INDUSTRIAL AUTOMATION WITH BRAINWAVE – MULTIMAX AN ADAPTIVE MODEL BASED PREDICTIVE CONTROLLER

M. Huzmezan, [*] **T. Janiewicz and D. Meade** [**]

[*] *University of British Columbia, Electrical and Computer Engineering*
{huzmezan}@ece.ubc.ca
[**] *Universal Dynamics Technologies Inc.*
{tjaniewicz, dmeade}@brainwave.com

Abstract. Controlling a system in a model based framework means to obtain its model, design a controller and close the loop. This process can be executed online by an indirect adaptive controller. This paper describes the structure, the algorithms, the C++ implementation and the user interfaces adopted for such a controller. The applicability of this controller ranges from pulp and paper to biomedical engineering with explicit benefits in the direction of a control strategy which is characterized by: a systematic tuning procedure, reduced cross-couplings between channels and minimized closed-loop overshoot and settling time.
Laguerre orthonormal basis functions identification has been extended for multivariable systems and further used to produce a valid linear process model. This model is used at sampling time in a multivariable predictive controller to produce a control move which achieves reference tracking in the presence of measured and unmeasured disturbances and actuator constraints. This controller exhibits a modular structure allowing use on delayed multivariable (MIMO) processes with self regulating or integrating response characteristics in the context of enforceable constraints upon inputs. An analysis of the parameters involved in the controller significantly reduced them allowing for short commissioning time.

Keywords. Multivariable Control, Indirect Adaptive Control, Laguerre Identification, Model Based Predictive Control

1. INTRODUCTION

Traditional non-adaptive controllers are generally employed for most industrial process control applications. The proportional-integral-derivative controller or PID loop is especially cheap and easy to implement. Even though its operations are simplistic a PID loop can be remarkably effective at keeping the process variable close to the setpoint. But, the simplicity of the PID controller also makes it prone to failure when challenging MIMO applications dominated by delay or with integrating characteristics are to be tackled. Accounting for the multivariate aspect of the process in an adaptive fashion represents a significant challenge for most process engineers.

MIMO adaptive controllers can outperform their fixed parameter counterparts in terms of efficiency by eliminating errors faster and allowing the process to be operated closer to its constraints where profitability is highest.

Very few adaptive controllers are capable of updating their control strategies on line (i.e. while the process is running). In fact only a careful look at these will entitle the user to define the extend to which adaptation is allowed to chance the process model or directly the controller parameters on line. Only recently adaptive controllers have been accepted as reliable and trustworthy by the plant's engineers who only now are convinced that, for instance, the multivariable characteristic of the controllers they have selected works as promised. Some examples of such commercial controllers, in alphabetical order, are:

- EXACT and Connoisseur from The Foxboro Company (www.foxboro.com)
- BrainWave from Universal Dynamics Technologies (www.brainwave.com)
- CyboCon from CyboSoft (www.cybocon.com)
- DMCPlus from Aspentech (www.aspentech.com)
- INTUNE from ControlSoft (www.controlsoftinc.com)
- KnowledgeScape from KnowledgeScape Systems (www.kscape.com)
- RCMPT from Honeywell (www.honeywell.com)

Usually some kind of manual or automatic identification operation is an indispensable first step towards effective adaptive control. Getting an initial model is still a challenge since the operator expectations is to have them in closed loop at the time of purchase. Some controllers are educated by the operators based on hints developed on existing knowledge or assumptions about the process. Others are trying to answer these questions themselves by conducting empirical tests on the process before start-up. Therefore weighted network of orthonormal Laguerre functions, radial basis function models, multilayer perceptron artificial neural network or ARX models are hidden behind the scene, allowing the user to concentrate on the essential problems related to the process.

The other issue is how the controller should combine its observations with the information supplied by the operators to design its own control strategy. There are many answers to that question but most of them fall into one of three basic categories: i) indirect, model-based adaptive control; ii)direct adaptive control or iii) rule-based or artificially intelligent adaptive control

For example, some controllers are relying on a process model to create a suitable control law, but it uses several rules to determine when it is likely to have sufficient data to create the model correctly. Others can be configured to make use of a process model as well or constructing it just for informational purposes. Some controllers are using the process model as the basis for designing their own control law but are taking an expert systems approach to create it.

