ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 159-164

A Comparative Design of a MIMO Neural Adaptiv e Rate Damping for a Nonlinear Helicopter Model *

P.A. Gili, M. Battipede

Politecnico di Torino, Aeronautical and Space Department,
C.so Duca degli Abruzzi 24, 10129 Torino, Italy

Abstract. Using a nonlinear 15-state helicopter model in 6 DOF, two different neural control systems, both acting as rate damping, have been designed and compared. They are both based on the reference model direct inverse scheme, but they differ each from the other for the identification of the inverse model: the first one is a MIMO feedforward two-layered neural network, while the second one is a combination of three MISO feedforward two-layered neural networks connected in parallel. The strong dynamic cross-coupling, that characterizes the model, has enabled us to verify the actual MIMO capability of both the neural rate damping configurations. However the multi-MISO version has demonstrated to have a more robust adaptive ability.

1. Introduction

The helicopter represents a very complex dynamic system, whose special features can be summarized as follows: it can operate in a very wide range of flight conditions; its response characteristics and the piloting strategy may vary significantly depending on flight condition; in most flight conditions, the command response is characterized by very strong cross-couplings in all axes, so that each control action needs to be associated with a proper compensation action; it is dynamically unstable in almost the whole flight envelope; its command system architecture and the inertia associated with the rotoric flow field introduce a measurable intrinsic delay between the command action and the response. These characteristics affect not only the activity of the pilot but also that of the control systems that equip the helicopter. The neural technology has already demonstrated to be an effective alternative to conventional control techniques for fixed-wing aircraft applications (a wide bibliografy on this subject is reported [1]). The helicopter flight dynamics control perhaps represents one of the most evident situations where neural technology is favoured over conventional methods: it allows adaptive and fault-tolerant control, with a considerably reduced gain-scheduling activity; it handles MIMO (Multi-Input-Multi-Output) control, without increasing the complexity of the structure; it can be applied even when the plant is characterized by strong non-linearities.

In [2] the neural technology was used to implement a MIMO terrain-follo wing autopilot, for a 9-state nonlinear helicopter model in 6 DOF. This full-authority controller was based on the *reference model direct inverse* scheme reported in

*Research performed with CNR contribution

figure 1 (also known as *predictor-corrector*); the inverse model was identified with a MIMO feedforw ard two-layered neural network, trained on-line, while the emulator was identified with a MIMO feedforw ard two-layered neural network, trained off-line. The same scheme is used to implemen t the rate damping that will be described in the third paragraph. The aim of the work is to compare the performance of two different versions of this controller: the first one is substantially similar to the autopilot of [2], the second one is based on the same structure but the inverse model is identified by three MISO (Multi-Input-Single-Output) networks connected in parallel.

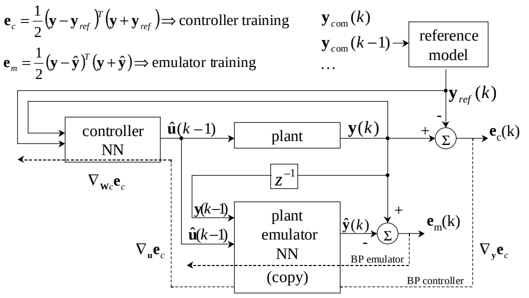

$\mathbf{e}_c = \frac{1}{2}(\mathbf{y}-\mathbf{y}_{ref})^T(\mathbf{y}+\mathbf{y}_{ref}) \Rightarrow$ controller training
$\mathbf{e}_m = \frac{1}{2}(\mathbf{y}-\hat{\mathbf{y}})^T(\mathbf{y}+\hat{\mathbf{y}}) \Rightarrow$ emulator training
$\mathbf{y}_{com}(k)$
$\mathbf{y}_{com}(k-1)$
...
reference model
$\mathbf{y}_{ref}(k)$
controller NN
$\hat{\mathbf{u}}(k-1)$
plant
$\mathbf{y}(k)$
+
-
$\mathbf{e}_c(k)$
$\nabla_{\mathbf{w}_c}\mathbf{e}_c$
z^{-1}
$\mathbf{y}(k-1)$
$\hat{\mathbf{u}}(k-1)$
plant emulator NN
$\hat{\mathbf{y}}(k)$
$\mathbf{e}_m(k)$
BP emulator
(copy)
BP controller
$\nabla_{\mathbf{u}}\mathbf{e}_c$
$\nabla_{\mathbf{y}}\mathbf{e}_c$

