2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)

State-feedback control algorithms for a CNC machine

Dora Sabau, Petru Dobra
Automation Department
Technical University of Cluj Napoca
Cluj-Napoca, Romania

***Abstract*—The controller of a small size Computer Numerical Controlled (CNC) system is designed and tested. The system is considered to have a Multi Input Multi Output(MIMO) model and two control design techniques are presented for it: Pole-placement and Linear Quadratic Regulator(LQR). The LQR problem is proposed to be solved by choosing the weights matrices using the total energy of the system. The simulation results are evaluated and the controller that meets the desired behavior is implemented on the real machine.**

***Index Terms*—LQR, Pole-placement, energy, MIMO, CNC, control**

I. Introduction

Numerical control machines started to be used since 1950s. Because of the rapid development of the industry, these machines were improved annually. Currently, Computer Numerical Control(CNC) machines are very popular in the industrial field in domains such as: aeronautical industry, energy and power technology and manufacturing industry. These machines provide a high level of automation and consistent motion control, accuracy, high precision and power, fast machining in accordance with modern tools [1]. CNC systems are, regularly, compound of: the machine itself and the command system. The control system incorporates new techniques which reduce the allocated time to a project and allow a complex control of the process. The requirements these control systems have nowadays are: precise control of the position, with safe velocity control and good acceleration and deceleration properties [2]. In order to meet these requirements, the controller that is used has to give good precision and to have high resolution [3], [4].

As shown in literature [5], [6], conventional Proportional-Integrative-Derivative(PID) controller, State-Feedback control, Feed-Forward control, Linear Quadratic Regulator(LQR), Fuzzy Logic Control or even more advanced control techniques such as optimized PD control using genetic algorithms or adaptive control, are used to control CNC machines.

In order to obtain the control law for positioning systems with translation axes, it is important to have a mathematical model for the process [7]. For this, two particular directions are taken into consideration. The first one is to consider the system as a decoupled one, which means each axis is handled as an independent process. The second approach takes into consideration the coupled-axis, so the process is treated in terms of Multi-Input-Multi-Output(MIMO) model [8].

In this paper, a MIMO model with two inputs and two outputs is taken into consideration. In chapter 2, based on the acquired data consisting on the input and the output signals from the machine, a mathematical model is obtained. The parameter identification results, obtained using System Identification ToolBox from Matlab, are presented in chapter 3. Chapter 4 focuses on designing and simulating two state-feedback control techniques: pole-placement method and Linear Quadratic Regulator(LQR) and chapter 5 and 6 show the experimental results and the conclusions.

II. MATHEMATICAL MODEL FOR CNC

The workstation in the laboratory, which represents a small-size CNC machine is shown in Fig.1.

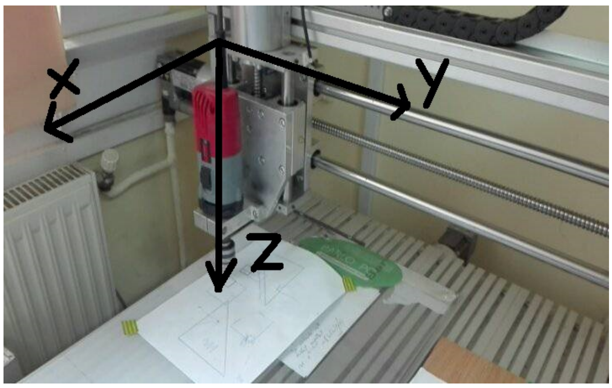

Fig. 1: CNC machine

The positioning system compounds of three axes of translation: X, Y and Z. Z-axis is driven by a stepper motor and has the role to place the tool at the right distance on the vertical direction. X and Y-axis are driven by BrushLess Direct Current(BLDC) motors which receive commands from MC206X controller. The controller is connected to the PC via an USB link, as shown in Fig.2 and can be programmed using TrioBasic programming Language, which gives the possibility to develop control commands fast and easy.

