Internal Model Control and Max-Algebra: Controller Design

Jean-Louis Boimond, Jean-Louis Ferrier

Internal Model Control and Max-Algebra: Controller Design

Jean-Louis Boimond and Jean-Louis Ferrier

Abstract— **This note proposes an internal model control for linear discrete-event systems over max-algebra. We shall concentrate on the controller block of this control structure.**

I. Introduction

We are interested in the control of discrete-event systems (DES's) which can be modeled by deterministic timed-event graphs (TEG's). It is well known that this particular class of graphs (a subset of the more general class of Petri nets [3], [4]) can be linearly described in max-algebra. Moreover, this algebra allows interesting concepts from conventional linear system theory to motivate the study of DES's [2], [3].

This paper deals with the following control problem: to fire at the latest date the process input so that the firing dates of the process output occur at the latest before the desired ones (described by the reference input). An attractive solution to this problem exists and is given in [2, Section X] and [3, Section 5.6] in the particular case where all the values of the desired process output are available and the model is exactly known. The control system offers a strong analogy with the adjoint system of conventional optimal control theory. Our motivation is to consider here some more general assumptions:

- Only the past values of the desired process output are available which prevents us from applying the previous solution;
- Mismatch between the process and its model can exist which leads us to consider a feedback control structure rather than the open-loop control structure used in the previous solution. We choose the internal model control (IMC) structure used in conventional control theory because it is recognized as very useful to take into account imperfect modeling.

This note is organized as follows. Section II deals briefly with the linear model representation in max-algebra. Basic IMC structure is introduced in Section III; the stability guarantee, in case of important mismatch between the process and its model, is not addressed here. Besides the model, this control structure includes a controller. Its design in max-algebra is described in Section IV. The unavailability of all the future firing dates of the controller input prevents us from having the exact solution to the previous control problem. However, in the proposed method we try to control the model in such a way that the future firing dates of the model output occur as close as possible to the ones defined by the predicted controller input. This input is the reference input modified to take into account mismatch between the process and its model. With such a procedure, the difference between the process output and the reference input depends on the prediction quality of the modified reference input. Specific problems of prediction are not discussed in this note; we only define the instants when the predicted modified reference input must be available. Section V applies the IMC design to a short example.

Manuscript received October 11, 1994; revised July 5, 1995 and September 12, 1995.

The authors are with the Laboratoire d'Ingénierie des Systèmes Automatisés (LISA), Notre Dame du Lac, 49000 Angers, France.

Publisher Item Identifier S 0018-9286(96)02258-1.

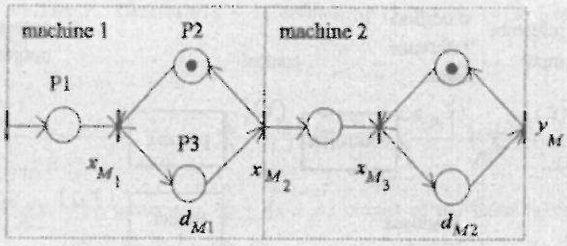

Fig. 1. A TEG.

II. Linear Algebraic Representation

Some results related to the modeling of deterministic TEG [3], [4] in max-algebra are presented in this section; for a general survey see [1]–[4].

To describe completely the behavior of a TEG, it suffices to record the sequences of its transition firing dates. We assume that transitions are fired as soon as they can be fired. For a transition labeled z, we define $z(k)$ as the date when transition z is fired for the kth time. For example, we consider two machines described by the deterministic TEG of Fig. 1.

