Proceedings of the
2005 IEEE Conference on Control Applications
Toronto, Canada, August 28-31, 2005

Graphical Based Predictive Control Design

Luisella Balbis, Reza Katebi and Ricardo Dunia

Abstract— Graphical programming provides a suitable way to design and configure advanced control applications. This paper illustrates the configuration of graphical model predictive control applications. The system in study is controlled with linear as well as nonlinear predictive control methods, and it demonstrates the convenience of a Graphical Interface for the development of these popular control design techniques. The system built uses the LabVIEW identification, control design optimisation and simulation toolkits for design verification and deployment. The control solutions can be easily imported to a real time platform for industrial applications. The predictive control with constrains is presented in details in this paper and its performance is illustrated in a case study. The example shows how graphical interface control design increases the reliability and robustness of the controller implementation.

I. INTRODUCTION

Predictive Control has been implemented in a wide variety of industrial applications, including food processing, pharmaceuticals, chemicals, aerospace and automotive manufacturing processes [2][5][9][10][12][13]. The Predictive Control strategy makes use of a system model to calculate an optimal control profile by minimizing a cost function based on predictive model outcomes. The model variables that are used to calculate the cost function are also subject to the physical and operational constrains of the system.

From the implementation perspective, the tendency during past years has been the use of industrial predictive control software that is installed and configured in work stations. This is the case of large control applications in Refineries and Petrochemical Complexes. However, as predictive control has provided large benefits in short period of time, its application has become more popular and accessible to relatively small dynamic systems. Furthermore, as PC's became more reliable, its graphical capabilities and easy to use software has originated a driving force for process engineers to incorporate basic calculations into PC hardware. Among these calculations are simple control applications that run online with the process operations.

The configuration of predictive controllers requires the interaction and transfer of information of many pieces of code that could have been developed by different engineering and manufacturing companies during several years of research and validation. Therefore, programming code developed for system identification, control design and analysis, optimization and field target deployment are usually made in different programming languages and may not take into account the same system assumptions. This diversity of efforts difficult the interaction and maintenance of the advanced control application. In this paper, the notion of a graphical based integrated predictive control design toolkit is introduced. This advanced controller incorporates system identification, control design, simulation, verification, validation and real time implementation in a single environment under the same graphical programming language. Furthermore, PC hardware is used in all stages, including data acquisition and controller deployment.

The system identification tool is used to define pseudorandom signals of variable amplitude to perturb the inputs of the system to obtain an empirical model based on the actual output response data. The graphical interface integrates the injected signals with the measured responses to determine if enough data has been acquired for a reliable model prediction. The control design tool follows system identification and is used to analyze the identified model and provides its basic dynamics characteristics, like poles and zero locations, percentage of overshoot and settling time. The predictive controller design is also part of control design and graphically illustrates the effect of tuning parameters like prediction and control horizon, cost function weights and reference trajectory on the predictive response. A simulation node environment graphically shows the effect of nonlinear terms and constraints accounted by the optimizer. The simulation node is also used to deploy the controller into a hardware target for the real time implementation of the predictive controller.

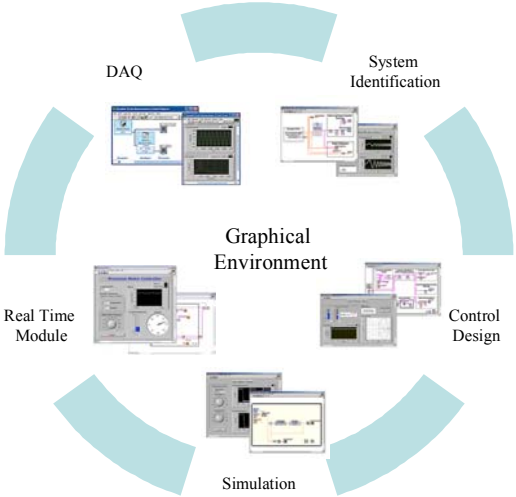

Figure 1. Integration of Predictive Control tools

Manuscript received February 28, 2005.

L. Balbis is a Ph. D. student at the Industrial Control Centre, University of Strathclyde, Glasgow, U.K. (tel: +44 141 548 2378; fax: +44 141 548 4203; e-mail: lbalbis@ eee.strath.ac.uk).

