COMPARATIVE RESPONSE FOR PROCESS CONTROL SYSTEM "BALL ON BEAM" DESIGNED BY LQR AND MPC METHOD

Popescu Ion Marian, ***University Constantin Brâncuşi, Tg-Jiu,România***
Runceanu Adrian, ***University Constantin Brâncuşi, Tg-Jiu,România***

Abstract: *Designing a control structure for the Ball on Beam offer important results and conclusions at the level of experience of design and implementation. However, the comparison between the two algorithms implement on the same process control, highlight the advantages or disadvantages of their level of functioning. The process Ball on Beam presented, although at first seems just a "toy", but, because it is a process unstable, nonlinear, underactuated, can provide a basis for implementing and testing real-time control of many principles.*

Key words: *LQR, GPC, Ball on Beam.*

1. MATHEMATICAL MODEL OF THE PROCESS "BALL ON BEAM"

Design of process control algorithms start from equation (1) taken from [1]. The physical parameters of the system are: m - Mass of the ball, I_B - The moment of inertia in rotation of the ball from the center's own, I_A - The moment of inertia of the beam to the fulcrum O, l_0 - The distance between the ball position, and fixed pivot, L - Length of beam, B - Attached to the engine flywheel arm length, R_a - The resistance of the rotor winding, R - Rolling radius of the ball resting on the beam. We note: $a = \frac{I_B}{R}$; $b = m + \frac{I_B}{R^2}$; $c = mg$; $d = ml_0^2 + I_B + I_A$; $e = \frac{K_t K_b L}{R_a B}$; $f = \frac{K_t}{R_a}$

$$\begin{cases} \dot{x} = \begin{bmatrix} \Delta\ddot{\alpha} \\ \Delta\dot{\alpha} \\ \Delta\ddot{l} \\ \Delta\dot{l} \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \frac{be}{ab-bd} & -\frac{cb}{ab-bd} & 0 & \frac{bc}{ab-bd} \\ 1 & 0 & 0 & 0 \\ -\frac{ae}{ab-bd} & \frac{cd}{ab-bd} & 0 & -\frac{ac}{ab-bd} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} -\frac{bf}{ab-bd} \\ 0 \\ \frac{af}{ab-bd} \\ 0 \end{bmatrix} u \\ \Delta l = y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \end{cases} \quad (1)$$

2. IMPLEMENTATION OF CONTROL SYSTEM, USING LQR METHOD (Linear Quadratic Regulator)

Starting from the state equation of (1), linearized around a stationary operating point for the "Ball on Beam", the state feedback structure is shown in *Fig.1*.
Thus, we seek a command of the form:

$$u(t) = [-k_1 - k_2 - k_3 - k_4] \cdot \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T$$

We consider the linearized system and quadratic objective function [2], [3]:

$$J = \frac{1}{2}\int_0^{t_f} \left(x^T(t)\, Q\, x(t) + u^T(t)\, R\, u(t) \right) dt \qquad (2)$$

