Optimal strategy for control a pilot plant reactor

PAULO SALGADO[1], PAULO AFONSO[2] and MANUELA CUNHA[3]
Departamento das Engenharias
[1]CITAB- Universidade de Trás-os-Montes e Alto Douro
Quinta dos Prados, 5001-801 Vila Real
[2]ESTGA - Universidade de Aveiro
Apartado 473, 3754-909 Águeda, Portugal
[3]Escola S/3 S. Pedro
Rua Morgado Mateus, 5000-455 Vila Real
PORTUGAL

Abstract: - In this paper, a real-time optimal control technique for control the liquid level and temperature in a continuous stirred tank reactor (CSTR) in a pilot plant is proposed. The control system makes use of the Pontryagin's Minimum Principle and from a new back propagation algorithm of the final co-state error, allowing the training of an adaptive fuzzy inference system to estimate values for the optimal co-state variables. These strategies are employed for designing approximate optimal controllers via a control variable discretization. This approach allows the on-line solution of the optimal control problem by training the system, which can be used on a close-loop control strategy. The control system design is developed by minimizing a quadratic performance index selected for the desired operating conditions. The control simulations results are presented.

Key-Words: - Neuro-fuzzy controller; Optimal control; Learning automatic

1 Introduction

Recent technological innovations have caused a considerable interest in the study of dynamical processes and its optimization. In industry, the aim of control depends on economical reasons. In many practical situations, an optimal controller can minimize (or maximize) certain desired criterion and satisfy some physical constraints at the same time.

During last decades, this problem was been widely studied in theoretical and numerical perspectives. In principle, there are two numerical classes of methods to treat problems of optimal control, the first one which uses the Pontryagin maximum (or minimum) principle to derive the necessary conditions for the optimizer. These conditions take the form of a multipoint boundary-value problem for the state and the additional adjoint equation. The second class includes the so-called direct methods. These methods have in common in the first step a discretization transforms of an infinite-dimensional optimal control problem into a finite-dimensional optimization task, also called an NLP problem. The latter is solved usually by some optimization code, e.g. an SQP solver.

In many physical and engineering systems, engineers are hindered by strong nonlinearity from application of linear control theory. In these cases, the problem of designing optimal controllers reduces to solving algebraic Riccati equations (AREs), which solution are usually easy to solve [1]. Moreover, for nonlinear systems, the optimization problem can be stated as nonlinear PDEs - Hamilton-Jacobi (HJ) [2], which are usually hard to solve.

Despite the relative success of these optimal control strategies for knowledge base processes, these methods when applied to real control systems have a weak performance due to model mismatch and unknown disturbances.

In the past few decades, as the interest in fuzzy systems (FS) has increased, researchers have considered the stability analysis of these fuzzy systems using a variety of modeling and control frameworks.

The popularity of FS models arises not only from its simplicity, but essentially from the fact of an alternative approach to traditional nonlinear modeling.

Additionally, the fuzzy inference systems emerged as one of the most useful approaches to collect human knowledge and expertise on control and to transform the collected knowledge into a basis for developing controllers [3].

A fuzzy logic controller is usually a fuzzy inference system establishing a static mapping from the state variables input values to the actuators output values [3].

Co-state variables play a key role in finding the optimal control when using Pontryagin's Minimum Principle (PMP). However, looking at the problem on fuzzy logic grounds, the co-state variables appears to behave like the output of an expert system that knows which sequence of values will minimize the cost function.

In this approach, the (adaptive) fuzzy inference system can be used to generate actuator values, but its primary function is to generate estimates of the co-state variables.

A number of stable and optimal fuzzy controllers were developed for linear systems by using the PMP with quadratic cost function. Wang [4] developed the optimal fuzzy controller for linear time-invariant systems. Based on the conventional linear quadratic optimal control theory, Wu and Lin [5] presented a design method of the optimal controllers for continuous- and discrete-time fuzzy systems. Later, Wu and Lin [6] developed a design scheme of the optimal fuzzy controller under finite- or infinite-horizon by using the calculus-of-variation method. In these works, a local and global approach for stable fuzzy controller design methods for both continuous and discrete-time fuzzy systems under both finite and infinite horizons using traditional linear optimal control theory is presented.