In terms of effectively computing the future control move most these controllers are using the plant model information, current inputs, outputs and measured disturbances, as well as input and output constraints, to solve a Linear Programming (LP) or a Quadratic Programming (QP) problem, respectively. While doing this to ensure the feasibility of the optimization problem a soft constraint approach is usually taken by assigning weights on each output which are increased as it approaches its constraint. With the same problem in mind some of these controllers are adopting a zone type approach rather than at specific set points tracking. When optimization is involved, to allow for different degrees of complexity and hence speed of computation, the afore mentioned controllers are taking into account hard input constraints ignoring the output constraints or they look at all constraints as active but assimilate them as soft constraints.

In the context of other commercial adaptive controllers the main contributions of this development are focused on: i) the extensions of the Laguerre orthonormal basis functions identification for multivariable systems and its employment for the identification of a valid linear process model; ii) the model augmentation with a disturbance model and its use at each sampling time in a multivariable predictive controller to produce the control move which achieves reference tracking in the presence of measured, unmeasured disturbances and actuator constraints iii) the modular structure allowing for different MIMO processes such as delayed multivariable with self regulating or integrating response characteristics iv) the C++ development and v) the development of an adequate graphic user interface (GUI) answering the requirements of process control engineers such as user-friendliness.

In Section 2 of this paper the controller architecture and features are disclosed. For a complete understanding of its implementation in Section 3 we present the essential details on its internal model followed by insights on the control computation in Section 4 and ways to optimize its calculation in Section 5. The utility of the MIMO adaptive predictive controller is illustrated through a brief study in Section 6 followed by conclusions in Section 7.

2. CONTROLLER ARCHITECTURE AND FEATURES

The MIMO advance predictive controller introduced in this paper is capable of controlling up to

12 cross-coupled loops or combinations thereof. Its design is described in detail in (Huzmezan, 1998). The controller bases its future control moves on models and requested set points. From the initial estimates, internal models of the process are built to predict the behavior of the controlled outputs. Several prediction and control parameters, detailed in the next sections of this paper, are provided to optimize the performance. The estimated dynamics can be refined further by the use of process identification features. In the case of non-linear plant dynamics, common during various phases of production, different models can be configured and triggered automatically by an external module.

The controller is a multi-threaded software, expanded to a MIMO version from the previous SISO commercial product. To understand more the capabilities of the controller, the control thread flowchart is provided in Figure 1.

