Adaptive Neurofuzzy Control of a Robotic Gripper with External Disturbances*

J. A. Domínguez-López, R. M. Crowder, R. I. Damper and C. J. Harris
School of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, UK
e-mail: {jadl00r|rmc|rid|cjh}@ecs.soton.ac.uk

Abstract – *Robotic grippers are commonly required to grasp and manipulate loads under a wide range of conditions, without the load slipping from the end effector, and avoiding damage to the load. We have previously demonstrated the adaptive neurofuzzy control of a simple gripper with a two-input, one-output action. When the robotic gripper is integrated with a multi-degree of freedom robot, the controller could face the curse of dimensionality. To show that satisfactory control was still feasible, we undertook extensive simulations of this gripper mounted on a multi-degree of freedom robot. We also report enhanced control by using the gripper's acceleration as an additional input.*

Keywords: Intelligent control, neurofuzzy systems, robotics, reinforcement learning

1 Introduction

In many applications, robotic grippers must manipulate loads under a wide range of conditions including unexpected external disturbances, which unless compensated for, will result in damage to the load. Here, we consider two types of disturbance: (1) an impact on the load itself, as would be experienced if material is being transferred to the load, for example by pouring; (2) acceleration of the end effector as the robot on which it is mounted moves.

Many techniques can be used to achieve satisfactory gripper control. Some use analytic solutions and cannot be easily implemented in real-time applications, particularly when dynamic adaptation to random external disturbances is an important requirement. Also, the analytic approach cannot be used if variables such as the load's weight and end effector acceleration are unknown. To overcome this problem, fuzzy controllers have been developed, using gripper variables as inputs [8].

We discuss four approaches to gripper control, developed and validated by simulation: (i) Hybrid Neurofuzzy Control (Section 2); (ii) Hybrid Neurofuzzy Control incorporating information about end effector acceleration (Section 3); (iii) Hybrid Neurofuzzy Hierarchical Control, also with information about end effector acceleration (Section 4); (iv) Hybrid Neurofuzzy Hierarchical Control incorporating end effector acceleration information and a scheme for limiting it (Section 5).

In comparing performances between controllers, a simulated gripper was subjected to a range of identical load disturbances. The gripper was a simple two-fingered, single degree of freedom system with slip and force sensors, based on the experimental system developed in our previous work [6, 7]. Full details of the simulation (kinematics equations) can be found in Appendix A of [4]. The simulation was validated against earlier results of the gripper handling a range of weights [5, 6, 7]

The gripper was simulated holding a 0.1 kg object. During simulation the end effector was initially stationary. An external transient force of 10 N was applied to the load 3 seconds into the simulation, and a second force of -10 N was applied at 5 seconds. Between 6 and 10 seconds, the gripper was moved, thereby applying acceleration forces along its z-axis.

2 Hybrid neurofuzzy controller

To accommodate rapid adaptation to environmental changes, the controller uses a hybrid learning approach which combines supervised and reinforcement learning. This combination allows the system to adapt faster, as the use of prior knowledge helps to achieve quicker neurofuzzy learning [5]. The reinforcement learning part of the hybrid algorithm is based on the well-known GARIC architecture [2].

The structure of the hybrid controller is shown in Figure 1, where a (neurofuzzy) Supervised Learning Network (SLN) augments the conventional GARIC structure. It consists of a fuzzy controller (the Action Selection Networks, ASN) that operates as the actor and a neural network (the Action Evaluation Network, AEN) that criticises the actions made by the ASN, in an actor-critic framework. The outputs of these two networks feed into the Stochastic Action Modifier (SAM) which solves stochastically the exploration-exploitation dilemma: Neither exploration of the parameter space to learn new capabilities nor exploitation of what has