where $Q = Q^T, Q \geq 0$ si $R = R^T, R > 0$

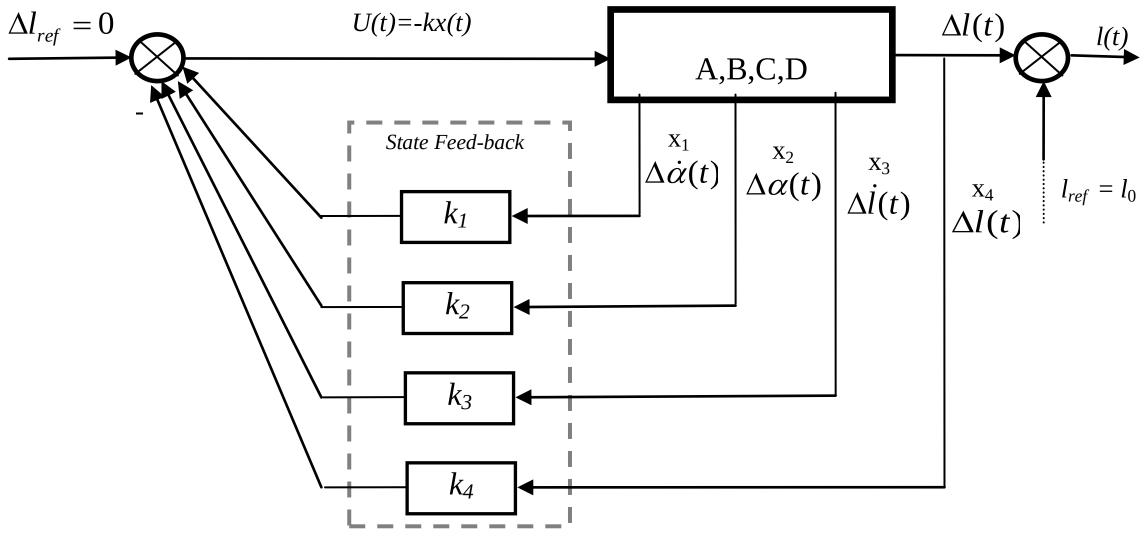

Fig.1. The state feedback structure

Using Matlab-Simulink programming environment, obtain the command for two variants of the weights in the matrix Q, shown in *Table 1*:

Table 1. Design results for different weights of the variable $x_4 = l$	
The parameters of control algorithm for system response shown in *Fig.2* ***(Version 1)***	The parameters of control algorithm for system response shown in *Fig.3* ***(Version 2)***
Q = 20 0 0 0 0 1 0 0 0 0 10 0 0 0 0 5 R = 1	Q = 20 0 0 0 0 1 0 0 0 0 10 0 0 0 0 50 R = 0.10000000000000
k1 = 1.701898817593371 k2 = 22.090825754154228 k3 = -5.521746679549753 k4 = 3.284724152681749	k1 = 9.410742579569869 k2 = 65.022363083483668 k3 = -26.182837532749915 k4 = 23.259302052164692

Using the parameters obtained in *Table 1*, the responses simulated in Matlab-Simulink is represented comparative in *Fig.2* and *Fig.3*. It can be seen that, the state variables, rotational speed beam, beam angular position, speed ball, are stabilized to "0" after a period of time, when it reaches a point of equilibrium. However, the "Version 2", which has a higher weight "= 50" in the matrix Q for the position of the ball, evidently stabilize faster and more accurate of the ball position along the rod, than "version 1" which has a smaller share "= 5".

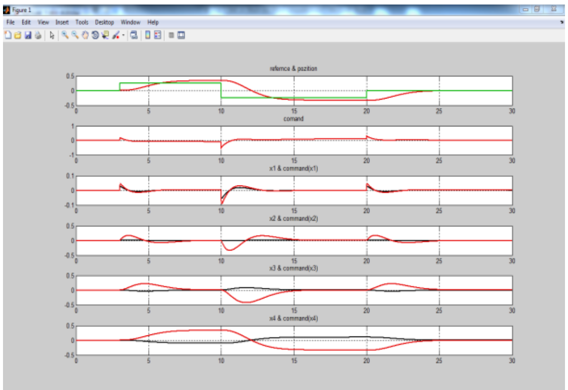

Fig.2. Simulated response of the system Ball on Beam in "Version 1"

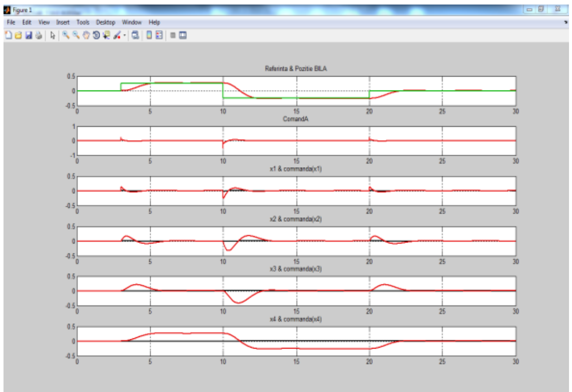

Fig.3. Simulated response of the system Ball on Beam in "Version 2"

3. IMPLEMENTATION OF CONTROL SYSTEM, USING MPC METHOD (Model Predictive Control)

Predictive control algorithms have a considerable development in recent years from the research community but also in the industrial environment. Many companies that provide hardware and software solutions, in the field of automation systems such as Honeywell, Matlab, and so on, have included substantial library of advanced algorithms, including algorithms that are also based on Model Predictive Control type - MCP (Model [Based] Predictive Control).

