U.P.B. Sci. Bull., Series C, Vol. 76, Iss. 2, 2014 ISSN 2286 – 3540

HIGH PERFORMANCE SOLUTIONS FOR ACQUISITION, DESIGN AND CONTROL OF A MULTIVARIABLE INDUSTRIAL PROCESS

Gabriel FLOREA[1], Ciprian LUPU[2]

In this paper our goal is to present high automated control solutions and a comparison study between pole placement method and LQG optimal control for a MIMO industrial process. All results for system identification and control will be given by robust algorithms designed in MATLAB and integrated in a unique software platform used for real-time data acquisition and process as well.

Keywords: Data Acquisition, System Identification, Automated Control, MIMO, Pole Placement, LQG

1. Introduction

The multivariable control approach starts with identification of direct transfer channels and its energy level comparative with secondary ones. Based on this preliminary analysis we can decide after the optimal control strategy. An ideal case is when MIMO proccess can be „split” in multiple independent SISO systems ignoring the indirect energy transfer between channels.

This can be done by introducing a measurement of channels coupling like RGA (Relative Gain Array) [4]:

$$k_{ij} = \frac{\Delta y_i}{\Delta u_j}, \Delta u_k = 0, k \neq j \tag{1}$$

[1]Eng., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: gabriel.florea@honeywell.com

[1] Assoc. Prof., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: cip@indinf.pub.ro

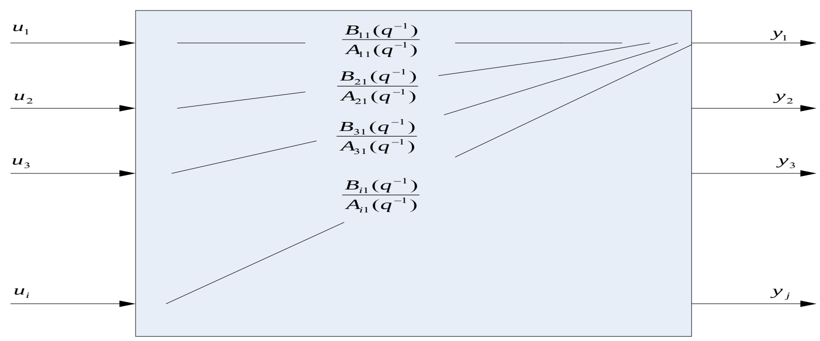

Fig. 1: Multivariable system reprezentation

where Δu_j represents the variation of two stationary imput values and Δy_i is the stationary output value when no changes applied on others outputs. Other alternative case for measure the intensity of energy transfer between input and output can be:

$$\widetilde{k}_{ij} = \frac{\Delta y_i}{\Delta u_j}, \Delta y_k = 0, k \neq i \tag{2}$$

where all others outputs remained unmodified.

For a multivariable time-discrete system with two inputs and two outputs we can use next representation for stationary gain array:

$$K = H_D(1) = \begin{bmatrix} \frac{B_{11}(1)}{A_{11}(1)} & \frac{B_{12}(1)}{A_{12}(1)} \\ \frac{B_{21}(1)}{A_{21}(1)} & \frac{B_{22}(1)}{A_{22}(1)} \end{bmatrix} \tag{3}$$

Some a priory analysis sets can drive us to one or other control strategy (decentralized or decoupling SISO controllers or multivariable optimal controllers). If direct transfer expressed by $\frac{B_{11}(q^{-1})}{A_{11}(q^{-1})}, \frac{B_{22}(q^{-1})}{A_{22}(q^{-1})}$ is more powerful then indirect ones $\frac{B_{12}(q^{-1})}{A_{12}(q^{-1})}, \frac{B_{21}(q^{-1})}{A_{21}(q^{-1})}$, or generally speaking:

$$K_{ii} = \frac{A_{ii}(1)}{B_{ii}(1)} >> K_{ij}, \forall i, j \in \mathbf{N}, i \neq j \tag{4}$$

two SISO independent controllers could be used.