In this paper, several computational techniques for real-time optimal control of non-linear systems are presented. The control policies make use of the Pontryagin's Minimum Principle and the temporal backpropagation. These techniques are combined to improved the feedforward supervised Fuzzy System of the state co-variables of the optimal problem.

This study considers the application of optimal control strategy to an experimental pilot plant reactor apparatus. It involves a continuous stirred tank reactor (CSTR) with a capacity of 80 liters, fitted with a cooling jacket and a hydraulic stirring system. This pilot plant has been the subject of several studies and was used as a benchmark to test control algorithms and other tools [7].

The purpose of this work is to implement an optimal control algorithm, using the learning optimal strategy as described in more detail in [8].

In this article, a Fuzzy Model (FM) is used as the nonlinear controller of co-state variable of optimal control problem. The FM is trained to directly minimize the performance index subjected to plant outputs, states and inputs. The optimization is carried out using a gradient scheme that is computed employing the recently developed concept of convergence of state and co-state optimal trajectories.

2 The Fuzzy Inference System

A fuzzy controller, as well as a FIS, comprehends the following three elements: the membership functions, which fuzzify the physical inputs, the inference engine, with a rule base decision; and the defuzzifier, that converts the fuzzy control decisions into physical control signals.

Various structures and learning algorithms are capable of implementing fuzzy inference engine and can be used as co-state variable controller. Without any loss of generalization, the fuzzy system uses the singleton fuzzifier, the product inference engine and the centre-average defuzzifier.

Consider a system $y = f(\boldsymbol{x})$, where y is the output variable and $\boldsymbol{x} = [x_1, \cdots, x_n]^T \in R^n$ is the input vector. Let $\boldsymbol{U} = [\alpha_1, \beta_1] \times \cdots \times [\alpha_n, \beta_n]$ be the domain of input vector $\boldsymbol{x}$. The problem to solve is the following: consider the input-output data pairs $(\boldsymbol{x}_k, y_k)$, $k = 1, 2, \cdots, n_p$, where $\boldsymbol{x}_k \in \boldsymbol{U}$ and $y_k \in V = R$. Our objective is to design a fuzzy system $g(\boldsymbol{x})$ based on these input-output pairs that approximates the unknown function $f(\boldsymbol{x})$, based on collected data points.

The relationships in the FS are based on a collection of if-then rules of the type:

$$R_{i_1,\cdots,i_n} : \text{IF } x_1 \text{ is } A^1_{i_1} \text{ and } \cdots \text{and } x_n \text{ is } A^n_{i_n} \text{ THEN } y \text{ is } C_{i_1,\cdots,i_n} \quad (1)$$

where $A^j_{i_j}$ in U_j and $C_{i_1 \pi, \cdots, i_n}$ in V are linguistic terms characterized, respectively, by fuzzy membership functions $A^j_{i_j}(x_j)$ and $C_{i_1,\cdots,i_n}(y)$, and the index set is defined by

$$I = \{ i_1, i_2, \cdots, i_n \mid i_j = 1, 2, \cdots, N_j;\ j = 1, 2, \cdots, n \} \quad (2)$$

Fuzzy system $g(\boldsymbol{x})$ is constructed through the following steps:

Step1: Partition of the input space — For each j, $j = 1, \cdots, n$ define N_j fuzzy sets in $[\alpha_j, \beta_j]$ using the following triangular membership functions: $A^j_r(x_j) = \mu(x_j; \overline{x}^{r-1}_j, \overline{x}^r_j, \overline{x}^{r+1}_j)$, for $r = 1, \cdots, N_j$. After completing this task, the domain space is partitioned by a grid of triangular membership functions. The antecedent of fuzzy rule $R_{i_1,\cdots,i_n}$ is a fuzzy set $A_{i_1,\cdots,i_n} = \times^n_j A^j_{i_j} \in \boldsymbol{U}$, with the membership functions $A_{i_1,\cdots,i_n}(\boldsymbol{x}) = A^1_{i_1}(x_1) * \cdots * A^n_{i_n}(x_n)$, where * is the min or product T-norm operator and $i_1, \cdots, i_n \in I$.