Figure 1: Predictor-corrector scheme

2. Helicopter mathematical model

The 15-state nonlinear helicopter model in 6 DOF (extensively described in [3]), identifies a twin engine helicopter belonging to the 2.5 tons class. The equations of motion can be written in a compact and explicit form, as follows:

$$\dot{\mathbf{X}} = f(\mathbf{X}, \mathbf{U}) \tag{1}$$

where $\mathbf{X}$ represents the state vector, while $\mathbf{U} = (\theta_0 \;\; \theta_{1s} \;\; \theta_{1c} \;\; \theta_{0t})^T$ is the control vector which is made up of the main and tail rotors pitch angles, that are supposed to be applied directly on the blades. This means that the dynamics connected to the pilot's action, to the actuators and to the mechanical interlinks has not been modeled.

In the flight simulations which will be presented, the initial condition is a stable and non-minimum-phase trim point, where the helicopter (equipped with a SAS that acts on the latero-directional variables) exhibits three oscillatory damped modes, the phugoid ($\omega_{ph} = 0.1319\,rad/s \;\; \xi_{ph} = 0.0923$), the short period ($\omega_{sp} = 2.3789\,rad/s \;\; \xi_{sp} = 0.6064$), the Dutch roll ($\omega_{dr} = 2.9161\,rad/s \;\; \xi_{dr} = 0.3287$) and two subsidences, the spiral mode ($\xi_s = 1.0703$) and the roll mode ($\xi_r = 1.2055$).

3. The MIMO adaptive Rate damping

The rate damping is the heart of any helicopter automatic flight control system (AFCS). Due to the command system architecture (the swash plate) it acts mainly on the angular rates p, q and r of the overall system. The rate damping has a twofold task: to avoid the angular motions to arise, unless the pilot maneuvers on purpose (for instance as a consequence of a gust); to improve the angular motion response to a pilot maneuver. The rate damping operation is very simple: sensors detect angular rates values different from zero (the reference value); the error signal, calculated from the difference between the actual and the reference signal, is fed to the controller, which activates the cyclic (θ_{1s} e θ_{1c}) and the tail rotor collective (θ_{0t}) actuators. The maneuver commanded by the controller cancels the undesired angular rates, but, since this happens with a certain delay, as regarding disturbance, the attitude may have changed. The rate damping is not able to bring the helicopter back to the original attitude and this is one of the reasons why the rate damping is not an independent control system. For this reason, during simulations the attention will be focused on the rates variables, while the attitude variables trends will not be considered.

As shown in figure 2 the rate damping is a MIMO control system with three control variables (θ_{1s}, θ_{1c} and θ_{0t}) and three controlled variables (p, q and r). During simulations, the main rotor collective pitch is supposed to be held at fixed values.

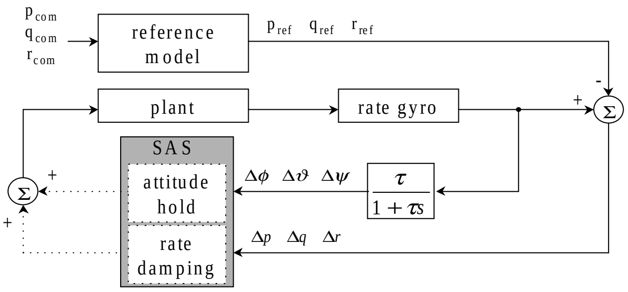


Figure 2: Rate damping working scheme

3.1. MIMO identification of the inverse model

The most simple way to implement the MIMO rate damping consists in identifying the inverse model with a single MIMO network, having three inputs (p, q and r) and three outputs (θ_{1s}, θ_{1c} and θ_{0t}). The inverse model is identified through a scheme that can be drawn back to the ARX type model [4]. It has been chosen a feedforward two-layered neural network, where the input layer is the second order regression vector (19 neurons, i.e. 18 plus the bias), the hidden layer 16 neurons have a bipolar hyperbolic tangent activation function, while the 3 output layer neurons have a linear activation function. On the whole the network is nonlinear.

