Proc. 2002 Australasian Conference on Robotics and Automation
Auckland, 27-29 November 2002

Online Learning of Autonomous Helicopter Control

Gregg Buskey[a,b], Jonathan Roberts[a], Gordon Wyeth[b]

[a] CSIRO Manufacturing and Infrastructure Technology
P.O. Box 883
KENMORE 4069, Queensland, Australia
[b] Dept. Computer Science and Electrical Engineering
University of Queensland
ST LUCIA, Queensland, Australia
Email: Gregg.Buskey@csiro.au

Abstract

This paper details the development of an online adaptive control system, designed to learn from the actions of an instructing pilot. Three learning architectures, single layer neural networks (SLNN), multi-layer neural networks (MLNN), and fuzzy associative memories (FAM) are considerd. Each method has been tested in simulation. While the SLNN and MLNN provided adequate control under some simulation conditions, the addition of pilot noise and pilot variation during simulation training caused these methods to fail. FAMs alone were found to be suitable under all simulation conditions. The performance of an FAM velocity control being used by a higher level way-point controller in simulation is presented. As a first step toward performing real helicopter tests, an FAM attitude controller has also been developed in simulation, and its tracking performance presented.

1 Introduction

This paper describes the development of an online learning control system for a scaled size 60 helicopter. Pilot signals during flight are used as training examples for the learning system. Several control architectures have been tested, including single layer neural networks (SLNN), multilayer neural networks (MLNN), and fuzzy associative memories (FAM).

A wide range of classical and non-conventional control methodologies have been applied to scaled size helicopter control, including linear PD/PID [18, 1, 21], sliding mode control [7], input/output linearization [9, 16], gain scheduling [19], fuzzy control [20, 14, 11, 15, 6], neural networks [5, 8, 16, 11, 10], markovian policy search [2], and combinations thereof. It should be emphasised that only a few of the non-conventional studies have moved beyond simulations.

While many of the neuro/fuzzy approaches incorporate some form of limited authority, online adaption, the majority of control authority is learnt offline. This offline controller learning has been performed using actor/critic type reinformcement learning [8], policy search methods based on MDP models built from flight data [2], and back propagation of errors through neural forward models [10]. Only two methods have used pilot control signals to directly train a feedback controller [14, 5]. While the FAMs in [14] showed promise in simulation, the controllers failed to stabilise the real helicopter. The MLNN in [5], on the other hand, managed to stabilise the real helicopter for brief periods, however later investigations showed that the system had many hidden, potential instabilities [3, 4].

Figure 1: JR Ergo 60 Helicopter Testbed

2 Approach Outline

Our previous work [5] used a MLNN, trained by a pilot maintaining stable hover for approximately 6 minutes, to learn attitude stabilisation. While ~18 seconds of stable autonomous hover was achieved, it was later found that some potentially destabilising associations had been developed. This is thought to be due to the lack of experience and excitation during training [3].

The method described in this paper uses specific desatbilisation/restabilisation sequences to provide the controller with a wider experience envelope during learning. Both SISO attitude controllers and MISO velocity controllers have been developed (Figure 2). Rate feedback was not included, as it was shown in [13] that there is little margin for rate feedback to improve performance. Controller

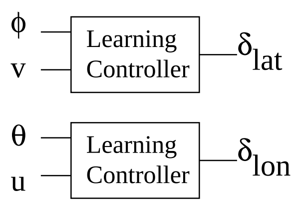

Figure 2: Velocity control structures

training is performed during the restabilisation phase. In the case of the attitude controller, this involves taking the aircraft from a rolled/pitched attitude to level. In the case of the velocity controller, this involves bringing the aircraft to rest from a non-zero velocity.

The suitability of each of the methods described in Section 1 has been tested in simulation. As will be detailed in Section 4, only the FAMs have been found to be suitable for future online learning experiments on the real platform.

3 Platform Description

This section describes both the real helicopter, and the simulation environment used to conduct controller testing. State variables include linear velocities $[u\,v\,w]$, angular velocities $[p\,q\,r]$, attitudes $[\phi\,\theta\,\psi]$ and position $[X\,Y\,Z]$. Control signals include lateral cyclic δ_{lat}, longitudinal cyclic δ_{lon}, tail rotor collective δ_{tail}, and main rotor collecitve δ_{col}.