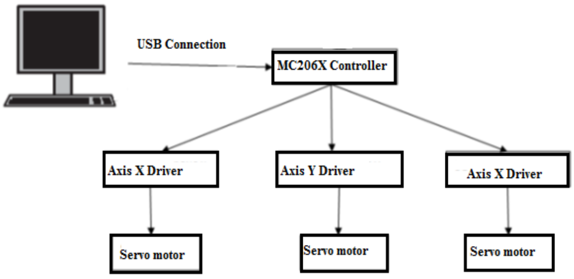

Fig. 2: Controller connections

The purpose of this paper is to consider a mathematical model for 2D positioning system(taking into consideration X and Y-axis) in order to design a control law for it.

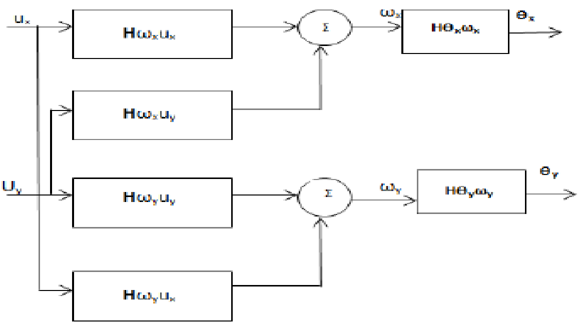

Fig. 3: Block diagram representation for the model used for parameters estimation

The system can be represented, as shown in Fig.3, by a Multi-Input-Multi-Output process with two inputs and two outputs, as it follows:

- Inputs: u_x, u_y -represent the drive command, where index x or y suggest the corresponding axis.
- Outputs: θ_x, θ_y -represent the position of rotors of the motors that drive each axis.

Also, we have the designations that suggest the type of the signal or the relationship between signals:

- ω_x, ω_y: angular velocity;
- $H_{\omega_x u_x}$, $H_{\omega_y u_y}$, $H_{\theta_x \omega_x}$,$H_{\theta_y \omega_y}$ the transfer functions from u to ω and from ω to θ, for each axis
- $H_{\omega_x u_y}$, $H_{\omega_y u_x}$ the mutual effect between the two axes

The block diagram represented in Fig.3 illustrates the simplified scheme that represents the dependency between electrical and mechanical part of the system. The whole system can be represented by two coupled subsystems, each one corresponding to one axis. The mutual effect between the axes appears because of the mechanical connection between the two.

The transfer function from the command u to the speed ω, for each axis, can be approximated by a first order term, taking into consideration only the time constant Tm, representing the mechanical part of the drive:

$$H_{\omega_x u_x} = \frac{K_{M_x}}{T_{M_x}s+1}; H_{\omega_y u_y} = \frac{K_{M_y}}{T_{M_y}s+1} \quad (1)$$

The second transfer function shows the relationship between the speed and the position and is represented by an integrator:

$$H_{\theta_x \omega_x} = \frac{1}{s}; H_{\theta_y \omega_y} = \frac{1}{s}; \quad (2)$$

The mutual effect between the two axes can be approximated with proportional gains:

$$H_{\omega_y u_x} = K_{\omega_y u_x}; H_{\omega_x u_y} = K_{\omega_x u_y} \quad (3)$$

The behavior analysis and the control design algorithm will be made using state space model:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (4)$$

The notations used are as follows:

- x-state vector
- A-the state matrix
- B-the input matrix
- u-the vector of the input signals
- C-the output matrix
- y-the vector of the output signals
- D-the direct transfer

The state space variables are considered the measured variables: the rotor speed and position for each axis. The input signals vector is represented by the command signal 'u' for each axis. The outputs are considered only the variables that need to be controlled: the position of each axis.

