FROM IDENTIFYING AND MODELING TO CONTROL: THE MULTICONTROLLER, AN ADVANCED CONTROL LOOP STRATEGY FOR CRYOGENIC SYSTEMS AT CERN

Sébastien Cabaret[1, 2]

[1]UPJV
33, rue St Leu, 80039 Amiens Cedex 1, France
[2]CERN
CH-1211, Geneva 23, Switzerland

Sebastien.cabaret@cern.ch

Abstract

CERN's (the European Organization for Nuclear Research) new challenge, the Large Hadron Collider (LHC) will produce particles collisions inside four dedicated experiments with which they will carry out physics research. Cryogenic controls have developed a framework named UNICOS (Unified Industrial COntrol System). It is an object oriented development for industrial process control technologies based on PLC-SCADA solutions (Programmable Logic Controller). The MultiController object has been integrated in the latest UNICOS framework to offer various advanced control loop strategies. It gives to the user a series of advanced control algorithms: Smith Predictor, PFC, RST and GPC. Additionally the MultiController offers full tuning possibilities via a Human Machine Interface (HMI). Process identification is a key point to elaborate the control signal. An advanced tool suite named 'Advanced Automation Tool Kit' allows several control functionalities to work in a PLC environment. It provides PLC objects for system simulation, online system identification, and online system recording processes. The MultiController combined with the 'Advanced Automation Tool Kit' gives to the process engineer a complete solution to tune a system with advanced controllers.

Keywords: Model Identification, Advanced Control Algorithms, Programmable Logic Controller, Human Machine Interface.

Presenting Author's biography

Sébastien Cabaret is a PhD student at CERN for the IT-CO (Information Technology – COntrol) group. Sébastien Cabaret has first been graduated in an Engineering French school and has worked for the RAL (Rutherford Appleton Laboratory) in the U.K. as a control engineer for the Proximity Cryogenic System of an Experiment named ATLAS at CERN. Since 2005, he is attached at CERN laboratory to accomplish its doctorate in advanced control applied for Programmable Logic Controllers.

ISBN 978-3-901608-32-2

1 Introduction

CERN's (the European Organization for Nuclear Research) new challenge, the Large Hadron Collider (LHC) will produce particles collisions inside four dedicated experiments with which they will carry out physics research.

Cryogenic technologies are used by LHC projects (superconductive magnets) to allow physicists to go deeper into their theory. In order to have a high energy Collider, the CERN Cryogenic control people provides extreme thermal conditions to the detectors and the LHC by means of an efficient control system philosophy. In this way they have initiated an industrial based framework named UNICOS. The UNICOS framework is an object oriented development for industrial process control technologies based on PLC-SCADA solutions [1] [2].

The first UNICOS framework version was able to provide PID control loops. The PID controller is simple and reliable and is able to solve up to 80 per cent of control loop systems. In cryogenics environments the PID controller is sometimes insufficient to solve critical problems (inverse response, long dead times, non-linear systems).

This paper describes new advanced control features developed for PLC-SCADA solutions at CERN.

The paper starts by presenting the design of the MultiController object for the UNICOS framework. It introduces the strategy to use a monolithic object design for multiple control algorithms and the corresponding HMI representation. After it explains the 'Advanced Automation Tool Kit' developed for the modeling, identification and validation. Thereafter, this paper gives the results of a pressure regulation setup for Schneider PLC used with UNICOS. Finally, it shows the advanced control algorithms proposed through the MultiController object and its results.

2 MultiController Object Design

Due to the need of advanced control loop strategies the MultiController object has been designed. It is an object programming solution for PLCs and SCADA systems and offers many advantages in terms of usability, functionality and extensibility [3]. The object design is the result of requests from multiple users and previous experience with existing PID controllers.

2.1 Multiple algorithms in a unique monolithic object – a simple object evolution

The MultiController object has a single interface for all regulation algorithms. The object structure is implemented with a set of parameters used for all possible algorithms (*Fig. 1*). The way the parameters are treated is dependant on the selected regulation method. The same parameters can be used differently by each advanced control strategy. This design allows the addition of new control loop algorithms without changing the object interface.