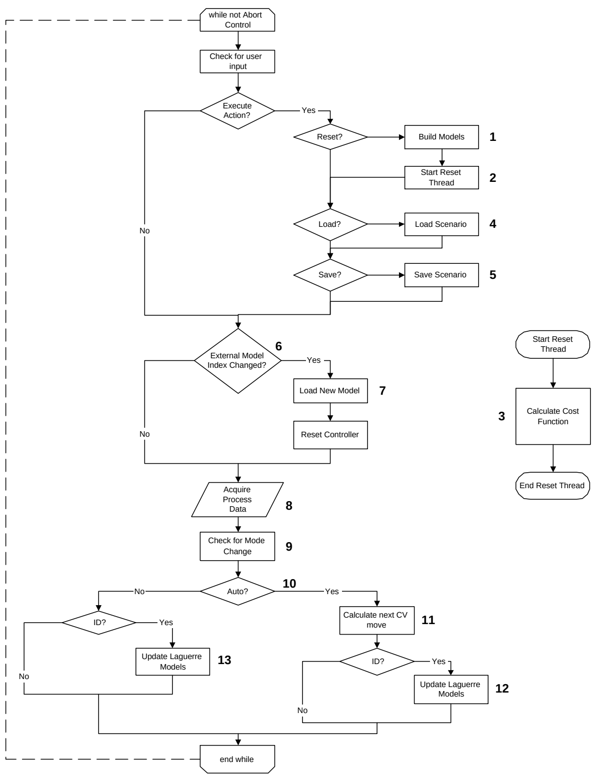

Fig. 1. MIMO Brainwave Controller Thread

After a successful setup of the controller, which will be discussed in relation to the user interface, a control loop is executed. Inside the controller main thread, various checks are in place triggering execution of requested actions. Action flags are triggered through the user interface. There are four general actions set by the user: i)*Reset* which triggers *model building (1)* and *cost function calculation(3)*; ii) *Load* which will *Load Scenario (3)*- a pre-defined set-up of models and tuning parameters; iii)*Save* which will save the current set-up as a *Scenario (4)* and iv)*Mode Switch* which switches from manual control to *Auto (10)*. For multi-modelling support an *External Model Index (6)* may be provided (inside the PLC or other connecting device) which will trigger a new scenario to be loaded. There are up to 20 scenarios that can be configured, each with different process estimates and tunings. At this point the controller will *acquire process data (8)* (i.e. the values for all inputs, all outputs and all measured disturbances). Depending on the user's choice, the controller will enter either *manual* or *auto (10)* mode. In the *manual* mode the controller monitors the process without influencing it. If identification is enabled, the values read are used to *update off-line Laguerre models(13)* using the identification algorithm described in Subsection 3.3. In the *auto* mode, the values read from the process are used to *calculate the next CV move (11)* based on the algorithm presented in Section 4. If, at the same time, identification is enabled, an identification algorithm is used to *update on-line Laguerre models (12)*. The above sequence is then repeated at every sample time, which is only restricted by the execution time of the controller main thread (about 50 ms for an average complexity system on a 1GHz Intel^{TM} processor).

3. THE CONTROLLER INTERNAL MODEL

The models in Braiwave are built using Laguerre series, an orthonormal set of functions defined as in continuous time as:

$$f_i(t) = \sqrt{2p}\frac{e^{pt}d^{i-1}}{(i-1)!dt_{i-1}}[t^{i-1}e^{-2pt}] \qquad (1)$$

where i is the order of the function and p is the time scale. See (Dumont and Zervos, 1986) for more details. A model is built by choosing appropriate linear combination of the first N Laguerre functions and integrated over some interval $[a, b]$.

3.1 *Self-regulating processes*

In the case of Braiwave a discrete representation of the Laguerre functions has been used as shown below.

$$l(k+1) = al(k) + bu(k) \qquad (2)$$

where $l(k)$ is the state vector, $u(k)$ is the input. The matrix a is a lower triangular, $n \times n$, with the property that every diagonal or sub-diagonal has the same number:

$$a = \begin{bmatrix} \tau_1 & 0 & \dots & 0 \\ \frac{-\tau_1\tau_2 - \tau_3}{T} & \tau_1 & \dots & 0 \\ \vdots & & & \\ \frac{(-1)^{N-1}\tau_2^{N-2}(\tau_1\tau_2+\tau_3)}{T^{N-1}} & \dots & \frac{-\tau_1\tau_2 - \tau_3}{T}\tau_1 & \tau_1 \end{bmatrix} \quad (3)$$

whereas b is defined as

$$b^T = \begin{bmatrix} \tau_4 & -\frac{\tau_2}{T}\tau_4 & \dots & (-\frac{\tau_2}{T})^{N-1}\tau_4 \end{bmatrix} \quad (4)$$

where

$$\tau_1 = e^{-pT}$$
$$\tau_2 = T + \frac{2}{p}e^{-pT} - 1)$$
$$\tau_3 = -Te^{-pT} - \frac{2}{p}(e^{-pT} - 1)$$
$$\tau_4 = \sqrt{2p}\frac{1-\tau_1}{p}$$

The output is approximated by a weighted sum of the state vector $l(k)$

$$y(k) = cl(k) \quad (5)$$

where $c = \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix}$ is a vector representing the Laguerre network coefficients. In case of a time varying system c becomes $c(k)$. An $m \times p$ MIMO system can be represented as an array of state equations (2), one for every input u_i, $i = 1, \dots, m$ and an array of output equations (5) one for every output y_j, $j = 1, \dots, p$:

$$L(k+1) = AL(k) + BU(k) \quad (6)$$

where

$$L(k) = \begin{bmatrix} l_1(k) & l_2(k) & \dots & l_m(k) \end{bmatrix}^T$$
$$U(k) = \begin{bmatrix} u_1(k) & u_2(k) & \dots & u_m(k) \end{bmatrix}^T$$
$$Y(k) = \begin{bmatrix} y_1(k) & y_2(k) & \dots & y_p(k) \end{bmatrix}^T$$

A is a block diagonal matrix with a on the diagonal repeated m times, similarly for B while C is of the form:

$$C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \vdots & \vdots & \dots & \vdots \\ c_{p1} & c_{p2} & \dots & c_{pm} \end{bmatrix}$$

Note that, due to the orthonormality of the Laguerre network, a and b can be the same for every state equation, thus the number of individual state vectors is m (no of inputs) rather than $m \times p$. This reduces the number of flops from $(m+p)^2$ to m^2 when computing predictions. Furthermore, this representation makes the system fully observable and controllable, and the state reduction procedures such as minimal realization are unnecessary. The MIMO system has the following block representation