3.1 JR Ergo Size 60 Helicopter

The JR Ergo size 60 helicopter (Figure 1) has a main rotor diameter of ~1.5m, stands 0.6m tail with modified landing gear and avionics housing [17], and has a lift capability of approximately 10kg. Sensors include stereo vision, and an in-house designed IMU comprised of a 3 axis magnatometer, three accelerometers, and three gyros [17]. The main control/data-logging computer is a Celeron III 800Mhz. Limited in-flight data communication (9600 baud) is maintained via a radio serial link [17].

3.2 Simulator

The simulator is based on a second order scaled size helicopter model (linearised within the hover envelope) developed at CMU [12]. The model is a 14 state system with second order main rotor dynamics. State feedback is provided by several independant threads mimicking the actual on-board sensors (i.e. velocities from vision, atttitude from IMU). Each has its own additive noise, sample rate, and delay. Pilot control is simulated using a variable feedback controller. Noise is added to the pilot signals, equivalent to inducing $6°s^{-1}$ roll and pitch rates. Parameters of the controller are also randomly varied by ±50% to simulate pilot variation during flight. In so doing we have tried to include as much potential variability during training as possible.

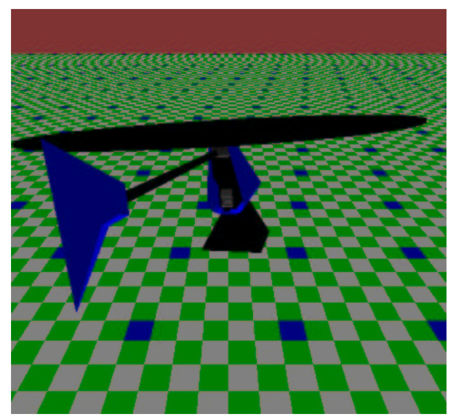

Figure 3: Simulator

4 Control Structure

This section describes the three architectures tested for online learning. All architectures use seperate controllers for lateral and longitudinal movement. State inputs are described generally as x and control demands are described generally as δ. The error between learning controller outputs o and desired (pilot) control signals δ is defined as e.

4.1 Single Layer NN

Given the offline learning success of the SLNN networks used in [2], the first architecture tested for online learning of velocity control comprised two, SLNN networks (Figure 4); simple error correction learning was used to adjust the weights:

$$\Delta w_i(t) = \eta x_i(t)e(t) + \alpha\Delta w(t-1)$$

Training without pilot noise or variation produced working controllers. However, the inclusion of pilot noise, and more so pilot variation, resulted on neural weights oscillation wildly. Fine tuning learning paramaters $[\eta, \alpha]$ somewhat improved the situation. However, the need for parameter tuning begs the question, "why not simply tune traditional controller parameters?" Consequently, the SLNN architecture was deemed unacceptable for online learning.

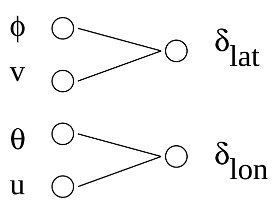

Figure 4: SLNN Velocity Controller

4.2 Multi-layer NN

The MLNN control structure was the same as that used for the SLNN, except of course for the addition of a hidden layer; these networks were trained using the well known back propagation algorithm. While the level of weight oscillation during training was reduced (when compared to the SLNN), oscillations by weights associated with velocities in particular, persisted. Consequently, MLNN were also considered unacceptable for online learning.

4.3 Fuzzy Associative Memories

Linear velocities and attitudes were partitioned into 5 overlapping fuzzy sets (Figure 5). Two seperate FAMs were trained to control δ_{lat} and δ_{lon} (Figure 6). If $\{\mu_{\phi 1}, \ldots, \mu_{\phi 5}\}$ and $\{\mu_{v1}, \ldots, \mu_{v5}\}$ describe the degree of membership in each set, for the crisp state values of attitude and linear velocity respectively, then the degree of membership C_{ij} in each FAM cell M_{ij} is defined as

$$C_{ij} = \mu_{\phi i}\mu_{vj}$$

The output of the FAM controller is defined as

$$o = \frac{\sum_i \sum_j C_{ij} y_{ij}}{\sum_i \sum_j C_{ij}}$$

where y_{ij} is the value associated with FAM cell M_{ij}. Unlike the previous FAM "teach by show" work in [14], we

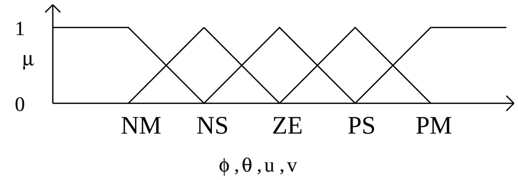

Figure 5: Overlapping fuzzy sets for attitudes and velocities.