This is the easiest way to perform a multivariable system control, because the process analysis and the theory behind are well known for this kind of approach (data acquisition, system identification, controller design and computation). Of course, most of the multivariable systems don't have this opportunity to be analyzed like decentralized ones and a new strategy is recommended to be used (e.g. optimal quadratic control, H_∞, etc.). In this paper we'll do a comparative closed loop performance analysis between SISO decentralized controllers and multivariable controllers (Multivariable Poles Assignment and LQG). All major aspect will be taken into consideration: data acquisition design, system identification strategy, software and hardware constraints for each case in part.

For industrial application there are standard automated solution (SCADA-PLC) provided by top suppliers like Siemens or Rockwell, which are based on a PID control strategy which cover more or less all field requirements. We would like to step forward and improve some existing practical and theoretical results (WinPIM,PC_Reg (Advantech), Matlab, LabView/Labwindows CVI) in one software platform which is capable to provide us the possibility to perform a complete system control. This tool can be used for academically propose as well as for industrial application.

2. Software platform arhitecture

As we mentioned before this software platform can be used for:

- Real time data acquisition;
- System Identification;
- Process control;

The main software application is developed in NI LabWindows CVI and is interfaced with different modules like acquisition drivers, data storage (SQL Server), identification and control algorithms (MATLAB).

Starting with the first layer, which represents the acquisition software module, we can extract data (inputs and outputs) from process sensors and actuators using different types of acquisition devices (e.g. NI USB 6008). The structure of data and the methodology depends on architecture chosen

Monovariable loops: different PRBS generators attached on each input and only one is active on a period of time. The PRBS – Pseudo Random Binary Sequence theory is presented in [6]; For each output we'll perform an acquisition experiment like:

1. Set the registry value (N), Amplifier (A), sampling time (T_e), and frequency division (d) for PRBS;

2. Start the acquisition process and collect data:

$$D = \left\langle (u_i + u_{PRBS_i}), (y_i) \right\rangle, i = \overline{1:2^N - 1} \quad (5)$$

   Note: The other inputs which have not a PRBS activated in that experiment should be on a minimum level (10%-15%) of manual amplitude;

Multivariable loop: different PRBS generators attached on each input all active for experiment. For each output we need to:

1. Set the registry values for each PRBS ($N_1, N_2, ..., N_i$), amplifier values ($A_1, A_2, ..., A_i$), sampling frequencies ($T_{e_1}, T_{e_2}, ..., T_{e_i}$), frequency divisions ($p_1, p_2, ..., p_i$) where i = no of inputs);
2. Start the acquisition process and collect data:

$$D = \left\langle \left(u_{1_i} + u_{PRBS_{1_i}}\right), ..., \left(u_{j_i} + u_{PRBS_{j_i}}\right), \left(y_{k_i}\right) \right\rangle, i = \overline{1:2^{\frac{N}{p}} - 1}\ j = \overline{1:M} \quad (6)$$

where M = no of inputs;

Note: The total number of data sets is equal with number of outputs;

Those data sets used by identification module can be stored inside a database (SQL) by the second software layer. Here we can save all the experimental result, mathematical models expressed by its polynomials coefficients or the controller parameters (RST, MIMO Pole Placement or LQG).

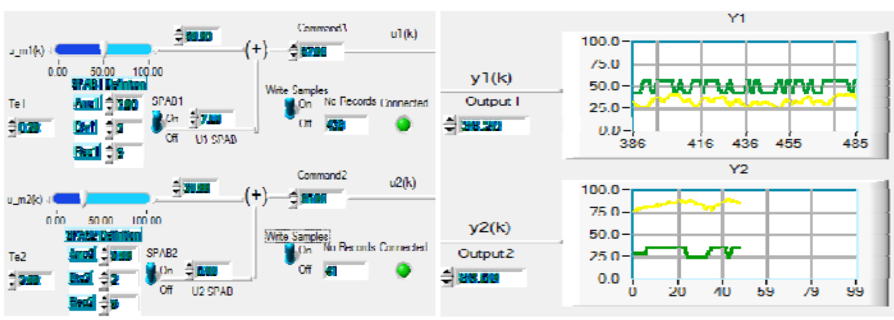

Fig.2: Acquisition module interface

The second layer contains all the function for saving and loading internally the data used after in computation module. This functionality can be extended for a remote connection (TCP/IP) between a client interface and server, where the mathematical model calculated by identification module can be saved on a remote SQL server.