A weakness of this type of control algorithms is that implementation requires complex mathematical treatments that are not very popular at practicing engineers. Current development of microprocessor systems allowed predictive control algorithms to be implemented successfully and fast processes.

The majority MPC algorithms have three elements to be processed to obtain the final command: Operational methodology [4], [5] for most of the predictive control algorithms is illustrated in *Fig.4.*

a. The prediction model
b. The cost function.
c. Getting the control law

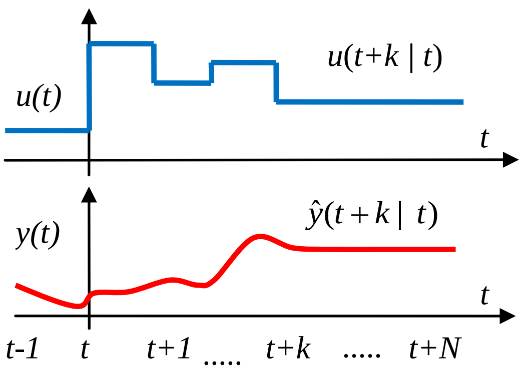

Fig.4. Operational methodology MPC

Thus,. is obtained a command that will minimize a quadratic cost function, consisting of the difference between the predicted output and a trajectory prediction system. As will be seen below, it provides an analytical solution (without constraints), which can be applied to open-loop unstable processes that contain poles approach of instability, processes characterized by non minimum phase models, processes characterized by unknown or variable dead time, processes with structural or parametric uncertainties. It incorporates the concept of control horizon, taking into account a certain weight of control in cost function. Depending on the choice of these weights available will get a variety of control objectives. The most processes can be described numerically by CARMA models (Controller Auto-Regresive Moving Average):

$$A(z^{-1}) \cdot y(t) = z^{-d} \cdot B(z^{-1}) \cdot u(t-1) + \\ + C(z^{-1}) \cdot \frac{e(t)}{\Delta}$$

(3)

For simplicity, the polynomial $C(z^{-1}) = 1$ where $\Delta = 1 - z^{-1}$, $u(t)$ and $y(t)$ are the sequences of input and output; $e(t)$ is perturbation signal; d is the delay time of the process, considered as the number of sampling periods; $A(z^{-1})$, $B(z^{-1})$, $C(z^{-1})$ are written using the polynomial delay operator z^{-1}. MPC generalized predictive control algorithms, derived from finding a sequence the control that minimizes the cost function multistage form:

$$J(N_1, N_2, N_u) = \\ = \sum_{j=N_1}^{N_2} \delta(j) \cdot [\hat{y}(t+j \mid t) - w(t+j)]^2 + \\ + \sum_{j=N_1}^{N_u} \lambda(j) \cdot [\Delta \cdot u(t+j-1)]^2$$

(4)

where $\hat{y}(t+j \mid t)$ is output as the prediction process to step j calculated after current time t. So we obtain output the simulated The values N_1, N_2 are the beginning and end of the period that constitutes the horizon cost function. The value N_u is the number of steps for the control horizon. The values $\delta(j)$, $\lambda(j)$ are sequence weighting cost function. The signal $w(t+j)$ is the reference trajectory. which is desired to realize in the future. This trajectory can be expressed as:

$$w(t+k) = \alpha \cdot w(t+k-1) + \\ + (1-\alpha) \cdot r(t+k) \qquad (5)$$

where $k = \overline{1, N}$ and α is a weighting parameter between 0 and 1. Predictive control objectives consist in calculating a sequence of future control $u(t)$, $u(t+1)$,.... so that the output at the time of next *t+j*, $y(t+j)$ of the process, simulated by the process equations, to approach the reference trajectory $w(t+j)$. This is achieved by minimizing the cost function in equation (4). In order to optimize the cost function will use the optimal prediction (simulated) output sequence $y(t+j)$ for $j \geq N_1$ and $j \leq N_2$. We start from the trivial Division with Rest Theorem:

*object that is divided = object that divides ** **Quotient the division** + **Rest**

In this context we consider *object that is divided* =1 and *object that divides* $= \Delta \cdot A(z^{-1})$. Thus we get:

$1 = \Delta \cdot A(z^{-1})$ * **Quotient** + **Rest**

We consider the following diophantine equation, obtained by dividing the polynomial 1 polynomial $\Delta \cdot A(z^{-1})$:

$$1 = \Delta \cdot A(z^{-1}) * E_j(z^{-1}) + z^{-j} \cdot F_j(z^{-1}) \quad (6)$$

This equation will be solved, and polynomials **Quotient the division** $E_j(z^{-1})$ and **Rest** $F_j(z^{-1})$ are uniquely defined, with grades $(j-1)$ respectively n_a. These are obtained by successive division of the polynomial 1 and the polynomial $\Delta \cdot A(z^{-1})$ until the rest can be factored as $z^{-j} \cdot F_j(z^{-1})$, and the remaining quotient is $E_j(z^{-1})$.

process equations on a time horizon of *j* steps of sampling before (in the future) from the moment *t*.

After processing we have:

$$y(t+j) - F_j(z^{-1}) \cdot y(t) =$$
$$-E_j(z^{-1}) \cdot B(z^{-1}) \cdot \Delta \cdot u(t+j-d-1) +$$
$$+E_j(z^{-1}) \cdot e(t+j) \qquad (7)$$

In (7), if the degree of the polynomial $E_j(z^{-1})$ is $(j-1)$ intuitively it can be said that the terms $e(t+j)$ are all in the future and thus, the best prediction of the term $y(t+j)$ is:

$$y(t+j) = F_j(z^{-1}) \cdot y(t) + \\ + E_j(z^{-1}) \cdot B(z^{-1}) \cdot \Delta \cdot u(t+j-d-1) \qquad (11)$$

Returning to the notation relative to the current time we have:

$$\hat{y}(t+j \mid t) = G_j(z^{-1}) \cdot \Delta \cdot u(t+j-d-1) +$$
$$+F_j(z^{-1}) \cdot y(t) \qquad (12)$$

where $G_j(z^{-1}) = E_j(z^{-1}) \cdot B(z^{-1})$

It can be shown that the polynomials recursion for $E_j(z^{-1})$, $F_j(z^{-1})$ can be processed by software. Next we consider $E_j(z^{-1})$ and $F_j(z^{-1})$ are polynomials quotient and the rest of the successive division $\Delta \cdot A(z^{-1})$ until the rest of the division can be factored as: $z^{-j} \cdot F_j(z^{-1})$.

These polynomials can be expressed as:

$$F_j(z^{-1}) = f_{j,0} + f_{j,1} \cdot z^{-1} + f_{j,2} \cdot z^{-2} + \\ .... + f_{j,na} \cdot z^{-na}$$

$$E_j(z^{-1}) = e_{j,0} + e_{j,1} \cdot z^{-1} + e_{j,2} \cdot z^{-2} +$$
$$.... + e_{j,j-1} \cdot z^{-(j-1)} \qquad (13)$$

We assume that the same procedure is used for obtaining polynomials quotient and the rest for *(j+1)* successive polynomial division for 1 to polynomial $\Delta \cdot A(z^{-1})$ until the rest of the division can be factored as: $z^{-(j+1)} \cdot F_{j+1}(z^{-1})$, with the following expression:

$$F_{j+1}(z^{-1}) = f_{j+1,0} + f_{j+1,1} \cdot z^{-1} + \\ + f_{j+1,2} \cdot z^{-2} + ... + f_{j+1,na} \cdot z^{-na} \qquad (14)$$