R. Katebi is a Reader at the Industrial Control Centre, University of Strathclyde, Glasgow, U.K. (e-mail: r.katebi@ eee.strath.ac.uk).

R. Dunia, Ph.D. is senior software engineer for Labview R&D at National Instruments, 11500 N. Mopac Expwy. Austin, TX 78759, USA (e-mail: rick.dunia@ni.com).

Figure 1 illustrates the integration of these pieces under the same graphical programming environment, which brings major benefit in the design and configuration of predictive controller applications. This work shows how having equivalent system models, graphical representations, timing scales and transparent process/hardware connections along the different stages of the model based controller design considerably increases the reliability and robustness of the controller implementation.

Next section provides the predictive tool overview and explains in details the control algorithm implemented in this paper. Section III shows how the different toolboxes are integrated under the same programming language. Predictive control is demonstrated in Section IV using an analog dual process simulator. Section V presents the conclusions and future work.

II. Predictive Control Tool

A. Predictive Control Algorithm

Predictive Control technology makes use of a system model and a cost function to define an optimal input profile based on future references. Therefore, the creation and validation of the system model and the realistic representation of the operating costs based on future system outputs are the first tasks in the design of a predictive controller.

The process model plays a fundamental role in output prediction along a prediction horizon interval defined by the lower and upper prediction horizon N_w and N_p, respectively. The cost function considers model outputs calculated for several samples in the future. Such predictive outputs should be based on a reliable system model in order to calculate accurately the cost function.

This research work considers a discrete state space model of the following form:

$$\begin{aligned} x(k+1) &= f\{x(k), u(k), d_p(k)\} \\ y(k) &= h\{x(k), u(k), d_m(k)\} \end{aligned} \quad (1)$$

where $f(\cdot)$ and $h(\cdot)$ are non linear functions. The vectors $x(k)$, $u(t)$ and $y(k)$ are the states, manipulated inputs and outputs, respectively. Because some model inputs are not adjustable and/or known, the vectors $d_p(k)$ and $d_m(k)$ are used to define input disturbances to the state dynamics (process noise) and to the outputs (measurement noise), respectively.

There are different ways to define the cost function [4] [15]. The most common expression is the quadratic cost function, as the optimization problem becomes convex due to positive definite conditions imposed on the weighting matrices. In predictive control, the optimal control action is obtained by minimizing a multistage cost function defined over a certain horizon. Note that the horizon considered for the adjustment of the manipulated inputs, known as control horizon N_u, is not necessary equal to N_p. Nevertheless, the causality characteristic of all physical systems imposes that $N_u \le N_p$.

The general quadratic cost used in predictive control weights the future errors between the reference trajectory and predicted plant output, $\hat{y}(k+i/k) - r(k+i)$, as well as the predicted control effort $\Delta u(k+i/k)$

$$J(k) = \sum_{i=Nw}^{Np} \left\| \hat{y}(k+i/k) - r(k+i) \right\|^2_{Q(i)} + \sum_{i=0}^{Nu-1} \left\| \Delta\hat{u}(k+i/k) \right\|^2_{R(i)} \quad (2)$$

where

$Q(i)$: Positive Semi-Definite Error Weighting Matrix.

$R(i)$: Positive Semi-Definite Control Weighting Matrix.

$\hat{y}(k+i/k)$: Vector of Predicted Output Signals.

$r(k+i)$: Vector of Future Set-Point.

$\Delta\hat{u}(k+i/k)$: Vector of Future Control Actions.

The future set point values are known or estimated over the prediction horizon. The predicted output can be calculated knowing the process model, the estimated state, the past inputs and outputs up to the current sample time k and future control signals, optimal inputs. For the prediction of the output the knowledge of the state of the plant is fundamental. In most of the cases this state can not be measured directly at the plant. Moreover, the measurements of the plant output can be affected by noise. In these cases the presence of an observer that calculates the state vector is required.

In unconstrained linear cases the optimization of the cost criteria leads to an analytical solution. However, the majority of dynamical systems are subject to constraints specified on the inputs, outputs and inputs increments, all expressed in the following form:

$$\begin{aligned} u_{\min} &\le u(k) \le u_{\max} \\ y_{\min} &\le y(k) \le y_{\max} \\ \Delta u_{\min} &\le \Delta u(k) \le \Delta u_{\max} \end{aligned} \quad (3)$$

In this case a numerical solution is obtained on line applying successive quadratic programming techniques, like Sequential Quadratic Programming, SQP.