The firing of transition u means that a part is given to the input stock (place P1) to be manufactured by machine 1. The firing of transition x_{M_1} denotes the loading of a part in machine 1 when this one is free (token in place P2). The holding time d_{M1} associated with place P3 indicates the working time of machine 1. The firing of transition x_{M_2} means that machine 1 has just completed its work. Machine 2, which can manufacture a part in d_{M2} seconds, works like machine 1. We state that

$$x_{M_1}(k+1) = \max\{x_{M_2}(k), u(k+1)\}$$
$$x_{M_2}(k+1) = d_{M1} + x_{M_1}(k+1).$$

For x_{M_1} to be fired $k+1$ times, it is necessary that u also be fired $k+1$ times, whereas x_{M_2} needs only to be fired k times since one token is already available in place P2. On the other hand, if x_{M_1} produces a token at a time t, this token will not be available before $t + d_{M1}$ for use by x_{M_2}. The max operation reflects the behavior of the connections. Finally, the equality results from the assumption that transitions are fired immediately when they can be fired.

Operations max and + are written in max-algebra as $\oplus$ and $\otimes$, respectively, to clearly underline the linearity of a deterministic DES. The elements of max-algebra are the real numbers and minus infinity (denoted ε). Zero is denoted e to refer to the unity element of $\otimes$. More generally, the model we consider is described by the following linear equations in max-algebra [3]:

$$\begin{cases} \underline{x}_M(k+1) = A \otimes \underline{x}_M(k) \oplus B \otimes \underline{u}(k+1) \\ \underline{y}_M(k) = C \otimes \underline{x}_M(k). \end{cases} \quad (1)$$

The model state (denoted $\underline{x}_M$) is the n-vector $(x_{M_1} \ \cdots \ x_{M_n})^t$. The control (denoted $\underline{u}$) is the p-vector $(u_1 \ \cdots \ u_p)^t$, and the model output (denoted $\underline{y}_M$) is the q-vector $(y_{M_1} \ \cdots \ y_{M_q})^t$. The matrices A, B, C have appropriate dimensions with entries in $\Re \cup \{\varepsilon\}$.

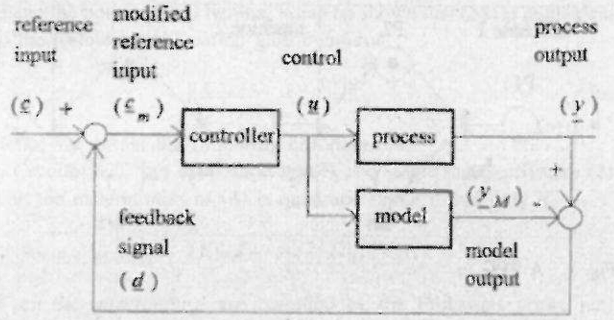

Fig. 2. Basic IMC structure.

As an example, the deterministic TEG of Fig. 1 is described in max-algebra by

$$\begin{cases} \underline{x}_M(k+1) = \begin{bmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & d_{M1} & \varepsilon \\ \varepsilon & d_{M1} & d_{M2} \end{bmatrix} \otimes \underline{x}_M(k) \\ \qquad \oplus \begin{bmatrix} e \\ d_{M1} \\ d_{M1} \end{bmatrix} \otimes u(k+1) \\ y_M(k) = [\varepsilon \quad \varepsilon \quad d_{M2}] \otimes \underline{x}_M(k) \end{cases} \tag{2}$$

where $x_M = (x_{M_1} \quad x_{M_2} \quad x_{M_3})^t$ with $\underline{x}_M(0) = (\varepsilon \quad \varepsilon \quad \varepsilon)^t$. Such an initial model state means that tokens of the initial marking are available immediately.

Henceforth, the sign $\otimes$ is omitted as in usual linear algebra.

III. Basic Internal Model Control

An open-loop control structure cannot guarantee to keep the process output close to the reference input in case of mismatch between the process and its model (always present in practice). To solve this problem, the open-loop control structure is improved. The resulting control, whose structure is depicted in Fig. 2, is called the basic IMC [5].

The modified reference input (denoted $\underline{c}_m$) is equal to the reference input (denoted $\underline{c}$) minus the difference between the process and model outputs (denoted $\underline{y}$, $\underline{y}_M$, respectively). This difference (denoted $\underline{d}$ and called the feedback signal) is due to an imperfect process modeling, since the same control $\underline{u}$ is applied to both process and model. With such a control structure, the difference between the process output and the reference input is also equal to the one between the model output and the modified reference input because $\underline{c}_m = \underline{c} - (\underline{y} - \underline{y}_M)$. Hence, to keep the process output as close as possible to the reference input, the model output must behave as closely as possible to the modified reference input which means that the controller is designed to be as close as possible to the inverse of the model (limitation is principally due to the necessary causality of the controller).