*0-7803-8566-7/04/$20.00

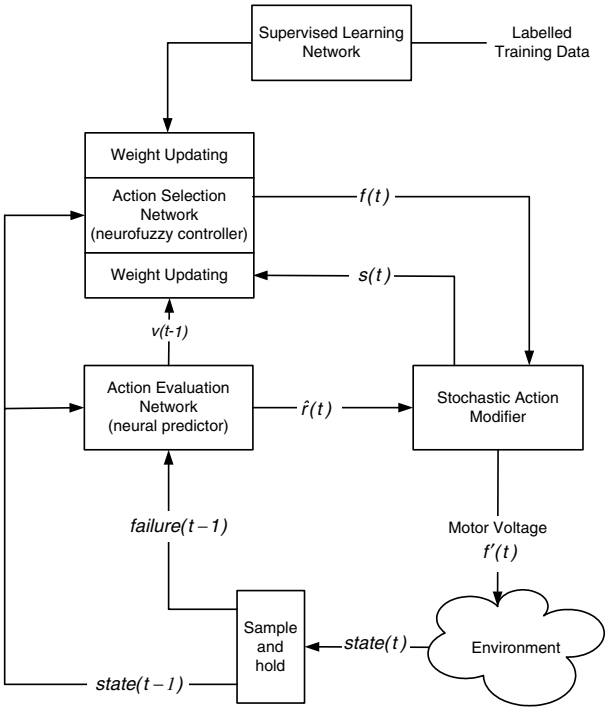


Figure 1: Hybrid system with a Supervised Learning Network (SLN) added to the basic GARIC architecture.

already been learned can be pursued exclusively.

Action Selection Network: This is the fuzzy controller that performs all control operations. It is implemented as a neurofuzzy controller so its parameters can be updated according to the signal received from the Action Evaluation Network. Using a (neuro)fuzzy network as the critic gives transparency to the system.

The ASN output, $f(t)$, determines the 'provisional' gripper motor voltage, and hence applied force. This 'provisional' value is subject to random modification by the Stochastic Action Module as described below. Updating of the ASN modifiable parameters (i.e., the rule confidence vector of connection weights between rule and defuzzification layers) is in the direction which increases the future reward $v(t)$, predicted by the AEN. To effect gradient descent optimisation, the weight update should be proportional to $\frac{\partial v(t)}{\partial w_{ij}(t)} > 0$, giving:

$$\Delta\omega_{ij}(n) = \eta \sum_{k=0}^{n} m^{n-k} \hat{r}(k) s(k) \frac{\partial v(k)}{\partial \omega_{ij}(n)} \qquad (1)$$

where k is an index that runs from the initial time 0 up to the current time n [9, p. 170], $\hat{r}(t)$ is the 'internal' reinforcement signal, $s(t)$ is an output of the SAM, and η and m are the learning and momentum constants which were empirically set to 0.75.

Action Evaluation Network: This is a neural predictor, as shown in Figure 2. The AEN indicates the current state 'goodness', mapping the input state vector to the reward signal from the environment $r(t)$. This mapping uses a Markov

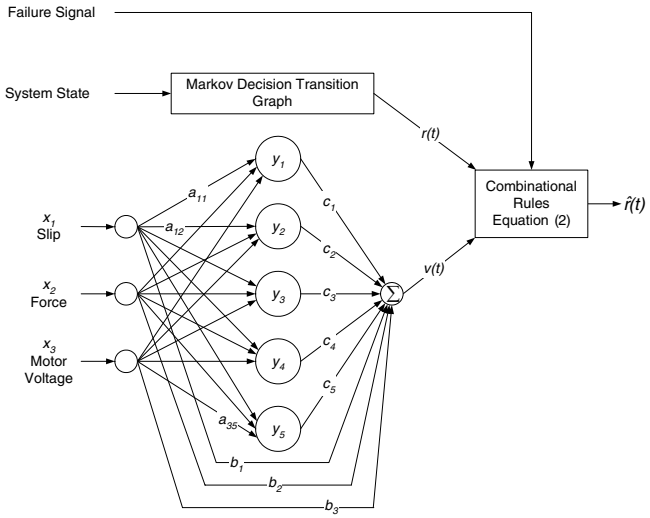


Figure 2: The Action Evaluation Network: see text for description.

decision transition graph [12, p. 66] and produces a scalar score, $\hat{r}(t)$, which predicts future reinforcements, $v(t)$. This is combined with $r(t)$ to generate an 'internal' reinforcement signal, $\hat{r}(t)$:

$$\hat{r}(t) = \begin{cases} 0 & \text{start state} \\ -r(t) - v(t-1) & \text{failure state} \\ -r(t) + \gamma v(t) - v(t-1) & \text{otherwise} \end{cases} \qquad (2)$$

where γ, set here to 0.47, is a discount rate used to control the balance between long-term and short-term consequences of the system's actions [9, p. 606]. Here, 'start' state means the state encountered on power-up, and 'failure' state indicates that the system has to be restarted, after dropping or crushing the load.