φ \ v	NM	NS	ZE	PS	PM
NM	y_{11}	y_{12}			
NS					
ZE			δ_{lat}		
PS					
PM					y_{55}

Figure 6: FAMs for δ_{lat}

have chosen these cell values to be crisp δ control demands rather than fuzzy singleton values.

The FAMs are trained using a simple, weighted average approach. For each input/output training example $[x_t, \delta_t]$, the principle FAM cell $\overline{M}_{ij_t}$ is found to be that which satisfies

$$maxC_{ij} \ \forall i, j \mid x_t$$

The cell value running average for the principle cell $\overline{y}_j$ is adjusted according to the corresponding pilot demand δ_t, and the degree $\overline{C}_{ij_t}$ to which the current state x_t belongs to $\overline{M}_{ij_t}$ according to

$$\overline{y}_j = \frac{\sum_t \delta_t \overline{C}_{ij_t}}{\sum_t C_{ij_t}}$$

This yields an average over all training examples weighted according to their degree of membership within the principel FAM cell. The number of state excursions N into each FAM cell, before training is considered complete, is a design choice. While $N = 1$ is theoretically sufficient to find an examplar for each cell, larger N results in a controller that represents the general actions of the teacher; i.e. any pilot mistakes will be absorbed by the averaging.

5 Simulation Results

Since only the FAM approach was deemed acceptable for online learning, the results will be restricted as such. Velocity control results are first presented. As an intermediate step toward real helicopter experiments, the design of the more simple attitude controller is then presented. Results for both the longitudinal and lateral motion axes were very similar. Consequently, only the lateral motion axis is presented.

5.1 Velocity Control

The lateral cyclic FAM trained with sensor noise, pilot noise and pilot variation is represented as a surface in Figure 7. Note particularly, how a very negative velocity and very negative roll attitude results in a highly positive demand (control stick pushed hard left to induce negative roll), and vice versa. Not clear from the figure is that the demand for $[\phi, v] = [0, 0]$ is almost 0.

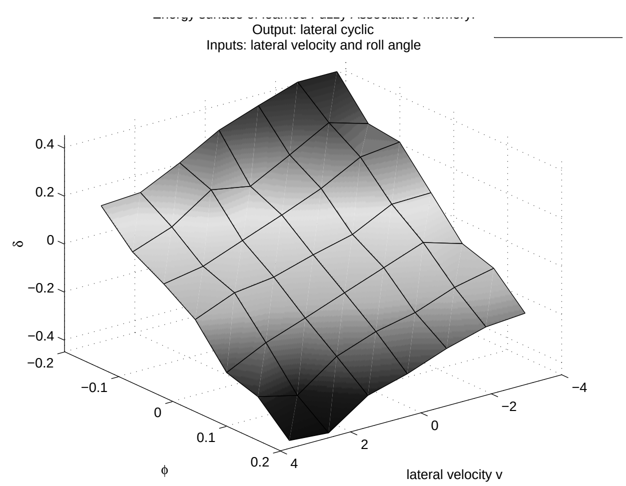

Figure 7: Velocity Control FAM Surface

The FAM velocity controller represented by this surface was successfully used, as a velocity demand interface, by a higher level SISO linear position controller. A comparison of the FAM output and the teacher output during this three waypoint mission is shown in Figure 8.