IV. Basic Internal Model Control in Max-Algebra

We try to control the firing dates of the discrete-event process output ($\underline{y}$) by firing control ($\underline{u}$) at appropriate instants. The reference input ($\underline{c}$) denotes the desired firing dates of the process output. The sequences of these firing dates are naturally nondecreasing, moreover only the firing dates of the reference input which occur before the instant when the control is computed are supposed available to compute the control.

In the previous section, we have seen that the basic IMC principle can be applied to a system when we can design a controller close to the inverse of its model. Hence, the use of this principle for DES raises the problem of model inversion in max-algebra. Due to the feedback signal design, we can note that the IMC structure induces a nonlinearity in max-algebra.

Definition (Characteristic Number): For $h = 1, \cdots, q$, let vector C_h be the hth row of matrix C, the characteristic number of the model output y_{M_h} whose behavior is described by (1); if it exists, it is the smallest integer, denoted δ_h, such that $C_h A^{\delta_h} B \neq \underline{\varepsilon}$.

In the sequel to this paper, we assume that δ_h $(h = 1, \cdots, q)$ exist.

Notation 1: Let us define

$$\Delta = \begin{bmatrix} C_1 A^{\delta_1} B \\ \vdots \\ C_q A^{\delta_q} B \end{bmatrix}$$

and

$$\Gamma = \begin{bmatrix} C_1 A^{\delta_1 + 1} \\ \vdots \\ C_q A^{\delta_q + 1} \end{bmatrix}.$$

Δ and Γ are $q \times p$ and $q \times n$ matrices, respectively; Δ_h and Γ_h are the hth row of matrices Δ and Γ, respectively.

Theorem: For $h = 1, \cdots, q$, we have

$$y_{M_h}(k + \delta_h + 1) = \Gamma_h \underline{x}_M(k) \oplus \Delta_h \underline{u}(k+1). \tag{3}$$

Proof: By definition of δ_h, we have

$$C_h B = \cdots = C_h A^{\delta_h - 1} B = \underline{\varepsilon}$$

and

$$\Delta_h \neq \underline{\varepsilon}.$$

Hence, (1) implies that

$$y_{M_h}(k + j) = C_h A^j \underline{x}_M(k)$$

for

$$j = 0, \cdots, \delta_h$$

then

$$\begin{aligned} y_{M_h}(k + \delta_h + 1) &= C_h A^{\delta_h} \underline{x}_M(k+1) \\ &= \Gamma_h \underline{x}_M(k) \oplus \Delta_h \underline{u}(k+1). \end{aligned}$$

Remark 1: The first δ_h firing dates of the model output y_{M_h} $(h = 1, \cdots, q)$ only depend on initial model state. We have

$$y_{M_h}(j) = C_h A^j \underline{x}_M(0)$$

for

$$j = 0, \cdots, \delta_h.$$

According to (3) we see that control $\underline{u}(k+1)$ can influence at the earliest the future model output $y_{M_h}(k + \delta_h + 1)$ $(h = 1, \cdots, q)$. Hence, the design of a controller that is as close as possible to the model inverse requires the knowledge of the desired future model output, i.e., the future modified reference input $c_{m_h}(k + \delta_h + 1)$ $(h = 1, \cdots, q)$ when control $\underline{u}(k+1)$ is calculated. But these values are unknown since they depend on both future reference input $c_h(k + \delta_h + 1)$ $(h = 1, \cdots, q)$ (not available by assumption) and future feedback signal $d_h(k + \delta_h + 1)$ $(h = 1, \cdots, q)$, i.e., control $\underline{u}(k+1)$. Hence, the controller we propose is split into two blocks called Prediction and Inversion (see Fig. 3).