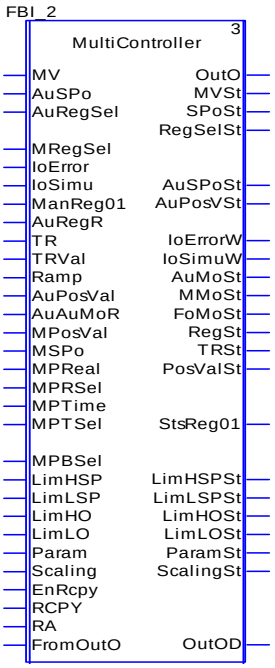

Fig. 1 – MultiController object in a Schneider PLC

2.2 An efficient tuning mechanism and a unique HMI with different views

The object development process is a twofold task. On one hand it consists of building a PLC object with the core implementation of the algorithms. On the other hand it deals with the HMI and its possibilities in terms of tuning and parameterized options [4].

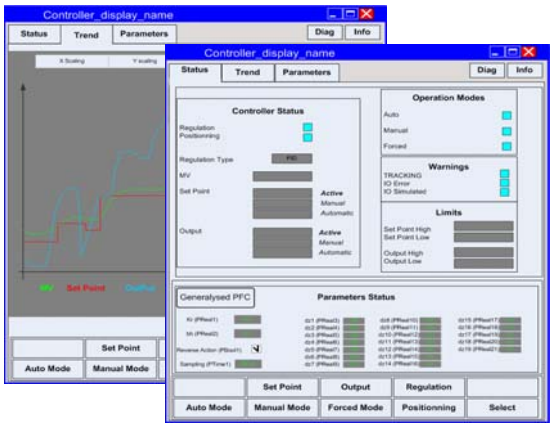

Fig. 2 – MultiController faceplate (status and trend views)

The object programming approach of the MultiController through the SCADA schema is a single monolithic representation by means of a custom faceplate, a unique set of trends, and a unique recipe mechanism. It allows for a global control of the regulation loop via one centralized object representation in the HMI using different views *(Fig. 2)*.

ISBN 978-3-901608-32-2

3 Advanced Control Algorithms in MultiController Object

The MultiController object has the following advanced control algorithms implemented: Smith Predictor, RST, PFC, and GPC (PID algorithm is also integrated into the MultiController, but is not considered as an advanced feature of the object).

3.1 Smith Predictor

3.1.1 Classical Smith Predictor structure

The Smith Predictor has been proposed [5] to compensate systems with long dead-times. It consists of finding a fictive structure *(Fig. 3)* so that the delay is concealed from the closed loop system.

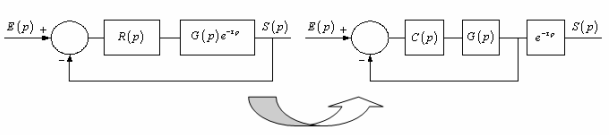

Fig. 3 – Smith Predictor principle: "from a real to a fictive structure"

The smith Predictor can be represented so that R(p) sees F(p):

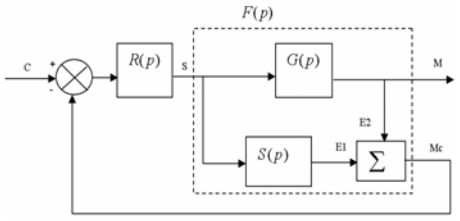

Fig. 4 – Second Smith Predictor functional view

3.1.1.1 Second order system application

The system G is represented by:

$$G(p) = \frac{G_s e^{-\tau p}}{(1+Tp)^2} \tag{1}$$

In this particular case R is a PI corrector and S is a dead-time compensator. The Smith Predictor applied for a second order is shown in the next figure (*Fig. 5*):

