

# DISCRETE-TIME ROBUST POLE-PLACEMENT DESIGN THROUGH GLOBAL OPTIMIZATION

Miguel Ayala Botto <sup>\*,1</sup> Robert Babuška <sup>\*\*</sup> José Sá da Costa <sup>\*</sup>

*\* Technical University of Lisbon, Instituto Superior Técnico  
Department of Mechanical Engineering, GCAR  
Av. Rovisco Pais, 1049-001 Lisboa, Portugal  
tel: +351 218 419 028, fax: +351 218 498 097  
e-mail: ayalabotto@ist.utl.pt*

*\*\* Delft University of Technology, Faculty ITS, Control Laboratory  
Mekelweg 4, P.O. Box 5031, 2600 GA Delft, The Netherlands  
tel: +31 15 27 85117, fax: +31 15 27 86679  
e-mail: r.babuska@its.tudelft.nl*

Abstract: A robust pole placement controller design method is presented for discrete-time systems with parametric model uncertainties contained within known bounds. The design methodology is based on the minimization of a cost function by using genetic algorithms. It allows for a thorough assessment of robust performance in addition to robust stability. The effectiveness of this technique is shown for a real-time experimental laboratory-scale helicopter system. *Copyright © 2002 IFAC*

Keywords: robust control, pole assignment, uncertainty, genetic algorithms, helicopter dynamics.

## 1. INTRODUCTION

This paper describes a design methodology which extends the basic pole placement method to discrete-time systems with parametric uncertainties. Pole placement controller design is a well known technique to control LTI systems. This method is intuitive and simple to use and can be equally well applied to both continuous as well as discrete-time systems (Ackermann, 1993). The basic idea behind it is the design of state feedback such that all poles of the closed-loop system assume prescribed values. The success of the pole-placement design is strongly dependent on the availability of an accurate model of the system under study. As modeling is a well-known bottleneck, there is a strong demand for robust pole-placement design that can take model uncertainty into account, while satisfying the closed-loop stability and performance spec-

ifications. Model uncertainty can be generally classified into two different types: parametric uncertainty, which represents imprecision of parameters within the model, and unstructured uncertainty, which represents unmodeled dynamics. In this paper the former one will be considered by assuming that each uncertain parameter is allowed to vary within some known bounds.

There is quite a rich range of literature on pole placement for uncertain systems; see (Söylemez, 1999) for a recent review of the developments for continuous-time systems. Not included in this review, but worth mentioning, are the contributions of Evans and Xianya (1985), Soh *et al.* (1987), Soh (1989) and Garcia *et al.* (1996) who give procedures for robust pole assignment in the presence of known but bounded parameter variations in the plant. Keel *et al.* (1988) and Solak and Peng (1995) study how the pole placement controller affects the robust stability of the resulting closed-loop system. In (Keel *et al.*, 1988), an iterative

---

<sup>1</sup> To whom all the correspondence should be sent.

procedure is also given for adjusting the parameters of the controller so that robustness requirements are met. With regard to discrete-time systems, on the other hand, the number of notable references is not very high. The technique described by Solak (1994) aims at determining a pole placement controller such that the real stability radius of the closed-loop characteristic polynomial is maximized. Halpern *et al.* (1995) consider systems with norm-bounded parametric uncertainties and describe how to design overparametrized pole assignment controllers in order to reduce the effect of uncertainties on the closed-loop characteristic polynomial. Given a system with both structured and unstructured uncertainties, the problem of placing its poles into the smallest possible circular region is dealt with in (Kim *et al.*, 1996). Finally, the results of (Soh *et al.*, 1987; Garcia *et al.*, 1996) apply to both continuous-time and discrete-time systems.

The common shortcoming of the works cited above is that the important and non-trivial task of pairing the nominal closed-loop poles with their perturbed counterparts is not treated. The only serious attempt in this direction appears to have been made by Söylemez and Munro (1997) by introducing a concept called *pole coloring* which relies on global optimization techniques in order to obtain a pole assignment controller that achieves the pairing between nominal and perturbed closed-loop poles and thus allows for a thorough assessment of robust performance in addition to robust stability. The novelty of the present paper is the use of a simplified and slightly modified version of this concept for discrete-time systems. The proposed control scheme has been successfully applied to real-time control of a laboratory-scale helicopter system.