Step 2: Learning of the Rule base — For each antecedent, with index $i_1,\cdots,i_n \in I$, find the subsets of the training data where the membership function $A_{i_1,\cdots,i_n}(\boldsymbol{x})$ is not null. If the number of points found is not zero, then rule $R_{i_1,\cdots,i_n}$ is added to the rule base, represented by a table of indexes: $RB=\left\{i_1,\cdots,i_n \in I : A_{i_1,i_2,\cdots,i_n}(x_k)>0, k=1,\cdots,n_p\right\}$.

Step 3: The fuzzy system —The fuzzy system can thus be represented by:

$$g(\boldsymbol{x},\boldsymbol{\theta})=\sum_{l=1}^{M} A_l(\boldsymbol{x})\cdot\boldsymbol{\theta}_l \Big/ \sum_{l=1}^{M} A_l(\boldsymbol{x}) \qquad (3)$$

where, θ_l is the point in *V* at which $C_l(y)$ achieves its maximum value and $l\in RB$ is the index of the rule and *M* is the total number of fuzzy rules.

3 The Optimal Control Algorithm

Consider the nonlinear discrete dynamic system as:

$$x_{k+1}=g(x_k)+h(x_k)u_k \qquad (4)$$

where $g:\mathbb{R}^n\to\mathbb{R}^n$ and $h:\mathbb{R}^n\to\mathbb{R}^{n\times m}$ are continuous over $\mathbb{R}^n$. Assume that $g_k+h_k u_k$ is Lipschitz continuous on a set $U\in\mathbb{R}^n$ containing the origin, and that system is stabilized in the sense that there exists a continuous control on *U* that asymptotically stabilizes the system. It is desired to find a sequence of u_k, which minimizes the cost function:

$$J=\Phi(N,\boldsymbol{x}_N)+\sum_{k=1}^{N-1} L^k(\boldsymbol{x}_k,\boldsymbol{u}_k) \qquad (5)$$

where $L^k(\boldsymbol{x}_k,\boldsymbol{u}_k)=(\boldsymbol{x}_k-\boldsymbol{r}_k)^T R_k(\boldsymbol{x}_k-\boldsymbol{r}_k)+u_k{}^T Q_k u_k$ and $\Phi(N,\boldsymbol{x}_N)=(\boldsymbol{x}_N-\boldsymbol{r}_N)^T R_N(\boldsymbol{x}_N-\boldsymbol{r}_N)$. r_k is the desired state at *k* sample, R_k and Q_k are matrices that allow to weight attainment of the desired state versus control effort. The control problem is to find the control sequence $\boldsymbol{u}_k^*$ that minimizes the above criterion or cost function *J*, in the interval [1,*N*]. This solution is found by solving simultaneously the following equations:

$$x_{k+1}=\frac{\partial H^k}{\partial\lambda_{k+1}}=f^k(x_k,u_k) \qquad (6)$$

$$\lambda_k=\frac{\partial H^k}{\partial x_k}=\left(\frac{\partial f^k}{\partial x_k}\right)^T\lambda_{k+1}+\frac{\partial L^k}{\partial x_k} \qquad (7)$$

$$0=\frac{\partial H^k}{\partial u_k}=\left(\frac{\partial f^k}{\partial u_k}\right)^T\lambda_{k+1}+\frac{\partial L^k}{\partial u_k} \qquad (8)$$