In the third layer we developed the MATLAB algorithm for identification and control based on wanted approach:

1. Monovariable system identification: we consider that the mathematical model is composed only by the transfer function for direct channels:

$$H(q^{-1}) = \begin{bmatrix} \frac{B_{11}(q^{-1})}{A_{11}(q^{-1})} & 0....0 & 0 \\ 0 & & \\ . & \frac{B_{22}(q^{-1})}{A_{22}(q^{-1})} & 0 \\ . & & \\ 0 & 0 & \frac{B_{mm}(q^{-1})}{A_{mm}(q^{-1})} \end{bmatrix} \quad (7)$$

To determine each transfer function we'll call the identification module that will load the experimental data inside MATLAB environment. *Sysident* MATLAB routine will return a valid mathematical structure and model that could be represented like: ARX, ARMAX and Box-Jekins. The basic algorithm for parameter estimation used is Minimizing Prediction Errors.

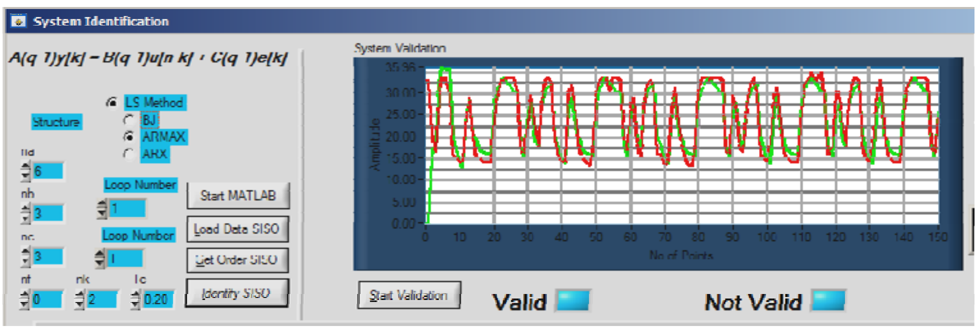

Fig.3: SISO Data Identification

2. Multivariable system identification: using the MIMO representation (Fig. 1) our goal will be to identify as many MISO subsystems as number of output we have. This case is a MATLAB approximation of MIMO systems and it's used by *SysIdent2* routine:

- ➢ $D = iddata(y_i, (u_1, u_2, ..., u_{nu}), Te_i)$ (8)
- ➢ $M = ARMAX(D, na, [nb_1, nb_2, ..., nb_{nu}], nc, [nk_1, nk_2, ..., nk_{nu}])$ (9)

$$A_{1,i}(q^{-1}) = A_{2,i}(q^{-1}) = ... = A_{j,i}(q^{-1}) = A_i(q^{-1}); i, j \in \mathrm{N} \quad (10)$$

the linear regression for ARMAX multivariable models is:

$$\mathbf{A}_j(q^{-1})\mathbf{y}_j = \sum_{i=1}^{nu} \mathbf{B}_{i,j}(q^{-1}) \cdot \mathbf{u}_i + \mathbf{C}_j(q^{-1})\mathbf{e}_j, \forall nu \in \mathbf{N} \quad (11)$$

$$\mathbf{H}(q^{-1}) = \begin{bmatrix} \frac{\mathbf{B}_{11}(q^{-1})}{\mathbf{A}_1(q^{-1})} & \frac{\mathbf{B}_{12}(q^{-1})}{\mathbf{A}_1(q^{-1})} \dots & \frac{\mathbf{B}_{1,\mathbf{ny}}(q^{-1})}{\mathbf{A}_1(q^{-1})} \\ \dots & \dots & \dots \\ \frac{\mathbf{B}_{nu,1}(q^{-1})}{\mathbf{A}_{ny}(q^{-1})} & \frac{\mathbf{B}_{nu,2}(q^{-1})}{\mathbf{A}_{ny}(q^{-1})} \dots & \frac{\mathbf{B}_{nu,ny}(q^{-1})}{\mathbf{A}_{\mathbf{ny}}(q^{-1})} \end{bmatrix}, \forall\, nu, ny \in \mathbf{N} \quad (12)$$

$$\mathbf{G}(q^{-1}) = diag \begin{bmatrix} \frac{\mathbf{C}_1(q^{-1})}{\mathbf{A}_1(q^{-1})} \\ \dots\dots \quad \dots\dots \quad \dots \\ \frac{\mathbf{C}_{ny}(q^{-1})}{\mathbf{A}_{ny}(q^{-1}} \end{bmatrix}$$