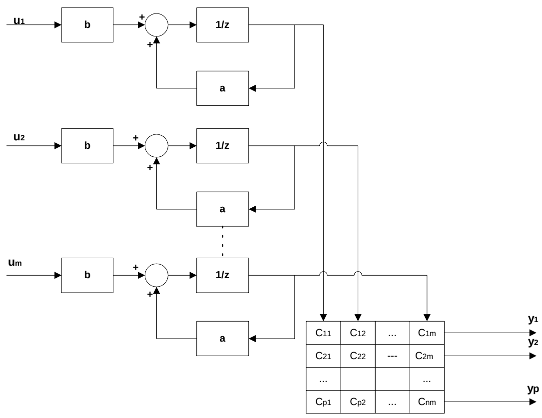

Fig. 2. Multivariable Prediction System

This multivariable model is used to predict the system's response. Predictions are made by iterating the above state space equations N_2 steps ahead:

$$Y(N_2) = CL(N_2)$$

To arrive at $L(N_2)$ we iterate the state vector L

$$L(2) = AL(1) + BU(1)$$
$$L(3) = AL(2) + BU(2) = AL(A(1) + BU(1)) + BU(2)$$
$$L(4) = AL(3) + BU(3) =$$
$$A(A(AL(1) + BU(1)) + BU(2)) + BU(3)$$
$$\vdots$$

$$L(N_2) = A^{N_2-1}L(1) + A^{N_2-2}BU(1) + \dots$$
$$+ABU(N_2-1) + BU(N_2) \quad (7)$$

In order to make the controller more or less aggressive it can assumed that only inputs u_1 to u_{N_u} $N_u < N_2$ will change while all others will remain zero.

3.2 *Integrating processes*

As defined above the matrices A, B and C are the model state space representations characterizing a stable (i.e. self-regulating) process. When the process is marginally stable the matrices are augmented with an integrator (i.e. poles on the unit circle). The controller assumes that if an output has an integrating response to one input, then it has an integrating response to all inputs. This is a safe assumption, since a self-regulating disturbance is trivial to reject with an integrating input. Consequently in the model of the system the integrators are placed at the output of the signal. In Figure 2 the integrators would be placed between the C matrix and the corresponding outputs y_i.

3.3 *Model Learning*

The on or off-line model learning capability establishes this software as an adaptive controller. The process model may be either refined or even built from scratch. Model identification is performed using recursive least squares. By using the variance of the states and the prediction error, the best possible fit is identified so that the prediction error is driven to white noise. The algorithm that identifies the vector of Laguerre network coefficients for a SISO cell within the MIMO system, as shown in (M.E. Salgado and Middleton, 1988), is characterized by the following equations:

$$c(k+1) = c(k) + \frac{\alpha P(k) l(k+1)}{\lambda + l(k+1)^T P(k) l(k+1)} e(k)$$

$$P(k+1) = \frac{1}{\lambda}\left[P(k) - \frac{P(k)l(k+1)l(k+1)^T P(k)}{\lambda + l(k+1)^T P(k) l(k+1)}\right] + \beta I - \delta P(k)^2$$

where α, λ, β and δ are constants, P(0) is a diagonal matrix with 10s on the diagonal and $e(k)$ is the prediction error.

4. BUILDING THE CONTROL MOVE

The ability to predict the future response of the process allows the controller to come up with control moves which will effectively keep the process at set point even for challenging processes with long time delays and integrating characteristics. The controller adjusts its moves based on the error between the actual output of the process and the desired output. Figure 3 shows the observer which is implemented to monitor and adjust the states of the model based on the prediction error between the predicted and actual output. The error can be magnified or reduced by adjusting the *observer gain matrix* appropriately . This is equivalent to an observer pole placement, see Figure 3.

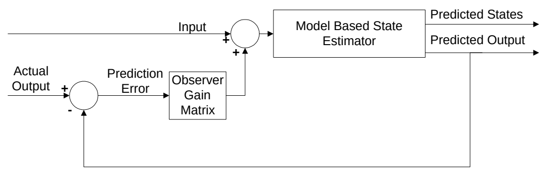

Fig. 3. The observer

The control moves are determined by looking at the predicted future error, which is the difference between the predicted future output and the desired future output (reference). The user can specify the region over which these error values will be summed. The region is bounded by the *initial* (N_1) and *final* (N_2) *prediction horizon*. The sum of the squared predicted errors inside the prediction interval can be written as:

$$\sum_{i=N_1}^{N_2} ||\hat{Y}(i) - \hat{S}(i)||^2$$

where $\hat{Y}(i)$ is the predicted output at update i and $\hat{S}(i)$ is the reference at update i.