Limit conditions equations:

$$\left(\frac{\partial L^0}{\partial x_0}+\left(\frac{\partial f^0}{\partial x_0}\right)^T\lambda_1\right)^T \mathrm{d}x_0=0;\left(\frac{\partial\Phi}{\partial x_N}-\lambda_N\right)^T \mathrm{d}x_N=0 \qquad (9)$$

where $H^k=L^k+\boldsymbol{\lambda}_{k+1}^T\cdot\boldsymbol{f}^k$ is the Hamiltonian functions and $\boldsymbol{\lambda}_k\in\mathbb{R}^n$ is a vector of Lagrange multipliers. Accordingly to common usage one will designate $\boldsymbol{\lambda}_k$ as the co-state variables.

Now, (7)-(9) can be rewritten:

$$\lambda_k=A_k\lambda_{k+1}+R_k(x_k-r_k) \qquad (10)$$

$$u_k=-Q_k^{-1}B_k\lambda_{k+1}, \qquad (11)$$

$$\lambda_N=R_N(x_N-r_N) \qquad (12)$$

where $A_k=\left(\partial f^k/\partial x_k\right)^T$ and $B_k=\left(\partial f^k/\partial u_k\right)^T$.

Given (9), (12) allows one to write the control variable at time *k* as:

$$u_k=-Q_k^{-1}h(x_k)\lambda_{k+1} \qquad (13)$$

Now the approach proposed in this paper may be made explicit. One takes λ_{k+1} as the output of a fuzzy inference system $\boldsymbol{\Lambda}$ that at instant *k* generates an estimate of λ_{k+1}^*, having as inputs the observed state x_k and the time to go *N-k*: $\lambda_{k+1}=\hat{\lambda}_{k+1}^*=\boldsymbol{\Lambda}(x_k,N-k)$. From equation (10) and (17) we have

$$\boldsymbol{x}_{k+1}=\boldsymbol{g}_k-H_k\boldsymbol{\lambda}_{k+1} \qquad (14)$$

where $H_k=h(x_k)Q_k^{-1}h^T(x_k)$.

In order to have no violation of the control variables constrains in *k* sample, the Q_k matrix has its values strong increased to penalize the violations.

In general, finding solutions is not an easy task due to the equations interdependence, which implies the used of forward and backward time sequences.

If, by adaptation or learning through fuzzy inference system, along successive runs or training iterations of the system from $k=0$ to *N*, the estimates converge to the optimal ones, then any of the control laws becomes optimal. This can be done and is the subject of the next section.

4 The learning algorithm

From (12) it follows that for optimal trajectories one must necessarily have $\lambda_N^*=R_N(x_N^*-r_N)$.

Theorem: Let $E=\lambda_N-R_N(x_N-r_N)$ be the error or difference between the end value of the state and the co-state variable trajectories. So, it is a necessary condition for the trajectories $x^*(k)$ and $\lambda^*(k)$ to be optimal: $E = 0$.

It is also possible to prove that this is a sufficient condition. If $E \to 0$, then $x_k \to x_k^*$ and $\lambda_k \to \lambda_k^*$, i.e. the trajectories of the state and co-state variables converge to the optimal ones. It follows that to attain optimal state trajectories, it is necessary that the error E converge to zero. This objective is achieved by adjusting the final λ_N co-state variables in order to minimize:

$$E^2 = \left(\lambda_N - R_N\left(x_N - r_N\right)\right)^T \left(\lambda_N - R_N\left(x_N - r_N\right)\right) \quad (15)$$

The gradient descent algorithm was employed to determine the adjustments to the final co-state value:

$$\lambda_N^{q+1} = \lambda_N^q - 2\alpha E^q \frac{\partial E^q}{\partial \lambda_N^q} \quad (16)$$

where, $q = 0,1,2,\ldots$ is the training iteration number and α is a scalar step-size variable.
For all q:

$$\frac{\partial E}{\partial \lambda_N} = I - R_N \frac{\partial x_N}{\partial \lambda_N} \quad (17)$$

where I is the identity matrix.