5.2 Attitude Control

Attitude control is implemented using simple, 1 dimensional, fuzzy associative memories. Figure 9 shows the lateral cyclic FAM represented as a control surface. This FAM attitude controller was successfully used by a higher level MISO linear position/velocity controller. The track-

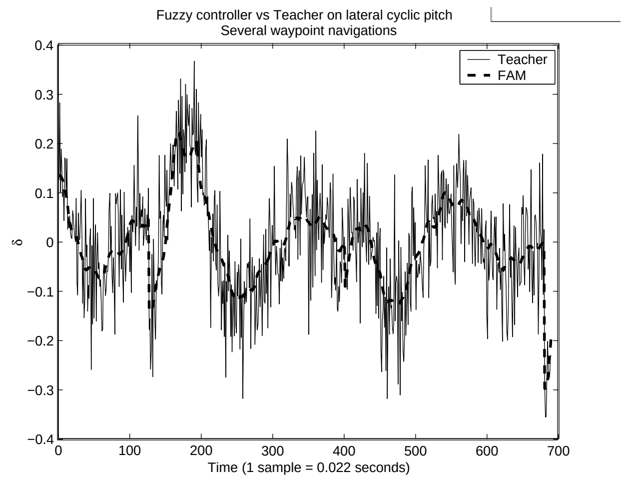

Figure 8: Comparison of FAM controller and Teacher during waypoint maneuvers

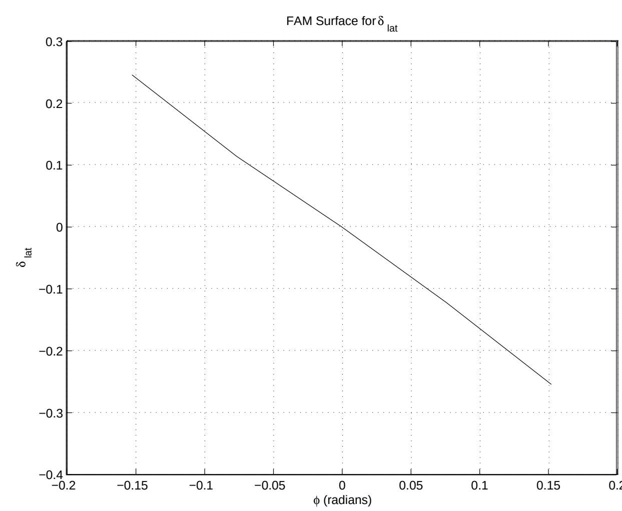

Figure 9: Attitude Control FAM Surface

ing response of the attitude controller is shown in Figure 10.

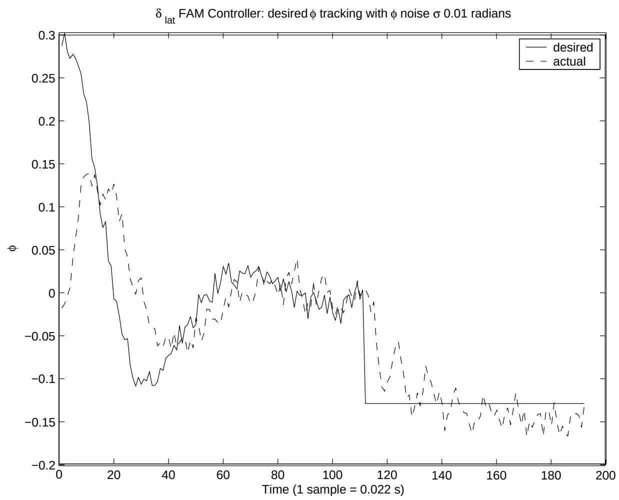

Figure 10: Attitude FAM controller step response

6 Conclusion

This paper details the preliminary experiments toward developing an online learning algorithm for autonomous helicopter control. It has been shown in simulation that single-layer and multi-layer neural networks are unlikely to be suitable for learning from a teacher; teacher variation results in network weights continually oscillating. Tuning learning parameters can damp these oscillations, however this begs the question, *Why not simply tune proportional controller gains?*

This work has found (in simulation) that fuzzy associative memories using a weighted average learning approach, are insensitive to significant pilot variation during training. Velocity and attitude controllers have been successfully trained (in simulation) by a teacher controller. The next step is to test the FAM algorithms on the real JR Ergo helicopter platform.

Acknowledgments

The authors would like to thank the whole automation team for their invaluable assistance and support. In particular, Peter Corke, Craig Worthington, Les Overs, Stuart Wolfe, Steven Brosnan, Pavan Sikka, Graeme Winstanley, Mathew Dunbabin, Elliot Duff, and our pilot Fred Proos.