It is also possible to set the number of control moves that the controller will take to get to the set point by adjusting a parameter called *control horizon* (N_u). Squaring and summing the changes in the control moves gives:

$$\sum_{i=1}^{N_u} ||\Delta U(i)||^2$$

A norm becomes a weighted norm if a weighting matrix is introduced and multiplied with the vector, thus for a given vector a, $||a||_Q^2 = a^T Q a$. If we weigh the error and the movement with weighting matrices Q and R respectively, and add the two sums, we arrive at the cost function employed within the Brainwave Multimax controller:

$$J(\Delta u) = \sum_{i=N_1}^{N_2} ||\hat{Y}(i) - \hat{S}(i)||_Q^2 + \sum_{i=1}^{Nu} ||\Delta U(i)||_R^2$$

The tuning matrices Q and R allow greater flexibility in the solution of the cost function. Q penalizes the error while R penalizes the movement.

The above cost function is optimized with respect to ΔU. By differentiating and solving for ΔU the next set of optimal control moves is obtained. The cost function, also called *optimizer and predictor* bases its calculations on state vectors predicted by the observers, see Figure 4. Input constraints are implemented via a multivariable anti-windup scheme which was proved to be equivalent to an on-line optimization for common processes (Goodwin and Sin, 1984).

Once the cost function is built it remains the same until we change one of the models, Q, R, or one of the horizons. At each update, the following takes place: i) the state vector is estimated and adjusted; ii) the output is estimated and iii) the next move is calculated.

Measured disturbances prediction is similar to the primary process prediction. For the sake of compatibility with the previous version of the controller, measured disturbances are assumed to affect one output at a time. There can be up to three measured disturbances per output in the current controller structure.

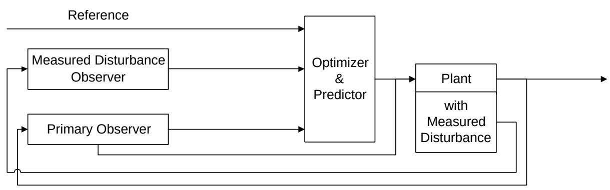

Fig. 4. The controller with the plant in the loop in the presents of measured disturbances

5. OPTIMIZING THE COMPUTATION

Building the cost function is a computationally intensive process. Matrices of dimensions in the hundreds or thousands are multiplied. Due to the fact that most of these matrices are more than 80% sparse, a sparse matrix structure has been developed. In the structure only the non-zero entries of the matrix are stored in the form of a link list, as shown in Figure 5, together with the entry's row and column position. In most cases this is a significant memory and time saver. The A matrix:

$$A = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & a \end{bmatrix} \quad (8)$$

for a 4×4 system is a 60×60 block diagonal matrix, as described in equation (6). Its diagonal consists of four 15×15 lower diagonal a matrices each with only 120 non-zero elements. Therefore, the A matrix has then $60 \times 60 = 3600$ elements, $120 \times 4 = 480$ non-zero. Thus A is 87% sparse. If