The summands at the right side of (17) can be solved iteratively as:

$$\frac{\partial x_{k+1}}{\partial \lambda_N} = A_k \frac{\partial x_k}{\partial \lambda_N} - H_k \frac{\partial \lambda_{k+1}}{\partial \lambda_N} \quad (18)$$

$$\frac{\partial \lambda_k}{\partial \lambda_N} = \frac{\partial \lambda_k}{\partial \lambda_{k+1}} \frac{\partial \lambda_{k+1}}{\partial \lambda_N} \quad (19)$$

with $\partial x_0 / \partial \lambda_N = 0$ and $\partial \lambda_N / \partial \lambda_N = I$.

From (10) and (14), the equation (19) can be rewrite as:

$$\frac{\partial \lambda_k}{\partial \lambda_N} = \left(I + R_k H_k\right)^{-1} A_k \frac{\partial \lambda_{k+1}}{\partial \lambda_N} \quad (20)$$

From the new value of λ_N^{q+1} a new backward co-state trajectory is computed trough equation (10). With the new value of co-state variable λ_k^{q+1}, for k=1,..,N, a new state trajectory is also computed. As far as $E \to 0$ the trajectories of the state and co-state variables converge to the optimal ones.

However, looking at the problem on fuzzy logic grounds, the co-state variables appears to behave like the output of an expert system that knows which sequence of values will minimize the cost function. This can be understood as a control strategy based on an adaptive fuzzy inference system, which generates at each time k, in the control time interval, an estimated value for the co-state variable at time $k+1$.

A training iteration can be defined as a sequence of control actions from $k = 0$ to $k = N$. Then, along successive training iterations the fuzzy inference system rules may be changed in order to generate estimates converging to the true optimal values of the co-state variables, tracking the adaptation of co-state variables (16). This implies the converging of the state variables values to the optimal.

5 The CSTR

This study considers the application of optimal control strategy to an experimental pilot plant reactor apparatus. It involves a continuous stirred tank reactor (CSTR) with a capacity of 80 liters, fitted with a cooling jacket and a hydraulic stirring system. Here an exothermic zero order chemical reaction, A→B, is experimentally simulated in the vessel [7].
As seen in Fig. 1, the reactor feed consists of two inlet water streams, with feed rates F_1 and F_2 and temperatures T_1 and T_2, while the outlet stream with flow rate F_3 flows by gravity through valve Vc_5.

The reactor features a jacket equipped with a spiral baffle and a hydraulic stirring system. The flow rate of the cooling fluid, F_j, with inlet temperature T_{j0} and outlet temperature T_{j2}, is determined by the aperture of the valve Vc_3. An electric resistor is used to heat supply, controlled by a PWM signal U_T. The speed of the agitator is controlled by manipulating valve Vc_4.

The total reactor mass balance is given by:

$$\frac{dh}{dT} = \frac{1}{A_r}\left(F_1 + F_2 - F_3\right) \quad (21)$$

where $A_r = \pi r^2$ [m^2] is the reactor area base, $V = V_0 + Ah$ [m^3] is the reactor liquid volume, with $V_0 = 4.2 \times 10^{-3} m^3$, and the reactor radius is $r = 0.232$ m. The dynamics of the reactor temperature can be described by

$$\frac{dT_r}{dT} = \frac{1}{\beta_r}(-Q_R + Q_G) + \eta U_T \quad (22)$$

with $\beta_r = \rho C_p V + \alpha_r$, where α_r and α_j represent the contribution of the wall and spiral baffle jacket thermal capacitances, and $\eta = 0.01$. Q_R is the rate of heat removal and Q_G is the rate of heat generated by the reaction, given by

$$Q_R = -\rho C_p \left(F_1(T_1 - T_R) - F_2(T_2 - T_r)\right) + UA(T_r - T_j) \quad (23)$$

$$Q_G = (-\Delta H_r) V k_0 e^{-E_n/(RT_r)} \quad (24)$$

Here ρ is the density, C_p is the specific heat capacity of the fluid, U is the overall heat transfer coefficient and A is the heat transfer area which is related to the liquid level and reactor radius r as: $A \simeq \pi r(r+2h)$.

