QUANTIZED-STATE CONTROL.
A Method for Discrete Event Control of Continuous Systems. Part II

Ernesto KOFMAN†

†*Laboratorio de Sistemas Dinámicos*
FCEIA - Universidad Nacional de Rosario
Riobamba 245 bis - (2000) Rosario - Argentina
Email: ekofman@eie.fceia.unr.edu.ar

Abstract— **In this paper, the practical topics related to the design and implementation of Quantized State Controllers are treated. The basic idea about the way of programming these controllers in digital devices is introduced. The algorithm that allows the design of the controller and converters achieving regional stabilization around the equilibrium points (deduced in (Kofman, 2001)) is applied in an example, obtaining some promising simulation results. Finally, some practical issues related to the choice of the quantization, the implementation of the converters and the delays in the real implementation are discussed and some advantages of the QSC scheme over the traditional discrete time controllers related to the reduction of the computational cost are showed through another simple example.**

Keywords— **Digital Control, Quantization Effects, Discrete Event Control, DEVS.**

I. INTRUDUCTION

The traditional design of digital controllers always arrives to *discrete time* models. However, there is another class of models that, under certain conditions, can be implemented over digital devices. This class corresponds to the *discrete event* systems, within the DEVS framework (Zeigler, 1976), (Zeigler et al., 2000).

The introduction of *Quantized State Control* (QSC) in (Kofman, 2001) shows a way of designing DEVS models to be used as controllers for continuous plants. The basic idea is starting with the design of a continuous controller and then adding quantizers[1] at the output of the integrators of this controller. The new controller with the quantizers constitutes a *Quantized State System* (QSS) (Kofman and Junco, 2000) that can be exactly represented by a DEVS model. The QSC scheme also uses an asynchronous sampling technique (Sayiner et al., 1993) that only performs A/D conversions when the difference between the analogical input and the digital output is bigger than some threshold (quantization size of the converter). Figure 1 shows the block diagram of a plant with a QSC controller.

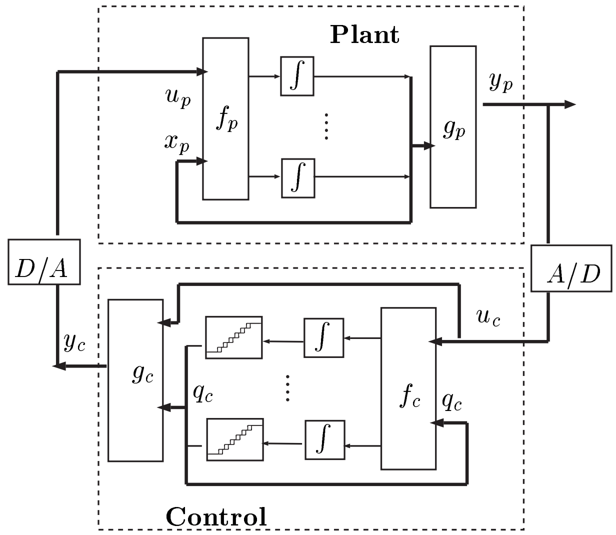

Figure 1: Block Diagram of the plant with the QSC

From the theoretical point of view, the QSC controllers show some advantages over the discrete time controllers in presence of quantization effects introduced by A/D and D/A converters. The reason of these advantages is in the fact that the QSC controllers can guarantee regional and even semiglobal "stabilization"[2] of nonlinear plants around the equilibrium points while traditional discrete time controllers only guarantee local "stabilization" (Hou et al., 1997), (Hu and Michel, 1999).

However, this result related to the quantization effects is not the only advantage of the QSC scheme. There is another advantage related to the computa-

[1] A quantizer is defined as a system whose input and output are related by a quantization function with hysteresis (Kofman, 2001) (Kofman and Junco, 2000)

[2] In presence of quantization effects, "stabilization" cannot be assured. In fact, only "ultimately boundedness" of the solutions can be guaranteed.

tional cost of the implementation that in some cases could be very important .