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} = \begin{pmatrix} \omega_x(t) \\ \theta_x(t) \\ \omega_y(t) \\ \theta_y(t) \end{pmatrix} \quad (5)$$

$$u(t) = \begin{pmatrix} u_x(t) \\ u_y(t) \end{pmatrix} \quad (6)$$

$$y(t) = \begin{pmatrix} \theta_x(t) \\ \theta_y(t) \end{pmatrix} \quad (7)$$

Transforming the transfer function model represented in Fig.3 into a state space model with the previous notations and considerations, we obtain:

$$A = \begin{pmatrix} -\frac{1}{T_{M_x}} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{T_{M_y}} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (8)$$

$$B = \begin{pmatrix} \frac{K_{M_x}}{T_{M_x}} & K_{\omega_x u_y} \\ 0 & 0 \\ K_{\omega_y u_x} & \frac{K_{M_y}}{T_{M_y}} \\ 0 & 0 \end{pmatrix} \quad (9)$$

$$C = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (10)$$

$$D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (11)$$

III. Parameter Identification

Using TrioBasic commands, we have the measured signal obtained by following a path at the maximum speed of the rotor of each axis. The acquired data signals are represented in Fig.4, with the mention that the signals are scaled for a good representation of all the signals in the same figure.

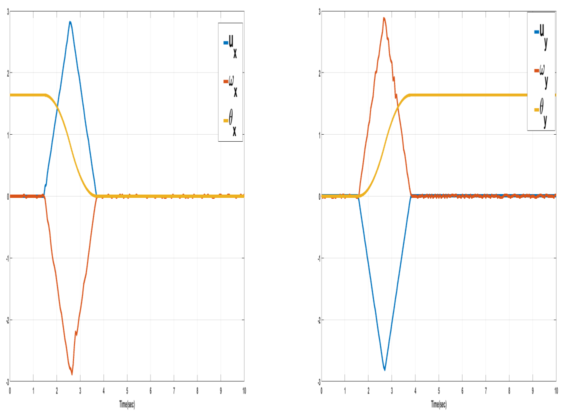

Fig. 4: Acquired data for each X and Y-axis

Using the models described above and applying the identification methods [9] based on minimization of prediction error and the least squares algorithm, the model parameters are obtained using System Identification Toolbox in Matlab [10].

Once the fitting, autocorrelation and cross-correlation tests are passed, the estimated model is obtained and transformed into transfer functions using 'Zero Order Hold' discretisation method. So, we have:

- For X-axis:

$$H_{\omega_x u_x} = \frac{1054}{s + 40.85} \quad (12)$$

$$H_{\theta_x \omega_x} = \frac{1}{s} \quad (13)$$

- For Y-axis:

$$H_{\omega_y u_y} = \frac{2188}{s + 87.81} \quad (14)$$

$$H_{\theta_y \omega_y} = \frac{1}{s} \quad (15)$$

- Mutual effect:

$$H_{\omega_y u_x} = K_{\omega_y u_x} = 24.46; H_{\omega_x u_y} = K_{\omega_x u_y} = 26.65 \quad (16)$$

The state space model will have:

$$A = \begin{pmatrix} -40.85 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -87.81 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix};$$
$$B = 10^3 * \begin{pmatrix} 1.05 & 0.267 \\ 0 & 0 \\ 0.0245 & 2.188 \\ 0 & 0 \end{pmatrix}; \quad (17)$$
$$C = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix};$$
$$D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix};$$

IV. Controller Design

A. Pole Placement

Closed-loop pole assignment is a common technique used to design controllers for MIMO processes. Specifying the closed loop eigenvalues, a state-feedback matrix K is computed, so that the closed loop system satisfies the condition [11]:

$$(A - BK)x = \lambda_i I x \quad (18)$$

where λ_i are the desired eigenvalues.

Also, to eliminate the stationary heading error when tracking a reference signal, a pre-filter matrix F is calculated using the following equation:

$$F = -(C(A - BK)^{-1}B)^{-1} \quad (19)$$

To allocate the eigenvalues [-80 -80 -120 -120], the matrices K and F have the following values:

$$K = \begin{pmatrix} 0.1510 & 9.1107 & -0.0013 & -0.1110 \\ -0.0017 & -0.1019 & 0.513 & 4.388 \end{pmatrix} \quad (20)$$

$$F = \begin{pmatrix} 9.1107 & -0.1110 \\ -0.1019 & 4.388 \end{pmatrix} \quad (21)$$

The step response is simulated and represented in Fig.5.