Control $\underline{u}(k+1)$ is calculated in the Inversion block so that the future firing dates of the model output $y_{M_h}(k + \delta_h + 1)$ $(h = 1, \cdots, q)$ occur as close as possible to the ones defined by the predicted modified reference input. Let us note that in practice the control components are not necessarily calculated at the same instant

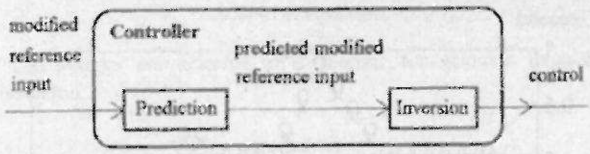

Fig. 3. Controller.

because their occurrences can be different. Therefore, it is interesting to estimate the future modified reference input $c_{m_h}(k+\delta_h+1)$ $(h=1,\cdots,q)$ in the Prediction block at each calculation of a control component $u_i(k+1)$ to have all the possible past values of the modified reference input. For $i=1,\cdots,p$, the values of the predicted modified reference input, which are used when the control $u_i(k+1)$ is calculated, are denoted by

$$\underline{c}_m^{p[u_i(k+1)]} = \{c_{m_1}^{p[u_i(k+1)]}(k+\delta_1+1) \ \cdots \ c_{m_q}^{p[u_i(k+1)]}(k+\delta_q+1)\}^t. \quad (4)$$

Practically, we compute the $k+1$th firing date of the control component u_i as soon as possible after the kth firing date of the control component u_i.

The design of the Prediction block is not the aim of this note even if the prediction quality of the modified reference input is important for the success of the model inversion method. In the example we consider in Section V, the next firing date of the reference input, used to calculate the control, is supposed to be known. Hence, only the feedback signal (d) needs to be predicted. To simply predict correctly a constant behavior of the feedback signal, we use its latest-known value when the control is calculated. It is clear that such a prediction method is rudimentary.

Let symbol $\odot$ refer to the multiplication of two matrices in which the min-operation is used rather than the max-operation [3]: for all matrices F, G with entries in $\Re \cup \{-\infty, +\infty\}$, we have

$$(F \odot G)_{ij} = \min_q \{F_{iq} + G_{qj}\}.$$

By convention, we have

$$(-\infty) \otimes (+\infty) = (-\infty)$$

but

$$(-\infty) \odot (+\infty) = (+\infty).$$

Let us recall an important result due to the Residuation theory [1]–[3]: Given a $q \times p$ matrix Φ and a q-vector $\underline{\beta}$ with entries in $\Re \cup \{-\infty, +\infty\}$, the greatest subsolution of $\Phi\underline{\alpha} = \underline{\beta}$ exists and is given by $\underline{\alpha} = (-\Phi^t) \odot \underline{\beta}$.

Corollary 1: Let Δ^i be the ith column of the matrix Δ

$$u_i(k+1) = (-\Delta^i)^t \odot \{\underline{c}_m^{p[u_i(k+1)]} \oplus \Gamma \underline{x}_M(k)\} \quad i=1,\cdots,p \quad (5)$$

occurring at the latest so that $\Delta_h^i u_i(k+1)$ occurs at the latest date before

$$\Gamma_h \underline{x}_M(k) \oplus c_{m_h}^{p[u_i(k+1)]}(k+\delta_h+1) \quad h=1,\cdots,q.$$

The proof is a direct application of the Residuation theory.

Notation 2: For $h=1,\cdots,q$, let us define l_h such that

$$\Delta_h^{l_h} u_{l_h}(k+1) = \Delta_h \underline{u}(k+1). \quad (6)$$

We can note that the possible value of l_h is not necessarily single, in this case any possible value can be considered.

