INTELLIGENT GOVERNORS FOR CONTROL OF CONSTRAINED NONLINEAR SYSTEMS

Leizer Schnitman and Takashi Yoneyama

ITA - Divisão de Eng. Eletrônica - Depto de Sistemas e Controle. São José dos Campos - SP

Abstract— In this article, the control of constrained nonlinear systems is tackled using reference governors. The classical governor is designed using a Lyapunov-like approach. It is then replaced by a new blocks which are based on artificial intelligence techniques. The objective is to allow future embedding of neurofuzzy adaptation features to accommodate uncertainties of the model.

Key Words— Constrained control, Nonlinear control, Fuzzy control, Neural control

1 INTRODUCTION

The control of nonlinear systems has received widespread attention in the last years as indicated by a number of excellent textbooks such as (Khalil, 1996), (Slotine and Li, 1991), (Sontag, 1990), (Zabczyk, 1992). Classical linearization methods work well when the model is accurate and the input reference signal is well conditioned and have low amplitude. However, this may not be the case for a general class of input commands such as large input steps or when uncertainties are considerable.

In order to treat the problem of constraints, the use of the reference governor (RG) approach is proposed (Bemporad, 1998), (Gilbert and Kolmanovsky., 1999), (McNamee and Pachter, 1999), (Miller et al., 2000), (Rossiter and Kouvaritakis, 1998), and a case of a large input step size is used as an example.

The main idea is to guarantee the constraint satisfaction for a general class of input commands while stability is assured by a local controller. An inner loop controller (ILC) is required so that it provides an adequate performance around the desired set-points but without considering the constraints. Thus, this controller is responsible for the system performance (e.g. stability or asymptotic stability) in the regions where the system is free of constraints. Therefore, if the boundaries of a specified region is not reached, then the actions of an adequate ILC shall be sufficient to assure the system stability and the RG action is not needed.

As the proposed system is nonlinear and the ILC does not consider the constraints, changes such as input steps in the reference signal $r(t)$ must be bounded and well conditioned. Thus, the RG receives the desired reference $R_d(t)$ and provides to the ILC a secondary input reference $r(t)$ as close as possible to the desired reference $R_d(t)$ but taking into account the constraints satisfaction while guaranteing that $r(t) \to R_d(t)$ when $t \to \infty$.

Because the RG theory is affected by model uncertainties, the idea is to replace it by an intelligent block using artificial intelligence techniques

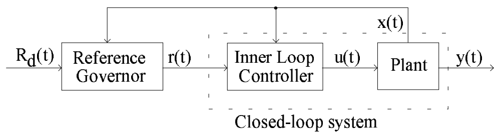

Figure 1. The Reference Governor Scheme

such as Artificial Neural Networks or Fuzzy Logic Controllers. Adaptation features are then provided and the main idea is to conceive laws that are able to fine tuning the proposed governors so as to improve the control action. Therefore, the intelligent controllers may be able to treat model uncertainties.

In the section 2 the RG theory is revisited and section 3 presents an specific structure of an ILC to be applied. Section 4 provides a practical example. Intelligent governors are proposed in section 5. Numerical results are shown in section 6 and the conclusions are presented in section 7.

2 THE REFERENCE GOVERNOR

2.1 The aim of the RG

The use of the RG requires an ILC which guarantees safe operation and adequate performance around the desired equilibrium points without considering the constraints. It means that if the constraints are not violated, the use of an adequate ILC is enough to satisfy the control objectives and the use of the RG is not required.

However, for a general class of input command such as large input step size, the ILC is unable to avoid constraints violation. Thus, when a possible constraints violation is detected, the idea is to enable the use of the RG. In order to avoid the constraints violation the RG receives the desired reference signal $R_d(t)$ produces a reference signal $r(t)$ which is supplied to the ILC, as shown in figure 1. The RG also keeps $r(t)$ signal as close as possible to the desired $R_d(t)$ but subject to constraints while guaranteeing that $r(t) \to R_d(t)$ when $t \to \infty$.