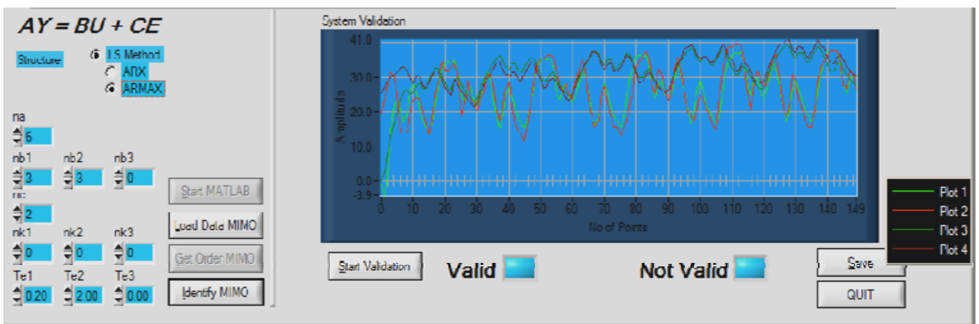

Fig.4: MIMO System Identification

Now, about multivariable control and design we'll present two possibilities in accordance with the preliminary analysis exposed in relations (1:4):

1. *Decentralized SISO* control if relation 4 is true, where for each independent direct loop we'll design an independent controller (Fig. 2) and the solution proposed is a PID with two degree of liberty (pursuit and regulation) named RST:

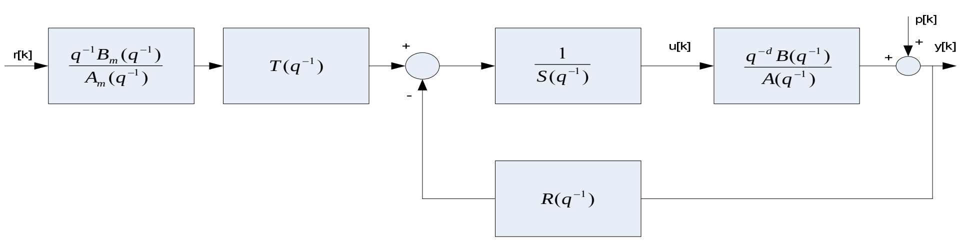

Fig.5: SISO RST Controller

For regulation polynomials ($R(q^{-1}), S(q^{-1})$) calculation we need to choose the desired poles in closed loop and specify the characteristic polynomial:

$$P(q^{-1}) = 1 + p_1 q^{-1} + \dots + p_{np} q^{-np} \quad (13)$$

Basically we start on the basis second order continuous time system and by discretization we assure next condition upon natural frequency and dumping [6]:

$$\begin{cases} 0.25 \le \omega_0 T_e \le 1.5 \\ 0.7 \le \xi \le 1 \end{cases} \tag{14}$$

At the end $R(q^{-1}), S(q^{-1})$ are the unique solutions of next equation ("Bezout Identity"):

$$A(q^{-1})S(q^{-1}) + q^{-d}B(q^{-1})R(q^{-1}) = P(q^{-1}) \tag{15}$$

Ideally, the output system should follow the reference trajectory y^*, but to avoid any command overload or saturation we define the desired behavior:

$$y^*(t) = \frac{q^{-d-1}B_m(q^{-1})}{A_m(q^{-1})} r(t) \tag{16}$$

Transfer function calculation between y^* and y will lead to our optimal choice:

$$T(q^{-1}) = GP(q^{-1})$$
$$G = \begin{cases} \frac{1}{B(1)}; B(1) \ne 0 \\ 1; B(1) = 0 \end{cases} \tag{17}$$

This method it's implemented in a MATLAB algorithm named *SISO_RST* and can be load in the main application in Get Controller option.