This network fine-tunes the rule confidence vector. Its input variables are the normalised measurements of the slip rate, the applied force to the load and the applied motor voltage. The hidden layer activation function is a sigmoidal function. Using a gradient descent algorithm, the formulae for updating the AEN weights are:

$$\begin{aligned} a_{ij}(t) &= a_{ij}(t-1) + \beta_1 \hat{r}(t) y_j(t-1)(1 - y_j(t-1)) \\ &\qquad \operatorname{sgn}(c_j(t-1)) x_i(t-1) \\ b_i(t) &= b_i(t-1) + \beta_2 \hat{r}(t) x_i(t-1) \\ c_j(t) &= c_j(t-1) + \beta_2 \hat{r}(t) y_j(t-1) \end{aligned}$$

for $i = 1, 2, 3$ and $j = 1, \ldots, 5$, where β_1 and β_2 were set empirically to 0.68 and 0.45 respectively. The signum function, sgn( ), takes the value 1 when its argument is positive and the value 0 otherwise.

Stochastic Action Modifier: This gives a stochastic deviation to the output of the ASN, so the system can have a better exploration of the state space and a better generalisation ability [2].

Supervised Learning Network: This is a neurofuzzy controller trained off-line with error back-propagation [11]. We used the labelled training data previously collected [4, 5] to train the SLN *ab initio*. The resulting weights were also used as the initial weights of the ASN. The learning rate and momentum for back-propagation were both 0.5, and training terminated when the mean squared error between actual and desired output voltages was less than $0.2\,V^2$.

During operation, input and output information are collected on-line. As 'good' control actions—defined as maintaining stable grip over 3 seconds—are discovered, the input-output data are concatenated onto the supervised training dataset, and training of the SLN reinitiated, without stopping system execution. Once the SLN (re)training has finished (i.e., stop criterion is satisfied), its new rule weight (confidence) vector is compared with that of the ASN. Specific weights of the ASN can be overwritten by corresponding weights of the SLN if the difference is lower than or equal to 5%. The value of 5% was set empirically, although the system was not especially sensitive to this value.

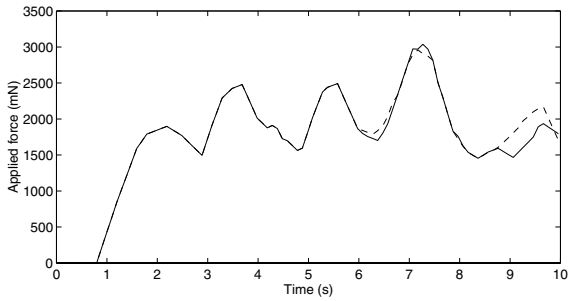
Following twenty minutes of training, the force applied by the basic neurofuzzy controller and the load slip are shown (as solid lines) in Figure 3(a) and 3(b), as a result of the vertical acceleration applied after 6 seconds and shown in Fig. 3(c). While the system manages to maintain a satisfactory grasp, the lack of information on the gripper's acceleration means the controller cannot respond correctly, and hence the load slips considerably.

3 Using acceleration information

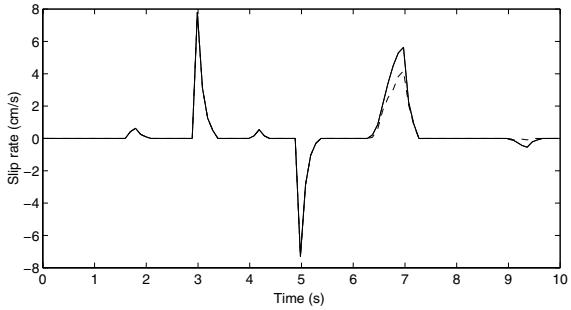
If the controller has knowledge of the gripper's acceleration (by some measurement), it will be able to react to this disturbance and thereby improve performance. A controller meeting this criterion was developed by incorporating end effector acceleration into one single fuzzy machine (Figure 4). The controller was based on the previously discussed architecture, although a revised AEN is required, as in Figure 5.

In Figure 3, the dashed lines show the performance of the system with acceleration information compared with the original system of Section 2 without. The new system is also able to maintain a stable grip despite the disturbances. However, it increases the applied force earlier when the end effector starts to accelerate, reducing the possibility of slippage, as shown in Figure 3(b). Both systems reduce load slippage due to negative acceleration; however, positive acceleration is still able to induce significant slip.

