Optimal Object Grasping using Fuzzy Logic

J. A. Domínguez-López, R. I. Damper, R. M. Crowder and C. J. Harris

Department of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ, UK
Emails: {jadl00r|rid|rmc|cjh}@ecs.soton.ac.uk

Abstract

Robotic end effectors are commonly required to hold and manipulate various objects under a wide range of conditions. To achieve a satisfactory grip, optimal force control is required to avoid the risk of the object slipping out of the end effector and to avoid any possible damage to the object. In this paper, we investigate the use of fuzzy logic controllers to achieve optimal grasping. Starting with a set of manually-designed fuzzy rules from previous work, these were analysed and a shortcoming identified and corrected. Backpropagation learning was then employed to train a neurofuzzy controller with and without *a priori* knowledge. In the former case, the *a priori* knowledge which defined the start point for learning was obtained from the manually-designed rules. In the latter case, training started from different initial random weight settings. The learned solutions were very similar although training was faster with *a priori* knowledge. The learned neurofuzzy controller had better performance than the manually-designed fuzzy controller, particularly in respect of its faster control action.

Keywords:

Robotics, grasping, fuzzy control, neurofuzzy learning.

Introduction

A key goal in advanced robotic systems is the provision of an end effector capable of considerable gripping dexterity, within an unstructured environment. This problem is of such complexity that we cannot hope to program a system to cope with all the situations that it might encounter in the (unstructured) environment. The problem is further complicated by the variability of the mechanical and electrical parts, introduced during production, together with those resulting from age and wear. Hence, an attractive approach is to achieve dexterity using control strategies based on learning through interaction between the end effector and its environment. We propose the use of neurofuzzy learning [1, 2] in this application. We first describe the implementation of two sets of manually-derived fuzzy rules, one of which was previously published in the literature [3] with the other being a modification to the published rules. Both are shown to be effective, but the latter correct a shortcoming in the earlier rules. These then form a baseline for assessment of the learned rules discussed later in this paper.

End Effector

Robotic end effectors are commonly used to restrain and manipulate a variety of objects under a wide range of conditions. To achieve a satisfactory grip, optimal force control is required to avoid the risk of the object slipping out of the end effector as well as any possible damage to the object. The problem of defining the required grip force is crucial and can be posed as an optimisation problem [4]. Many techniques have been proposed and used to solve this problem [5]. Some are analytic solutions and cannot easily be implemented in real-time applications, particularly when dynamic adaptation to external disturbances is an important requirement. Also, the analytic approach cannot be used if variables such as weight and end effector acceleration are unknown.

To overcome this situation, other approaches using fuzzy controllers have been developed using a range of sensors to measure the physical variables within the system. This is the approach adopted here, not least because fuzzy control can be used in conjunction with powerful automatic learning methods [1, 2].

The experimentation reported in this paper has been undertaken on a simple, low-cost, two-finger end effector, depicted in Figure 1. The end effector was fitted with a slip, position and force sensor as follows:

- The slip sensor is located at the left finger and is based on a rolling contact principle. As the object slips, it moves the roller together with the slotted code wheel. The slip speed is obtained measuring the code wheel slits frequency through. The slip sensor has an operational range of 0 to 80 $\mathrm{mm.s^{-1}}$ and sensitivity of 0.5 $\mathrm{mm.s^{-1}}$.

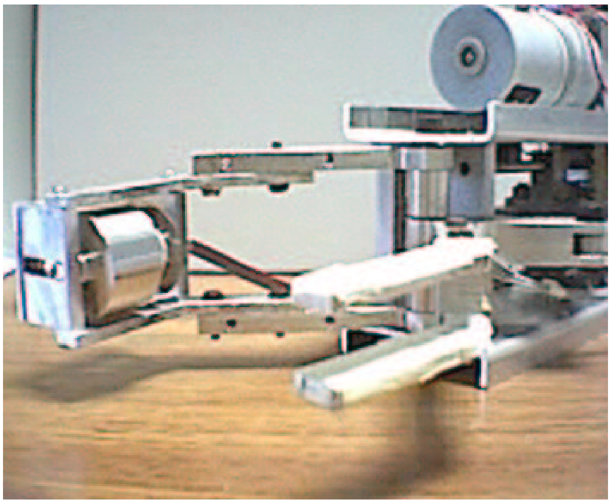


Figure 1: *End effector used for the experiments.*

- The position of the fingers is determined using a linear potentiometer mounted on the end effector's lead screw actuator.
- The applied force is measured using strain gauges on the end effector structure. The force sensors have a range of 0 to 2.5 N, with a sensitivity of 1.9 mN.