The RG design requires a function $V_r(x) \geq 0$

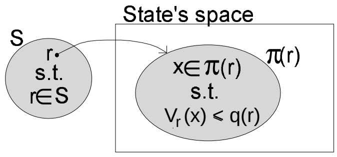

Figure 2. Mapping a reference r to a basin of safe operation

in the sense of energy.

Consider that for each instant $t = t_i$, $r(t)$ is constant and let it be equal to r and define $q(r) > 0$ to be a function which guarentees that $V_r(x)$ is bounded. It means that for each reference signal r, $q(r)$ represents an upper bound that should be computed and the constraints violation are avoided since $V_r(x) \leq q(r)$. Therefore, the RG must provides to the ILC an input reference r such that $V_r(x) \leq q(r) \;\; \forall t$.

Also, consider that the desired set-points (i.e. possible values for r) are restricted to a nonempty compact and convex set S and maps the state x according to $x(t) \in \pi(r) = \{x : V_r(x) \leq q(r)\}$. Figure 2 illustrates the main ideas.

Note that the use of an ILC is responsible to provide $\pi(r)$ as a basin of attraction, i.e. $x(t) \in \pi(r)$ such that: $0 \leq V_r(x)$, $\dot{V_r}\ (x) \leq 0$, and $V_r(x) = 0 \Longrightarrow x = x_{eq}(r)$, i.e., $\forall\ x(0) \in \pi(r)$ implies that

$$\begin{cases} x(t) \in \pi(r) \forall t \\ x(t) \rightarrow x_{eq}(r) \textit{ when } t \rightarrow \infty \end{cases} \quad (1)$$

where $x_{eq}(r)$ is an equilibrium point for the system when using r as a constant input reference signal.

2.2 The constraints

For instance, let $H(x,u) < 0$ represent the nonlinear constraint which in closed loop is characterized by $h(x,r) < 0$ and suppose that $x \in \pi(r) \Rightarrow h(x,r) < 0$. The objective is to find the largest $q(r)$ which satisfies $\pi(r)$ i.e., $\forall r \in S$ implies that $q(r) > 0$ and $x \in \pi(r) \Rightarrow h(x,r) < 0$.

2.3 The reference signal

Let $R_d(t)$ be the desired input reference, for instance, consisting of a single step:

$$R_d(t) = \begin{cases} r_i & \text{for} \;\; t \leq 0 \\ r_f & \text{for} \;\; t > 0 \end{cases} \quad (2)$$

and consider $|r_f - r_i|$ large enough. Also assume that the system is initially at the equilibrium point $x_{eq}(r_i)$. Note that there is no difficulty if $|r_f - r_i|$ is small enough, but the closed loop system can not guarantee safe operation for input steps of any size, and specially when $x_0(r_f) = x_{eq}(r_i) \notin \pi(r_f)$ as graphically described in figure 3.

Remark 1 *It is also considered that for* $t = t_i$, $R_d(t)$ *is constant and for sake of simplicity it is denoted* R_d.

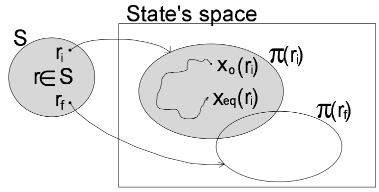

Figure 3. Example of a maping leading to unsafe operation for the closed loop system

Therefore, as shown in figure 1, for each instant $t = t_i$ the RG receives the desired reference R_d and must compute the new reference r to be supplied to the closed-loop system such that the constraints are satisfied, i.e., it computes r as close as possible to R_d but satisfying $x_0(r_f) \in \pi(r_f)$. Hence $r(t) \rightarrow R_d(t)$ when $t \rightarrow \infty$ and it guarantees that $h\,(x,r) < 0,\ \forall\ t \geq 0$.