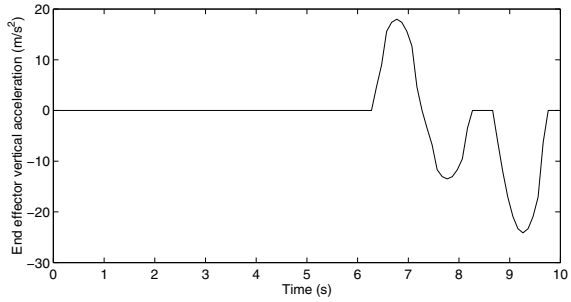
Comparing the performance of the two systems, when there is no end effector acceleration, both systems perform similarly. In the presence of end effector acceleration, however, the system with the acceleration information is able to reduce slippage, particularly in the negative direction. However, this improvement actually comes at the price of having 700 against 140 possible rules in the controller of Section 2. So there is a trade-off between simplicity and better performance. Nevertheless, this application is



(a) Force applied by the end effector



(b) Load slip



(c) End effector acceleration

Figure 3: Simulation results for controllers with (dashed) and without (solid) end effector acceleration

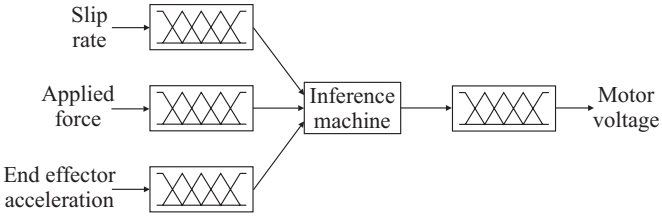


Figure 4: Conventional approach for a neurofuzzy controller with three inputs.

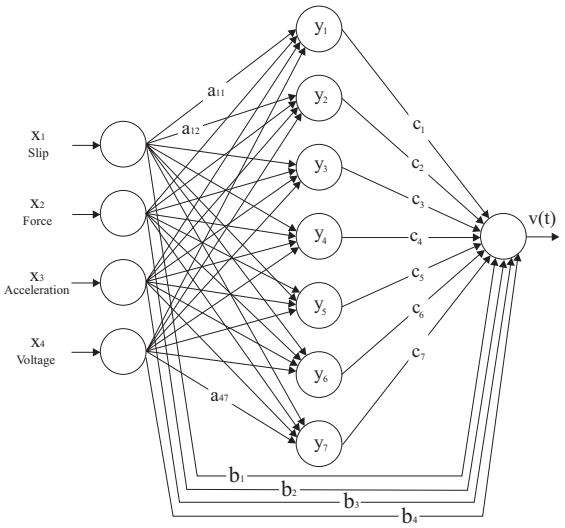


Figure 5: Action Evaluation Network for the three-input neurofuzzy controller.

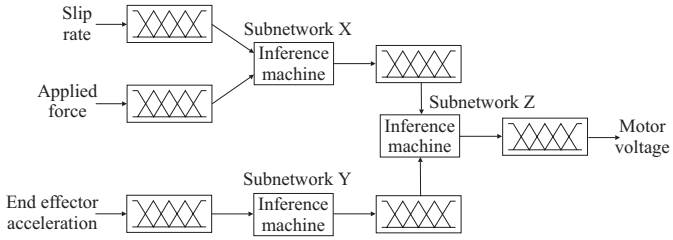


Figure 6: Conventional hierarchical model for the neurofuzzy controller with three inputs.

still considered a low-dimensional problem, which demands modest memory and processing time.

4 Hierarchical controller

Hierarchical control divides a problem into several simpler subproblems, making this an attractive approach to use for parsimonious neurofuzzy modelling [10, 3]. In Section 3, we saw that the addition of just one input to the neurofuzzy controller results in a bigger, more complex rule base—the so-called *curse of dimensionality* [1]. Figure 6 shows the neurofuzzy hierarchical structure commonly used to mitigate the curse of dimensionality. The outputs of the subnetworks X and Y form the inputs of the subnetwork Z. With this approach, the addition of an extra input variable increases linearly the number of rules. However, the training of this network is difficult as the outputs are complex nonlinear functions of the weights [3].