Regardless of the cost function being considered, it is only the first element of the optimal control profile action that is fed to the plant in the form $u(k/k) = \Delta\hat{u}(k/k-1) + u(k-1)$. Then the whole cycle of output measurement, state estimation, prediction and optimization is repeated at the following sample interval.

The general structure of a model based predictive control is shown in Figure 2. Note that references as well as disturbances are the external inputs of the overall closed loop system, and that some of the disturbances could be measured and considered in the predictor model, as the dashed line in Figure 2 illustrates.

Furthermore, the tuning parameters of the predictive control are given by the predictive and control horizons, as

well as weights defined for the cost function. The optimization constrains also affect the control action profile as the potential inputs saturations and system limitations are taken into account in the optimizer.

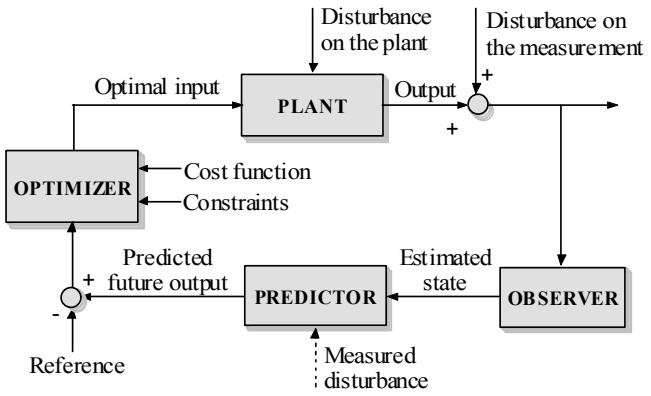

Fig. 2. Structure of the predictive controller

The different flavors of predictive control algorithms could differ [1] [6] [7], but these algorithms conserve the same global structure illustrated in Figure 2. The major features and objectives pursue in the design of the prediction control tool are stated next.

B. Contents and aim of the toolkit

The Predictive Control tool aims to provide all major features associated with predictive control, ranging from simple classical to rather sophisticated nonlinear methods that may be used on applications from process control to high speed machinery control.

From the economic standpoint, optimal conditions of manufacturing facilities are commonly found at the operational constrains of the process in study. Therefore, the reliability of an online constrained optimization package is essential for the success of a predictive controller. The accuracy of the system model is also important in the performance of a predictive controller. Most of the industrial dynamic models are data based, and measurement profiles forced by known input changes are used to build response data matrices that capture the systems dynamic characteristics.

The tool presented in this paper makes use of linear and non linear predictive control methods based on state space model formulation. The non-linear methods will be based in part on generalization of the linear algorithms to the case of state-dependent models. The design methods included are:

- Classical control with standard nonlinear component
- Multiple model nonlinear system control
- Linear Generalized Predictive Control
- Linear Quadratic Gaussian Predictive Control
- Nonlinear Quadratic Gaussian Control
- Predictive control for Hybrid Systems

Physical models based on first principle equations tend to provide nonlinear relations between forcing functions and system state responses. The control of nonlinear systems finds application in most of the engineering fields and aims to develop design procedures which guarantee that the feedback system satisfies the specified requirements not only locally, but in all the possible states region [4] [14]. The linear control methods should be considered as the starting point of any non linear analysis. Commercial software has provided reliable applications of linear model predictive control. Among the major players are DMC-Plus, from ASPEN Tech., Connoisseur by Foxboro-Invensys, and RMPCT by Honeywell. Some times nonlinear relations between variables are known in advanced and can be incorporated into a linear dynamic system. This is the case of sensors whose measurement readings are nonlinear function of the states. Other attempts to use linear models for the control of nonlinear systems are based on defining multiple linear models that switch depending of the operating regions. Gain scheduling, nonlinear model linearization at different operating regions and the use of different empirical models based on data taken at different points of operations are examples of using a family of linear functions to approximate nonlinear dynamics.