3 A specific ILC

Before analyzing the RG, an adequate ILC should be designed. For a constant input reference signal equal to r, consider an equilibrium point $x_{eq}(r)$ and an initial condition $x_0(r+\delta)$. The ILC must guarantee that there is some $\delta > 0$ such that $x(t)$ is asymptotically stable, i.e., $x(t) \rightarrow x_{eq}(r)$ when $t \rightarrow \infty$. Hence, a basin of attraction as required in (1) is then characterized.

Since the RG simply requires a local controller, simpler controllers such as the linear ones could be used. However, the exact feedback linearization theory can yield good performance in a larger operation region (e.g. (Khalil, 1996), (Slotine and Li, 1991), (Sontag, 1990), (Zabczyk, 1992)). Therefore, it is selected in order to design a specific ILC to be used at this work.

Consider a nonlinear system of form:

$$\dot{x} = f(x) + g(x).u \quad (3)$$

Using the proposed control theory, if equation (3) can be written as

$$\dot{x} = A_e.x + B_e \beta^{-1}(x)\,[u - \alpha(x)] \quad (4)$$

where A_e is an $n \times n$ matrix, B_e is an $n \times 1$ matrix, the pair (A_e, B_e) is controlable, $\alpha : \Re^n \rightarrow \Re$ is defined in a domain $D_x \subset \Re^n$, $\beta : \Re^n \rightarrow \Re$ is defined in a domain $D_x \subset \Re^n, \beta(x) \neq 0 \;\; \forall x \in D_x$; then one can use the following control law:

$$u = \alpha(x) + \beta(x).v \quad (5)$$

Thus, using equations (4) and (5) it yields the linear expression $\dot{x} = A_e.x + B_e.v$

For the linear system, a state feedback is then used for pole placement $v = r - K.x$ where K is a gain vector, then $\dot{x} = (A_e - B_e.K)\,.x + B_e.r$, or, in an equivalent form:

$$\dot{x} = A_c.x + B_c.r \quad (6)$$

where $A_c = A_e - B_e.K$

The block diagram of the ILC scheme is shown in figure 4.

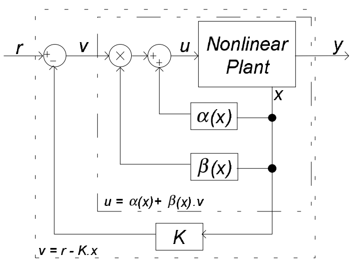

Figure 4. Inner loop controller scheme

4 APPLICATION EXAMPLE

The RG can be used with a large variety of plants and controllers. For sake of simplicity and focus on the intelligent approach of the governor, this article adopts a simple plant as an example which the nominal results basing on the nominal RG are already published in (Miller et al., 2000).

4.1 The plant

Consider a mass-spring damper, controlled by a current $i(t)$, as shown in figure 5. Tthe differential equation that govern the system has the form

$$\ddot{d}(t) = -\frac{k}{m}d(t) - \frac{c}{m}\dot{d}(t) + \frac{\alpha}{m}\frac{i(t)^\beta}{(d_0 - d(t))^\gamma} \quad (7)$$

For practical purposes, let the following values: $m = 1,54$ (mass), $k = 38,94$ (constant of elasticity), $c = 0,0659$ (damping coefficient), $i(t)$ =control variable (current), $d_0 = 0,0102$ (distance $d(t)$ for $i = 0$), $\alpha = 4,5 \times 10^{-5}$, $\beta = 1,92$, $\gamma = 1,99$.

Note that three constraints could be considered: d_{max} as the larger displacement of the mass m preventing the contact with the electromagnet; the control action only pulls the mass, it cannot push and finally the maximum current, i.e., $i(t)$ is bounded.