The aim of this paper is to discusse practical topics related to the implementation of the QSC controllers as well as introducing some examples in order to show the advantages of the scheme. In Section II the idea about the way of implementing a QSS in a digital device is explained. In (Kofman, 2001) an algorithm allowing the design of QSC controllers achieving goals related to final error, convergence speed and region of attraction was introduced. In Section III an example of the use of this procedure is developped and some simulation results are presented.

Then, in Section IV, a very simple example showing the reduction of the computational cost of the implementation is presented. Finally, in Section V, some remarks related to practical design and implementation of the converters and the errors introduced by the delays are discussed.

II. EXACT SIMULATION OF QSS

The addition of quantizers at the output of the integrators of a continuous system transforms it into a QSS. When the input variables have piecewise constant trajectories, the state trajectories are piecewise linear while the quantized variables[3] as well as the state derivatives have piecewise constant trajectories. In (Kofman and Junco, 2000) the formal definition of QSS, the proof of the mentioned properties and the DEVS model that exactly simulates a generic QSS are introduced.

Here, instead of presenting that DEVS model, the principles that rule the discrete event simulation of the QSS will be explained.

The basic idea for the discrete event simulation of the QSS is the representation by events of the changes in the quantized variables and state derivatives (it is possible because these variables have piecewise constant trajectories).

Consider a *quantized integrator*, that is, a simple integrator and a quantizer connected to its output so that the output of the system is the quantized variable $q(t)$ as Figure 2 shows.

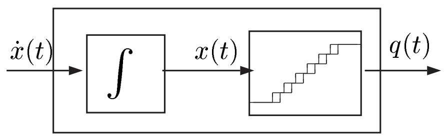

Figure 2: A Quantized Integrator

According to the properties explained above the trajectories of q and $\dot{x}(t)$ will be piecewise constant while the state variable x will have a piecewise linear trajectory. Figure 3 shows typical trajectories in a quantized integrator.

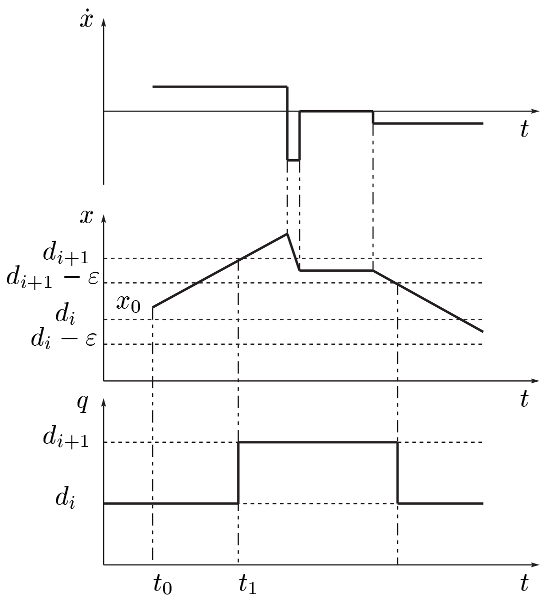

Figure 3: Trajectories in a quantized integrator

Performing a simulation of this behaviour is equivalent to predict when the variable q will change and what value it will adopt after that change. These calculations require the knowledge of the initial condition of the state variable (x_0). Due to the presence of hysteresis it is also necessary to know the initial value of the quantized variable ($q(t_0) = d_i$).

Two different cases must be taken into account in order to predict when the next change of q will occur. The first case is when there is no modification of the input value before the change of q. In this case, considering that the initial time of the simulation is t_0, the time of the next change can be calculated as:

$$t_1 = \begin{cases} t_0 + \dfrac{d_{i+1} - x_0}{\dot{x}} & \text{if } \dot{x} > 0 \\ t_0 + \dfrac{x_0 - (d_i - \varepsilon)}{\dot{x}} & \text{if } \dot{x} < 0 \\ \infty & \text{if } \dot{x} = 0 \end{cases} \tag{1}$$

where d_i and d_{i+1} are the actual and the next quantized values and ε is the width of the hysteresis window. The value of q after the change will be either d_{i+1} or d_{i-1} depending on the sign of $\dot{x}$. The new initial condition for x can be calculated as d_{i+1} or $d_i - \varepsilon$ also according to the sign of $\dot{x}$.