Corollary 2: Control $\underline{u}(k+1)$ defined by (5) occurs at the latest so that

$$y_{M_h}(k+\delta_h+1) = \Gamma_h \underline{x}_M(k) \oplus \Delta_h^{l_h}((-\Delta^{l_h})^t \odot \{\underline{c}_m^{p[u_{l_h}(k+1)]} \oplus \Gamma \underline{x}_M(k)\}) \quad h=1,\cdots,q. \quad (7)$$

More precisely, either $y_{M_h}(k+\delta_h+1)$ occurs at the latest before $c_{m_h}^{p[u_{l_h}(k+1)]}(k+\delta_h+1)$, i.e., at

$$\Delta_h^{l_h}((-\Delta^{l_h})^t \odot \{\underline{c}_m^{p[u_{l_h}(k+1)]} \oplus \Gamma \underline{x}_M(k)\})$$

or $y_{M_h}(k+\delta_h+1)$ occurs at the earliest after

$$\Delta_h^{l_h}((-\Delta^{l_h})^t \odot \{\underline{c}_m^{p[u_{l_h}(k+1)]} \oplus \Gamma \underline{x}_M(k)\})$$

i.e., at $\Gamma_h \underline{x}_M(k)$.

Proof: From (3) and (6), and due to (5), we easily obtain (7). If we suppose that

$$\Gamma_h \underline{x}_M(k) < \Delta_h^{l_h}((-\Delta^{l_h})^t \cdot \{\underline{c}_m^{p[u_{l_h}(k+1)]} \oplus \Gamma \underline{x}_M(k)\}) = \Delta_h^{l_h} u_{l_h}(k+1)$$

from (7) it is clear that

$$y_{M_h}(k+\delta_h+1) = \Delta_h^{l_h} u_{l_h}(k+1) > \Gamma_h \underline{x}_M(k).$$

On the other hand, by using Corollary 1 we see that $\Delta_h^{l_h} u_{l_h}(k+1)$ occurs at the latest before $\Gamma_h \underline{x}_M(k) \oplus c_{m_h}^{p[u_{l_h}(k+1)]}(k+\delta_h+1)$ which implies that $y_{M_h}(k+\delta_h+1)$ occurs at the latest before $c_{m_h}^{p[u_{l_h}(k+1)]}(k+\delta_h+1) > \Gamma_h \underline{x}_M(k)$. When

$$\Gamma_h \underline{x}_M(k) \geq \Delta_h^{l_h}((-\Delta^{l_h})^t \odot \{\underline{c}_m^{p[u_{l_h}(k+1)]} \oplus \Gamma \underline{x}_M(k)\})$$

from (7) we see easily that $y_{M_h}(k+\delta_h+1)$ occurs at $\Gamma_h \underline{x}_M(k)$.

Remark 2: Control $\underline{u}(k+1)$ defined by (5) needs the model state $\underline{x}_M(k)$ which is easily available in the IMC structure.

Remark 3: In the particular case where the model is exactly known and the values of the predicted modified reference input are exact, we clearly have $c_{m_h}^{p[u_i(k+1)]} = c_h(k+\delta_h+1)$ $(i=1,\cdots,p,$ $h=1,\cdots,q)$ which yields, according to Notation 2

$$c_{m_h}^{p[u_{l_h}(k+1)]}(k+\delta_h+1) = c_h(k+\delta_h+1) \quad h=1,\cdots,q.$$

Hence, by using Corollary 2 we see that $y_h(k+\delta_h+1)$ occurs at the latest before $c_h(k+\delta_h+1)$, i.e., at

$$\Delta_h^{l_h}\left((-\Delta^{l_h})^t \odot \left\{\begin{bmatrix} c_1(k+\delta_1+1) \\ \vdots \\ c_q(k+\delta_q+1) \end{bmatrix} \oplus \Gamma \underline{x}_M(k)\right\}\right)$$

or at the earliest after this time, i.e., at $\Gamma_h \underline{x}_M(k)$. Let us recall that the latter case is avoided when the adjoint system can be used (see Section I).

Remark 4: In single-input–single-output (SISO) case, Corollary 2 is simplified. The solution of $\Phi\alpha = \beta$ exists and is given by $\alpha = (-\Phi)\beta$. Thus control $u(k+1)$ occurs at the latest so that $y_M(k+\delta+1) = \Gamma \underline{x}_M(k) \oplus c_m^{p[u(k+1)]}(k+\delta+1)$.