Although linear systems have been the initial modeling technology used for predictive controllers, we would like to introduce the general non-linear dynamic model to illustrate the broad scope of applications that can fit into such a type of model structure. DOT products provide a Nonlinear continuous or discrete first principle model for the predictive controller that could take a form similar to Eq.(1). An extended Kalman filter is used for nonlinear state estimation. However, graphical features of such a controller are limited because make use of a textual based programming language. Therefore, the solution proposed in this paper takes advantage of a general model representation in a graphical programming language.

III. Software Integration and Implementation

LabVIEW platform is used to develop the toolkit. The main advantage of LabVIEW is that it offers a variety of toolkits for control purposes so that the whole design process from system identification to the simulation verification can be seamlessly implemented on the same platform.

The *System Identification* toolkit allows to build the model following an experimental approach and to validate the identified model. The data extracted from the physical system is intended to excite low and high frequency responses. In low frequency responses we would like to extract information about the steady state values of the system, while high frequencies are used to obtain the initial transients of the output responses which are essential for disturbance rejection.

The *Control Design* toolkit is used to verify the properties

of the empirical model being identified. It also helps to determine if the order of the model could be reduced and to calculate dynamic characteristics of the system, like percentage of overshoot, settling time, rise time and delay.

The *Simulation environment* allows testing the implementation of the controllers design for the nonlinear system in study. The nonlinear pieces that affect the control application can be included in the simulation diagram. Examples of such nonlinearities are process constrains, input saturations and property estimators used as controlled variable.

The predictive control toolkit provides to the graphical language platform (LabVIEW) a new set of functions that accomplish the state estimation, integration, modeling prediction and optimization calculations. The analysis of these algorithms leads to identify their general requirements and to define different modules, each of them performing a different task for the algorithm under investigation. This procedural abstraction is useful to detect in first instance the main Virtual Instruments (VI's) that will be part of the toolkit. In addition, sub-routines called sub-VI's are defined and facilitates modular and hierarchical programming. Figure 3 illustrates the different modules playing a major role in the predictive controller design and implementation.

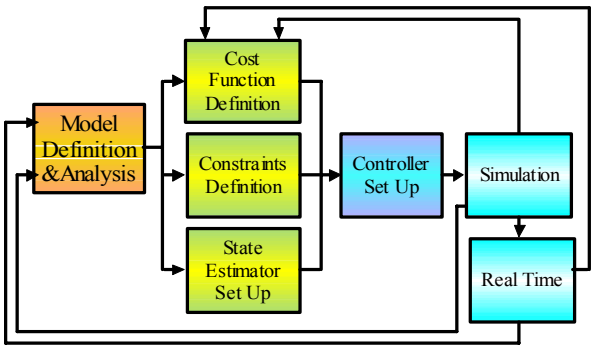

Fig. 3. Flow diagram of a predictive controller design and implementation

Existing LabVIEW toolkits are used for the model definition and analysis. For example, once a model has been built using the Identification toolkit, properties such as controllability and observability are investigated using Control Design toolkit. Appropriate VI's has been built for the constraints, cost function and controller set up. These VI's are off-line files and they have to run before the simulation starts. Once these calculations are performed the optimization algorithm starts.

From the user interface point of view, it should be mention that the predictive control toolbox is developed as future integration of the existing LabVIEW platform. Therefore the style adopted reflects well consolidate LabVIEW programming criteria, that make this product easy to learn and to use [16]

The VI for the cost function definition is shown in Figure 4, and builds a data object that contains the tuning parameter of the controller, i.e. lower and upper limits of the prediction horizon, control horizon, and the weight matrices $Q(i)$ and $R(i)$.

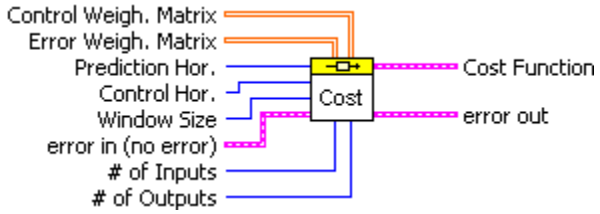

Fig. 4. Cost Function.vi

As mentioned before, an observer has to be used in order to estimate the state and predict the future behavior of the system. In Control Design toolkit a time variant and invariant Kalman filters are available. An extended Kalman filter could be easily incorporated for nonlinear state estimation. Measurement and covariance noise are required for state estimation and could be calculated using a recurrent algorithm. Furthermore, there is no need for noise signals to be Gaussian of independent.