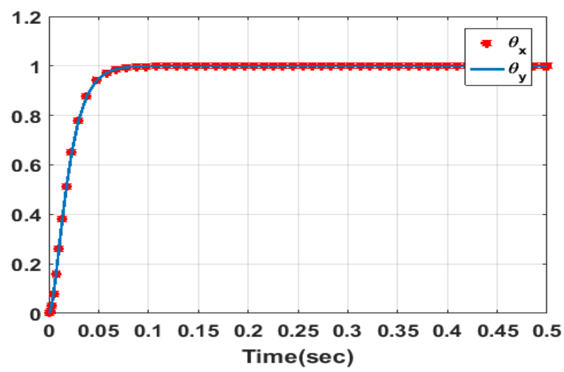

Fig. 5: Closed-loop step response using pole-placement

B. LQR Problem

Another way to design a controller for a MIMO process is using LQR problem. This problem is based on optimal control, which is concerned with operating a system at minimum cost. In this case, the system dynamics is described using linear differential equations (state space model) and the cost function is represented by a quadratic function as the following one:

$$J = \int_0^{\infty} (x^T(t)Qx(t) + u^T(t)Ru(t))dt \qquad (22)$$

where Q defines the weights on the state and is a semi-defined symmetric matrix and R defines the weights on the control input.

The aim is to minimize the cost function J and to find the optimal control law u* that should be used [12].

Based on the scientific articles in the literature, it turns out that regardless of the matrix Q and matrix R, the minimum cost function is obtained by solving a Ricatti equation [13]:

$$-\dot{P}(t) = Q - P^T BR^{-1}B^T P + A^T P + PA \qquad (23)$$

This equation is obtained by solving the Hamilton-Jacobi-Bellman (HJB) equation (23) with the boundary condition in (24).

$$\frac{\partial J^*(x(t),t)}{\partial t} + min_u H = 0 \qquad (24)$$

$$J^*(x(t_f), t_f) = 0, \text{where } t_f \text{ is the final time.} \qquad (25)$$

The optimal cost function J* is quadratic:

$$J^*(x,t) = x^T P(t)x \qquad (26)$$

So,

$$\begin{aligned} \frac{\partial J^*(x(t),t)}{\partial t} &= J_t^* = x^T \dot{P}(t)x \\ \frac{\partial J^*(x(t),t)}{\partial x} &= J_x^* = 2P(t)x \end{aligned} \qquad (27)$$

The Hamiltonian:

$$H = x^T Qx + u^T Ru + J_x^{*T}(Ax + Bu) \qquad (28)$$

To solve the problem, we have to minimize the Hamiltonian with respect to u:

$$\frac{\partial H}{\partial u} = 2Ru + (J_x^{*T} B)^T = 2Ru + B^T J_x^* = 0 \qquad (29)$$

which gives

$$u = -\frac{1}{2}R^{-1}B^T J_x^* \qquad (30)$$

After calculations and replacements, the HJB equation becomes:

$$x^T \dot{P}(t)x + X^T Qx + (-R^{-1}B^T Px)^T R(-R^{-1}B^T Px)$$

$$+2x^T P^T(Ax + B(-R^{-1}B^T Px)) = 0 \qquad (31)$$

which gives the Ricatti equation mentioned above.

Since we have the Ricatti equation, we have solved the optimal control problem. The feedback control law is:

$$u = -\frac{1}{2}R^{-1}B^T(2P(t)x) = -R^{-1}B^T P(t)x = -Kx \qquad (32)$$

where K is the gain matrix.

Like in the pole-placement case, if we want to implement a tracking controller, it is necessary to compute a matrix N the reference signals is multiplied by, so that the stationary error is null.

$$N = -(C(A - BK)^{-1}B)^{-1} \qquad (33)$$

In many cases, the weight matrix Q or R is chosen through successive attempts, so that the results fulfill the requirements.

This paper propose a method to chose the weights of matrix Q using the energy of the system [8], [14].

Having:

- the general differential equation of a motor, corresponding to each axis:

$$J\frac{\partial^2\theta}{\partial t^2} + D\frac{\partial\theta}{\partial t} + K\theta = u \qquad (34)$$

- the energy of the subsystem, consisting of each axis separately:

$$E = E_k + E_p;\ E_k = \frac{1}{2}J\omega^2;\ E_p = \frac{1}{2}K\theta^2 \qquad (35)$$

where E_k is the kinetic energy and E_p is the potential energy,

We can write the cost function J in LQR problem as the total energy of the system.