If there is a modification of the input value before the change of q, the situation can be treated as a change in the initial condition x_0 and a change in the initial time t_0. The increment in x_0 can be calculated as:

$$\Delta x_0 = \dot{x}(t_0) \cdot e \tag{2}$$

where e is the elapsed time from t_0 to the time of the change in the value of u.

[3]Output variables of the quantizers

The alternate use of equations (1) and (2) allows the exact simulation of the quantized integrator. In terms of the DEVS formalism, the rules explained for the first case (changes in the value of q) determine the *internal transition function* while the second case determines the *external transition function*. Equation (1) will be related with the *time advance function*. The state of the DEVS model will save the last values of x_0, d_i and $\dot{x}$.

Since DEVS allows the coupling of different models, a general QSS can be exactly simulated by a coupled DEVS structure. The model of the integrators can be built using the ideas explained above. The rest of the system, composed only by static functions, has a straightforward representation in DEVS.

The possibility of representing a QSS by a DEVS model and the fact that DEVS models can be simulated in real time (Zeigler and Kim, 1993) by a digital device[4] allow the use of QSS as digital controllers.

III. EXAMPLE OF QSC DESIGN

As it was mentioned in the introduction, one of the main advatages of the QSC scheme is the possibility of designing a controller taking into account the quantization effects of the converters and achieving regional "stabilization" of the plant around the equilibrium points and convergence in finite time to the ultimately bound.

In order to facilitate the design of the controllers achieving the mentioned goals, an algorithm based in the theorem of stability of QSC was introduced in (Kofman, 2001). The following example illustrates the use of this procedure.

Consider the plant:

$$\left\{ \begin{array}{rcl} \dot{x}_p(t) & = & x_p^2 + u_p \\ y_p(t) & = & x_p(t) \end{array} \right. \tag{3}$$

The goal is stabilizing the plant around the origin. The first step is designing a continuous controller. The following controller can achieve the mentioned goal.

$$\left\{ \begin{array}{rcl} \dot{x}_c(t) & = & -x_c - u_c \\ y_c(t) & = & x_c(t) - u_c(t) - u_c^2(t) \end{array} \right. \tag{4}$$

The resulting closed loop equations are:

$$\left\{ \begin{array}{rcl} \dot{x}_p(t) & = & -x_p + x_c \\ \dot{x}_c(t) & = & -x_p - x_c \end{array} \right. \tag{5}$$

It can be easily verified that the origin is the equilibrium point, and it is asymptotically and globally stable. By taking the Lyapunov function

$$V(x) = \frac{1}{2}x_p^2 + \frac{1}{2}x_c^2 \tag{6}$$

[4]The DEVS representation of QSS is exact. However, real time simulation of DEVS has errors related to temporal resolution and round-off introduced by the digital device

it follows that:

$$\dot{V}(x) = -x_p^2 - x_c^2 \tag{7}$$

The second step is identifying D, the region of convergence of the plant with the continuous controller. As it was mentioned before the stability is global and $D = \mathbb{R}^2$. The next step of the algorithm is chosing the region D_1 from which the trajectories will converge. Due to the global stability of the continuous closed loop system, the definition of this region will be only neccessary for the choice of the saturation bounds.