The rest of the division will change accordingly as $z^{-(j+1)} \cdot F_{j+1}(z^{-1})$, and the quotient is the quotient from the previous step to which was added another term of the current division. Thus we have:

$$E_{j+1}(z^{-1}) = E_j(z^{-1}) + e_{j+1,j} \cdot z^{-j} \qquad (15)$$

Using an observation of the division of polynomials we have:

$$e_{j+1,j} = f_{j,0} \qquad (16)$$

The coefficients of the polynomial $F_{j+1}(z^{-1})$ pot fi exprimaţi sub forma**:**

$$f_{j+1,i} = f_{j,i+1} - f_{j,0} \cdot \tilde{a}_{i+1}; i = 0,...,n_a - 1$$

We return to the equation (12)

$$\hat{y}(t+j \mid t) = G_j(z^{-1}) \cdot \Delta \cdot u(t+j-d-1) +$$
$$+F_j(z^{-1}) \cdot y(t) \qquad (17)$$

where $G_j(z^{-1}) = E_j(z^{-1}) \cdot B(z^{-1})$

We have, for (j+1), taking into account the relationship (15) and (16):

$$G_{j+1}(z^{-1}) = E_{j+1}(z^{-1}) \cdot B(z^{-1}) =$$
$$= \left(E_j(z^{-1}) + e_{j+1,j} \cdot z^{-j}\right) \cdot B(z^{-1}) =$$
$$= \left(E_j(z^{-1}) + f_{j,0} \cdot z^{-j}\right) \cdot B(z^{-1})$$
$$\Rightarrow G_{j+1}(z^{-1}) = E_j(z^{-1}) \cdot B(z^{-1}) + \\ + f_{j,0} \cdot z^{-j} \cdot B(z^{-1}) \qquad (18)$$

Given the notation $G_j(z^{-1}) = E_j(z^{-1}) \cdot B(z^{-1})$, we have:

$$G_{j+1}(z^{-1}) = G_j(z^{-1}) + f_{j,0} \cdot z^{-j} \cdot B(z^{-1})$$

Such terms are obtained:

$$g_{j+1,j+i} = g_{j,i+i} + f_{j,0} \cdot b_i; \qquad i = 0,...,n_b$$

Solving the problem of generalized predictive control is reduced to finding a sequence of control signals $u(t)$, $u(t+1)$,......, $u(t+N)$ ce vor fi obţinute prin optimizarea funcţiei cost din relaţia (4). If we consider that the system has a dead time expressed in a number of *d* sampling period, system output will be influenced by input signal $u(t)$ after a number of *(d+1)* sampling period. The value N_1, N_2 şi N_u that define prediction horizon can be defined as: $N_1 = d+1$, $N_2 = d+N$, $N_u = N$.

Output prediction horizon is influenced by the process dead time. If $N_1 < (d+1)$ then the terms of the cost function will depend only on the previous control signals (in the past). On the other hand, if $N_1 > (d+1)$, then do not take into account the first point of reference trajectory sequence.

În continuare vom considera următorul set de predicţii cu valorile lui j în intervalul specificat:

$$\hat{y}(t+d+1 \mid t) = G_{d+1}(z^{-1}) \cdot \Delta \cdot u(t) + F_{d+1}(z^{-1}) \cdot y(t)$$
$$\hat{y}(t+d+2 \mid t) = G_{d+2}(z^{-1}) \cdot \Delta \cdot u(t+1) + F_{d+2}(z^{-1}) \cdot y(t) \quad (19)$$
$$\vdots$$
$$\hat{y}(t+d+N \mid t) = G_{d+N}(z^{-1}) \cdot \Delta \cdot u(t+N-1) + F_{d+N}(z^{-1}) \cdot y(t)$$

Relations (19) can be written in vector form as follows:

$$Y = G \cdot U + F(z^{-1}) \cdot y(t) + G^{|}(z^{-1}) \cdot \Delta \cdot u(t-1) \quad (20)$$

It can be seen that the last two terms in equation (20) depend only on the initial conditions (past) and can be grouped as a free response within ***f*** below:

$$Y = G \cdot U + f \quad (21)$$

Can be considered if we have zero initial conditions, the free response ***f*** is zero. If we apply a step input to the input unit at time t, we have:

$$\Delta \cdot u(t) = 1; \quad \Delta \cdot u(t+1) = 0; \quad \ldots\ldots; \quad \Delta \cdot u(t+N-1) = 0$$