The rest of the paper is organized as follows. In Section 2 background information on pole placement control of discrete-time systems is given. Section 3 presents the methodology employed to design a robust pole placement controller with the pole coloring concept. Experimental real-time results are presented in Section 4. Finally, some conclusions are drawn in Section 5.

## 2. STANDARD POLE PLACEMENT CONTROLLER DESIGN

Consider a completely controllable and observable MIMO dynamic process with  $m$  inputs,  $n$  states and  $m$  outputs, described by the following linear discrete-time nominal model:

$$\mathcal{M}_o : \begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \end{cases} \quad (1)$$

The standard pole placement design method addresses the problem of driving the states of the system to zero. If, on the other hand, the objective is to make the states and the outputs of the system follow desired trajectories, then the pole placement controller needs to be

augmented by extra dynamics. Unless the plant has an integrating property, these extra dynamics are usually chosen to be one or two integrators so that steady-state errors can be eliminated. This paper addresses the latter of the above two problems by designing a unity feedback control scheme whose feedforward path consists of an integrator connected in series with the plant which has a pole placement controller around it. A detailed formulation of such a control scheme as a pole placement problem can be found, for instance, in (Ogata, 1987). The resulting closed-loop system is given by:

$$\mathbf{x}_c(k+1) = \mathbf{A}_c\mathbf{x}_c(k) + \mathbf{B}_c\mathbf{r}(k+1) \quad (2)$$

$$\mathbf{y}(k) = \mathbf{C}_c\mathbf{x}_c(k) \quad (3)$$

where  $\mathbf{r}(k)$  is the reference input vector,  $\mathbf{y}(k)$  is the output vector of the plant,  $\mathbf{x}_c(k) = [\mathbf{x}(k) \ \mathbf{u}(k)]^T$  is the closed-loop state vector, and:

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{K}_2(\mathbf{I}_m - \mathbf{A}) - \mathbf{K}_1\mathbf{C}\mathbf{A} & \mathbf{I}_m - (\mathbf{K}_2 + \mathbf{K}_1\mathbf{C})\mathbf{B} \end{bmatrix} \quad (4)$$

$$\mathbf{B}_c = [\mathbf{0}_{n \times m} \ \mathbf{K}_1]^T \quad (5)$$

$$\mathbf{C}_c = [\mathbf{C} \ \mathbf{0}_m] \quad (6)$$

with  $\mathbf{K}_2$  being the  $m \times n$  pole placement matrix and  $\mathbf{K}_1$  the  $m \times m$  integrator gain matrix, corresponding to the best overall system state-feedback matrices computed from:

$$[\mathbf{K}_2 : \mathbf{K}_1] = \left[ \hat{\mathbf{K}} + \begin{bmatrix} \mathbf{0} : \mathbf{I}_m \end{bmatrix} \right] \begin{bmatrix} \mathbf{A} - \mathbf{I}_n & \mathbf{B} \\ \mathbf{C}\mathbf{A} & \mathbf{C}\mathbf{B} \end{bmatrix}^{-1} \quad (7)$$

with  $\hat{\mathbf{K}}$  being the solution of the pole placement technique according to the Ackermann's formula (Ackermann, 1993), obtained once the desired closed-loop poles are specified.

## 3. ROBUST POLE PLACEMENT CONTROLLER DESIGN

If the entries of matrices  $(\mathbf{A}, \mathbf{B})$  of the nominal model  $\mathcal{M}_o$  are uncertain, then  $\mathbf{K}_1$  and  $\mathbf{K}_2$  need to be determined such that perturbations in the closed-loop poles as a result of the uncertainties do not violate the design specifications. This study considers parametric model uncertainties with the following model description:

$$\mathcal{M}_p : \begin{cases} \mathbf{x}(k+1) = \mathbf{A}(\mathbf{p})\mathbf{x}(k) + \mathbf{B}(\mathbf{p})\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \end{cases} \quad (8)$$

with  $\mathbf{p} = [p_1 \ p_2 \ \dots \ p_l]^T$  being the vector containing the uncertain system parameters, where  $p_j \in [p_j^- ; p_j^+]$ , for  $j = 1, \dots, l$ , with  $p_j^-$  and  $p_j^+$  being independent known bounds. As mentioned in Section 1, a vast array of techniques is available for robust pole placement design of such a type of systems in the continuous-time domain, most of them relying on the concept of pole region assignment by relaxation of the nominal