In the fourth step, the ultimately bound (D_2) must be chosen according to the desired steady state error. We will suposse that the goal is assuring the convergence of the trajectories to the region $D_2 = \{x/\|x\| < 1\}$. ($\|\cdot\|$ stands for norm 2)

The following steps are the calculation of the functions α and α_M according to:

$$\begin{aligned} &\alpha(x, \Delta x_c, \Delta y_p, \Delta y_c) = \\ &\quad \frac{\partial V}{\partial x_p}(x) \cdot f_p(x_p, g_c(x_c + \Delta x_c, g_p(x_p) + \Delta y_p) + \\ &\quad +\Delta y_c) + \frac{\partial V}{\partial x_c}(x) \cdot f_c(x_c + \Delta x_c, g_p(x_p) + \Delta y_p) \end{aligned} \tag{8}$$

and

$$\alpha_M(\Delta x_c, \Delta y_p, \Delta y_c) = \sup_{x \in D_3} [\alpha(x, \Delta x_c, \Delta y_p, \Delta y_c)] \tag{9}$$

It follows from (8), (3), (4) and (6) that:

$$\begin{aligned} &\alpha(x, \Delta x_c, \Delta y_p, \Delta y_c) = \\ &\quad -x_p^2 - x_c^2 + x_p(\Delta x_c - \Delta y_p - \Delta y_p^2 - 2x_p\Delta y_p + \\ &\quad +\Delta y_c) + x_c(-\Delta x_c - \Delta y_p) \end{aligned} \tag{10}$$

It can be seen from the equation above that $\alpha(x, 0, 0, 0) = -x_p^2 - x_c^2 = \dot{V}(x)$ as the proof of the theorem of stability in (Kofman, 2001) predicts.

The calculation of α_M according to the definition in (9) is quite difficult. However, a bound of this function can be easily obtained. It follows from (10) that:

$$\begin{aligned} \alpha(x, \Delta x_c, \Delta y_p, \Delta y_c) \leq -\|x\|^2 + \|x\|(|\Delta x_c| + |\Delta y_p|) + \\ +\|x\|(|\Delta x_c| + |\Delta y_p| + |\Delta y_p^2| + 2\|x\|\ |\Delta y_p| + |\Delta y_c|) \end{aligned}$$

Then, it results from (9) and the inequality above that:

$$\begin{aligned} &\alpha_M(\Delta x_c, \Delta y_p, \Delta y_c) \leq \sup_{\|x\| \geq 1} [-\|x\|^2 + \\ &\quad +\|x\|(2|\Delta x_c| + 2(\|x\| + 1)|\Delta y_p| + |\Delta y_p^2| + |\Delta y_c|)] \end{aligned}$$

Since outside the region D_2 the condition $\dot{V}(x) < -1$ is satisfied, the convergence speed s_1 (seventh step) can be chosen to be bounded between 0 and 1. Suppose that the choice is $s_1 = 0.5$. Then, the quantization must be chosen in order to satisfy $\alpha_M < -0.5$.

Taking quantization intervals $\Delta q = 0.07$ for all variables, the mentioned condition is achieved. The hysteresis window can be taken having the same width than the quantization interval and the relationships are still satisfied. If the restriction about the convergence speed is not taking into account and the goal is just assuring stability, that quantization interval of $\Delta q = 0.07$ is sufficiently small to guarantee convergence to the region given by $\|x\| < 0.4127$.

The simulation results of the plant with the QSC designed above, verify that the goals were achieved. The convergence region is much smaller than the region in which the convergence was assured. This is due to the fact that the conditions given in the proof of the Theorem of stability are conservative and the implementation through the calculation of the bound of α_M was also done in a conservative way.

The simulation was done for an initial condition $x_p = 10$. The results are shown in figures 4 to 8. The number conversions performed by the A/D converter were 173 for 30 seconds of simulation time. The minimun time between two successive conversions was 5.6 milliseconds (at the beginning of the simulation) while the maximum was greater than 2 seconds.