Neurofuzzy Controllers

A fuzzy controller has distinct advantages over more conventional approaches when the process is highly non-linear, very complex and/or a suitable mathematical model is not easily available. Furthermore, it has the advantage that its parameters can be easily updated if the plant operating points change [6]. Often, even a non-specialist user can maintain the rule-base for the controller; understanding the rules is made easier because they utilise linguistic rather than numerical variables.

Design and implementation of a fuzzy controller is conventionally divided into three basic steps: fuzzification, the application of the knowledge base and defuzzification. Fuzzification transforms an objective term into a fuzzy concept. The knowledge base processes the rules producing the list of fuzzy outputs using the current fuzzy input values. This has two components: a rule-base and a database. The rule-base contains the control-action rules expressed as IF-THEN (cause-action) statements, while the database stores information about the number of inputs and outputs, their operational range, and the definition of the memberships. Defuzzification translates the fuzzy concept back into a crisp system output, which is necessary because effector actions in the real world require numerical control values. There are various methods to calculate the crisp output value. Centre of sums (CoS) and centre of gravity (CoG) are the most often used defuzzification techniques. The CoS approach has some advantages over CoG for this work, and so we use it here. In particular, CoS is faster because it is simpler to program [7].

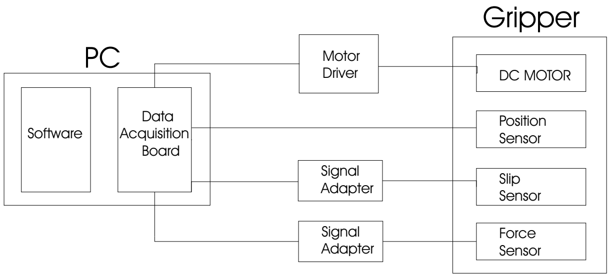


Figure 2: *Block diagram of the experimental system.*

Controller Implementation

The block diagram of the controller is shown in Figure 2. It was implement using a personal computer (Pentium 75 MHz, 64 MB RAM) fitted with a high-speed data acquisition card (Eagle Technology PC30GAS4). The card has a number of analogue and digital I/O channels that interfaces with the end effector's sensors and motor drive. Hardware signal processing was used to condition the outputs of the slip and force sensors prior to the data acquisition card.

The implemented fuzzy logic controller has two inputs, the object's slip and the applied force, and one output, the required motor torque. The operational range of each of these variables was determined experimentally, allowing the database to be manually created initially. The design was carried out after analysing the grasping problem. When the object is not slipping, the end effector should not squeeze it any more. The end effector has to apply additional force proportional to the degree of object slip but taking into account the absolute force already applied. When the applied force is high, the end effector has to grasp carefully to avoid crushing the object: It has to squeeze inversely proportional to the applied force. The database is then designed following these two requirements. Triangular membership functions were chosen for all signals because of their simplicity and economy [7].