Choosing as the state variables $x_1 = d(t)$, $x_2 = \dot{d}(t)$ and the control signal $u(t) = i(t)^\beta$, the equation (7) can be rewritten as

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{k}{m} & -\frac{c}{m}\end{bmatrix} \cdot \begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ \frac{\alpha}{m}\frac{1}{(d_0 - x_1)^\gamma}\end{bmatrix}.u \quad (8)$$

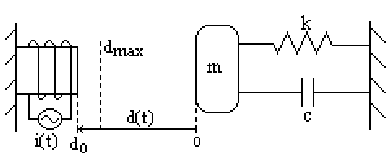

Figure 5. Plant mass-spring damper

or in an equivalent form:

$$\begin{cases}\dot{x}_1 = x_2\\ \dot{x}_2 = -\frac{k}{m}.x_1 - \frac{c}{m}.x_2 + \frac{\alpha}{m}\frac{1}{(d_0 - x_1)^\gamma}.u\end{cases} \quad (9)$$

which is clearly nonlinear.

4.2 The internal loop controller

As proposed before, the first task is to design an exact feedback linearization to some serve as the ILC. Thus, let the control law of form:

$$u(t) = \frac{(d_0 - x_1)^\gamma.(k.r - c_d.x_2)}{\alpha} \quad (10)$$

where $c_d \geq 0$ is a desired damping coefficient.

Using the control law proposed in equation (10), the expression (8) becomes:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{k}{m} & -\frac{(c+c_d)}{m}\end{bmatrix} \cdot \begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ \frac{k}{m}\end{bmatrix}.r \quad (11)$$

which is linear and stable because the constants k, m, c and c_d are always non-negative. Compare to equation (6) to rescue the practical values of A_c and B_c.

4.3 The constraints

In spite of the variety of possible constraints, the paper focuses on the physical constraint $d(t) < d_{max}$. Saturation problems with the electromagnet as well as with the current source are not considered. Different applications and detailed analysis of more complex systems are found in (Schnitman and Yoneyama, 2001a), (Schnitman and Yoneyama, 2001c), (Schnitman and Yoneyama, 2001b). In this paper, the constraint function is simply described as $h(x, r) = d(t) - d_{max} < 0$ which leads to the set $S \subset [0, d_{max}[$.

4.4 Design of the RG

A Lyapunov candidate function can be the energy function, i.e.,

$$V_r(x) = \frac{m}{2}x_2^2 + \frac{k}{2}(x_1 - r)^2 \quad (12)$$

Using (11) it can be immediately verified that $\dot{V}_r = -(c + c_d).x_2^2, \leq 0$, as expected.

As described in the last section, the main objective of the RG is to maximize $q(r)$ such that $\pi(r) = \{x : V_r(x) \leq q(r)\}$ satisfying the constraint $x \in \pi(r) \Rightarrow h(x, r) < 0$. As shown in figure 6, for a specific reference r, use the frontier $h(x, r) = 0$ to define $q(r)$, i.e. $q(r) = min_x$ $V_r(x)$ subject to $h(x, r) = 0$. In such case $q(r)$ is computed as:

$$q(r) = \frac{k}{2}(d_{max} - r)^2 \quad (13)$$

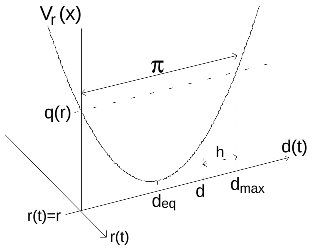

Figure 6. Graphic to analyse constraints

Define a constraint function

$$C = V_r(x) - q(r) \leq 0 \tag{14}$$

in order to consider $V_r(x) \leq q(r)$. Combining (12) and (13) one gets:

$$C = \frac{k}{2}\left(\frac{m}{k}x_2^2 + x_1^2 - d_{max}^2\right) + k.r.(d_{max} - x_1) \tag{15}$$

From $C \leq 0$, equation (15) yields $r \leq \rho(x_1, x_2)$ where

$$\rho(x_1, x_2) = \frac{1}{2}\left(d_{max}^2 - x_1^2 - \frac{m}{k}x_2^2\right) . (d_{max} - x_1)^{-1} \tag{16}$$