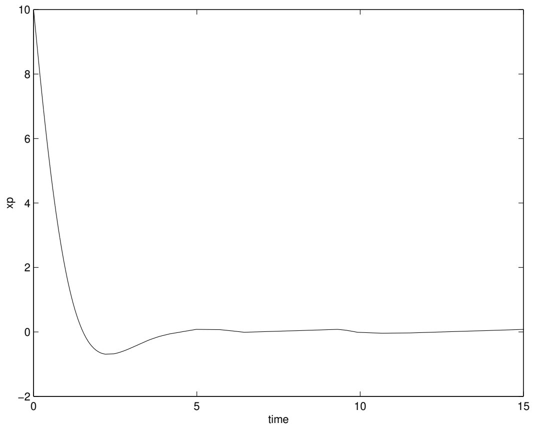

Figure 4: Plot of x_p for the plant with QSC

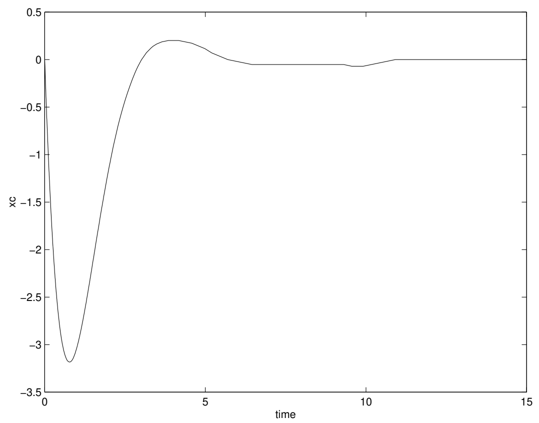

Figure 5: Plot of x_c in the controller

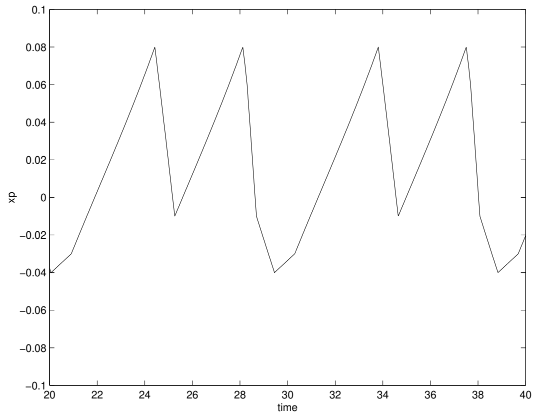

Figure 6: Final oscillations in x_p

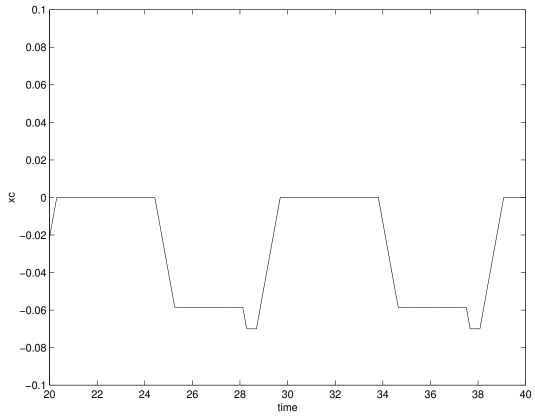

Figure 7: Final oscillations in x_c

IV. MORE ADVANTAGES OF QSC

As it was mentioned before, one of the most important advantage of the QSC scheme is related to the possibility of assuring regional and semiglobal convergence of the system trajectories. However, there is another advantage that in some cases could be important related to the time interval between samples and the resulting computational cost.

Consider the following first order system

$$\dot{x} = \mathrm{sgn}(x) + u \tag{11}$$

Suppose that the controller can measure the variable x, but the A/D converter only produces odd numbers (-3, -1, 1, 3, ...). Then, when x is in the shaded region of Figure 9 (between -1 and 1) the control system sees the value -1.

Consider that the goal is keeping the value of x between -2 and 2. The time to go from 1 to 2 (or from -1 to -2) with $u = 0$ is $t_1 = 1$. This implies that it is impossible to find a discrete time controller achieving the proposed goal with a sampling period greater than t_1. The reason of this is that the controller cannot distinguish if the value of x is positive or negative when it is in the shaded region and then, the sign of u could