The jacket features a spiral baffle made of steel. In our model, we lump the mass of this metal strip, which is spirally wound around the vessel wall. Assuming a uniform temperature, the evolution of the jacket temperature is represented by

$$\frac{dT_j}{dT} = \frac{1}{\beta_j}(\rho_j C_{pj} F_j (T_{j0} - T_j) + UA(T_r - T_j)) \qquad (25)$$

where C_{pj} is the specific heat capacity of the coolant, F_j is the coolant flow rate and $\beta_j = \rho_j C_{pj} V_j$.

A summary of the main process variables and model parameters is listed in Table 1. With these models the continuous state variables $x(t)=[h(t), T_r(t), T_j(t)]$ are predicted by Taylor ODE solver of 3th order $x_{k+1} = g(y_k) + g'(y_k)T + g''(y_k)T^2/2 + g'''(y_k)T^3/6$, where $\boldsymbol{y_k} = (x_k, u_k)^T$, $u_k = [F_3, U_T, F_j]$ is the actuator vector and T corresponds to the sampling time. The resulting discrete model is rewrite in the canonical form (4).

Table 1: Physical parameters

C_p , C_{pi}	4184.0 (J/(kg K))	T_{i0}	26.0 (ºC)
F_1	0.0 (l/min)	U	900.0 (W/(m^2 K))
F_2 , F_3	4.0 (l/min)	V_i	0.014 (m^3)
(E_A / R)	10080 (K)	α_i	7.0×10^5 (J/K)
k_0	6.20×10^{14} (mol/(m^3 s))	$(-\Delta H_r)$	33488.0 (J/mol)
T_1 , T_2	25.0 (ºC)	ρ , ρ_i	1000.0 (kg/m^3)

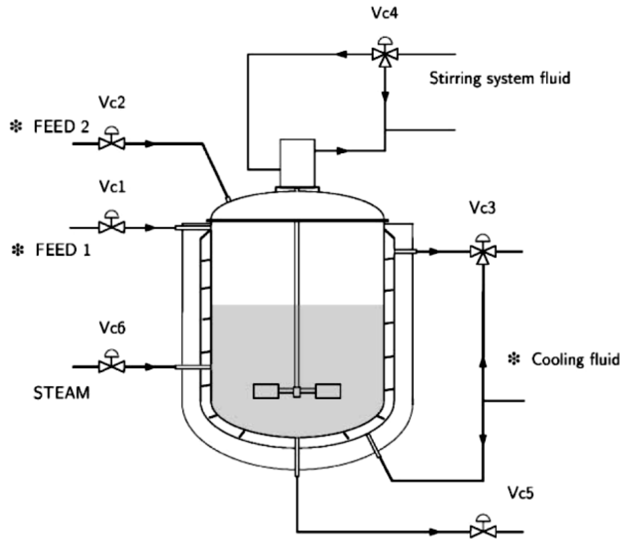

Fig. 1 – Simplified flowsheet of the pilot plant.

6 Simulation Results

The optimal co-state trajectories will be computed and will be used in the adaptation the co-state fuzzy controller. In the training phase of the fuzzy optimal controller, quadratic cost weights were $R = diag[1,1,0]$; $Q = diag[1,50,0.1] \times 10^{-2}$; $S = 4R$ and T= 10 s. A matrix Q_p is used to penalize the violation of input control constrains.

The learning process of section II was applied to implement the co-state fuzzy control, through 120 fuzzy rules (which output center parameters of fuzzy rules are adjusted). A set of simulation tests was performed by a sequence of steps change in the reference and the co-state discrete trajectory is obtained from the learning algorithm. The resulting trajectory is used to update the fuzzy control weights. This approach goes on continuously to reduce the feedforward control function error during the FM training and generates the optimal control trajectory.