Thus the expected output sequence is $[\hat{y}(t+1), \ \hat{y}(t+2) \ ,...., \ \hat{y}(t+N)]^T$ and coincides with the first column of the matrix G. In this context, the first column of the matrix G can be calculated in response to the input step of the process, when a unit step is applied. Free response time can be calculated recursively by:

$$f_{j+1} = z(1 - \Delta \cdot A(z^{-1})) f_j + B(z^{-1}) \cdot \Delta \cdot u(t-d+j) \quad (22)$$

cu $f_0 = y(t)$ şi $\Delta \cdot u(t+j) = 0$ for $j \geq 0$

The minimum of cost function of equation (25) with respect to ***U*** command is:

$$\frac{dJ}{dU} = 0; \ \frac{dJ}{dU} = H \cdot U + b^T = 0$$
$$\Rightarrow U = (G^T \cdot G + \lambda \cdot I)^{-1} \cdot G^T \cdot (W - f) \quad (26)$$

The control signal applied as process control is the first element of the vector U and is expressed as:

$$\Delta \cdot u(t) = K \cdot (W - f) \quad (27)$$

where K the first column of the matrix $(G^T \cdot G + \lambda \cdot I)^{-1} \cdot G^T$. This shows in an explicit manner, which can be seen from *Fig.5*, that if we have errors of prediction, which means $(W - f) = 0$, then the command does not vary and thus remains free evolution process [4], [5].

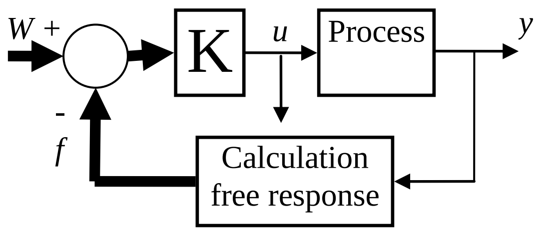

Fig.5. Control Law MPC

In order to understand fully the operation of a predictive control algorithm, presents the design further described in detail in [4], [5] and implemented at process simulation equations Ball on Beam.

The implementation is done in Matlab simulation environment using numerical data presented in the paper [1].

In Fig.6. system response is presented in response Ball on Beam controller presented for prediction horizon j=90 and weight control into cost function . Graphs have the following meanings:

- The green graph is the reference trajectory of the system
- Red graph is the response system

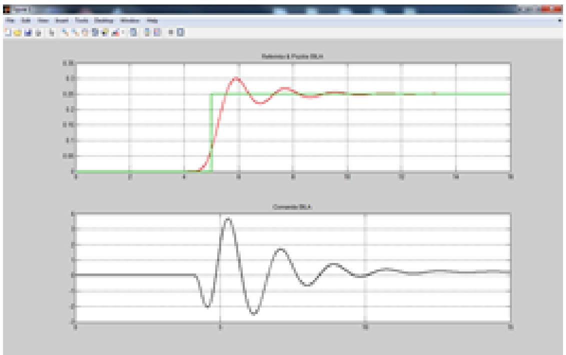

Fig.6. The system response to a prediction horizont j=90 si $\lambda = 0.2$

- Black graph is obtained predictive control

Also in *Fig.6.* can be seen as the control signal is activated before the exact time of the step change prediction horizont.

In *Fig.7.* system response is presented in response Ball on Beam controller presented for prediction horizont j=90 and weight control into cost function $\lambda = 0.8$.

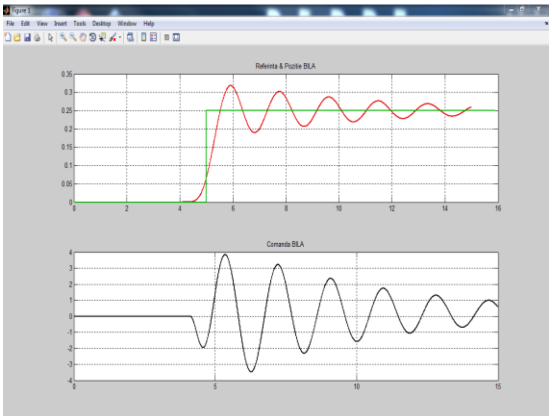

Fig.7. The system response to a prediction horizont j=90 şi $\lambda = 0.8$

It can be seen that the system becomes more oscillatory than in the previous case and increases response time.