2. MIMO controller if the relations (1:4) are not submitted we'll using in this case a procedure of controller design based on LQG architecture and poles placement as well as LQR algorithms [11, 12]:

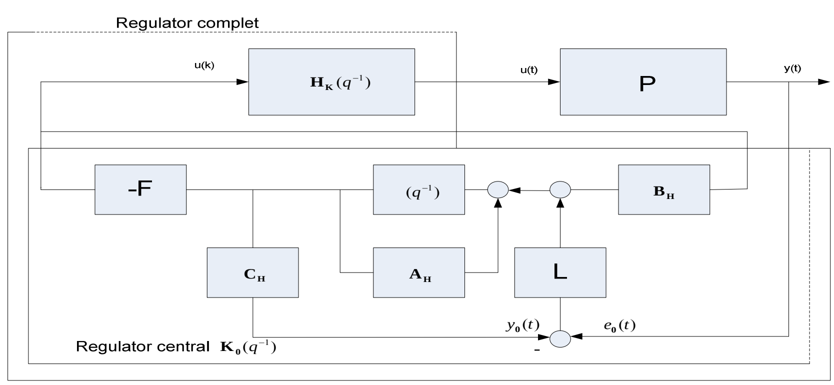

Fig.6: LQG controller structure

We'll start this analysis by imposing to the closed loop system same performances as we have done for SISO decoupled case: disturbances rejection and references tracking. For this a fixed part ($H_K(q^{-1})$- integral component) will be connected in cascade with initial system obtaining the augmented plant:

$$H(q^{-1}) = P(q^{-1})H_K(q^{-1}) \tag{18}$$

$$\begin{aligned} \mathbf{x}_H(t+1) &= \mathbf{A_H x_H}(t) + \mathbf{B_H u_K}(t) \\ \mathbf{y}(t) &= \mathbf{C_H x_H}(t) \end{aligned} \tag{19}$$

where:

$$\begin{aligned} \mathbf{x}_{H_K}(t+1) &= \mathbf{A}_{\mathbf{H}_K}\mathbf{x}_{\mathbf{H}_K}(t) + \mathbf{B}_{\mathbf{H}_K}\mathbf{u_K}(t) \\ \mathbf{u}(t) &= \mathbf{C}_{\mathbf{H}_K}\mathbf{x}_{\mathbf{H}_K}(t) + \mathbf{D}_{\mathbf{H}_K}\mathbf{u_K}(t) \end{aligned} \tag{20}$$

with state space matrix representation:

$$\begin{aligned} \mathbf{A}_H &= \begin{bmatrix} \mathbf{A}_{\mathbf{H}_K} & \mathbf{0} \\ \mathbf{B_P C}_{\mathbf{H}_K} & \mathbf{A_P} \end{bmatrix}; \mathbf{B_H} = \begin{bmatrix} \mathbf{B}_{\mathbf{H_K}} \\ \mathbf{B_P D}_{\mathbf{H}_K} \end{bmatrix} \\ \mathbf{C_H} &= [\mathbf{0} \quad \mathbf{C_P}]; \mathbf{D_H} = \mathbf{0} \end{aligned} \tag{21}$$

For this new system will design two multivariable controllers having the same state space form like:

$$\begin{aligned} \hat{\mathbf{x}}_\mathbf{K}(t+1) &= \mathbf{A_K x_K}(t) + \mathbf{B_K y}(t) \\ \mathbf{u}(t) &= \mathbf{C_K x_K}(t) \end{aligned} \tag{22}$$

with:

$$\begin{aligned} \mathbf{A_K} &= \begin{bmatrix} \mathbf{A_H} - \mathbf{B_H F} - \mathbf{LC_H} & \mathbf{0} \\ -\mathbf{B}_{\mathbf{H_K}}\mathbf{F} & \mathbf{A}_{\mathbf{H_K}} \end{bmatrix}; \mathbf{B_K} = \begin{bmatrix} \mathbf{L} \\ \mathbf{0} \end{bmatrix} \\ \mathbf{C_K} &= [-\mathbf{D}_{\mathbf{H_K}}\mathbf{F} \quad \mathbf{C}_{\mathbf{H_K}}]; \mathbf{D_K} = \mathbf{0} \end{aligned} \tag{23}$$

L and **F** matrix can be calculated using:

- Poles placement method : estimation poles p_{L_i} corresponds to the eigenvalues of the matrix ($A_H - LC_H$) and state-feedback poles p_{F_i} corresponds to the eigenvalues of matrix ($A_H - B_H F$). Using MATLAB "place" function we developed *MIMO_Place* routine that computes **L** and **F** matrix.
- LQG method: using separation principle we compute optimal Kalman estimation L_K [13]:

$$L_K = AY_K C^T (V + CY_K C^T)^{-1} \tag{24}$$

$$Y = Y^T \geq 0 \tag{25}$$

solution of difference Ricatti equation:

$$\begin{aligned} Y_{K+1} &= W + AY_K A^T - AY_K C^T (V + CY_K C^T)^{-1} CY_K A^T \\ Y_0 &= W \end{aligned} \tag{26}$$

where W represents the covariance matrix of process disturbances:

$$E\{w_d[k]w_d[k-d]^T\} = W\delta[k-d] \tag{27}$$

Optimal command is given by [W.H.T.M Aangement, 2003]:

$$\begin{aligned} u_K &= -K_K \hat{x}_K \\ K_K &= (R + B^T X_{K+1} B)^{-1} B^T X_{K+1} A \end{aligned} \tag{28}$$

where X_{k+1} is solution of difference Ricatti equation:

$$X_K = C^T QC + A^T X_{K+1} A - A^T X_{K+1} B(R + B^T X_{K+1} B)^{-1} B^T X_{K+1} A$$
$$X_N = C^T QC \tag{29}$$

We created a routine *MIMO_LQG* for computing optimal L_K and K_k by using:

DLQR and *DLQE* MATAB functions:

K=dlqr(sysHext.a,sysHext.b,Qc,Rc);

L=dlqe(sysHext.a,eye(size(sysHext.a)),sysHext.c,eye(size(sysHext.a)),eye(2));

In this section we presented some theoretical and practical aspects regarding SISO and MIMO control design for multivariable systems. It is very interesting to see some real results produced by our algorithms, so we propose in the next chapter a comparative closed loop performances analysis, pointing the software effort for implementation, as well.

3. Case study

We'll present in this chapter two complete and advanced automation solutions for a multivariable air flow and temperature system (ELWE LTR 710) with following input/output variables:

- Two inputs variables (speed of the fans and power of the heater);
- Five possible outputs, but only two of them to be measured and controlled (pressure and temperature);

For experimental data acquisition (identification and validation) we set first the sample rate for both loops using the strategy presented in relation (6) and taking into account the dynamics (fast and slow channels):

$$T_{e2} = k_2 T_{e1} ; k_2 \in \mathbf{N}$$
$$T_{e1} = k_1 T_{tact} ; k_1 \in \mathbf{N} \tag{30}$$

The PRBS sequences are:

$$u_{MAN1} = 50\%; A_1 = 10\%; R_1 = 0; D_1 = 2; L_1 = 255;$$
$$u_{MAN2} = 30\%; A_2 = 5\%; R_2 = 8; D_2 = 2; L_2 = 127 \tag{31}$$

Sample rate chosen: 0.2s for pressure loop and 2 s for temperature loop.

For the system identification point of view using the approach described by relation (12) we obtain the multivariable mathematical ARMAX model:

$$\mathbf{H}(q^{-1}) = \begin{bmatrix} \dfrac{B_{11}(q^{-1})}{A_1(q^{-1})} & \dfrac{B_{12}(q^{-1}(\ )}{A_2(q^{-1})} \\ \dfrac{B_{21}(q^{-1})}{A_1(q^{-1})} & \dfrac{B_{22}(q^{-1})}{A_2(q^{-1})} \end{bmatrix} \tag{32}$$

with:

$$
\begin{aligned}
&A_1(q^{-1}) = 1 - 0.8401q^{-1} + 0.1472q^{-2} + 0.0989q^{-3}; n_{A_1} = 4;\\
&B_{11}(q^{-1}) = 0.1153q^{-3} + 0.2094q^{-4}; n_{B_{11}} = 4;\\
&B_{21}(q^{-1}) = 0.0005q^{-3} + 0.0001q^{-4}; n_{B_{21}} = 4;\\
&C_1(q^{-1}) = 1 + 0.2542q^{-1} + 0.0247q^{-2}; n_{C_1} = 3;\\
&A_2(q^{-1}) = 1 - 1.5991q^{-1} + 0.9746q^{-2} - 0.2994q^{-3}; n_{A_2} = 4;\\
&B_{12}(q^{-1}) = -0.0505q^{-1} - 0.0335q^{-2}; n_{B_{12}} = 2;\\
&B_{22}(q^{-1}) = -0.0213q^{-1} + 0.1530q^{-2}; n_{B_{22}} = 2;\\
&C_2(q^{-1}) = 1 - 0.6599q^{-1} + 0.2901q^{-2}; n_{C_2} = 3
\end{aligned}
\tag{33}
$$