Manually-Designed Rules

Prior to this work a rule-base for the control of an end effector was developed by Dubey [3, 5]. Using the database values, the rule-base was designed to ensure stable grasping. The linguistic variables used for the term sets are simply value magnitude components: Almost Nil (*AN*), Small (*S*), Medium (*M*) and Large (*L*) for the fuzzy set slip while for the fuzzy set applied force they are *S*, *M* and *L*. The applied motor voltage term has the set

Table 1: *The original set of manually-designed fuzzy rules. Note that the voltage is proportional to torque.*

Voltage		Fingertip force		
		S	*M*	*L*
Slip	*AN*	M	S	VS
	S	L	M	S
	M	VL	L	M
	L	VVL	VL	L

Table 2: *The revised set of manually-designed fuzzy rules.*

Voltage		Fingertip force			
		Z	*S*	*M*	*L*
Slip	*Z*	S	VS	Z	NVS
	AN	M	S	VS	Z
	S	L	M	S	VS
	M	VL	L	M	S
	L	VVL	VL	L	M

members Very Small (*VS*), *S*, *M*, *L*, Very Large (*VL*) and Very Very Large (*VVL*). The latter has more set members so as to have a smoother output. Motor torque is directly proportional to motor voltage at standstill. The developed rules are given in Table 1.

The original set of rules was defined using the Fuzzy Logic Toolbox of MATLAB—a graphical interface for modelling and visualising the controller. Once the output had been tuned to a satisfactory model and the controller had been modelled, the model file was saved for later use by a fuzzy inference program. Analysing the original rule-base, when the applied force is constant, the applied voltage increases with the slip rate. This response is desired to ensure the object does not fall. On the other hand, with the same slip rate, the applied voltage increases as the applied force grows. This situation will result in an increase of the applied force so that the object will start to be crushed. A different, modified rule-base was used to overcome this problem. (This is simpler and faster than employing another algorithm to prevent crushing as was done in [5].) The modified rules are shown in Table 2.

To demonstrate that the modified rule-base improves the performance of the system, the original and modified rules were compared under the same conditions. It was found that the modified rule-base allowed faster grasping, and even more important, there was no need for an extra algorithm to relax the fingers to avoid crushing the object.

Since the long-term goal of our work is to achieve control by automatic learning through interaction between the end effector and its environment, the manually-derived solutions will act as a baseline for the assessment of the learned solution(s).

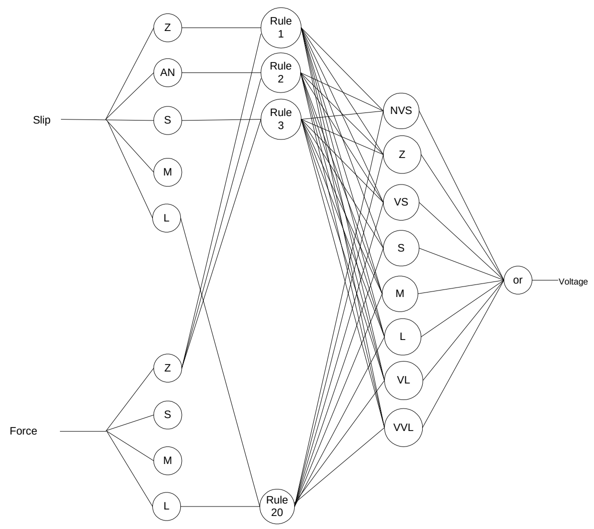


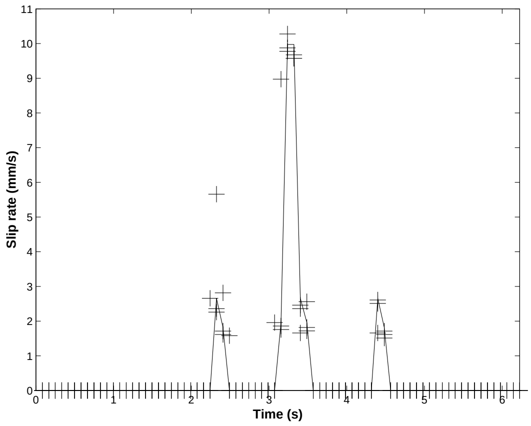
Figure 3: *The neurofuzzy network used to control the end effector.*