Fig. 2 shown the response of state variables; the horizontal lines are the reference signals. Three trajectories are presented, which are result of three distinct optimization strategies: from the co-state fuzzy control ('*'); from discrete nonlinear optimization of co-states variables ('o') and from the numerical optimization of actuations input variables ('+'). It is possible to observe the similarity of the responses, with cost function values, respectively, 546.5, 602.5 and 602.5. In Fig. 3 are represented the co-state trajectory obtained by referred optimization processes. The actions variables are represented on Fig. 4. The actuation are limited by: $0 \le F_3 < 20$; $U_T \ge 0$ and $0 \le F_1 < 5$.

7 Conclusion

In this paper the implementation of the non-linear quadratic optimal strategy to control the liquid level and temperature in a pilot plant CSTR is presented.

The methodology applied is based on Pontryagin's Minimum Principle. A learning algorithm optimally adjusts interactively the co-state variable values and the solutions are saved in a fuzzy inference system.

The proposed method allows attaining on-line close-loop calculation of the optimal control actions. This feature makes possible to design feedback strategies more robust than the standard off-line open-loop optimal ones, subject to model mismatches and unpredictable disturbances.

Acknowledgment: This work was supported by the Portuguese "Fundação para a Ciência e a Tecnologia (FCT)" under grant POSI/SRI/41975/2001 and CITAB.

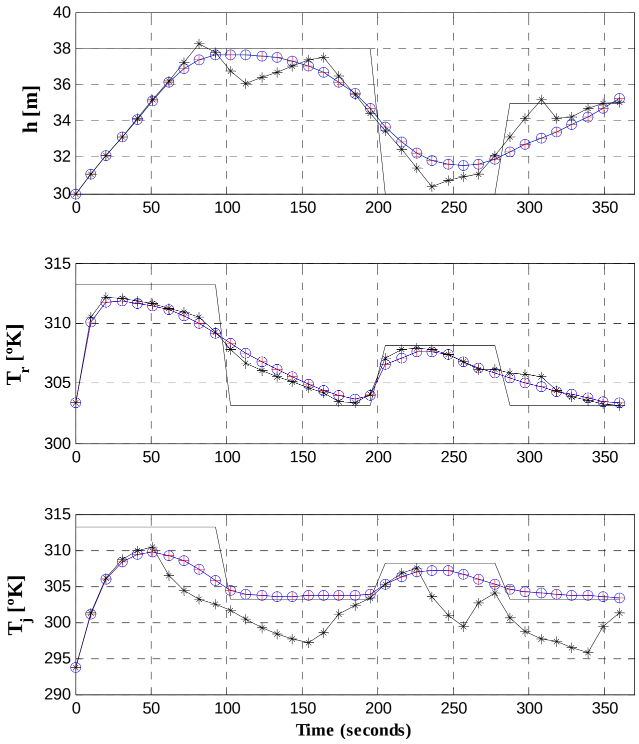

Fig. 2. : Trajectory of state variables, from: optimal fuzzy control ('*'); nonlinear optimization of co-state variables ('o') and of control inputs '+').