That means that Q will be:

$$Q = \frac{1}{2}\begin{pmatrix} J_x & 0 & 0 & 0 \\ 0 & K_x & 0 & 0 \\ 0 & 0 & J_y & 0 \\ 0 & 0 & 0 & K_y \end{pmatrix} \qquad (36)$$

For our system, we set:

$$Q = \frac{1}{10^3}\begin{pmatrix} 0.4744 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0.2285 & 0 \\ 0 & 0 & 0 & 0.05 \end{pmatrix},$$

$$R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

and we obtain:

$$K = \begin{pmatrix} 0.0059 & 0.0071 & 0 & -0.0001 \\ 0.0001 & 0.0001 & 0.0028 & 0.0071 \end{pmatrix} \qquad (37)$$

$$N = \begin{pmatrix} 0.0071 & -0.0001 \\ 0.0001 & 0.0071 \end{pmatrix} \qquad (38)$$

The simulink model shown in Fig.6 is used to simulate the behavior of the closed loop system.

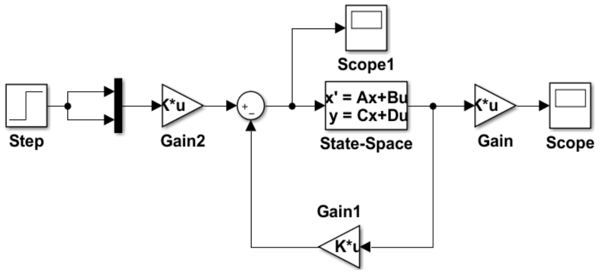

Fig. 6: Simulink Model for feedback control

The step response can be visualized in Fig. 7 where it can be observed that the settling time is much bigger than the one obtained with the pole-placement method.

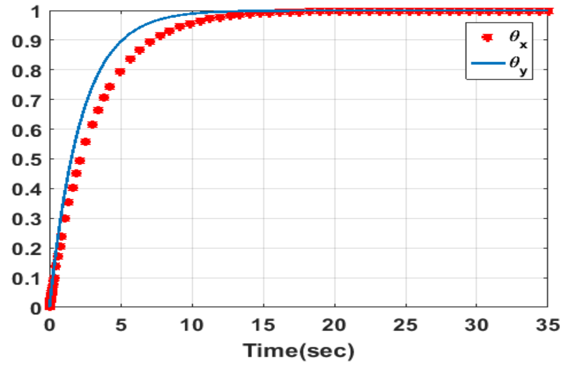

Fig. 7: Closed-loop step response using LQR

In order to obtain some results that are closer to the one obtained with the first method, several testes were made and some conclusions can be mentioned: in order to improve the response of the closed loop system using the energy LQR method presented above, it is necessary to change both R matrix and the weights that correspond to the position variable. The best results,which are almost the same as the one obtained with pole placement method, were obtained using the following matrices:

$$Q = \begin{pmatrix} 0.0005 & 0 & 0 & 0 \\ 0 & 0.85 & 0 & 0 \\ 0 & 0 & 0.0002 & 0 \\ 0 & 0 & 0 & 0.25 \end{pmatrix},$$

$$R = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix}$$

The simulated step response using the following control matrices:

$$K = \begin{pmatrix} 0.0059 & 0.0071 & 0 & -0.0001 \\ 0.0001 & 0.0001 & 0.0028 & 0.0071 \end{pmatrix} \quad (39)$$

$$N = \begin{pmatrix} 0.0071 & -0.0001 \\ 0.0001 & 0.0071 \end{pmatrix} \quad (40)$$