Figure 7 shows the proposed neurofuzzy hierarchical structure for the gripper controller with acceleration information; the product of the outputs of the subnetworks A and B gives the overall output. The design of this structure is based on the previous results, which have shown that the gripper controller has to increase the motor voltage when the acceleration increases.

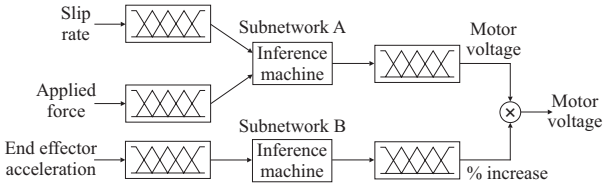


Figure 7: Proposed hierarchical model for the neurofuzzy controller with three inputs.

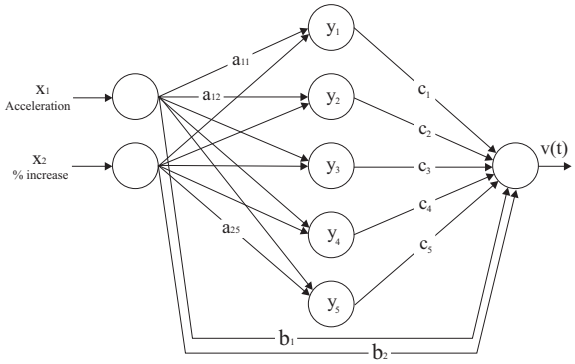


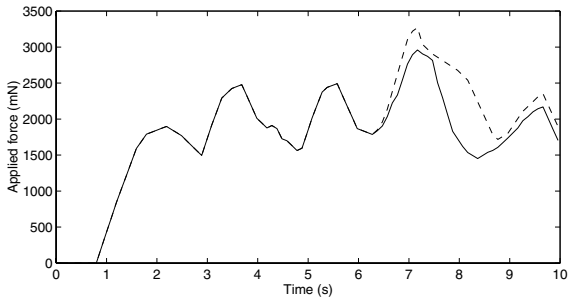
Figure 8: Action Evaluation Network for the neurofuzzy subnetwork B.

In a neurofuzzy hierarchical structure the rule base increases linearly, so the density of the end effector acceleration fuzzy set can be finer. The total possible number of rules of the entire network is equal to 188. Accordingly, there has been a considerable reduction of the rule base in comparison with the approach of Section 3 (cf. 700 rules). The training of subnetworks A and B is identical to the training of the previous neurofuzzy systems, but subnetwork B has a revised AEN, as shown in Figure 8.

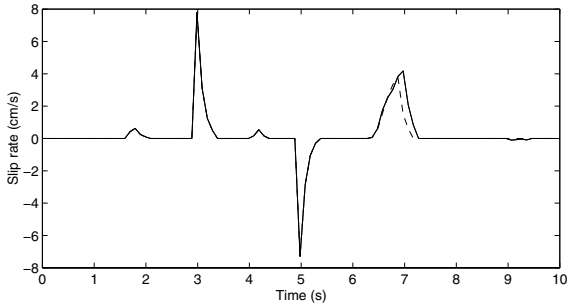
Figure 9 compares the performance of the controller discussed in Section 3 (solid lines) with the controller discussed in this section (dashed lines). Figures 9(a) and 9(b) show the force applied by the basic neurofuzzy controller and the load slip for the two systems in response to the acceleration applied shown in Fig. 9(c) (identical to that in Fig. 3(c)). The controller based on the hierarchical approach is capable of stable gripping despite the induced disturbances. The controller is able to manage the negative accelerations and to reduce the slippage for the positive end effector accelerations. The neurofuzzy hierarchical system not only reduces the number of rules but also gives a slight improvement in system performance.

5 Limiting acceleration

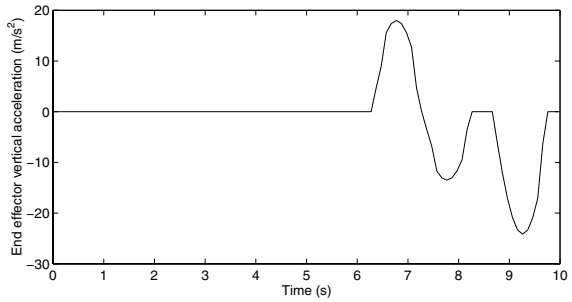
As end effector acceleration can induce slippage, an approach to reducing slip is to restrict the gripper's acceleration. Accordingly, a framework to control the maximum end effector acceleration is proposed, as shown in Figure 10. As with hierarchical systems in general, such a structure facilities the development of intelligent control system for robots, as the problem is subdivided into several simpler



(a) Applied force



(b) Load slip



(c) Vertical acceleration

Figure 9: Simulated results for the controller of Section 3 (solid) compared with the hierarchical controller (dashed).

subproblems.