Figure 5 shows the VI for the controller set up. It builds an object that contains all the parameters for the control law implementation. Note that the cost function parameters cluster is built in the cost function VI in Figure 4.

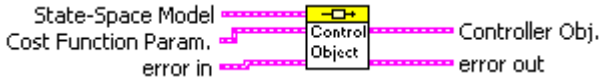

Fig. 5. Controller Object.vi

A cluster of arrays contains the constraints upper and lower limits. The cluster of constrains, together with the controller object, the reference signal, the initial input $u(0)$ and the estimation of the states, will be used by the predictive controller to implements the control law, shown in Figure 6.

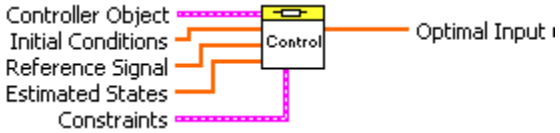

Fig. 6. Controller .vi

The controller will transform the inequalities in Eq(3) and will compute the weighting matrices for the quadratic programming solver. The main disadvantage of quadratic programming approach is that it requires solving a constrained optimization problem on line, with high computation effort.

While linear predictive control techniques have been applied successfully in a variety of real time applications, the nonlinear predictive control problem is a relatively recent topic in research. The numerical complexity of optimization algorithms for non linear models is still an open issue.

The dynamic behavior of the controller can be tested and verified by embedding the controller in the Simulation environment.

Simulation adjusts and asses the desired performance. If the performance requirements are not satisfied, the control strategy choice and modeling process shown in Figure 3, should be reviewed.

The controller will be tested in a dual circuit simulation equipment, from which online data is set and acquired. LabVIEW offers a Data Acquisition (DAQ) platform that acquires signals in real time even though they are controlled by programs running in Windows. However, there are cases in which software and operating systems must behave deterministically. For this purpose LabVIEW Real Time Module allows execution on NI RT series hardware, including RT Series Plug-in Devices, PXI embedded controllers, RT Compact FieldPoint and Compact Vision controllers. The traditional complexity of building embedded system is overcome by the simply architecture of a LabVIEW Real Time system.

On a Windows based machine the application is developed with the usual graphical approach, by simply choosing the functions needed in the application and graphically connecting them. Once the application is ready, it can be downloaded to the target processor running a real time operating system. Moreover, the modular nature of LabVIEW programming allows easily scaling from simple application to complicated control systems, as modifications and additions are fast and simple to implement.

IV. DEMONSTRATION EXAMPLE

This section shows how a complete predictive control application is built using the developed toolkit and how it can be easily carried out due to and integrated graphical environment.

The controller is a Constrained Generalized Predictive Controller in its linear form [7] [8][11]. In order to verify the accuracy of the controller performance different case studies has been proposed, among which:

- Stable Minimum Phase
- Stable Non Minimum Phase
- Unstable Minimum Phase
- Unstable Non Minimum Phase;
- System with delays.

For each the cases above a predictive controller has been design, tested and finally deployed in a real time control system.

The GPC control application for the non minimum phase stable system described next.

A. System Set Up

The system to control is a stable non minimum phase MIMO system with a delay of 0.5 second on the input and subject to input and output constraints:

$$0 \le u(k) \le 5$$

$$0 \le y(k) \le 5$$

The overall MIMO system is described by the continuous transfer function of Eq.(4). The system is formed by two output processes, Y_A and Y_B, both characterized by a gain of 0.75, and the presence of a positive zero at s=2 and the $1/(1+s)^2$ factor in the diagonal elements of the matrix transfer function. Moreover, process A presents two complex poles given by the roots of the polynomial $(1+0.25s+s^2)$. Non-zero off diagonal elements of the MIMO transfer function illustrates that the two output responses are coupled by the term $1/(1+s)$.

$$\begin{bmatrix} Y_A \\ Y_B \end{bmatrix} = \begin{bmatrix} 0.75 \cdot \dfrac{(1-0.5s)}{(1+s)^2 \cdot (1+0.25s+s^2)} & \dfrac{1}{1+s} \\ \dfrac{1}{1+s} & 0.75 \cdot \dfrac{(1-0.5s)}{(1+s)^2} \end{bmatrix} \begin{bmatrix} U_A \\ U_B \end{bmatrix} \quad (4)$$

The transfer function of Eq.(4) is discretized with sampling time 0.5 sec because the plant behavior is dominated by a pole pair with natural frequency equal to 1 sec, and converted to a state space model. Observability and Controllability properties are verified with the respective analysis VI's.