Finally, the RG control algorithm simply becomes:

$$\begin{cases} If \quad C \leq 0 \Rightarrow r = R_d \\ If \quad C > 0 \Rightarrow r = \rho(x_1, x_2) \end{cases} \tag{17}$$

4.5 Graphical analysis

Note that the numerical effort involved in RG is basically the computation of $\rho(x_1, x_2)$. However, it is possible to use the equation (16) to visualize the desired surface as shown in figure 7. This figure also shows the trajectories of the simulation with $d_{max} = 8 \times 10^{-3}$ and input step size from $R_d = 1$ to 7×10^{-3}. Figure 8 presents a

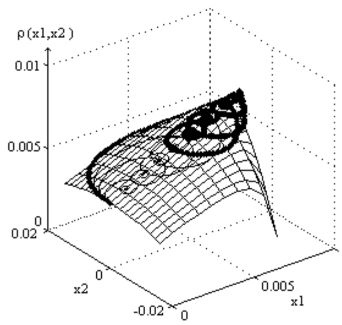

Figure 7. Desired surface and trajectories

two-dimensional plot for the same simulations. In both figures, the single dot means that the RG is idle (e.g. for small input step size) and the stronger dot means that the performance of RG is necessary to guarantee the nonviolation of the constraints.

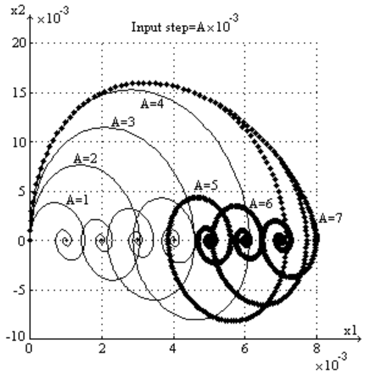

Figure 8. Phase-plane portrait under RG

Inspecting figures 7 and 8, it is possible to note that the RG is always necessary for input step size larger than $\frac{d_{max}}{2}$. In such cases (A=5,6,7), as expected, the trajectory are initially the same and differ only when the constraint boundaries become closer.

5 INTELLIGENT GOVERNORS

5.1 Input and output domains

This paper proposes to train a first order Takagi-Sugeno-Kang fuzzy structure and a neural structure in order to reproduce the nominal governor. The intelligent blocks have two inputs (x_1, x_2) and one output (ρ_f). The objective is to model the desired surface as described in the last section.

Therefore, before designing the fuzzy or neural governor it is important to define the input and output domain. In the present case the following data were found to be adequate: $x_1 = [0, d_{max} - \varepsilon]$, $x_2 = [0, 2.d_{max}]$, $\rho_f = \left[\frac{d_{max}}{2}, d_{max}\right]$; where $\varepsilon > 0$.

Once the solution surface has already been determined as shown in figure 7, a first proposition is simply to design intelligent systems which are able to reproduce it. It is important to note that the trajectories are also known. Hence, it is not necessary to reproduce the whole surface but simply the neighborhood of these trajectories.

5.2 Fuzzy Governor

For sake of simplicity, is assumed the uniform distribution of input membership functions (MF) as shown in figure 9 where $A = d_{max}$ for x_1 and $2.d_{max}$ for x_2.

In order to linearize the surface, the fuzzy rule base becomes:

$$If \quad x_1 \text{ is } MF_1^i \text{ and } x_2 \text{ is } MF_2^i \Rightarrow \rho_f = f_i(x_1, x_2) \tag{18}$$

where MF_1^i and MF_2^i are the MF associated to the antecedents of the i^{th} rule and $f_i(x_1, x_2)$ is a plane defined through the a_i, b_i and c_i parameters,

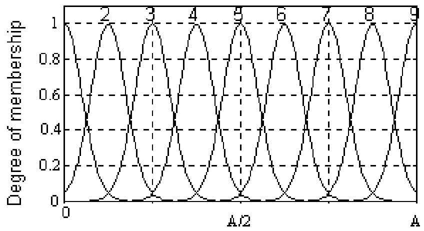

Figure 9. Uniform distribution of input MF

i.e.$f(x_1, x_2) = a.x_1 + b.x_2 + c$ which are associated to the consequence of the i^{th} rule .