Learning a Neurofuzzy Solution

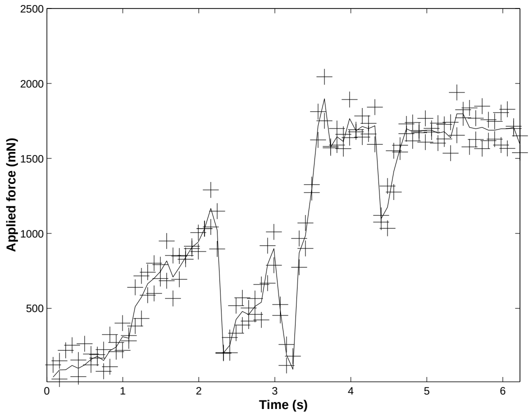
Figure 3 shows the neurofuzzy network used to control the end effector. The network structure is defined by the modified rule set of Table 2, i.e., there are 5 fuzzy membership values for slip and 4 for force, giving 20 hidden units each corresponding to a rule. The output is applied voltage (proportional to torque).

In the work described in this paper, the grasped object was an empty soup can. To obtain the training data for the neurofuzzy network, the end effector was controlled manually by the first author to accomplish the desired performance. The output value and the value of the inputs were stored as the desired output and the corresponding input values. Figures 4(a), 4(b) and 4(c) show the obtained performance for both slip rate and applied force inputs and the respective output (applied motor voltage). In each graphs, the full line represents the mode of the collected data while, for illustration of the variability, the crosses indicate a selection of the practical data obtained. A non-linear normalisation to the data is always recommended, especially when the ranges of the individual variables are very separate, and was therefore used here. If the data are not scaled, learning may be compromised [8, p177].

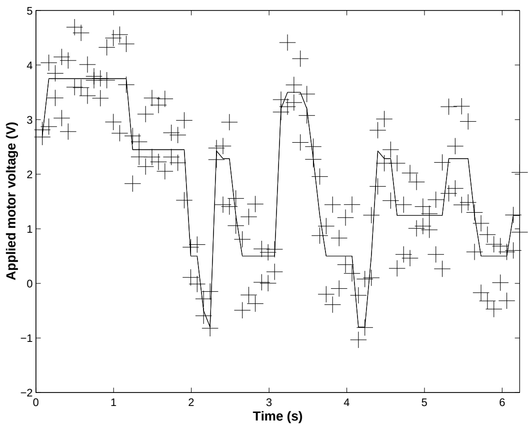
Network training used the well-known back-propagation algorithm [9]. Training was compared with and without *a priori* knowledge. (The possibility to incorporate such knowledge is one of the chief attractions of neurofuzzy learning.) Table 2 was used as the source of *a priori* knowledge. Rules entered in this table were given an initial confidence (weight) of 1; the rules that are not listed have a rule confidence equal to zero. With



(a) Input slip



(b) Input force



(c) End effector motor terminal voltage

Figure 4: *Data produced for training the neurofuzzy network, acquired when the end effector was manually controlled.*

a priori knowledge, the learning rate and momentum rate were 0.5. Without *a priori* knowledge, the learning rate was 0.21. In the latter case, training was repeated 15 times starting from different initial weight values and averages were taken. The stopping condition for learning was that the average squared error per epoch reduced to less than 0.2%.

The rule-base and rule confidences achieved after training are presented in Table 3. Each rule output has two rule confidences, the first is the value obtained when *a priori* knowledge was used to initialise the weight vector while the latter is without any knowledge. It can be seen that the learning procedure arrives at almost the same values in the two cases. Comparing Tables 2 and 3, we see some similarities but also some striking differences (e.g., for the case when slip and force are both Z). When *a priori* knowledge is used, the learning has merely to fine-tune to provided rule set and, consequently, learning is faster. Thus, training took 25 seconds without *a priori* knowledge, and 10 seconds with.

Performance of the Learned Solution

The performance of the learned neurofuzzy controller (using the rules learned without *a priori* knowledge) is illustrated in Figure 5. After the object has been grasped, slip is induced manually by pulling on it. Figure 5(a) shows five such interventions of various degrees of severity. As seen in Fig. 5(b), the first three interventions are of sufficient magnitude to cause the force applied to the object to drop suddenly, after which the neurofuzzy controller increases the applied force to regain control.