We considered for the first loop (pressure) a delay equal with 2 samples rate: $d_1 = 2 * T_{e_1} = 0.4s$;

With this model identified we'll continue by computing the multivariable controllers:

- MIMO pole placement:

Specify the poles desired for closed loop: as we explained in the 2[nd] chapter of this paper with only one additional point that we keep the initial poles of the initial system and add another 2 poles that are introduced by the integrator (relation 19).

After running *GetController* module from the main menu of application we'll obtain the multivariable controller:

$$
\begin{aligned}
u_1 &= \frac{R_{11}(q^{-1})}{R_2(q^{-1})} e_1 + \frac{R_{31}(q^{-1})}{R_2(q^{-1})} e_2\\
u_2 &= \frac{R_{12}(q^{-1})}{R_4(q^{-1})} e_1 + \frac{R_{32}(q^{-1})}{R_4(q^{-1})} e_2
\end{aligned}
\tag{34}
$$

Loading the obtained multivariable controller in the application environment and running the *Drive* the closed loop behavior becomes:

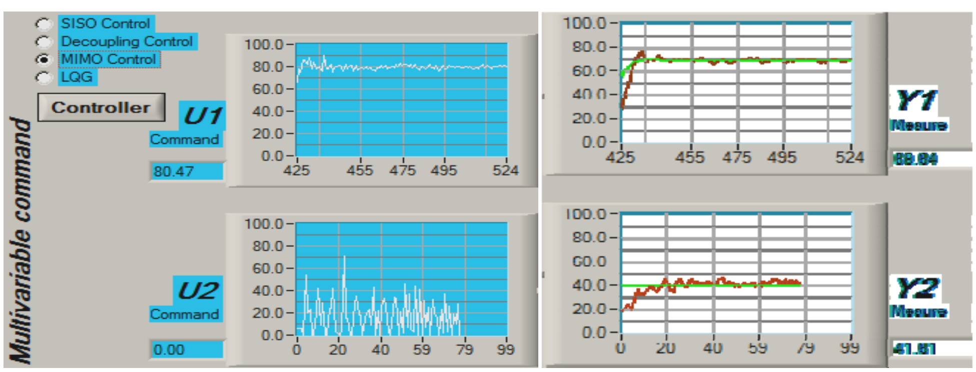

Fig. 7: Closed loop MIMO Pole Placement Control

We have stabilization times: $t_{S_1} = 4s; t_{S_2} = 20s$;

It's obvious from this picture a high level of command oscillations on both channels and in order to avoid that we'll present an optimal control like LQG;

- MIMO LQG: as described in the 2nd chapter the solution

of optimal LQG control will be given by separation principle described in our software MATLAB code. Running *GetController* module with LQG option we get:

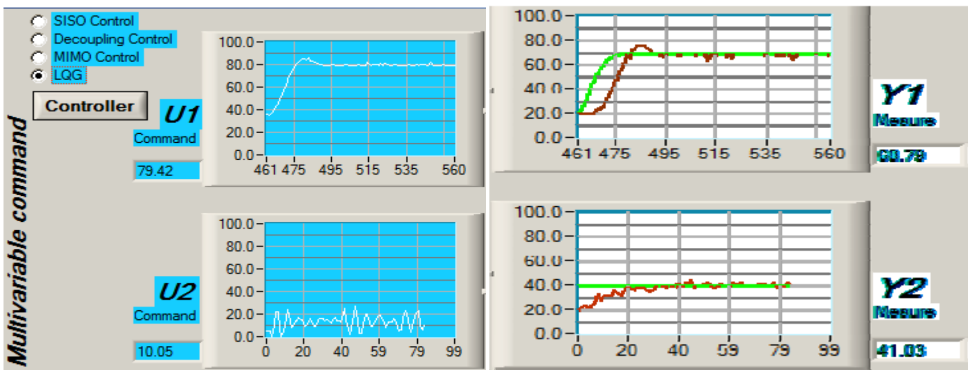

Fig.8: Closed loop MIMO LQG Control

We have in this case similar performances from stabilization time and response velocity point of view, but with less command oscillation on both channels, which can be a true advantage (low concussion at actuators level).