The tuning parameters chosen for the GPC algorithm are: $N_p = 30$; $N_u = 1$; $Q(t)$ = diag(100,100) ; $R(t)$=diag(10,10). Such tuning values guaranty closed loop stability response with acceptable setling time and overshoot specifications.

The plant and the measurements are affected by a disturbance, with covariance matrices H=diag(10^{-2} 10^{-2}) and G=diag(10^{-3} 10^{-3}) respectively. A Kalman Filter is used with a steady state gain calculated by the Kalman Gain.vi.

B. Simulation

The simulation model is based on the Discrete State Space.vi. and the Kalman Filter is embedded in the Simulation loop. Graph Utility functions allow display the outputs of the simulation on a chart as illustrated in Figure 8.

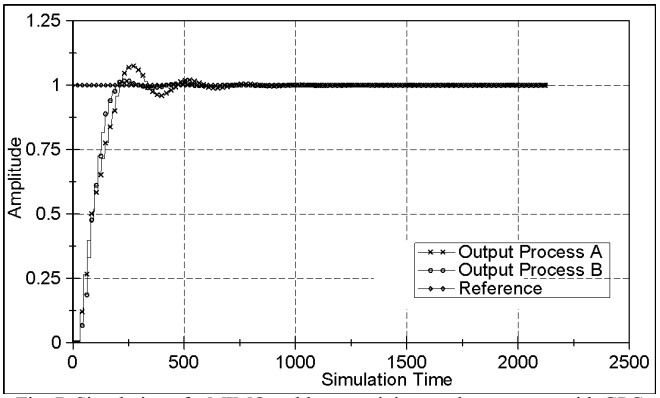

Fig. 7 Simulation of a MIMO stable non minimum phase system with GPC controller

The closed loop response of Figure 8 shows that both outputs reached the set point of r =1 simultaneously and

with very little trajectory difference as both output settling time specifications were identical. The fact that process A has complex poles makes it a little more difficult to control than process B.

C. Real Time Implementation

For the real time testing of the processes under investigation, we have used a KRi Dual Process Simulator [3]. This simulator is an analogue process simulator capable to reproduce two independent or coupled processes, stable or unstable. It could also introduce delays and noise on the processes and on the measurements. The process dynamics can be easily configured by a switch in the front panel. To perform the real-time testing, the output of the process simulator is acquired at each time step by the DAQ device and sent to the control application running in Windows.

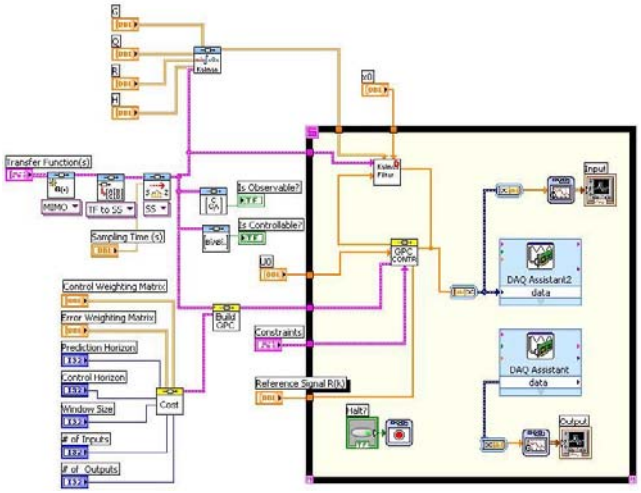

Fig 8 Block Diagram of control application

The overall block diagram for the real time implementation is depicted in Figure 8, whereas the results for the real time control are shown in Figure 9. Due to the complex poles of the system, the controller can not prevent a small overshoot, particularly in process A. However, the oscillatory, non-minimum phase dynamics are effectively dominated even in presence of constraints. The importance of having an accurate model for predictive control shows in Figure 7 and 9 that both, simulation and real time responses, are almost identical.