Although not shown explicitly here, it was noticeable that the system using the learned neurofuzzy controller was able to grasp the object faster than the system with the manually-designed fuzzy controller. It was also found that, immediately following the intervention, the slip rate was lower and the slip phase lasted less time with the neurofuzzy controller.

Discussion and Conclusions

Using a low-cost experimental end effector, we have demonstrated the capability of manually-derived fuzzy rules to implement satisfactory grasping performance, as well as overcoming an imperfection of an earlier set of rules. After testing with both the original and modified rule-bases, it was observed that the system with the modified fuzzy controller had a faster response, required less memory, and was much easier to update. The system was able to realise satisfactory grasping of an unknown object in an unstructured and unknown environment. It needs no prior information about the object or environment, which could become misleading in a rapidly changing situation. Information (e.g., weight, shape, coefficient of friction, structure, etc.) concerning the object to be

Table 3: *Rule-base and rule confidences found from training.*

		Fingertip force			
Voltage		*Z*	*S*	*M*	*L*
Slip	*Z*	VL (0.1, 0.1) VVL (0.9, 0.9)	S (0.05, 0.0) M (0.1, 0.1) L (0.8, 0.9) VL (0.05, 0.0)	NVS (0.2, 0.15) Z (0.8, 0.85)	NVS (0.9, 0.9) Z (0.1, 0.1)
	AN	L (0.2, 0.15) VL (0.7, 0.65) VVL (0.1, 0.2)	S (0.3, 0.2) M (0.6, 0.6) L (0.1, 0.2)	Z (0.1, 0.95) VS (0.9, 0.95)	Z (0.75, 0.8) VS (0.25, 0.2)
	S	M (0.2, 0.25) L (0.8, 0.75)	M (0.25, 0.3) L (0.65, 0.7) VL (0.1, 0.0)	S (0.4, 0.5) M (0.6, 0.5)	VS (0.4, 0.6) S (0.6, 0.4)
	M	L (0.08, 0.15) VL (0.9, 0.85) VVL (0.02, 0.0)	L (0.2, 0.2) VL (0.8, 0.8)	M (0.3, 0.3) L (0.7, 0.7)	S (0.3, 0.5) M (0.7, 0.5)
	L	VL (0.1, 0.15) VVL (0.9, 0.85)	L (0.1, 0.2) VL (0.9, 0.8)	L (0.8, 0.82) VL (0.2, 0.18)	M (0.9, 0.7) L (0.1, 0.3)

gripped and the end effector itself were never given to the control system.

Contrasting the performances of the manually-designed fuzzy and learned neurofuzzy controllers, it can be concluded that the latter technique gives system performance closer to that desired, especially in terms of speed of response. Other potential advantages are: more robust operation in unstructured environments where novel, unexpected situations routinely arise; adaptation to drift of end effector parameters and characteristics over its lifetime; faster implementation without requiring prior knowledge of operational conditions (database and rule-base). In particular, expanding on the last point, a neurofuzzy controller, which is a self-tuning mechanism that acquires knowledge by means of learning, avoids problems associated with manual definition of the rules for a fuzzy controller. Furthermore, it should have performance close to optimal because the rule-base and the rule confidences have been acquired through interaction with its environment, ensuring that it is well adapted to this environment. The final fuzzy model can still be interpreted by a set of linguistic rules which gives transparency to the approach.

Comparing the performance of the neurofuzzy system with and without *a priori* knowledge, it was found that the only significant difference was the required training time. The latter needed just a little more training (requiring $\sim$15 s) to achieve similar performance. So, it can be said that *a priori* knowledge helps but it is not indispensable. Information concerning the object to be gripped and the end effector itself were never given to the control system. The system was thus able to realise satisfactory grasping of an unknown object with minimum required fingertip force even in an unstructured and unknown environment.

Training of the neurofuzzy system used backpropagation—a gradient descent procedure which is well known to be susceptible to problems of local minima [10]. Hence, we do not know that an optimal solution has been obtained, although the fact that we obtained similar solutions with initial weights set by *a priori* knowledge and set randomly suggests that we may not be too far away from the optimum. Nonetheless, future work will use an optimisation technique less susceptible to local minima, such as the dynamic tunnelling algorithm [11].