Both the forward and the inverse model are trained off-line with the Levenberg-Marquardt method [5], based on the back-propagation technique. To make the controller adaptive, the inverse model must be trained on-line: the technique used is described in [2]. It belongs to the Recursive Identification methods category, where the weights update is based on the step by step calculation of the derivative matrix:

$$\mathbf{\Psi}(k) = \nabla_\Theta \mathbf{y}(k) \cong \nabla_\Theta \hat{\mathbf{y}}(k) = \nabla_{u(k-1)} \hat{\mathbf{y}}(k) \cdot (\nabla_\Theta \mathbf{u}(k-1)) \tag{2}$$

It is important to point out that the (i, j) element of the Ψ matrix, namely the derivative of the i^{th} output with respect to the j^{th} element of the Θ vector*, is calculated summing the contributions filtered through all the command signals:

$$\Psi(i,j) = 3\frac{\partial \hat{y}(i)}{\partial \Theta(j)} = \frac{\partial \hat{y}(i)}{\partial u(1)}\frac{\partial u(1)}{\partial \Theta(j)} + \frac{\partial \hat{y}(i)}{\partial u(2)}\frac{\partial u(2)}{\partial \Theta(j)} + \frac{\partial \hat{y}(i)}{\partial u(3)}\frac{\partial u(3)}{\partial \Theta(j)} \tag{3}$$

3.2. Multi-MISO identification of the inverse model

An alternative method to implement the MIMO rate damping consists in identifying the inverse model through three MISO networks, connected in a parallel stucture, following the scheme reported in figure 3. Each network has three input variables, the angular rates, and a single output variable, that is one of the three control variables. Adopting the same ARX model described above, each network turns out to have 12 neurons in the input layer; for the hidden layer it has been chosen to maintain the 16 neurons of the MIMO network. On the whole the inverse model dimension is increased, meaning that the number of connections is almost doubled: 675 for the three MISO networks vs. the 355 of the single MIMO network. Obviously also the computation cost is increased, anyway the ensuing advantages are not negligible.

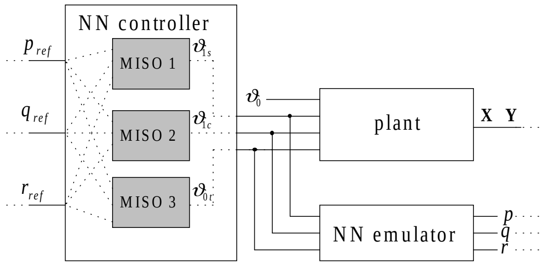


Figure 3: Inverse model identification through three parallel MISO networks

The inverse model weights update is performed through the calculation of three Ψ_m matrices, according to the form:

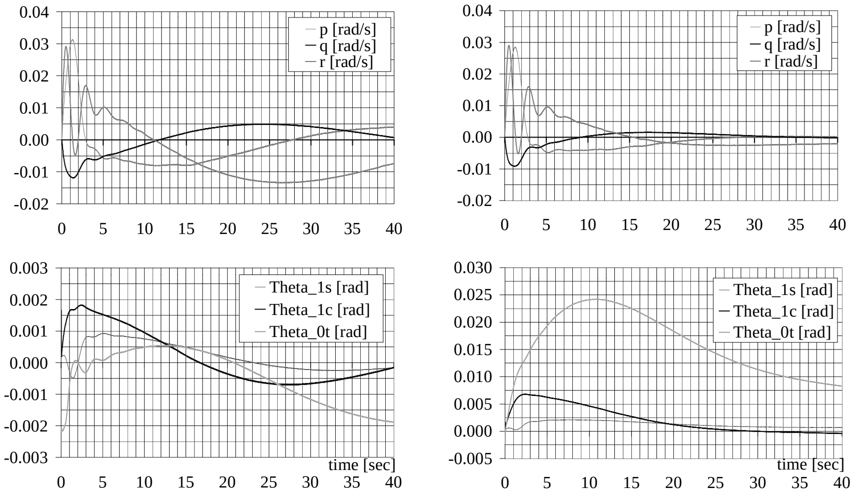
$$\mathbf{\Psi_m}(k) = \nabla_{\Theta_m} \hat{\mathbf{y}}(k) = \nabla_{u_m(k-1)} \hat{\mathbf{y}}(k) \cdot (\nabla_{\Theta_m} u_m(k-1)) \tag{4}$$

whose generic (i, j) element can be written as follows:

$$\Psi_m(i,j) = \frac{\partial \hat{y}(i)}{\partial \Theta_m(j)} = \frac{\partial \hat{y}(i)}{\partial u_m}\frac{\partial u_m}{\partial \Theta_m(j)} \tag{5}$$

which means that the effects of the m^{th} matrix weights on the outputs are filtered exclusively through the m^{th} control signal.