V. Example

We consider the two machines described in max-algebra by (2). Since $CB = d_{M1}d_{M2} \neq \varepsilon$, the characteristic number δ exists and is equal to zero; thus $\Delta = CB$ and $\Gamma = CA$. From (5), we have

$$u(k+1) = (-\Delta)\{c_m^{p[u(k+1)]}(k+1) \oplus \Gamma \underline{x}_M(k)\}$$

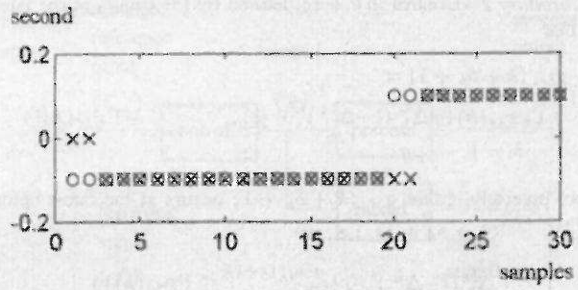

Fig. 4. Feedback signal (d) o predicted feedback signal (d^p) ×.

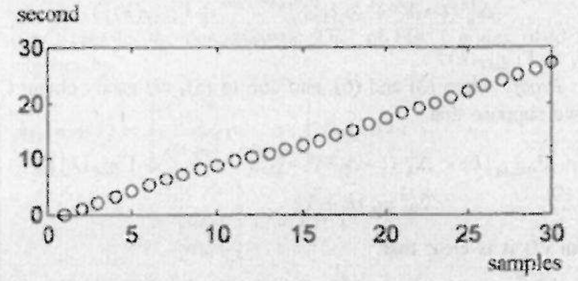

Fig. 5. Control (u) o.

with

$$(-\Delta) = (-d_{M1})(-d_{M2})$$

and

$$\Gamma = [\varepsilon \quad d_{M1} d_{M2} \quad 2d_{M2}].$$

Hence, according to Remark 4 and owing to the IMC structure (see Section III), we have

$$\begin{aligned} y(k+1) - c(k+1) &= y_M(k+1) - c_m(k+1) \\ &= \{c_m^{p[u(k+1)]}(k+1) \oplus \Gamma \underline{x}_M(k)\} \\ &\quad - c_m(k+1). \end{aligned}$$

In the simulation case, the reference input is defined by

$$\begin{aligned} c(k) &= c(k-1) + 1 && \text{for } k = 1, \cdots, 9 \text{ with } c(0) = 0 \\ c(k) &= c(k-1) + 0.5 && \text{for } k = 10, \cdots, 13 \text{ and} \\ c(k) &= c(k-1) + 1 && \text{for } k \geq 14. \end{aligned}$$

For convenience, only the manufacturing time of machine 2 (d_{M2}) is supposed to be imperfectly modeled: The value d_{M2} of the model, used for control design, is equal to 0.7 s, while the true value is equal to 0.6 s during the first 19 samples and to 0.8 s afterwards. The manufacturing time of machine 1 (d_{M1}) is exact and is equal to 0.2 s.

The reference input $c(k+1)$ is supposed to be known when control $u(k+1)$ is computed [just after $u(k)$], and hence only the feedback signal needs to be predicted. To predict it correctly when its behavior is constant, let $d^{p[u(k+1)]}(k+1) = d(j)$, where $d(j)$ is the last firing date of the feedback signal we dispose of when control $u(k+1)$ is computed.

Fig. 4 represents the behaviors of feedback signal and its prediction. Initially, the model is supposed to be perfect: Let $d^{p[u(1)]}(1) = 0$. We can observe that the change of the feedback signal at sample 20 causes a wrong prediction at samples 20 and 21 which is due to the poorness of the prediction method.

Fig. 5 represents the behavior of control. Fig. 6 represents the behaviors of the error between the model output and the reference input and of the error between the process output and the reference input.