The maximum allowable acceleration depends on the force applied to the load and the current motor voltage, because these two factors dictate if it is possible for the force applied by the gripper to increase. When the applied force and the motor voltage are high, it is not possible to accommodate an increase in gripper acceleration. As result, there is also a limit in the gripper's acceleration to maintain stable gripping. The neurofuzzy controller (2) (NFC2) shown in Figure 10 determines the maximum gripper acceleration that guarantees stable gripping.

The output of NFC2 is the maximum absolute acceleration ($|a_{\max}|$) which guarantees stable gripping. (The absolute value gives a more transparent network.) The output of the limiter is the maximum permitted end effector acceleration: $a_s = \min(|a_{\max}|, |a_d|)\,\mathrm{sgn}(a_d)$, where a_d is the desired acceleration.

It is recognised that limiting the robot's performance may compromise the robot's path-following capabilities. Hence,

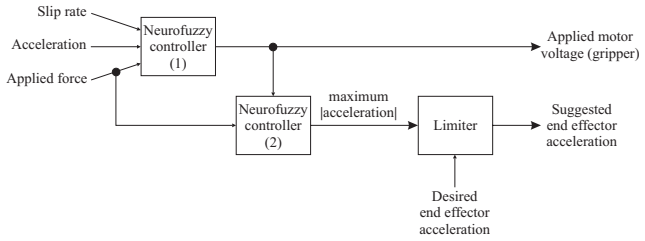


Figure 10: Hierarchical framework to improve the performance of the end effector by limiting its maximum acceleration.

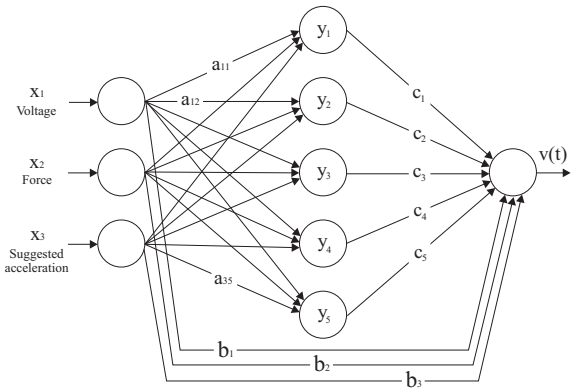


Figure 11: Action evaluation network for Neurofuzzy Controller 2.

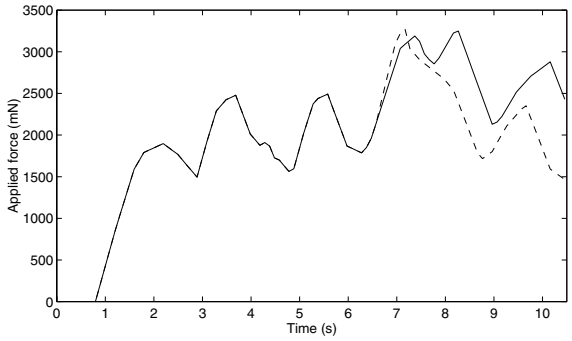
in a practical system, the output of NFC2 may need to be fed back to the robot's controller as opposed to the limiter approach used in the simulation.

Both NFCs in Fig. 10 have identical Markov decision transition graphs. NFC1 is trained first prior to training NFC2. If the two NFCs were trained simultaneously, it would be impossible to determine which controller was responsible for either failure or success. The form used for NFC1 is the neurofuzzy hierarchical controller described in Section 4. Figure 11 shows the AEN of NFC2.