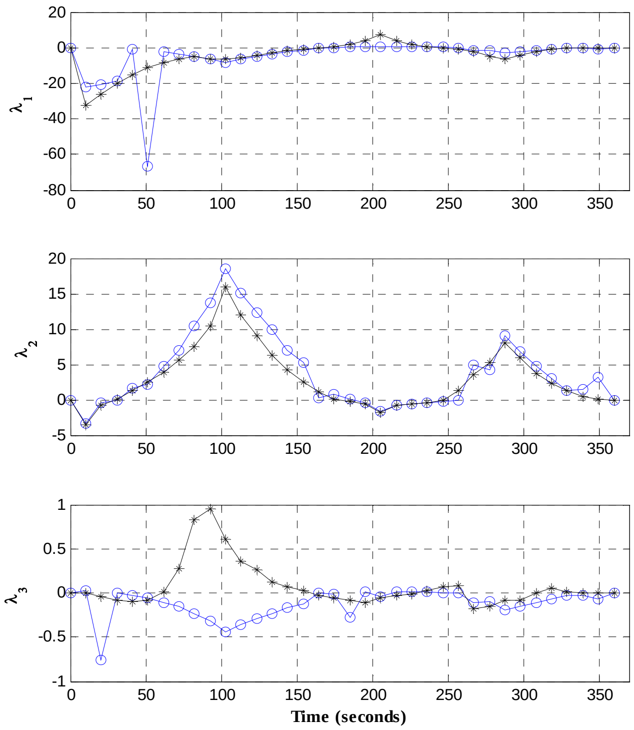

Fig. 3. : Trajectory of co-state variables computed by the proposed optimal control algorithm ('*') and by numeric nonlinear optimization ('o').

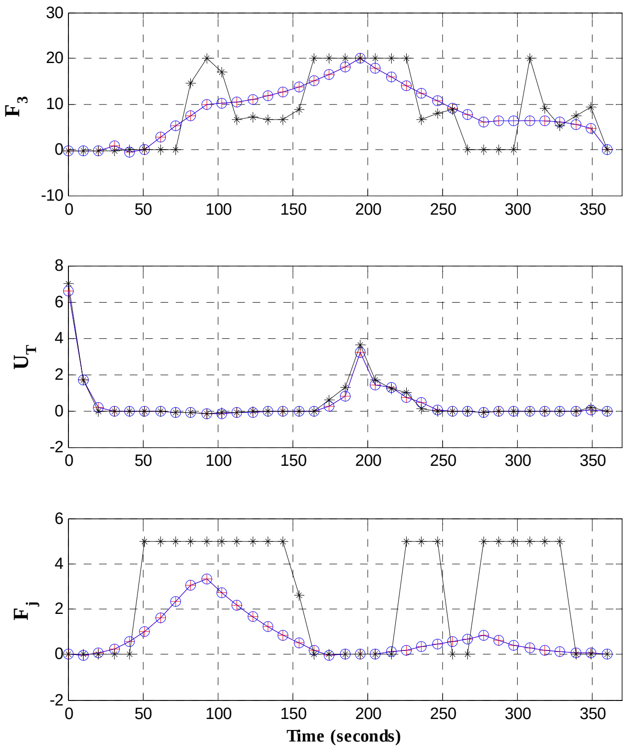

Fig. 4. : Actuations computed by: the optimal control algorithm ('*'); nonlinear optimization of co-state variables ('o') and of control inputs '+').

References:

[1] D. E. Kirk, Optimal Control Theory: An introduction, Prentice Hall, 1970.

[2] A. J. van der Schaft, "On a state space approach to nonlinear H_∞ control," Systems and Control Letters, vol.16, pp.1-8, 1991

[3] Li-Xin Wang, *A course in fuzzy systems and control*, NJ, Prentice-Hall PTR, 1997.

[4] L. X.Wang, "Stable and optimal fuzzy control of linear systems," IEEE Trans. Fuzzy Syst., vol. 6, pp. 137–143, Feb. 1998.

[5] S. J.Wu and C. T. Lin, "Optimal fuzzy controller design: Local concept approach," IEEE Trans. Fuzzy Syst., vol. 8, pp. 171–185, Apr. 2000.

[6] S. J.Wu and C.T. Lin, "Optimal fuzzy controller design: Global concept approach," IEEE Trans. Fuzzy Syst., vol. 8, pp. 713–729, 2000.

[7] Santos, L., Afonso, P., Castro, J., Oliveira, N., Bieglerb, L., "On-line implementation of nonlinear MPC: an experimental case study", Control Eng. Practice 9 (2001) 847–857

[8] Salgado. P. and at. al "Optimal closed loop control for nonlinear system using fuzzy logic", submitted for publication.