Rule extraction

In order to extract the fuzzy rules, use the data pairs (x_1, x_2) provided in figure 7 and the respective desired output $\rho(x_1, x_2)$ (or simply the desired trajectory and its neighborhood) and group each pair (x_1, x_2) which active the same inputs MF_1^i and MF_2^i to define the antecedent part of each rule. For each group (rule) complete the equations:

$$\begin{bmatrix} x_1(1) & x_2(1) & 1 \\ x_1(2) & x_2(2) & 1 \\ \vdots & \vdots & \vdots \\ x_1(n) & x_2(n) & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \rho_1(x_1(1), x_2(1)) \\ \rho_2(x_1(2), x_2(2)) \\ \vdots \\ \rho_n(x_1(n), x_2(n)) \end{bmatrix} \quad (19)$$

Now any standard technique to solve the system of equation (19) can be employed. For example, use least squares to estimate the best parameters $[a_i, b_i, c_i]$. Thereby, each rule is simply represented by the best plane that fits the desired surface region. (See details in (Schnitman and Yoneyama, 2000)).

5.3 Neural governor

A simple neural network is proposed. It has a single hidden layer with six neurons (hyperbolic tangent) and one linear neuron to perform the output signal. The weight and bias are initialized randomly and trained under © Matlab using the Levenberg-Marquardt procedure for 500 epochs. The obtained network has the following data:

$$w_1 = \begin{bmatrix} -0,7459 & 0,1146 \\ -5,6212 & -0,1268 \\ 1,5185 & 0,1911 \\ 0,1572 & 0,0210 \\ 0,3564 & -0,1953 \\ -0,1632 & -0,0253 \end{bmatrix} \times 10^{-3} \; ; \; b_1 = \begin{bmatrix} 17,4235 \\ 50,1825 \\ -78,0596 \\ -2,4893 \\ 1,6330 \\ 4,2108 \end{bmatrix}$$
$$w_2 = \begin{bmatrix} -5,7883 \\ 86,6601 \\ 47,1565 \\ 0,2473 \\ 0,0004 \\ 5,6788 \end{bmatrix} \; ; \; b_2 = [39,1439] \quad (20)$$

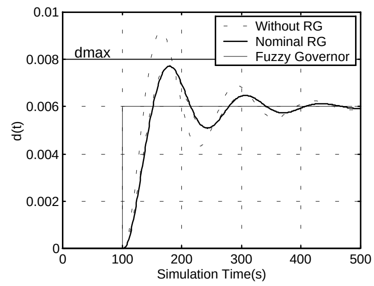

Figure 10. Fuzzy simulation results

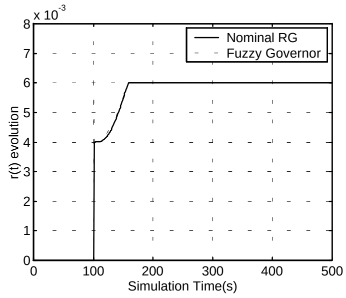

Figure 11. Fuzzy reference evolution

6 NUMERICAL RESULTS

For the practical values as defined in Section 4.1), set the constraint $d_{max} = 8 \times 10^{-3}$ and a single step to 6×10^{-3}. Figure 10 shows the simulation results using the fuzzy governor and figure 11 shows the governor 's output, i.e. how $r(t) \rightarrow Rd(t)$ for conventional and fuzzy governors. Figures 12 and 13 shows the obtained results when the neural governor is used and also comparing with the nominal governor performance.

7 CONCLUSIONS

The control of constrained nonlinear plants has many relevant practical applications. Despite the successful use of linearization methods to design nonlinear controllers, they usually do not accommodate constraints Hence, the RG approach may be a very useful tool to treat nonlinear constrained systems.