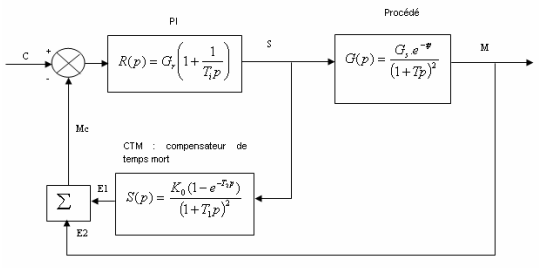

Fig. 5 –Smith Predictor applied for a second order system

The closed loop response is then:

$$H_{CloseLoop} = \frac{1}{1+\frac{T}{G_s G_{PI}}p+\frac{T^2}{G_s G_{PI}}p^2} = \frac{1}{1+\frac{2\xi}{w_n}p+\frac{1}{w_n^2}p^2}\Bigg|_{\substack{K_0=G_s \\ T_{PI}=T}} \tag{2}$$

3.1.2 Modified Smith Predictor for Integrator with long delay

The structure proposed by Matausek [6] is a simple and straightforward modification of the Smith Predictor for integrator systems with long dead-time *(Fig. 6)*. It allows a fast setpoint response and a satisfactory load disturbance rejection.

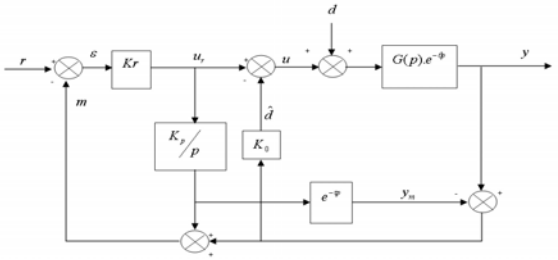

Fig. 6 – Matausek Modified Smith Predictor for integrator system

The process is considered so that:

$$G(p) = \frac{K_p}{p} \quad ; \quad \tau = \theta \tag{3}$$

Moreover the load $H_d(p)$ and setpoint $H_r(p)$ contributions present the good disturbance rejection $(0,\ t \to +\infty)$ and no steady state error:

$$H_r(p) = \frac{K_p.K_r.e^{-\tau p}}{1+K_p.K_r}$$
$$H_d(p) = \frac{K_p.\left[p+K_pK_r\left(1-e^{-\tau p}\right)\right]e^{-\tau p}}{\left(p+K_p.K_r\right)\left(p+K_p.K_0.e^{-\tau p}\right)} \tag{4}$$

The tuning proposed by Matausek is:

$$K_0 = \frac{\pi}{2.K_p.\tau} \quad ; \quad K_r = \frac{1}{K_p.T_r} \tag{5}$$

3.2 Generalized Predictive Control

The Generalized Predictive Control (GPC) proposed by Clarke *et al.* [7] [8] is a Model Based Control (MBC) strategy. The idea of GPC is to calculate a future sequence of control signals in such a way that it minimizes a cost function over a prediction horizon.

3.2.1 The j-step ahead predictor y(t+j)

From a particular operating point, even a non-linear system locally-linearized model as a CARIMA form [11]:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\frac{\xi(t)}{\Delta(q^{-1})} \tag{6}$$

where ξ(t) is an uncorrelated random sequence and Δ the differential operator 1-q^{-1}. From (6) we derive the j-step ahead predictor *y(t+j)*:

$$y(t+j) = \underbrace{F_j(q^{-1})y(t) + H_j(q^{-1})\Delta u(t-1)}_{Past} + \underbrace{G_j(q^{-1})\Delta u(t+j-1) + J_j(q^{-1})\xi(t)}_{Future} \tag{7}$$