closed-loop specifications. By defining an admissible region  $\Gamma$  for the location of the perturbed poles, it is possible, under certain conditions, to design a fixed feedback gain controller that guarantees simultaneous  $\Gamma$ -stabilization for a family of plant models. This idea may be further extended to consider the combination of several performance requirements as well, for instance, constraints on the rise time, settling time, overshoot, and so on. The immediate disadvantage of this procedure is that there is no direct assessment of the robust performance in the controller design. In view of this, Söylemez and Munro (1997) introduced the concept of *pole coloring* which will be adopted in this paper. The main idea is to consider the pairing between the nominal and perturbed closed-loop poles when computing the feedback gains, through the minimization of a cost function of the form:

$$J_a = \min_{q=1,\dots,n!} \left( \max_{i=1,\dots,n} f_i \right) \quad (9)$$

for all possible uncertain plant models  $\mathcal{M}_p$ . Here,  $n$  is the order of the closed-loop system, and each  $f_i$  is a robustness assessment function which increases as a perturbed closed-loop pole  $\lambda_i^p$  moves away from the nominal closed-loop pole  $\lambda_i^o$  with which it is paired. Given  $n$   $\lambda_i^o$  and as many  $\lambda_i^p$ , there are  $n!$  possible permutations for pairing them, and the minimization in (9) aims to choose the best one. The natural way to define  $f_i$  is to consider the perturbed poles to stay in disks centered around the nominal poles, so the following index can be used:

$$f_i = |\lambda_i^o - \lambda_i^p| \quad (10)$$

Note that as the distance between the  $\lambda_i^o$  and the associated  $\lambda_i^p$  reduces, the perturbed closed-loop system looks more like the nominal one. The power of this idea is reinforced by the combination in index  $f_i$  of all the above mentioned closed-loop performance specifications, which can be nicely achieved through a suitable normalization of each individual cost functions (Söylemez and Munro, 1997). Furthermore, it is possible in this way to consider dominant poles to stay in smaller disks and allow the far left poles of the complex plane to live in larger disks. While such measures have obvious significance for the continuous-time domain, there seems to be no immediate way of rewriting them in the  $z$ -plane. This fact stems from the evidence that a disk in the  $s$ -plane is not reproduced equally in the  $z$ -plane due to the nonlinearity involved in the transformation. Besides, some of the most intuitive performance specifications in the  $s$ -plane are lost when transposed to the discrete-time domain, for instance: constant real part lines in the  $s$ -plane are mapped into circles centered at  $z = 0$  in the  $z$ -plane; constant damping lines in the  $s$ -plane are mapped into logarithmic spirals in the  $z$ -plane.

For most practical situations, this problem can be reasonably well circumvented by approximating such a spiral by a circle whose center coordinates  $(x_c, 0)$  and radius  $r$  fulfill the relation:

$$x_c(1 - x_c) = \alpha r(1 - r) \quad (11)$$

with  $\alpha$  being an arbitrary constant less than unity (Ackermann, 1993). Figure 1 is an illustration of this idea taken from a recent study (Impram *et al.*, 2001). It depicts the  $\zeta = 0.2$  spiral and the circle with  $(0.17, 0)$ ,  $r = 0.75$  and  $\alpha = 0.75$ . Note that only a small region around  $z = 1$  is left out by this approximation which actually has an implicit advantage; it puts an upper bound on the settling time  $t_s$  (see the circle drawn in dashed lines which corresponds to  $t_s \approx 2.38$  sec).

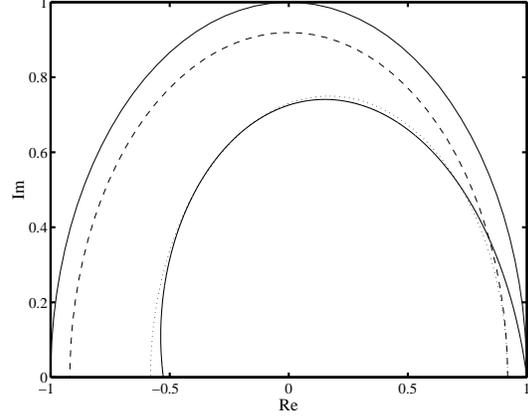


Fig. 1. The unit circle (solid line), the  $t_s \approx 2.38$  sec circle (dashed line), the  $\zeta = 0.2$  spiral (solid line), and the circle (dotted line) approximating the latter.