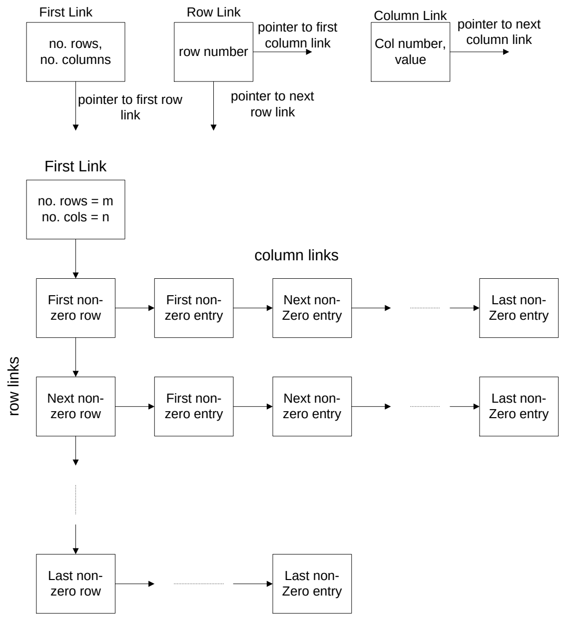

Fig. 5. The sparse matrix structure

we now try to predict the movement, say 41 steps ahead, then according to equation (7), the A^{40} matrix will have to be computed. This can lead to intense computations given that A grows with the system's dimension. Squaring A the conventional method leads to $60 \times 60 \times 60 = 216,000$ multiplications A^{40} is 39 times (i.e. 8,424,000). With the sparse matrix structure only the non-zero elements get multiplied. It can be shown that squaring the a matrix involves 680 non-zero calculations (the reader can verify that by taking into consideration the lower triangular structure of a the number of multiplications to compute a^2 is $\sum_{i=1}^{i=15} \sum_{j=1}^{i} j = 680$. From equation (8) can be seen that squaring A will involve squaring only the block diagonal, that is $680 \times 4 = 2,720$ calculations, obtaining A^{40},therefore, will take $2,720 \times 39 = 106,080$ calculations, a 98.8 % decrease from the original 8,424,000! The drawback of this scheme is the relative difficulty of finding the specified element in the link lists, however the considerable reduction in calculations far outweighs that drawback. In the end the calculation time is decreased tenfold or more. The implementation of the sparse matrix structure is an essential feature which makes the controller practical.

6. CONTROLLER PERFORMANCE

The MultiMax controller can be configured for processes of up to 12 inputs and 12 outputs. This would result in potentially 144 models to be built for the CV-PV relationships, plus up to 36 feedforward models (3 feedforwards per PV). An example will be shown here for a 2 input, 2 output process consisting of both and integrating and self-regulating type process. The relationships configured for this example are:

	PV_1	PV_2
CV_1	integrating ($dt = 5$, $\tau = 5$, $gain = 0.03$)	self-regulating ($dt = 10$, $\tau = 5$, $gain = 1$)
CV_2	integrating ($dt = 15$, $\tau = 5$, $gain = 0.015$)	self-regulating ($dt = 5$, $\tau = 5$, $gain = 1$)

This process simulation results in 4 models being configured in the controller, each identified either from open loop step changes, or from closed loop control process identification. In the example here, the process estimates are determined from open loop step changes. Owing to the cross coupling between the processes, only 2 step changes are required since both channels react to changes in either of the control variables.

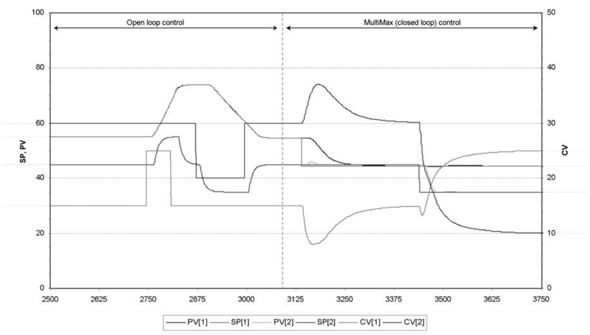

Fig. 6. Multimax Control

Figure 6 shows how this system works in both open loop and closed loop control with BrainWave - MultiMax. It can be seen that any changes to either of the actuators will result in both integrating and self-regulating process responses in the two channels. These open loop step changes allow accurate model identification. The control in open loop also demonstrates the cross coupling between the various channels, together with their characteristics.

Controlling the system with the MultiMax controller, configured with a control horizon of 5 updates, an *initial prediction horizon* of 20 updates and a *final prediction horizon* of 40 updates results in tight set point control in both channels. The tuning of the Q and R matrices are straightforward. In the case of the Q matrix, an identity matrix has been found most appropriate; for the R matrix, the tuning is equal to

$$R = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$$

Finally, the *observer gain matrix* for this example was equal to

$$K = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

With this tuning, very little overshoot of set point is also observed following significant set point changes, in addition to tight control of the set point in the unchanged channel. The example shows set point changes for both channels with the associated control moves that have been implemented by the controller based on its internal model estimates and tunings.

Attention was paid to the development of a graphic user interface (GUI) that answers the requirements of process control engineers involved in a sum of industries. The real time implementation is achieving sample times as low as 0.1 [s] for commonly encountered medium size MIMO systems. A strict management of the controller features is employed to ensure its user-friendliness.