3.2.2 The Diophantine equations

From (6) and (7) we obtain

$$A(q^{-1})J_j(q^{-1})\Delta(q^{-1})y(t+j) = B(q^{-1})J_j(q^{-1})\Delta u(t+j-1) + \xi(t)J_j(q^{-1})$$

$$\left[1 - q^{-j}F_j(q^{-1})\right]y(t+j) = \left[G_j(q^{-1}) + q^{-j}H_j(q^{-1})\right]\Delta u(t+j-1) + J_j(q^{-1})\xi(t+j) \tag{8}$$

to have the following Diophantine equations to be solved:

$$\begin{aligned} A(q^{-1})J_j(q^{-1})\Delta(q^{-1}) + q^{-j}F_j(q^{-1}) &= 1 \\ B(q^{-1})J_j(q^{-1}) &= G_j(q^{-1}) + q^{-j}H_j(q^{-1}) \end{aligned} \tag{9}$$

3.2.3 The cost function

The cost function *J* is defined to set up the future control sequence:

$$J = \sum_{j=N_1}^{N_2}\left(y(t+j) - w(t+j)\right)^2 + \lambda\sum_{j=1}^{N_u}\Delta u(t+j-1)^2 \tag{10}$$

w(t+j) is the setpoint at (t+j), N_1 is the minimum costing horizon, N_2 is the maximum costing horizon, N_u is the prediction horizon and λ is the control-weighting coefficient.

3.2.4 The matrix representation of an optimum j-step ahead predictor

The optimum j-step-ahead prediction [9] is given by:

$$\hat{y} = if(q^{-1}).y(t) + G.\tilde{u} + ih(q^{-1}).\Delta u(t-1) \tag{11}$$

with

$$if(q^{-1}) = \left[F_{N_1}(q^{-1}) \quad \dots \quad F_{N2}(q^{-1})\right]^T$$

$$ih(q^{-1}) = \left[H_{N_1}(q^{-1}) \quad \dots \quad H_{N2}(q^{-1})\right]^T$$

$$\tilde{u} = \left[\Delta u(t) \quad \dots \quad \Delta u(t+N_u-1)\right]^T$$

$$G = \begin{bmatrix} g_{N_1}^{N_1} & g_{N_1-1}^{N_1} & \dots & \dots \\ g_{N_1+1}^{N_1+1} & g_{N_1}^{N_1+1} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ g_{N_2}^{N_2} & g_{N_2-1}^{N_2} & & g_{N_2-N_u+1}^{N_2} \end{bmatrix} \tag{12}$$

The expectation of the cost-function of (10) can be written as:

$$J = \left[if(q^{-1}).y(t) + G.\tilde{u} + ih(q^{-1}).\Delta u(t-1) - w\right]^T \left[if(q^{-1}).y(t) + G.\tilde{u} + ih(q^{-1}).\Delta u(t-1) - w\right] + \lambda.\tilde{u}^T\tilde{u} \tag{13}$$

3.2.5 Cost Function minimization – control increment signal

The objective of GPC is to compute the future control sequence u(t), u(t+1),…, in such a way that the optimal j-step-ahead predictor is driven close to *w(t+j)* [9] [10]. This is accomplished by minimizing the cost function and making the gradient of J:

$$\frac{\partial}{\partial u}\left[if(q^{-1}).y(t) + G.\tilde{u} + ih(q^{-1}).\Delta u(t-1) - w\right]^T = G^T$$

$$\Rightarrow \quad \tilde{u} = M.\left[w - if(q^{-1}).y(t) - ih(q^{-1}).\Delta u(t-1)\right] \tag{14}$$

with

$$M = Q.G^T = \left[G^T.G + \lambda.I_{N_u}\right]^{-1}.G^T \tag{15}$$

Only the first value of the sequence of $\tilde{u}$ will be used to be compliant with the GPC strategy which repeats the procedure at each sampling time. The optimal control increment signal is then:

$$\Delta u_{opt}(t) = m_1^T.\left[w - if(q^{-1}).y(t) - ih(q^{-1}).\Delta u(t-1)\right] \tag{16}$$

with m_1, the first line of M.

3.3 Predictive Function Control

The Predictive Function Control (PFC) principles have been introduced in the early 1980's [11] [12]. It applies the same predictive strategy developed for the General Predictive Control (GPC) but uses different concepts to achieve the control signal. Giving the setpoint on a receding horizon, the predicted process output will reach the future setpoint following a reference trajectory *(Fig. 7)*. Additionally the PFC uses a model to build the control signal [13] [14].