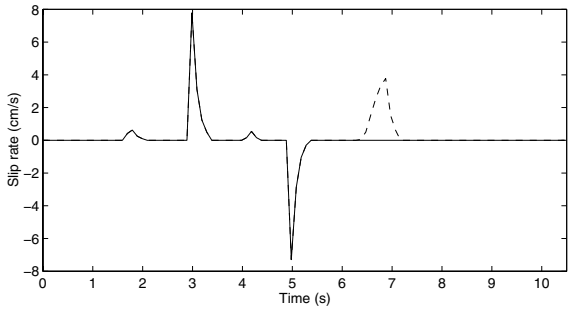
This controller required 45 minutes of training. The solid lines in Figures 12(a) and 12(b) show its performance against the standard disturbances (Fig 12(c), dashed line, which is limited as shown by the solid line). Despite the disturbances, the system is capable of maintaining a stable grip. As illustrated in Figure 12(b), the hierarchical end effector controller was able to eliminate entirely the load slippage due to end effector acceleration. These results should be compared with the dashed lines in Figs. 12(a) and 12(b) which are for the hierarchical controller of Section 4. Clearly, by limiting the acceleration, the system performance can be improved.

6 Conclusions

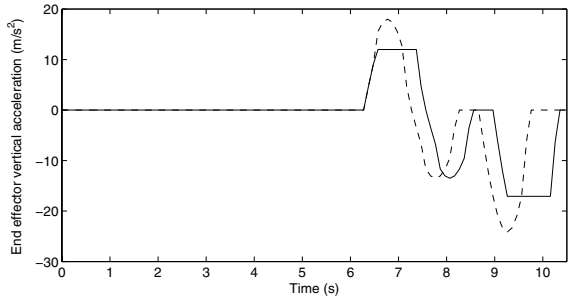
We have described the simulation of a hybrid supervised/reinforcement learning neurofuzzy controller for a simulated end-effector, validated against earlier work with a real system. Results show that when the system has



(a) Applied force



(b) Load slip



(c) Gripper acceleration: desired (dashed), limited (solid)

Figure 12: Simulated results for the hierarchical controller with acceleration limitation (solid) and that of Section 4 (dashed) in which there was no limitation.

knowledge of a disturbance (i.e., gripper acceleration), the controller is able to take action to reduce its impact, and improve the gripping action. Further, when the system is able to modify the disturbance (i.e., limiting the gripper's acceleration), again the gripper performance is improved.

The curse of dimensionality affects not only the system's transparency but also its learning speed. As the number of inputs increases, so the system's complexity increases too, with a corresponding increase in the number of possible rules, neurons and synaptic weights. A disadvantage of supervised learning is that the size of the training dataset has to be large if it is to represent adequately the input-output mapping. These two disadvantages can be reduced using parsimonious adaptive/product models.

References

[1] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, NJ, 1961.

[2] H. Berenji and P. Khedkar. Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Transactions on Neural Networks*, 3(5):724–740, 1992.

[3] M. Brown, K. M. Bossley, D. J. Mills, and C. J Harris. High dimensional neurofuzzy systems: Overcoming the curse of dimensionality. In *Proceedings of IEEE International Conference on Fuzzy Systems*, volume 4, pages 2139–2146, Yokohama, Japan, 1995.

[4] J. A. Domínguez-López. *Intelligent Neurofuzzy Control in Robotic Manipulators*. PhD thesis, University of Southampton, UK, 2004.

[5] J. A. Domínguez-López, R. I. Damper, R. M. Crowder, and C. J. Harris. Adaptive neurofuzzy control of a robotic gripper with on-line machine learning. Accepted for publication in *Robotics and Autonomous Systems*.

[6] J. A. Domínguez-López, R. I. Damper, R. M. Crowder, and C. J. Harris. Hybrid neurofuzzy online learning for optimal grasping. In *Proceedings of IEEE International Conference on Machine Learning and Cybernetics*, volume 2, pages 803–808, Xi'an, China, 2003.

[7] J. A. Domínguez-López, R. I. Damper, R. M. Crowder, and C. J. Harris. Optimal object gripping using fuzzy logic. In *Proceedings of International Conference on Robotics, Vision, Information and Signal Processing (ROVISP 2003)*, pages 367–372, Penang, Malaysia, 2003.

[8] V. N. Dubey, R. M. Crowder, P. H. Chappell, and D. R. Whatley. A robotic end effector for unstructured environments. In *Proceedings of the American Nuclear Society's 7th Topical Meeting on Robotics and Remote Systems*, pages 452–460, Augusta, GA, 1997.

[9] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ, 1999.

[10] G. V. S. Raju and J. Zhou. Adaptive hierarchical fuzzy controller. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(4):973–980, 1993.

[11] D. E. Rumelhart, G. E. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323(9):533–536, 1986.

[12] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2000.