In *Fig.8.* system response is presented in response Ball on Beam controller presented for prediction horizont j=50 and weight control into $\lambda = 0.2$. It can be seen that the system becomes unstabile.

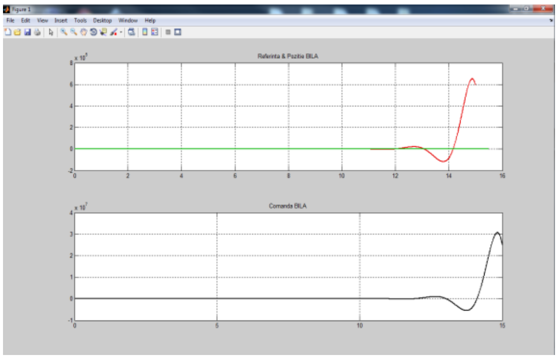

Fig.8. The system response to a prediction horizont j=50 şi $\lambda = 0.2$

4. COMPARISON BETWEEN RESPONSE SYSTEM DESIGNED USING LQR AND MPC

Structure simulation program developed into Simulink is shown in *Fig.9* and running the two systems, closed loop designed.

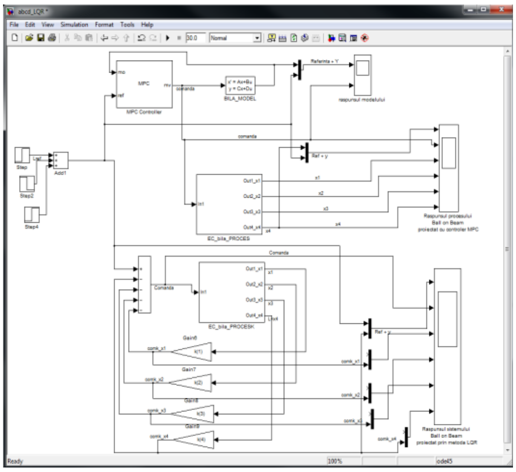

Fig.9. Comparative simulation program of the two systems

The first system is implemented with predictive controller and the second is designed as a system with state reaction after placing poles LQR method. A first comparative result is shown in *Fig.10.* One can see the characteristic reaction system after state (left chart) showing error stationary position. However, the MPC controller command system (right chart) shows large jumps in specific order discrete systems.

In both variants it can be seen that the state variables speed of rotation of the rod - x1, angular position rod - x2, speed of travel of the ball along the rod - x3, following a transitional regime stabilizes at zero when the ball has reached its stationary operation.