*Θ collects the inverse model weights matrices in vector shape



(a) (b)

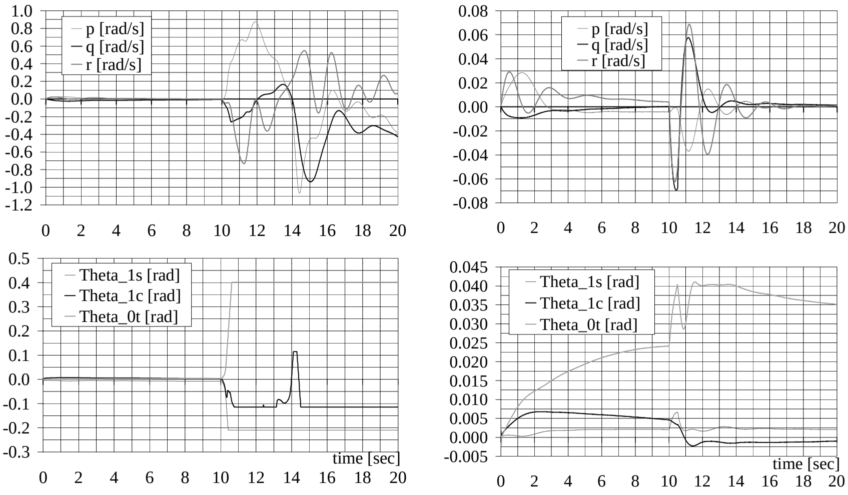
Figure 4: On-line training of the inverse model in calm air: MIMO (a) and multi-MISO (b)

4. Results and Conclusions

Results are presented in form of comparison between the angular rates trends. The lower part of each diagram represents the required control maneuver. Figure 4 shows the behaviour of the two controllers trained on-line in a situation of calm air (the helicopter is flying with a speed of $50\,m/s$ at the height of $100\,m$): their performances are comparable since both are able to control the MIMO dynamic system. The greater dimension of the multi-MISO controller would seem to penalize this solution, but the next test-case demonstrates that the multi-MISO configuration features more robust adaptive capabilities. Figure 5, in fact, reports the behavior of the two controllers in a gust event: a vertical gust of $5\,m/s$ suddenly occurs after 10 seconds from the controller activation; it goes on for 0.5 seconds and then disappears. In this situation the pure MIMO controller clearly demonstrates to be inadequate to control the MIMO dynamic system, while the multi-MISO controller continues to perform well. The explanation lies in the comparison between the (i,j) elements of the Ψ and Ψ_m matrices: in the Ψ matrix three different gradient descent directions are suggested (eq. 2) and the controller is obliged to follow the forth one, namely the mean one, that is not necessarily the best; on the contrary, in the Ψ_m matrix the learning algorithm gives a more detailed piece of information, directly selecting the optimum gradient descent direction (eq. 4).

List of Symbols

BP	Back-propagation algorithm
e	Error signal



(a) (b)

Figure 5: On-line training of the inverse model in a gust event: MIMO (a) and multi-MISO (b)

$\{p\,q\,r\}^T$	Angular rates vector (roll, pitch and yaw rate)
$\mathbf{u}$	Control signal
$\mathbf{U}$	Controls vector, i.e. main rotor collective pitch (θ_0), main rotor longitudinal and lateral ciclic pitch (θ_{1s} and θ_{1c}), tail rotor collective pitch (θ_{0t})
$\mathbf{X}$	State vector
$\mathbf{y}$	Output signal
$\mathbf{y}_1$	Hidden layer neurons
$\hat{\mathbf{y}}$	Approximate output signal
$\{\varphi\,\theta\,\psi\}^T$	Attitude variables vector (bank, pitch and yaw angle)
Θ	Inverse model weights vector
Ψ	Derivatives vector

References

[1] P. Gili, M. Battipede: An Adaptive Neurocontroller for a Nonlinear Combat Aircraft model, AIAA paper n. 98-4485, AIAA GN&C Conference, Boston MA (August 10-12 1998)

[2] P. Gili, M. Battipede: A MIMO Neural Adaptive Autopilot for a Nonlinear Helicopter Model, AIAA paper n. 99-4219, AIAA GN&C Conference, Portland OR, (August 9-11, 1999)

[3] M. Battipede: MIMO Neural Adaptive Control for Helicopter Flight Dynamics, Ph.D. diss., Polytechnic of Turin (November 1999)

[4] L. Ljung: System Identification: Theory for the User Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1987)

[5] M. Norgaard: Neural Network Based System Identification Toolbox, Tech. Report 97-E-851, Department of Automation, Technical University of Denmark (1997)