6. Conclusion

We have presented and described in this paper one complete software solution for real-time acquisition and control for multivariable systems. Two types of system identification analysis has been done due to the preliminary structure of the multivariable systems: decentralized or centralized. As a main topic, if the energy level of the cross transfer channels is consistently lower than direct transfer channels, we can obtain good control performances using a simple SISO RST strategy. But if this condition is not accomplished then we need to proceed by making an identification of the mathematical model using MATLAB *armax* function when the MIMO system is considered to be approximated with a class of MISO subsystems.

A MIMO feedback control comparison between poles placement method and optimal LQG has been made in the 3rd chapter where we presented the analytical results like transfer functions as the mathematical models for multivariable system (2 inputs/ 2 outputs), as well for the MIMO controller structure. In the end we extract some real-time behavior figures for the closed loop system using the drive option of the software platform.

A big advantage is the possibility to use this unique software platform for different type of experiments like data acquisition, system identification and control for both types of systems: SISO and MIMO. We developed strong and precise MATLAB algorithms which can be used in any industrial automation control in a simple manner without any other external software need. If there are special design requests, that the existing MATLAB algorithm library can not cover it, the user can develop new algorithms and integrate them without big programming efforts. In fact, this is the basic idea of this software tool to be continuously improved by user.

In the future our goal will be to improve the control part in order to integrate some robustness analysis and to complete the optimal controllers by adding $H_2$and H_∞ synthesis.

REFERENCES

[1] *Albertos P., and A. Sala*, (2004), *Multivariable Feedback Control*: Analysis and design, Springer, ISBN 1439-2232

[2] *Astrom K.J., and Wittenmark B.*, (1997): *Computer-Controlled Systems: Theory and Design*, 3rd edition, Prentice Hall, ISBN-10: 0133148998

[3] *Halvarsson B.*, (2010), Interaction Analysis in Multivariable Control Systems, Phd Thesis Uppsala Universitet

[4] *Jora B*, (2003), Sisteme automate interactive – Interactive automated systems, curs, Facultatea de Automatica şi Calculatoare, Bucuresti

[5] *Kailath T.*, (1980): Linear systems, Prentice Hall, Englewood Cliffs, ISBN 0-13-536961-4

[6] *Landau I.D., and Zito G.* (2006). Digital Control Systems - Design, Identification and Implementation, Springer-Verlag London, ISBN-10: 1846280559.

[7] *Ljung L.* (1999). System Identification 2nd edition – Theory for the user, Linkoping University, Sweden, Prentice Hall ISBN 0-13-656695-2

[8] *Lupu C., D. Popescu, B.Ciubotaru, C. Petrescu, G. Florea,* (2006). Switching solution for multiple-models control systems, Control and Automation, 2006. MED'06. 14th Mediterranean Conference on. IEEE, 2006.

[9] *Lupu C., Petrescu C., Florea G.* (2012). Solutions for Ratio Control of Nonlinear Blend Processes, 16th ICSTCC Sinaia.

[10] *Lupu C., Spinu A., Udrea A.* (2010). Intelligent precision control for air-heater systems, IFAC Workshop on Intelligent Systems, UPB.

[11] *Maciejowski, J.M.*, (1989): Multivariable feedback design – Addison Wesley Cambridge, ISBN 0-201-18243-2

[12] *Prochazka H.*, (2003): Synthese de regulateurs numeriques robustes multivariables par optimisation Convexe, Phd thesis, Grenoble.

[13] *Skogestad S., and I. Postlethwaite*, (2005), Multivariable Feedback Control: Analysis and Design, Wiley-Interscience, ISBN 978-0-470-01168-3

[14] *Stefanoiu D., Culita J., and Stoica P.* (2005). Fundamentele modelarii şi identificarii sistemelor, Ed. Printech, Bucuresti.

[15] *Wolovich W.A.*, (1974): Linear multivariable systems, Springer, ISBN 9780387901015.

*** *National Instruments*, www.ni.com

*** *MATLAB*, www.mathworks.com