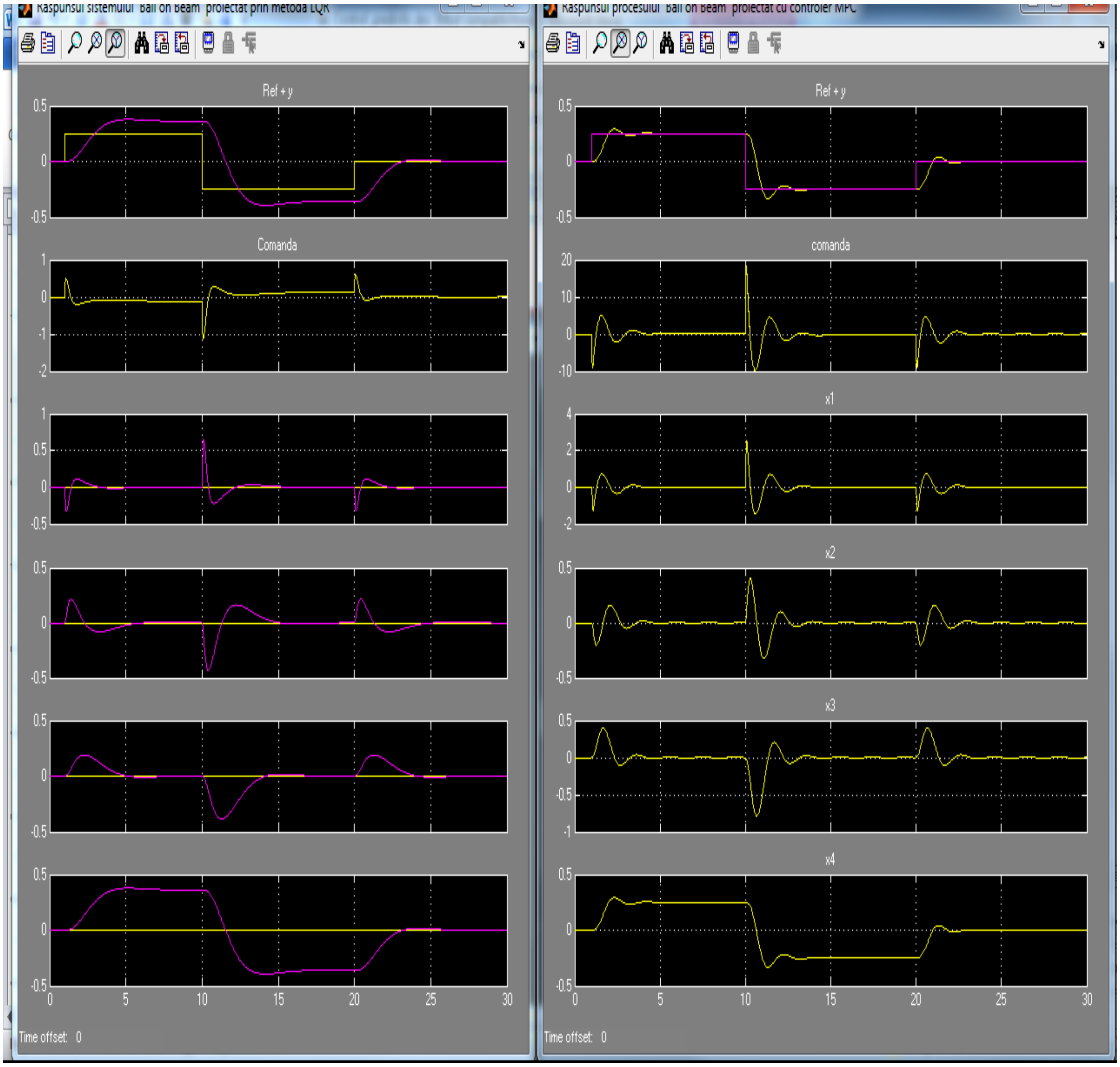

Fig.10. Comparative response of two structures designed

5. CONCLUSION

Currently manufacturing firms provides an extremely large range of hardware configurations, plus a variety of software tools.

The Ball on Beam process, because it is unstable, nonlinear, underactuated, may constitute a basis for real-time implementation and testing of various advanced control principles.

Basically, the best use of the Ball on Beam system is the practical realization of a comparison of the responses obtained by implementing different control principles that can lead to important conclusions.

REFERENCES

[1] Popescu Ion Marian," Mathematical modeling of the process "BALL ON BEAM"", Annals of the University "Constantin Brâncuşi" from Târgu Jiu, Engineering Series, Nr. 1/2012, pag.5-16, http://www.utgjiu.ro/revista/ing/pdf/2012-1/1_MARIAN_Popescu.pdf

[2]F. Andreev, D. Auckly, S. Gosavi, L. Kapitanski, A. Kelkar, W. White, "Matching, linear systems, and the ball and beam", Automatica, Volume 38, Issue 12, December 2002, Pag.2147-2152

[3] C. T. Chen, "Analog and Digital Control System Design-Transfer-Function, State-Space, and Algebraic Methods", 2006, State University of New York at Stony Brook, ISBN 13: 9780195310467, ISBN 10: 0195310462

[4] E. F. Camacho, C. Bordons," Model Predictive Control", 1999, Springer Verlag, ISBN 3-540-76241-8

[5] D. W. Clarke, C .Mohtadi, and P. S. Tuffs, "Generalized Predictive Control", Automatica, 25(6):859-875, 1989