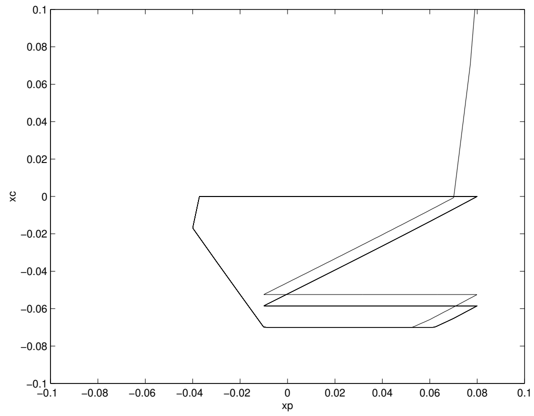

Figure 8: Final oscillations x_p *vs.* x_c

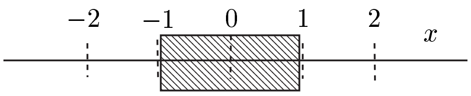

Figure 9: *Invisible* zone due to the quantization

be the same as the sign of x and the trajectory will abandon the desired region before the time t_1.

However, if we use QSC, the time between samplings can be done arbitrarily big. For instance, consider the following static control law

$$y_c = -u_c(1 + a) \tag{12}$$

where a is a positive constant. When x goes away of the shaded region the controller immediately detects the change and it inverts the sign of the derivative. The new speed on x is a. Then, x enters again the shaded region and the time to reach the origin is $1/a$. After that, x goes to the other bound of the shaded region, but with a new speed of $2+a$. When x leaves the shaded region again the controller inverts the sign of the derivative and we obtain a cyclic behaviour where x oscillates between 1 and -1. The time between successive A/D conversions is

$$t_2 = \frac{1}{a} + \frac{1}{2+a} \tag{13}$$

It is clear that taking a small, t_2 can be done arbitrarily big. Then, with this implementation the number of calculations in the controller and the size of the oscillation can be considerably reduced.

As it is mentioned in Section V,.the QSC implementation is not exact. In fact, there are delays related to the presence of converters and the digital processor. In this last example, these delays must be smaller than the minimum sampling period $t_1 = 1$ in order to obtain the proposed goal of keeping x between 2 and -2.

One could think that if it is possible to implement a QSC achieving such minimum delay, then it would be possible to implement a discrete time controller using that sampling period. However, the delay in the QSC controller is only the time required to detect the error and to perform the D/A conversion because the calculations can be done before the error was detected. This is possible because the QSC controller knows that its next input value can only adopt two different values[5].

A classical discrete time controller during each sampling period must perform the A/D conversion, calculate the next state and output of the controller and then perform the D/A conversion. The time required to finish all these tasks is much greater than the delay in the QSC scheme.

V. REMARKS AND PRACTICAL TOPICS

A. Making Use of the State Quantization in the D/A Conversion

The difference between the continuous designed control and the discrete event implementation is due to the effects of the quantization introduced by the state quantization and the presence of A/D and D/A converters. However, in some cases, the effects of quantization in the D/A converters can be eliminated.

The input of the D/A converters is given by

$$y_c = g_c(q_c, u_c) \tag{14}$$

Since q_c and u_c are quantized variables, g_c has a finite domain and then y_c can only take values in a finite set. This set is determined by the quantization values in q_c, u_c and by the function g_c. If that quantization values are chosen so that the mentioned set is a subset of the quantization values of the D/A converters, these converters will not introduce any error.

One of the cases where D/A errors can be avoided is when the output variables of the controller coincide with state variables of that system. In this case the mentioned error can be eliminated by choosing the same quantization in these state variables and in the corresponding D/A converters.

B. Practical Implementation of the A/D Converters

As it was mentioned, the A/D converters in the QSC schema do not work in the classical way. They only perform conversions when the difference between de analogical and digital variables is greater than the quantization interval.

A way of achieving this behavior is placing a counter, a D/A converter and a comparer instead of the A/D converter. The output of the counter should be connected to the D/A and the D/A output must be compared with the output of the system. When the

[5] In the example, if the last detected value of x was -1, then 1 and -3 are the only two values that the new input can adopt. Thus, the controller has two possible output values that could be calculated before the detection of the new input value. For a dynamic controller similar ideas can be applied.

difference is greater than some threshold (the quantization interval), the value in the counter is increased (or decreased according to the sign of the difference). The value of this counter corresponds to the digital output of the asynchronous A/D converter.

