_1 = \mu_{c_1}^{-1}(w_1)$$

  and for Rule 2:

$$z_2 = \mu_{c_2}^{-1}(w_2)$$

- Using a weighted averaging approach, z_1 and z_2 are combined to produce the combined result z^*.

CONCLUSIONS

- With the GARIC architecture, we have proposed a new way of designing and tuning a fuzzy logic controller.
- The process control knowledge can now be modeled using approximate linguistic terms and later refined through the process of learning from experience.
- GARIC combines the qualitative knowledge of human experts in terms of symbolic rules and learning strength of the artificial neural networks.
- The GARIC architecture is general enough for use in other rule-based systems which perform fuzzy logic inference.

FUZZY RULES FOR GUIDING REINFORCEMENT LEARNING

Hamid R. Berenji
NASA Ames Research Center

Pratap S. Khedhar
University of California at Berkeley

CONCLUSION

Knowledge + performance-driven learning

for **both** action evaluation and selection

- Easy to build in *a priori* knowledge
- Easy to tune approximate knowledge
- Generalizable to arbitrary characterization of state space
- Hierarchical techniques of knowledge structuring may be useful
- Integrated/uniform structure and algorithms for both ASN and AEN

- The current set of N neurons collectively vote to determine the net's prediction.
- The learning rule says that the update is proportional to the influence of neuron j, the (signed) error generated, and the corresponding input.

CLUSTERING IN PRODUCT SPACE
(H. R. Berenji, P. S. Khedhar)

- Generate initial set of fuzzy rules from raw data.
- Using radial basis functions and an extended clustering approach.
- $f : \Re^n \rightarrow \Re^m$ where *n* and *m* are the input and output dimensions.
- A neuron represents a fuzzy rule *r* :

If s_1 is ξ_{r1} and s_2 is ξ_{r2} .. **then** y_{r1} is $c_{r10} + c_{r11}s_1 + .. + c_{r1n}s_n$ and y_{r2} is $c_{r20} + c_{r21}s_1 + .. + c_{r2n}s_n$

SPACE SHUTTLE ATTITUDE CONTROL

- The controller is expected to perform four basic operations:

  1. Attitude hold or maintaining the desired attitude within a small region of the desired value, typically known as a deadband.

  2. Attitude maneuver or going from one attitude to another.

  3. Rate hold or maintaining a desired rate on a given axis.

  4. Rate maneuver or going from one rate value to another rate value for a given axis.

- Its on-orbit controller or Digital AutoPilot is based on modern digital control theory and is a highly optimized controller.

- It uses two types of thrusters (two levels of jet thrusts), known as primary and vernier, and operates with two different sets of deadband values.

- It can perform rate maneuvers in pulse as well as discrete modes. Typical perturbations acting on the system include gravity gradient, aerodynamic torques, and translational burns.

FUZZY CONTROL RULES (JET FIRING COMMANDS) FOR ATTITUDE CONTROL

Angle Error

Rate Error	NB	NM	NS	ZO	PS	PM	PB
NB	PM	PM	PS	PM			
NM	PM	PM	PS	PM			
NS	PS	PS	PS	PS			
ZO	PS	PS	PS	ZO	NS	NS	NS
PS				NS	NS	NS	NS
PM				NM	NS	NM	NM
PB				NM	NS	NM	NM

- The structure of GARIC for the Space Shuttle consists of the following:

  - In ASN, there are two inputs, error and error rate, each using seven labels, 31 rules with conclusions using five labels, and a single output. Hence, the network has 2, 14, 31, 5, 1 neurons in its five layers.

  - In AEN, there are two inputs, error and error rates, and a biased unit, 31 hidden layer nodes, and a single output. Hence, the network has 3, 31, 1 neurons in its three layers.

- For each rule, seven labels (NB, NM, NS, ZE, PS, PM, PB) are used for angle error and angle error rate and five labels (NM, NS, ZE, PS, PM) are used for jet firing commands.

- In a learning experiment, a failure occurs when the value of a state variable goes beyond the allowed deadband.

- Every time a failure occurs in a GARIC execution, the control is shifted to a supervisory control routine to bring the state of the system back to within the deadband.

RESULTS

- With a small number of trials (less than ten), GARIC learns to hold the error within a ± .4 deadband.
- A similar experiment was performed to train this newer controller to hold the error within a ± .3 deadband.
- This time five trials were needed for GARIC to learn this new task.
- Although an adaptive behavior has been added, the fuel consumption for the 100,000 time step simulation runs was about 222 lb.

CONCLUSIONS

- Fuzzy Logic and Neural Networks can be combined and used for intelligent control.
- Neural networks can provide adaptive performance for fuzzy logic controllers.
- Fuzzy logic can provide knowledge representation capabilities for neural networks.

TETHER CONTROL USING GARIC

- Tether control consists of three main operations:
  - Deployment Phase:
    * Conducting tether is used to deploy a payload.
    * e.g., Italian satellite weighing 525 kg, deployed to a distance of 20 km.
  - On-station Phase:
    * Acquire scientific and operational data.
  - Retrieval Phase:
    * Retrieve up to 2.4 km.
    * Dampen oscillations.
    * Completely retrieve the payload for reuse.