Having determined the desired region for the closed-loop poles, it is worth pointing out that if the objective is merely to put the  $\lambda_i^p$  inside this region, then this can be achieved by choosing a pole placement controller such that the cost function:

$$J_b = \max_{i=1,\dots,n} |\lambda_i^p - x_c| \quad (12)$$

satisfies  $J_b < r$  for all possible uncertain plant models,  $\mathcal{M}_p$ . However, since the aim here is also to pair the  $\lambda_i^p$  with the  $\lambda_i^o$ , a more sophisticated cost function is defined as:

$$J = \max_{\mathcal{M}_p} \begin{cases} J_a, & \text{if } r > \max_{i=1,\dots,n} |\lambda_i^p - x_c| \\ J_b, & \text{if } r < \max_{i=1,\dots,n} |\lambda_i^p - x_c| \end{cases} \quad (13)$$

where  $J_a$  is the same as in (9) with the  $f_i$  as given in (10), and  $J_b$  is as defined above in (12).

It is obvious that the above cost function is not analytically differentiable. Furthermore, it is, in general, multi-modal and hence it is not possible to use gradient-based optimization algorithms. Therefore, genetic algorithms (Goldberg, 1989) will be employed in this paper as a search mechanism to locate the global minimum of (13).

#### 4. EXPERIMENTAL RESULTS

A ‘helicopter’ laboratory setup was used to experimentally validate the proposed control technique. The

setup consists of a beam attached to a fixed pole such that it can freely rotate in the horizontal and vertical planes. At both ends of the beam, DC motors with propellers are mounted. One propeller is used to control the vertical angle (pitch), the other one the horizontal angle (yaw), see Figure 2. The objective is to control the motors such that attitude of the beam follows a specified reference trajectory.

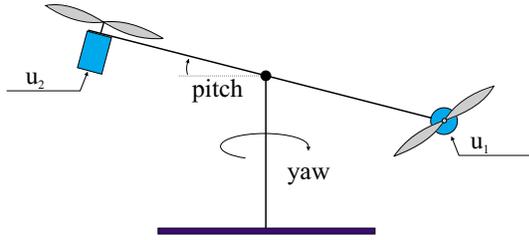


Fig. 2. The helicopter setup.

This system has two control inputs,  $u_1$  and  $u_2$ , which are the voltages applied to the motors to control the yaw and the pitch, respectively. These inputs are commanded through a D/A card from a computer. There are four measured outputs: the two angles of the beam (yaw and pitch, in radians) and the angular velocities of the two propellers. To evaluate the robust pole placement controller, only the pitch controller is considered, by setting  $u_1$  equal to zero. A linearized model was derived from the nonlinear model equations based on elementary physical insight. In the pitch plane, the beam is basically a physical pendulum which, when excited, exhibits poorly damped oscillations around its stable equilibrium. This is because the axis of rotation is above the center of gravity. The motor is modeled as a first-order linear system and the force generated by the propeller is proportional to the propeller's angular velocity  $\omega$ :

$$\tau\dot{\omega} + \omega = K_1 u_2 \quad (14)$$

$$\ddot{\alpha} + b\dot{\alpha} + K_2 \sin \alpha = K_3 \omega \quad (15)$$

with  $\alpha$  being the beam's angular position (pitch),  $\tau$  being the time constant of the motor (including the propeller),  $K_1$  the gain from the control signal to the propeller's velocity,  $b$  the damping (viscous friction) of the beam's motion,  $K_2$  a constant related to the influence of the gravity force and  $K_3$  the gain from the propeller's velocity to the angular acceleration of the beam.

These parameters were identified from input–output data measured on the system with a sampling period of 0.1 sec. Six data sets were recorded for different ranges of the pitch angle, yielding a family of six different discrete-time plants whose step responses are shown in Figure 3. It is clear that the system exhibits different damping and stationary gain through its domain of operation, which confirms its nonlinear dynamic characteristics. In the sequel these 6 models will be considered sufficient to describe the entire

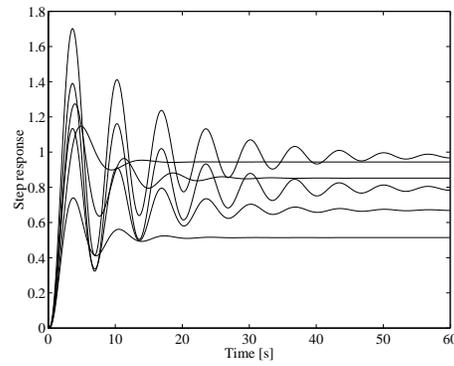


Fig. 3. Step responses from the six identified models. operating envelope of the system, and therefore will be used to design a robust pole placement controller.