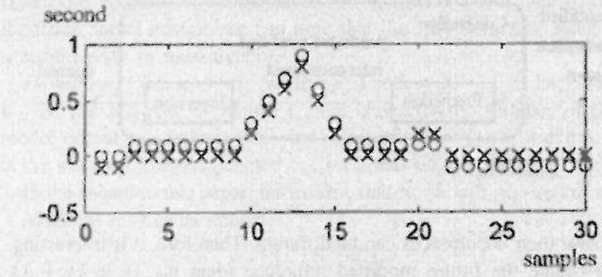

Fig. 6. Model output/reference input error $(y_M - c)$ o and process output/reference input error $(y - c)$ ×.

We observe in Fig. 5 that for samples 10–15, the control occurs as soon as possible, i.e., $u(k) = (-\Delta)\Gamma \underline{x}_M(k-1)$ for $k = 10, \cdots, 15$. Two reasons induce this behavior:

- For samples 10–13, machine 2 cannot produce enough parts to satisfy the production rate desired by the reference input. Such a problem causes a divergence between the process output and the reference input at the same samples (see Fig. 6).
- At samples 14 and 15, the machines produce parts as soon as possible to zero the error between the process output and the reference input (see Fig. 6).

On the other hand, it can be seen in Fig. 6 that the process output is equal to the reference input in spite of a manufacturing time mismatch, when for the same sample the prediction of feedback signal is exact (see Fig. 4), and the control does not depend on the model state.

VI. Conclusion

We propose the use of IMC structure to try and take into account imperfect modeling for deterministic DES's. Such systems are described by linear equations in max-algebra. In the basic IMC structure, the controller design raises the problem of model inversion because the difference between the process output and the reference input is equal to the one between the model output and the controller input, i.e., the modified reference input. Due to the IMC structure, we cannot design the controller by using the adjoint system [2, Section X], [3, Section 5.6] which is ideally adapted to an open-loop control structure. The proposed controller is based on the future (predicted) modified reference input, and it is split into two blocks, called Prediction and Inversion. Our work concerns only the Inversion block where Residuation theory plays an essential role. Firing the model output as late as possible before desired time instants is not always assured, the control firing dates occur at the latest, so that for each model output:

- If the future firing date of the model output can occur before the one defined by the predicted modified reference input, then it occurs at the latest time before the one defined by the predicted modified reference input (only at in SISO case);
- Otherwise, the future firing date of the model output occurs at the earliest after the one defined by the predicted modified reference input.

An important point is the robustness of the IMC. How can we guarantee stability in spite of the important mismatch between the process and its model? A possible approach would consist in predicting the reference input separately from the feedback signal. Hence, robustness would depend on the technique used to predict the feedback signal.

ACKNOWLEDGMENT

The authors are grateful to C. Ferrier for valuable linguistic discussion.

REFERENCES

[1] R. A. Cuninghame-Green, *Minimax Algebra* (Lecture Notes in Economics and Mathematical Systems), vol. 166. New York: Springer-Verlag, 1979.

[2] G. Cohen, P. Moller, J. P. Quadrat, and M. Viot, "Algebraic tools for the performance evaluation of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 39–58, 1989.

[3] F. Baccelli, G. Cohen, G. J. Olsder, and J. P. Quadrat, *Synchronization and Linearity. An Algebra for Discrete Event Systems*. New York: Wiley, 1992.

[4] T. Murata, "Petri nets: Properties, analysis, and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[5] M. Morari and E. Zafiriou, *Robust Process Control*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

Principle of Proportional Damages in a Multiple Criteria LQR Problem

Yuri B. Shtessel

***Abstract*—A multiple criteria linear quadratic regulator (LQR) problem is considered. The principle of proportional damages is worked out to minimize a norm of a difference between ideal and optimal values of the vector-performance criteria, while the distribution of the losses (damages) of the performance criteria are under control. A multiple criteria LQR problem solution via principle of proportional damages is obtained as a linear form of state variables and is invariant to the norm which is minimized.**