C. Delays Introduced by the Real Devices

One of the most important features of the QSC implementation is related to the fact that it takes into account the effects of quantization in the converters. It seems to be a methodology that considers what really happens in a real implementation. However, it was implicitly assumed that the A/D and D/A converters as well as the digital device work in an ideal way without introducing any delay. It is clear that all real devices need some time to calculate and give their output values. The mentioned delays will produce that the performance of a real implementation of a QSC differs from the predicted one.

For instance, the exact implementation of the controller designed in the example (equation (4)) would require an infinity processing speed since it has a direct path from the input to the output. However, as it is explained in the example of Section IV,.the delays can be minimized by taking into account some properties of the scheme.

VI. CONCLUSIONS

The stability and convergence properties and the algorithm for designing QSC controllers deduced in (Kofman, 2001) and illustrated with the example of Section 9 show some theoretical advantages of the QSC scheme over classical discrete time controllers. Moreover, the reduction of the computational cost of the implementation, discussed through the example in Section IV,. shows a very important practical reason for the use of the developed methodology.

Although the principles of the implementation of the QSC controllers seems to be quite difficult (at least for people that is not used to the DEVS formalism) the programming of them is rather simple. Another advantage is the fact that for the design of the QSC controllers, classical techniques of design of continuous controllers can be applied (for instance, in the example of Section 9, the original continuous controller was designed via *exact linearization* (Khalil, 1996)). Then, according to the algorithm of design, an standard Lyapunov analysis to determine the regions of attraction and the ultimately bounds is performed. This kind of analysis is very common in nonlinear control systems design.

As it was mentioned in the conclusions of (Kofman, 2001), the future research is pointed to the particularization of the method for linear systems, where obtaining closed formulas for the design is the most important expected result. In this case, it would be also interesting arriving to less conservative conditions of design.

Some work must be also done in order to optimize the methods for real time simultion of DEVS taking into account the properties of QSC, since the minimization of the delays in the implementation is extremely important. The remarks of Section IV about the fact that the controllers have some information about the next input event and the possibilities of precalculating the next output consitute a starting point in this direction.

From the practical point of view, the next step is implementing QSC controllers in real systems in order to verify that the good theoretical properties that were deduced and validated via simulation are still conserved in real applications.

REFERENCES

Hou, L., Michel, A., and Ye, H., “Some qualitative properties of sampled-data control systems,”, *IEEE Trans. Automat. Contr.* 1721–1725 (1997).

Hu, L. and Michel, A., “Some qualitative properties of multirate digital control systems,”, *IEEE Trans. Automat. Contr.* 765–770 (1999).

Khalil, H. *Nonlinear Systems.* New Jersey: Prentice-Hall, 2nd edition (1996).

Kofman, E. “Quantized-State Control. A Method for Discrete Event Control of Continuous Systems. Part I,”. Technical Report LSD0101, LSD-UNR. Submitted to RPIC’01. Available at www.eie.fceia.unr.edu.ar/∼ekofman (2001).

Kofman, E. and Junco, S. “Quantized State Systems. A DEVS approach for continuous system simulation,”. Technical Report LSD0004, LSD, UNR. To appear in *Transactions of SCS.* Available at www.eie.fceia.unr.edu.ar/∼ekofman (2000).

Sayiner, N., Sorensen, H., and Viswanathan, T. “A non-uniform sampling technique for A/D conversion,”. In *Proceedings of ISCAS’93*, 2 1220–1223 (1993).

Zeigler, B. *Theory of Modeling and Simulation.* New York: John Wiley & Sons (1976).

Zeigler, B. and Kim, J., “Extending the DEVS-Scheme knowledge-based simulation environment for real time event-based control,”, *IEEE Trans. on Robotics and Automation*, 3(9), 351–356 (1993).

Zeigler, B., Kim, T., and Praehofer, H. *Theory of Modeling and Simulation. Second edition.* New York: Academic Press (2000).