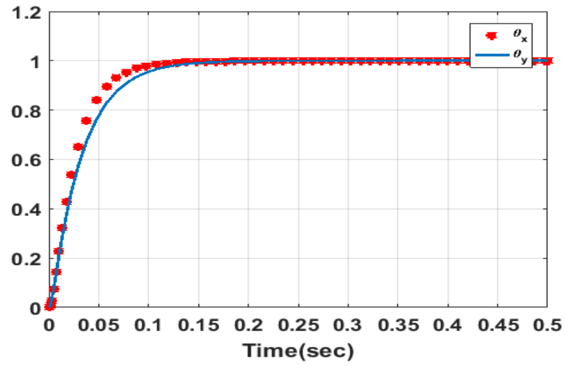

Fig. 8: Closed-loop step response using modified LQR weight matrices

V. Experimental Results

Based on the scenarios simulated and presented in the previous chapter, for experimental test (39) and (40) are used. As presented in the first chapter, the motors that drive the axes are controlled with MC206X Motion Coordinator. This have five servo gains [15]: Proportional gain: P_Gain, Integral gain: I_Gain, Derivative gain: D_Gain, Output velocity gain: OV_Gain, Velocity Feed-Forward Gain: VFF_Gain. The block diagram that represents the closed loop control system that can be implemented is represented in Fig. 9.

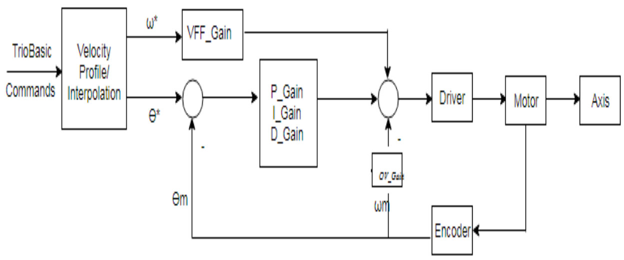

Fig. 9: Servo loop diagram

For implementation, we take into consideration the first two elements from matrix K for X-axis and last two elements for Y-axis, because those are the one that correspond to the velocity and position gain for each axis.

So, we set: For X-axis: $P\_Gain = 9.21, OV\_Gain = 0.21$, and for Y-axis: $P\_Gain = 4.55, OV\_Gain = 0.13$;

During the experimental tests, several cases were taken into consideration: following a circle path with different radius or squares paths with different size. The resultant path was reproduced based on the signals and was represented, in each case, compared to the ones that were obtained using P_{Gains} set after several tries. [16]

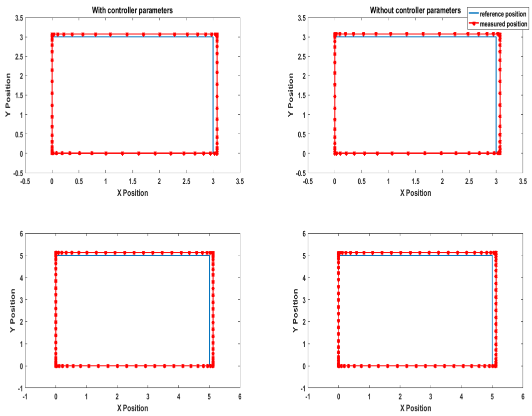

Fig. 10: Results with and without using the designed controller-squares drawn

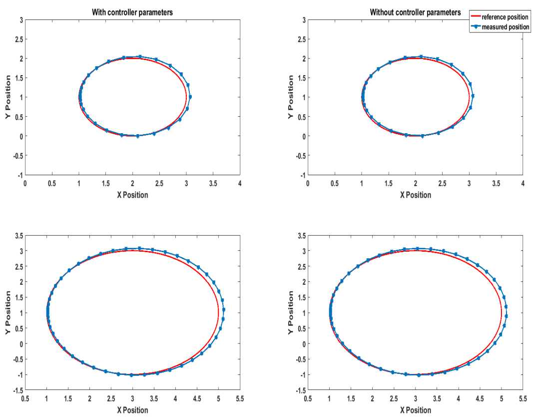

Fig. 11: Results with and without using the designed controller-circles drawn

In the figures above, it can be seen that in each case a small error exists. For each case, a fit performance parameter was calculated and the result were represented in Table.1.