The nominal closed loop poles,  $\lambda_i^o$ , were chosen to be  $\{0, 0.5743 \pm j0.1247, 0.926 \pm j0.093\}$ . This choice gives the rise time of 1.55 sec and the damping ratio  $\zeta$  of about 0.58 which corresponds to about 10% overshoot, the settling time of approximately 5.15 sec and a dead-time of one sampling period. The dead-time is due to the implementation of the real-time controller (using the Real-Time Toolbox of MATLAB) and is thus regarded as a pure delay in the system's input. Besides, notice that the pair  $\{0.926 \pm j0.093\}$  is much more dominant than the other poles. Given that there is parametric uncertainty in the plant, it would be quite unrealistic to think that the perturbed closed-loop system can be made to have the above performance characteristics. Therefore, the procedure described in Section 3 is here applied, by first defining the center coordinates and the radius of the circle that best approximates the constant damping line,  $\zeta = 0.58$ , in the z-plane.

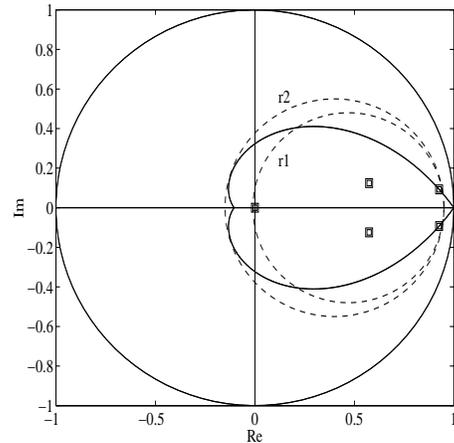


Fig. 4. The unit circle, the  $\zeta = 0.58$  spiral (solid line), the nominal closed-loop poles ( $\square$ ) and two approximating circles with center coordinates  $x_c = 0.47$  and  $x_c = 0.4$  with radius  $r_1 = 0.48$  and  $r_2 = 0.55$ , respectively (dashed lines).

Figure 4 shows two possibilities that clearly reflect the dilemma involved in this choice: in order to encircle all the nominal closed-loop poles while providing

the best damping line approximation, the circle with the smallest radius should be adopted. In this case, however, a valid region in the  $z$ -plane is left out from the circle. This should not be a major concern, however, since it is expected from the optimization index defined in (13) that the pairing of the nominal with the perturbed poles will occur as soon as all perturbed poles are inside the circle. Simulations as well as experimental results have indeed shown that the choice for the  $r_1$  radius circle resulted in the best control performance.

The parameters of the genetic algorithm were set as follows: 10-bit binary encoding in a population size of 100 individuals, with a crossover rate of 0.9 and a mutation rate of 0.007. Reproduction is performed by stochastic universal sampling and a generation gap of 0.7. Finally, the values of  $J$  corresponding to each member of the population were converted into a fitness value through  $\sigma$ -scaling in order to prevent premature convergence. After a series of computer runs, the minimum of (13) is attained at 0.3628 for the approximating circle with radius  $r_1$ . Figure 5 shows the convergence of the optimization algorithm up to this point.

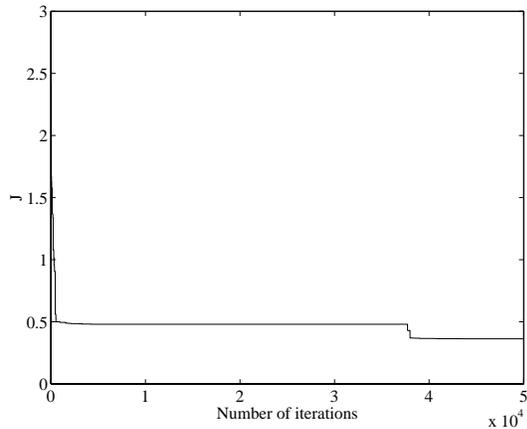


Fig. 5. Convergence of the genetic algorithm.