COMPLEXITIES

- In vacuum, zero-g, and under gravitational and magnetic forces.
- Time varying dynamics of a long, flexible, variable length tether, the orbiter and the payload.
- Unlike tether length and tether tension, oscillation cannot be directly measured or controlled.

Longitudinal and Librational Oscillation in a Tethered payload system

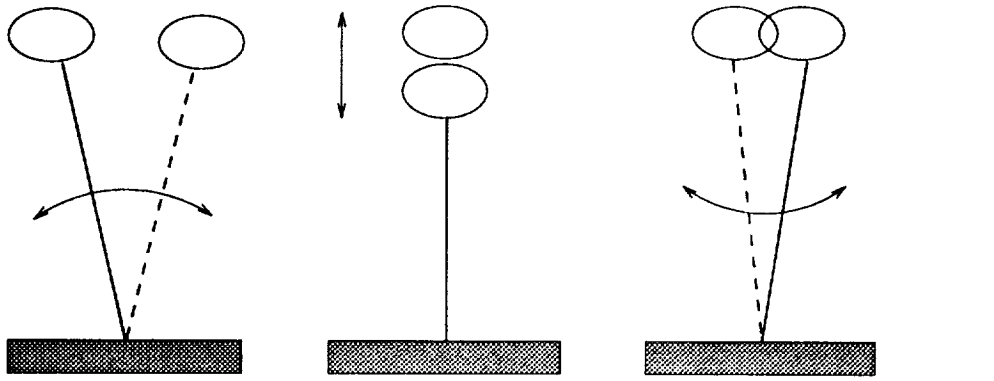


Fig 1 Longitudinal and Librational Oscillations in a Tethered payload system

- ASN
  - Inputs: Length error and length rate error, each uses seven labels.
  - Output: Change in voltage to be applied to motors. Seven labels for conclusion.
  - Number of rules: 49
  - Network architecture: Network has 2, 14, 49, 7, 1 neurons in its five layers.
- AEN
  - Inputs: Length error, length rate error and bias unit.
  - Output: Single output.
  - Number of hidden neurons: 49
  - Network architecture: Network has 3, 49, 1 neurons in its three layers.

APPLYING GARIC FOR TETHER CONTROL

- Failure: Length error greater than two degrees.

- On failure, use a supervisory controller to bring error within the specified limits.

TETHER CONTROL RULES

LENGTH ERROR

LENGTH RATE ERROR

	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NS	NS	ZO
NM	NB	NM	NM	NM	NS	ZO	PS
NS	NB	NM	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PB	PB
PS	NM	NS	ZO	PS	PM	PM	PB
PM	NS	ZO	PS	PM	PM	PM	PB
PB	ZO	PS	PS	PB	PB	PB	PB

Figure 3 : Fuzzy control rules for Tether control.

RESULTS

- Learning experiments were performed during deployment phase (i.e., 16200 secs).

- GARIC learned to maintain the deadband in a small number of trials (less than five).

FQ-LEARNING: A REINFORCEMENT LEARNING APPROACH TO FUZZY DYNAMIC PROGRAMMING

Hamid R. Berenji
Computational Sciences Division
NASA Ames Research Center
Moffett Field, CA

CONCLUSION

- Fuzzy Logic as the base for soft computing
- Fuzzy Logic as a powerful tool for knowledge representation in computational intelligence
- The key for computational intelligence

FUZZY SYSTEMS THAT CAN LEARN!!!!!!

REFERENCES

1. Berenji, H. R., "An Architecture for Designing Fuzzy Controllers Using Neural Networks," *International Journal of Approximate Reasoning*, Vol. 6, No. 2, Feb. 1992, pp. 267-292.

2. Berenji, H. R., "Fuzzy Q-Learning: A New Approach for Fuzzy Dynamic Programming Problems," in *Third IEEE International Conference on Fuzzy Systems*, Orlando, FL, June 1994.

3. Berenji, H. R. and Khedkar, P., "Learning and Tuning Fuzzy Logic Controllers Through Reinforcements," *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, 1992.

4. Berenji, H. R. and Khedkar, P., "Fuzzy Rules for Guiding Reinforcement Learning," in *IPMU*, Palma de Mallorca, Spain, July 1992.

5. Berenji, H. R. and Khedkar, P., "Clustering in Product Space for Fuzzy Inference," in *Second IEEE International Conference on Fuzzy Systems*, San Francisco, CA, March 1993.

6. Berenji, H. R., Malkani, A. and Copeland, C., "Tether Control Using Fuzzy Reinforcement Learning," in *Fourth IEEE International Conference on Fuzzy Systems*, Yokohama, Japan, March 1994.

7. Berenji, H. R., Jani, Y., Lea, R. N., Khedkar, P., Malkani, A. and Hoblit, J., "Space Shuttle Attitude Control by Fuzzy Logic and Reinforcement Learning," in *Second IEEE International Conference on Fuzzy Systems*, San Francisco, CA, March 1993.