Acknowledgements

J. A. Domínguez-López wishes to acknowledge the support of the Mexican National Council for Science and Technology (CONACyT) for its sponsorship.

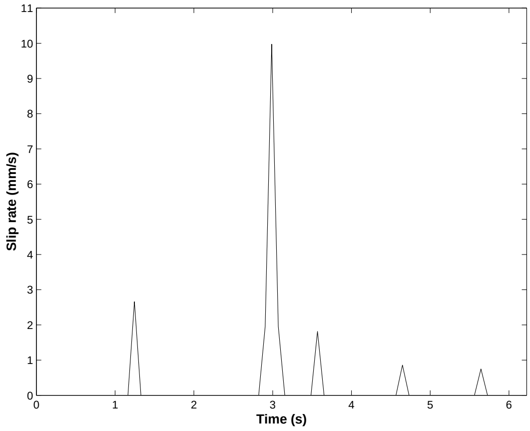
References

[1] M. Brown and C. J. Harris. *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall, Hemel Hempstead, UK, 1994.

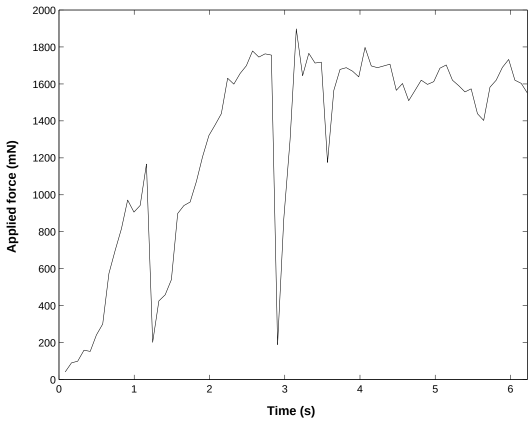
[2] C. Harris, X. Hong, and Q. Gan. *Adaptive Modelling, Estimation and Fusion from Data: A Neurofuzzy Approach*. Springer-Verlag, Berlin and Heidelberg, Germany, 2002.

[3] V. N. Dubey, R. M. Crowder, and P. H. Chappell. Optimal object grasp using tactile sensors and fuzzy logic. *Robotica*, 17(6):685–693, 1999.

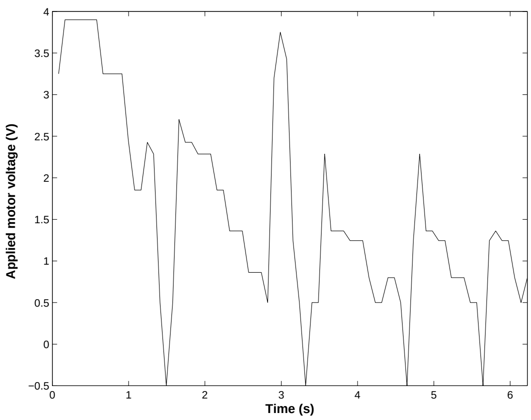
[4] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *IEEE International Conference on Robotics and Automation*, pages 348–353, San Francisco, CA, 2000.



(a) Object slip



(b) Applied force



(c) End effector motor terminal voltage

Figure 5: *Performance of the end effector using the trained neurofuzzy controller.*

[5] V. N. Dubey. *Sensing and Control Within a Robotic End Effector*. PhD thesis, Department of Electrical Engineering, University of Southampton, Southampton, UK, 1997.

[6] J. Jantzen. Design of fuzzy controllers. Technical Report 98-E 864 (design), Technical University of Denmark, Department of Automation, Lyngby, Denmark, 1998.

[7] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer-Verlag, London, UK, 1993.

[8] W. Pedrycz. *Computational Intelligence: An Introduction*. CRC Press, Boca Raton, FL, 1998.

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

[10] M. Gori and A. Tesi. On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):76–86, 1992.

[11] Y. Yao. Dynamic tunneling algorithm for global optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1222–1230, 1989.