The corresponding location of the perturbed poles is shown in Figure 6. In this plot one can see the clusters of perturbed poles around the dominant poles  $0.926 \pm j0.093$ , giving a clear image of the pole pairing concept.

The corresponding unitary step responses of the closed-loop system are shown in Figure 7. From this plot, it is observed that the rise time is between 1.8 and 2.1 sec, the overshoot has a maximum of 8.22%, while the settling time is in the worst case of 6.2 sec. In view of this, it can be concluded that, besides the rise-time, generally all the other nominal design specifications are reasonably well satisfied. Moreover, from the comparison with the original open-loop step responses previously shown in Figure 3, one may conclude that the robust pole placement controller provided an overall reduction on the system's uncertain behaviour.

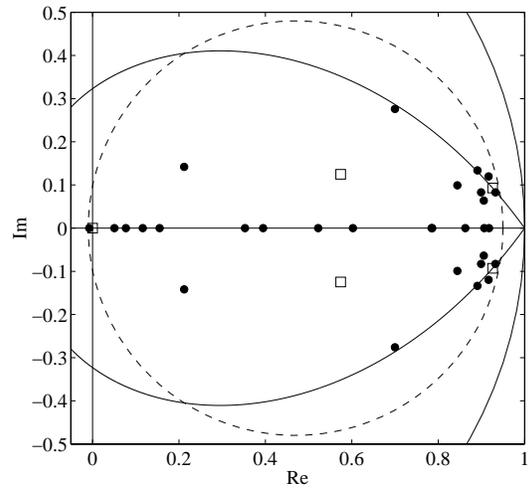


Fig. 6. Perturbed (●) and nominal (□) poles of the closed-loop system for the approximating circle with radius  $r_1$  (dashed line).

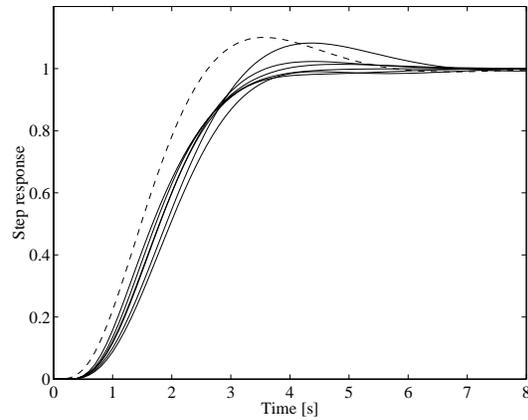


Fig. 7. Step responses of the nominal (dashed line) and perturbed (solid lines) closed-loop system.

The experimental results show a comparison between the performance of a standard pole placement controller (Section 2) and the robust pole placement controller (Section 3). The standard controller is based on a nominal model obtained as the average of the six identified models. The results are shown in Figures 8 and 9, respectively.

The gains found for the standard pole placement controller were  $\mathbf{K}_1 = 1.54$  for the integrator gain, and  $\mathbf{K}_2 = [16.30, 15.77, 1.68, 0.68]$  for the state vector  $\mathbf{x}(k) = [\alpha(k), \dot{\alpha}(k), \omega(k), u_2(k-1)]$  feedback gains. It can be seen from Figure 8 a high activity in the control action. Besides, the overall closed-loop tracking performance exhibits large overshoots while steady-state errors are often visible.

In contrast, the robust pole placement controller resulted in  $\mathbf{K}_1 = 0.88$  for the integrator gain, and  $\mathbf{K}_2 = [11.83, 10.65, 1.27, 0.49]$  for the state vector feedback gains, which are considerably smaller than the ones obtained from the standard pole placement design. This has a positive impact in reducing the control action activity as can be seen in Fig-

This work was in part supported by the Ministério da Ciência e da Tecnologia de Portugal under the POCTI, FCT program.