A snapshot from the controller face plate shown in Figure 7 is giving a flavor of the way the setup window blends in the *trender* for the process responses together with the model viewer which allows the user to supervise the accuracy of the online identification process.

Based on the control objectives of the operator, the tunings of the controller can be adjusted to maintain tighter set point control at the expense of more actuator movement, or to limit actuator movement at the expense of set point control. Additionally, changes to the *observer gain matrix* will affect set point tracking at the expense of unmeasured disturbance rejection. The numbers used in the simulation shown here represent fairly conservative numbers for the tuning parameters.

7. CONCLUSIONS

The advanced model based predictive controller BrainWave MultiMax was developed in a modular structure for use on multi-input/multi-output MIMO processes with possible integrating responses, exhibiting long delays and time constants. The controller was developed and tested using a flexible Matlab Simulink test-bed, which enabled us to do most of the preliminary testing in this environment and subsequent implementation in C++. A thorough analysis of the parameters involved in the controller provided golden rules for a number of tuning parameters, dramatically reducing the commissioning time. Particular attention was paid to the development of the graphic user interface (GUI) in order to make it user-friendly. The implementation in real time of the controller can

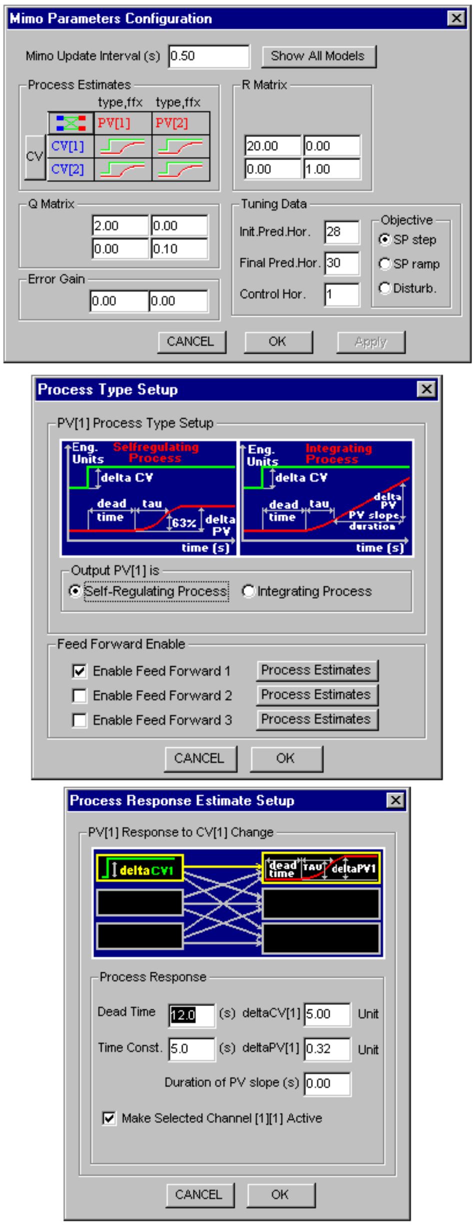

Fig. 7. A snapshot of the controller face plate reflecting the tender, model viewer and controller setup windows

achieve sample times as low as 0.1 [s] for a medium size MIMO system.

The applicability of this controller ranges from pulp and paper to biomedical engineering. The main benefits of this control strategy are: a systematic tuning procedure, reduced cross-couplings between channels and minimized closed-loop overshoot and settling time. Prohibitive costs of predictive control and a low number of commercially available multivariable adaptive controllers prevent industries that operate on small profit margins to take advantage of such technology. This is exactly the niche market this controller is targeting.

In terms of applications the paper shows the benefits of the aforementioned controller to a challenging multivariable process characterized by strong coupling between channels, significant delays in all channels and an integrating characteristic.

8. REFERENCES

Dumont, G.A. and C.C. Zervos (1986). Adaptive Controllers based on Orthonormal Series Representation. In: *Proceedings of the* 2_{nd} *IFAC Workshop on Adaptive Systems in Control and Signal Processing, Lund, Sweden.*

Goodwin, G.C. and K.S. Sin (1984). *Adaptive filtering, prediction and control.* Prentice-Hall Inc.

Huzmezan, M. (1998). Theory and Aerospace Applications of Constrained Model Based Predictive Control. PhD thesis. University of Cambridge.

M.E. Salgado, G.C. Goodwin and R.H. Middleton (1988). Exponential Forgetting and Resetting. *International Journal of Control* **47**(2), 477–485.