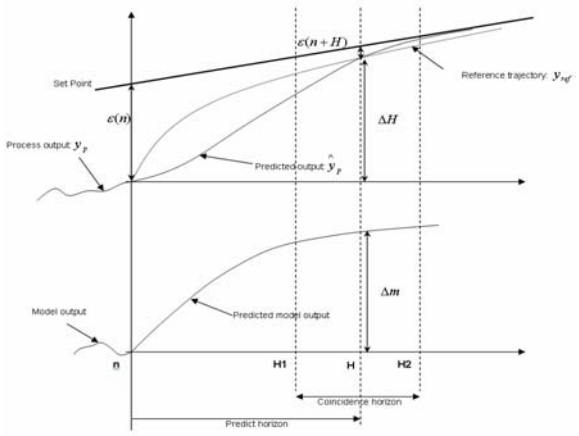

Fig. 7 – Predictive Function Control principles

The control signal *v* is then determinated using the base functions defined as follow:

$$v(n+i) = \sum_{k=0}^{k_M}\mu_k U_{Bk}(i) \tag{17}$$

3.3.1 First Order system application

Consider the process output y_p modeled by a first order system given by S(n) with a reference trajectory ε(n) and an exponential decrement λ:

$$S(n) = a_m S(n-1) + K(1-a_m)E(n-1)\Big|_{a_m = e^{-T_e/T}} = S(n)_L + S(n)_F$$
$$\varepsilon(n) = e^{-nT_e/T}$$
$$\varepsilon(n+H) = \varepsilon(n).\lambda^H \quad (18)$$

At a predict coincidence point H we have:

$$\varepsilon(n) = C(n) - s(n) = C(n) - y_p(n)$$
$$\varepsilon(n+H) = C(n) - y_{reference}(n+H)$$
$$\Delta H = y_{reference}(n+H) - y_p(n) \quad (19)$$

The command equation becomes:

$$\left(C(n) - y_p(n)\right)\left(1-\lambda^H\right) = S_M(n+H) - S_M(n) \quad (20)$$

Considering a step function for the base function, we build the control signal *v(n)*:

$$v(n) = \frac{\left(C(n) - y_p(n)\right)\left(1-\lambda^h\right) + S_M(n) - a_m^h S_M(n)}{K(1-a_m^h)} \quad (21)$$

3.3.2 Generalized PFC

The generalized PFC is applicable for asymptotic stable systems given by its convolute representation (so called MA systems):

$$y_P(n) = a_1^P.u(n-1) + \ldots + a_i^P.u(n-i) + \ldots + a_N^P.u(n-N) = \sum_{i=1}^{N} a_i^P.u(n-i) \quad (22)$$

The model is then:

$$y_M(n) = \sum_{i=1}^{N} a_i^M.u(n-i) \quad (23)$$

If the step function is the base function at a unique coincidence point H, the equation (23) can be split:

$$y_M(n+H) = \underbrace{u(n).\sum_{i=1}^{H} a_i^M}_{Future} + \underbrace{\sum_{i=H+1}^{N} a_i^M.u(n-i+H)}_{Past} \quad (24)$$

Using the reference trajectory defined in (18) we obtain the control signal *u(n)*:

$$u(n) = \frac{\left(C(n) - y_p(n)\right)\left(1-\lambda^H\right) - A_H^T.U(n) + A^T.U(n)}{\sum_{i=1}^{H} a_i^M} \quad (25)$$

with

$$A_H^T = \left\{a_{H+1}^M, \ldots, a_N^M\right\}$$
$$A^T = \left\{a_1^M, \ldots, a_N^M\right\}$$
$$U^T(n) = \left\{u(n-1), u(n-2), \ldots, u(n-N+H)\right\} \quad (26)$$

3.4 RST Controller

The RST controller representation is extremely useful for PLC implementation due to its simple structure [15]. The polynomial approach in q overcomes the usual inconvenience introduced by the sampling time *(Fig. 8)*.