V. CONCLUSION

A predictive control toolbox using a graphical programming language (LabVIEW) has been presented in this work. The advantages of using a graphical based programming language for control applications have been underlined. The easy applicability of the proposed framework and the significant performance obtained using a linear predictive control algorithm has been clearly illustrated by a case study.

Future work will be the development of real time applications using PXI embedded and the exploration of LabVIEW potential for the design of Internet-based control systems.

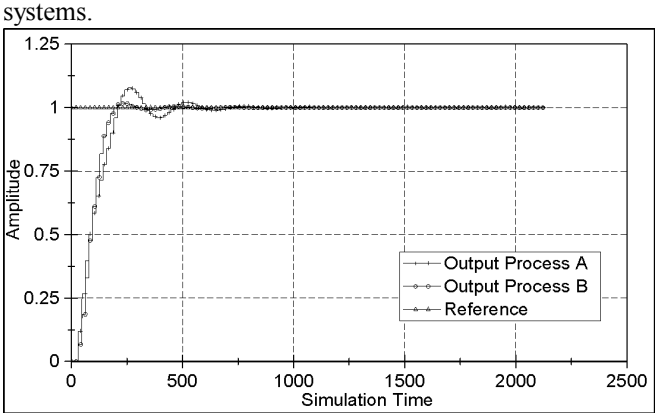

Fig. 9. Real time response of a MIMO stable non minimum phase system with GPC controller

ACKNOWLEDGMENT

The authors would like to thank National Instruments Inc. for the financial help and technical support.

REFERENCES

[1] E. F Camacho and C. Bordons, "Model Predictive Control ", London, Springer, 1999

[2] D.A. Linkens and M. Mahafouf, "Generalized Predictive Control (GPC) in Clinical Anesthesia", Dept. Automatic Cont. and System Eng., Sheffield Univ., U.K, 1994.

[3] Dual Process Simulator Model KI-101 User Guide, KentRidge Instruments, 1992-1996

[4] R. Findeisen and F. Allgower, "An Introduction to Nonlinear Model Predictive Control", 21st Benelux Meeting on System and Control, Veldhoven, 2002

[5] J.T. Wen, S. Seereeram, D.S. Bayard, ``Nonlinear Predictive Control applied to Spacecraft Attitude Control," 1997 American Control Conference, Albuquerque, NM, June, 1997.

[6] JM Maciejowski:, "Predictive Control with Constraints", Prentice Hall, 2002

[7] M. Morari and J.H. Lee, Model Predictive Control: Past, present and future", in *Computers and Chemical Eng*., vol. 23, no.4/5, pp. 667-682, 1999

[8] A.W Ordys. and David W. Clarke, "A state-space description for GPC controllers", in *International Journal of Systems Science*, vol.24, no.9, 1993.

[9] J Richalet Young, "Industrial Applicationd of Model Based Predictive Control ," in Automatica., vol. 29, no. 5, pp. 1271–1274, 1993.

[10] Allgöwer, F. and A. Zheng, editors. Nonlinear Model Predictive Control, Birkhäuser, 2000.

[11] Clarke, D and C. Mohtadi and P. S. Tuffs, Generalized Predictive Control I , The Basic Algorithm, Automatica vol 23, no 2, pp 127, 1987.

[12] Qin, S.J. and T.F. Badgwell, "An Overview of Industrial Model Predictive Control Technologies", Proceedings CPC-V, CACHE. AIChE, AICHE, pp 232-256, (1997).

[13] Gómez, J.C. and A. Jutan. "Identification and Model Predictive Control of a pH Neutralization Process based on Linear and Wiener models". 13th IFAC Symposium on System Identification SYSID2003, Rotterdam, August 2003.

[14] Rawlings, J., Meadows, E. & Muske, K. (1994), Nonlinear model predictive control: a tutorial and survey, in `Proceedings Adchem '94', Kyoto, Japan.

[15] Garcia, C., Prett, D. & Morari, M. (1989), `Model predictive control: Theory and practice-a survey', Automatica 25(3), 335-348.

[16] http://zone.ni.com/devzone/conceptd.nsf/webmain/31D31213959B38BA86256BC2006653EA?OpenDocument&node=