## 6. REFERENCES

- Ackermann, J. (1993). *Robust Control: Systems with Uncertain Physical Parameters*. Springer-Verlag, London, UK.
- Evans, R. J. and X. Xianya (1985). Robust regulator design. *International Journal of Control* **41**, 461–476.
- Garcia, G., J. Daafouz and J. Bernussou (1996). Output feedback disk pole assignment with positive real uncertainty. *IEEE Transactions on Automatic Control* **AC-41**, 1385–1391.
- Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.
- Halpern, M. E., R. J. Evans and R. D. Hill (1995). Pole assignment with robust stability. *IEEE Transactions on Automatic Control* **AC-40**, 725–729.
- Impram, Serkan T., Miguel Ayala Botto and José Sá da Costa (2001). Robust pole placement controller design for the flexible transmission system. *Submitted to the European Journal of Control*.
- Keel, L. H., S. P. Bhattacharyya and J. W. Howze (1988). Robust control with structured perturbations. *IEEE Transactions on Automatic Control* **33**, 66–78.
- Kim, B. S., H. S. Han and J. G. Lee (1996). Pole placement of uncertain discrete systems in the smallest disk by state feedback. In: *Proceedings of the IEEE Conference on Decision and Control*. Vol. 4. Kobe, Japan. pp. 4558–4563.
- Ogata, K. (1987). *Discrete-Time Control Systems*. Prentice-Hall, Englewood Cliffs, NJ.
- Soh, Y. C. (1989). Robust pole-placement for uncertain interval systems. *IEE Proceedings: Control Theory and Applications* **136**, 301–306.
- Soh, Y. C., R. J. Evans, I. R. Petersen and R. E. Betz (1987). Robust pole assignment. *Automatica* **23**, 601–610.
- Solak, M. K. (1994). Robust pole-placement in discrete-time systems. In: *Proceedings of the American Control Conference*. Vol. 1. Baltimore, MD. pp. 756–760.
- Solak, M. K. and A. C. Peng (1995). A note on robust pole placement. *IEEE Transactions on Automatic Control* **AC-40**, 181–184.
- Söylemez, M. T. (1999). *Pole Assignment for Uncertain Systems*. UMIST Control Systems Centre Series, Research Studies Press, Baldock, UK.
- Söylemez, M. T. and N. Munro (1997). Robust pole assignment in uncertain systems. *IEE Proceedings: Control Theory and Applications* **144**, 217–224.

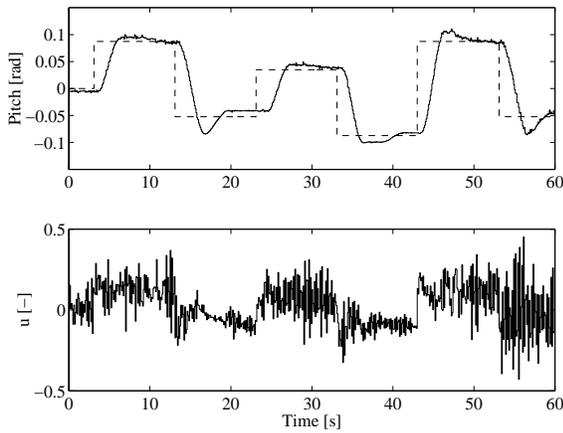


Fig. 8. Control performance of the standard pole placement controller.

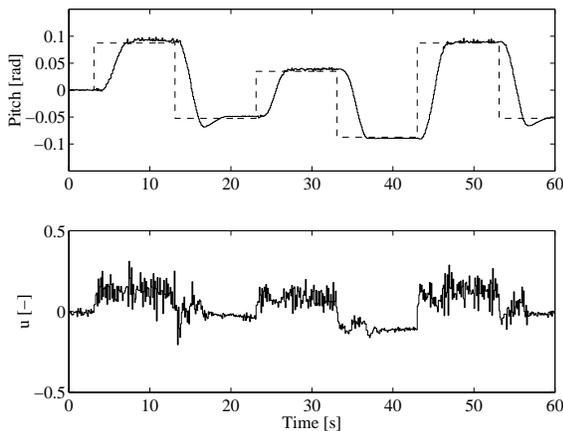


Fig. 9. Control performance of the robust pole placement controller.

ure 9, while providing a more robust controller design. Moreover, a reasonable tracking performance could now be achieved, with small overshoots and very small steady-state error. Note that the noisy behavior observed for positive setpoint values is due to a peculiar noise characteristic of the pitch angle sensor.

## 5. CONCLUSIONS

This paper presented a robust pole-placement design technique for discrete-time systems with uncertain parameters. The controller design makes use of the concept of pole pairing to compute the controller coefficients by means of the minimization of an appropriately defined objective function. For this purpose, genetic algorithms have been employed as the optimization technique. Experimental real-time results are presented for a laboratory-scale helicopter system, showing the effectiveness and potential of this approach in comparison with the standard pole placement controller.