I. INTRODUCTION

The problem considered belongs to the domain of linear systems with quadratic performance criteria. This is, for example, a linearized flight control problem [1]. The set of linearized equations of the motion of a rigid aircraft can be considered as three interacted subsystems. Each subsystem describes pitch, yaw, and roll angle rates. Behavior of these angle rates can be characterized by three quadratic performance criteria. These criteria can be minimized by means of corresponding choice of control functions: aileron, rudder, and elevator positions. Apparently, each control function affects all performance criteria. Consequently, an air-vehicle flight control is a typical multiple criteria control problem. Decoupling control strategy, when initially subsystems are decoupled via decoupling control and then optimal control functions are identified in each subsystem separately, is very popular. Unlike this traditional approach, we propose to employ interconnections between subsystems to improve the performance of each subsystem. In this work we will look for the compromise minimization control solution based on a procedure that calculates Pareto optimal solutions as the approximation to an ideal point [2]–[5]. The distance between ideal and actual values of the vector-performance criteria is formed as some norm in the criteria space. This norm is usually minimized [5] to synthesize the compromised control strategy. This approach is attractive because of its mathematical and physical clarity. However, some disadvantages exist. First, it is not obvious how to specify the norm in the criteria space. The compromised solution should depend on the form of this norm. Second, minimizing the norm we do not control the distribution of the differences between ideal and optimal (compromised) values of each criterion. It can lead to unacceptable compromised behavior in some subsystems. The principle of proportional damages is worked out [6], [7] to minimize a norm of a difference between ideal and optimal values of the vector-criteria, while the distribution of the losses (damages) of each criterion is under control. A multiple criteria linear quadratic regulator (LQR) problem solution via principle of proportional damages is obtained as linear form of state variables and is invariant to the norm which is minimized.

Manuscript received January 13, 1995.
The author is with the Department of Electrical and Computer Engineering, University of Alabama, Huntsville, AL 35899 USA.
Publisher Item Identifier S 0018-9286(96)02262-3.

II. PROBLEM FORMULATION

Suppose a linear time-varying multi-input–multi-output system, consisting of N interacting subsystems, is described by the systems of the differential equations

$$\dot{x} = A(t)x(t) + B(t)u(t)$$
$$x(0) = x_0 \qquad (1)$$

where the entries of matrices $A(t) \in R^{n\times n}$, $B(t) \in R^{n\times m}$ are continuous functions of time.

Each subsystem is characterized by the quadratic performance index (criterion)

$$J_i(x, u) = \frac{1}{2}\int_0^T [x^T(t)Q_i(t)x(t) + u^T(t)R_i(t)u(t)]\,dt$$
$$\forall i = \overline{1, N} \qquad (2)$$

where time-dependent matrices $Q_i(t) \in R^{n\times n}$, $R_i(t) \in R^{m\times m}$ are positive semidefinite and positive definite, correspondingly, while the performance of (1) is characterized by the vector-criteria

$$J(x, u) = \{J_1(x, u), J_2(x, u), \cdots, J_N(x, u)\}. \qquad (3)$$

We wish to find an optimal feedback control law $u^*(t, x)$ such that (3) is minimized. We will think of (3) as minimized in the sense of the principle of proportional damages [5], [6].

III. PRINCIPLE OF PROPORTIONAL DAMAGES

One of the widespread approaches to look for the compromise feedback control law $u^*(t, x)$ is based on a procedure that calculates Pareto optimal solutions as an approximation to an ideal point J^0 which is introduced in criteria space as follows [2]–[5]:

$$J^0 = \{J_1^0, J_2^0, \cdots, J_N^0\} \qquad (4)$$

where

$$J_i^0 = J_i(x, u^i),\ J_i(x, u^i) \le J_i(x, u)$$
$$\forall i = \overline{1, N}. \qquad (5)$$

Very often a distance between (3) and (4) is represented as in the following norm:

$$\rho(J, J^0) = \left[\sum_{i=1}^{N} |J_i(x, u) - J_i^0|^p\right]^{1/p}, \quad p \in [1, \infty). \qquad (6)$$