References

[1] O. Amidi. *An Autonomous Vision-Guided Helicopter*. PhD thesis, Dept of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, 1996.

[2] J.A. Bagnell and J.G. Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *International Conference on Robotics and Automation*, 2001.

[3] G. Buskey. Online learning algorithm development for the csiro autonomous helicopter. Technical report, CSIRO Manufacturing and Infrastructure Technology, Pinjarra Hills, Australia, July 2002.

[4] G. Buskey, J. Roberts, P. Corke, M. Dunbabin, and G. Wyeth. The csiro autonomous helicopter project. In *International Symposium on Experimental Robotics*, Sant'Angelo d'Ischia, Italy, July 2002.

[5] G. Buskey, G. Wyeth, and J Roberts. Autonomous helicopter hover using an artificial neural network. In *IEEE International Conference on Robotics and Automation*, pages 1635–1640, Seoul Korea, May 2001.

[6] C. Cavalcante, J. Cardoso, J.J.G. Ramos, and O.R. Neves. Design and tuning of a helicopter fuzzy controller. *Proceedings of 1995 IEEE International Conference on Fuzzy Systems*, 3:1549–1554, 1995.

[7] T. Chen and T.S. Li. Simplex-type fuzzy sliding-mode control. *Fuzzy Sets and Systems*, 124:249–261, 2001.

[8] R. Enns and J. Si. Helicopter tracking control using direct neural dynamic programming. *Proceedings of the International Joint Conference on Neural Networks*, 2:1019–1024, July 2001.

[9] T.J. Koo and S. Sastry. Outputtracking control design of a helicopter model based on aproximate linearization. *IEEE Conference on Decision and Control*, 4:3635–3640, 1998.

[10] K. KrishnaKumar and J. Neidhoefer. Immunised neurocontrol. *Expert Systems With Applications*, 4(3):201–214, 1997.

[11] D. McLean and H. Matsuda. Helicopter station-keeping: comparing lqr, fuzzy-logic and neural-net controllers. *Engineering Applications of Artificial Intelligence*, 11:411–418, 1998.

[12] B. Mettler, T. Kanade, and M.B. Tischler. System identification modeling of a model-scale helicopter. Technical Report CMU-RI-TR-00-03, Robotics Institute - CMU, March 2000.

[13] B. Mettler, T. Kanade, M.B. Tischler, and W. Messner. Attitude control optimization for a small-scale unmanned helicopter. In *AIAA Guidance, Navigation and Control Conference*, Denver, Colorado, August 2000.

[14] J.F Montgomery and A. Bekey. Learning helicopter control through "teaching by showing". *IEEE Conference on Decision and Control*, 4:3647 – 3652, 1998.

[15] C. Phillips, C.L. Karr, and G. Walker. Helicopter flight control with fuzzy logic and genetic algorithms. *Engineering Applications of Artificial Intelligence*, 9(2):175–184, 1996.

[16] J.V.R. Prasad, A.J. Calise, Y. Pei, and J.E. Corban. Adaptive nonlinear controller synthesis and flight test evaluation. In *Proceedings of the 1999 IEEE International Conference on Control Applications*, pages 137–142, Kohala Coast-Island of Hawaii, August 1999.

[17] J.M. Roberts, P.I. Corke, and G. Buskey. Low-cost flight control system for a small autonomous helicopter. In *International Conference on Robotics and Automation*, 2003 - Submitted for review.

[18] C.P. Sanders. Hierarchical control of small autonomous helicopters. *IEEE Conference on Decision and Control*, 4:3629 – 3634, 1998.

[19] K. Sprague, V. Gavrilets, D. Dugail, B. Mettler, and E. Feron. Design and applications of an avionics system for a miniature acrobatic helicopter. In *Digital Avionics Systems Conference*, pages 3.C.5–1 – 3.C.5–10, Daytona Beach Florida, October 2001.

[20] M. Sugeno, H. Winston, and I. Hirano. Intelligent control of an unmanned helicopter based on fuzzy logic. In *51st Annual Forum of the American Helicopter Society*, Frt Worth, May 1995.

[21] B. Woodley and H. Jones. A contestant in the 1997 international aerial robotics competition. aerospace robotics laboratory, stanford university. In *AUVSI*, July 1997.