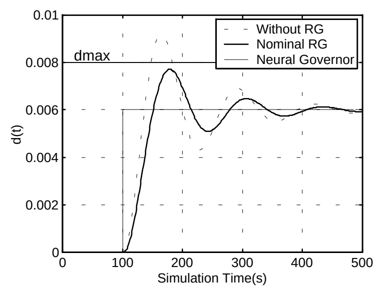

Figure 12. Neural simulation results

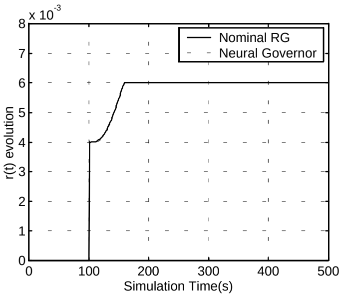

Figure 13. Neural reference evolution

On the other hand, the original RG approach is restricted to the cases where the models are well known. Thus, the proposed governors may accommodate model uncertainties and the structure is suited for addition of a neurofuzzy learning mechanism. Moreover, as described in the section 5, the rules and MF that compose the fuzzy rulebase as well as the data pair used to neural training are extracted directly from the simulation of the nominal blocks, so that the intelligent governors are able to reproduce the nominal performance for the ideal case.

The objective is now to propose adaptation laws in order to improve the controller performance. On the other hand, notice that providing an adaptation law which simply tries to minimize a cost function (usually based on the error) may be easy to be obtained; however, the learning about the constraints will probably be forgotten, unless the constraints compose the cost function. Moreover, an intelligent governor simply reproduce a desired solution surface. Hence, even if the adaptation laws be found, the mathematical formulation for its convergence and the new proof of the stability may also be hard.

ACKNOWLEDGEMENTS

Acknowledgements The authors are indebted to FAPESP for support under grant 98/16074-1.

References

Bemporad, A. (1998). Reference governor for constrained nonlinear systems, *IEEE Trans. on Automatic Control* **43**(3): 415–419.

Gilbert, E. G. and Kolmanovsky., I. (1999). Set-point control of nonlinear systems with state and control constraints: A lyapunov-function, reference-governor approach, *Proc. of the IEEE 38th Conference on Decision and Control* pp. 2507–2512.

Khalil, H. K. (1996). *Nonlinear Systems*, 2nd edn, Prentice Hall.

McNamee, J. and Pachter, M. (1999). Efficient nonlinear reference governor algorithms for constrained tracking control systems, *Proc. of the American Control Conference* pp. 3549–3553.

Miller, R. H., Kolmanovsky, I., Gilbert, E. G. and Washabaugh, P. D. (2000). Control of constrained nonlinear systems: A case study, *IEEE Control Systems Magazine* **1**(20): 23–32.

Rossiter, J. and Kouvaritakis, B. (1998). Reference governors and predictive control, *Proc. of the American Control Conference* pp. 3692–3693.

Schnitman, L. and Yoneyama, T. (2000). Extracting fuzzy rules from simulation data, *Submitted to Revista Controle e Automação FRG 321:1100* .

Schnitman, L. and Yoneyama, T. (2001a). Fuzzy control of an unstable nonlinear system under state and input constraints, *CIMA- Fuzzy Logic and Applications Symposium* .

Schnitman, L. and Yoneyama, T. (2001b). Neural control of a magnetic levitation system, *Proceedings of the 4th Nonlinear Control Network (NCN) Workshop - Nonlinear and Adaptive Control* .

Schnitman, L. and Yoneyama, T. (2001c). Um controlador fuzzy para um sistema não linear, instável e sujeito a restrições de estado e controle, *Anais do II Seminário Nacional de Controle e Automação* .

Slotine, J.-J. E. and Li, W. (1991). *Applied Nonlinear Control*, Prentice Hall.

Sontag, E. D. (1990). *Mathematical Control Theory*, Springer Verlag.

Zabczyk, J. (1992). *Mathematical Control Theory*, Birkhäuser.