		Using designed controller		No designed controller	
		X Axis	Y Axis	X Axis	Y Axis
Circle	R=1cm	93.2%	96%	92%	92.7%
	R=2cm	95%	98%	93%	97%
Square	L=3cm	80%	80%	79%	76%
	L=5cm	84.5%	86%	82%	84.1%

TABLE I: Fit parameter calculated for each case

VI. Conclusion

As a conclusion, based on the latest results presented in the end of the previous chapter, we can say that the designed controller gives a better precision in terms of following a path. However, we also observed that the error is bigger in case of square path and that could be because at the corners of the square the acceleration and deceleration are big. Also, a reason for the error occurrence is the ball screw translational axis construction and the friction that appears.

In this paper, two methods for designing a controller for an small size CNC system are presented. Both of the methods are good for a MIMO system and both of them are easy to implement. For the LQR problem, an energy based algorithm of choosing the weight matrices was proposed. At the first simulation, the controller seemed to work good but slower than the one obtained with pole placement algorithm. Starting from the parameters obtained using energy based matrix, the weights were modified in order to obtain almost the same result as in the first method. So, the energy based LQR seems to be a good technique in terms of control and stability, but if more precise performances is needed, several tests are needed. So, based on this, we can emphasis that choosing Q and R is part art, part science.

As a future development, other tests and control algorithms have to be tried in order to obtain a better fitting. Also, friction models and compensation should be done for the same purpose.

References

[1] Kücük Hüseyin Koc, Emine Seda Erdinler, Ender Hazir, and Emel Öztürk. Effect of cnc application parameters on wooden surface quality. *Measurement*, 107:12–18, 2017.

[2] Armin Afkhamifar, Dario Antonelli, and Paolo Chiabert. Variational analysis for cnc milling process. *Procedia CIRP*, 43:118–123, 2016.

[3] Xianfeng Chen and Weiming Zhang. Research on the interpolation technology of cnc system. *Procedia engineering*, 174:1252–1256, 2017.

[4] Zhiqian Sang and Xun Xu. The framework of a cloud-based cnc system. *Procedia CIRP*, 63:82–88, 2017.

[5] SK Jha et al. Comparative study of different classical and modern control techniques for the position control of sophisticated mechatronic system. *Procedia Computer Science*, 93:1038–1045, 2016.

[6] Zhen-yuan Jia, Jian-wei Ma, De-ning Song, Fu-ji Wang, and Wei Liu. A review of contouring-error reduction method in multi-axis cnc machining. *International Journal of Machine Tools and Manufacture*, 2017.

[7] Kaan Erkorkmaz and Wilson Wong. Rapid identification technique for virtual cnc drives. *International Journal of Machine Tools and Manufacture*, 47(9):1381–1392, 2007.

[8] Yoram Koren. Cross-coupled biaxial computer control for manufacturing systems. *Journal of Dynamic Systems, Measurement, and Control*, 102(4):265–272, 1980.

[9] Dumitru Popescu, Amira Gharbi, Dan Stefanoiu, and Pierre Borne. Linear identification of closed-loop systems. *Process Control Design for Industrial Applications*, pages 21–92, 2017.

[10] Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.

[11] Honghai Wang, Jianchang Liu, Xia Yu, Shubin Tan, and Yu Zhang. New results on pid controller design of discrete-time systems via pole placement. *IFAC-PapersOnLine*, 50(1):6703–6708, 2017.

[12] Stefano Longo, Eric C Kerrigan, and George A Constantinides. Constrained lqr for low-precision data representation. *Automatica*, 50(1):162–168, 2014.

[13] Loïc Bourdin and Emmanuel Trélat. Linear–quadratic optimal sampled-data control problems: Convergence result and riccati theory. *Automatica*, 79:273–281, 2017.

[14] João Marcos Kanieski, Emerson Giovani Carati, and Rafael Cardoso. An energy based lqr tuning approach applied for uninterruptible power supplies. In *Circuits and Systems (LASCAS), 2010 First IEEE Latin American Symposium on*, pages 41–44. IEEE, 2010.

[15] Trio motion technology motion coordinator technical reference manual. 2008.

[16] Mariana Ratiu and Alexandru Rus. Modeling of the trajectory-generating equipments. In *Engineering of Modern Electric Systems (EMES), 2017 14th International Conference on*, pages 216–219. IEEE, 2017.