The RST controller is driven by the following equation:

$$S(q^{-1}).u_k = T(q^{-1}).v_k - R(q^{-1}).y_k \quad (27)$$

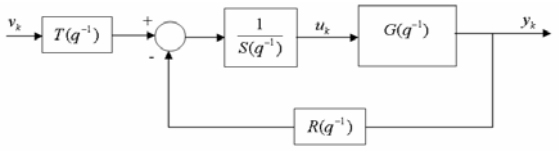

Fig. 8 – the RST controller

The RST controller is often used to calculate robust closed loop response by pole placement. The structured control signal introduced by the RST representation is done so that any controller can be represented through the RST formalized schema.

As an example, the RST representation of the GPC can be found; from equation (16) we can see that:

$$\Delta u_{opt}(t)\left[1 + m_1^T.ih(q^{-1}).q^{-1}\right] = m_1^T.\left[q^{N_1} \quad \ldots \quad q^{N_2}\right]^T w(t) - m_1^T.if(q^{-1}).y(t) \quad (28)$$

So by identification with (27) we obtain the RST form of the GPC controller:

$$\begin{cases} S(q^{-1}) = \Delta.\left[1 + m_1^T.ih(q^{-1}).q^{-1}\right] \\ R(q^{-1}) = m_1^T.if(q^{-1}) \\ T(q) = m_1^T.\left[q^{N_1} \quad \ldots \quad q^{N_2}\right]^T \end{cases} \quad (29)$$

4 The 'Advanced Automation Tool kit'

The model identification and the validation processes are usually offline workarounds [16] [17] [18]. Matlab/Simulink from Mathworks [19] company is one of the best simulation tools to accomplish theses tasks. However the online approach is not supported.

The principle of the 'Advanced Automation Tool Kit' consists on making the online identification and modeling processes trough a PLC environment. By an object oriented development the user is able to determinate the basics tasks needed for the model identification and its validation.

4.1 The model identification for Schneider PLC objects

4.1.1 The input sequence for the data acquisition

Clearly the Maximum Length Sequence is the ideal input signal for the process identification. However, the sequence of three steps is most of the time sufficient to identify the dynamic behavior of the system. Following this assumption the 'Advanced Automation Tool Kit' provides an object build in the

ST language for Schneider PLCs. It produces a three step sequence signal with determinate time scales (*Fig. 9*).

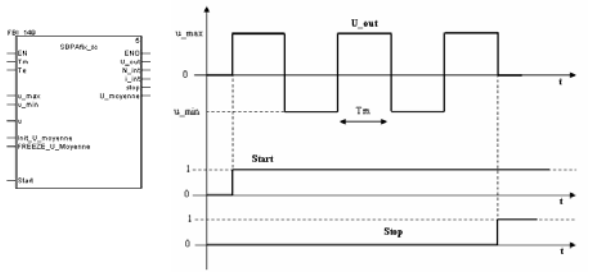

Fig. 9 – A three step sequence signal for Schneider PLC object

4.1.2 The Recursive Last Square (RLS) method

The RLS method is based on the linear regression [17]. This principle in automation is especially built for the estimation of ARMA system parameters plants:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) \tag{30}$$

The discrete model is defined by:

$$y(t+1) = \theta^T \phi(t) \tag{31}$$

with θ the parameter vector and $\phi(t)$ the measures vector.

The predictor (a priori) is build as follows:

$$\hat{y}^0(t+1) = \hat{y}(t+1|\hat{\theta}(t)) = \hat{\theta}(t)^T \phi(t) \tag{32}$$

The predictive error (a priori) is written with the relation:

$$\varepsilon^0(t+1) = y(t+1) - \hat{y}^0(t+1) = \varepsilon^0(t+1, \hat{\theta}(t)) \tag{33}$$

The last square criteria minimization J is then:

$$\min_{\theta(t)} J(t) = \frac{1}{t}\sum_{i=1}^{t}[y(i) - \hat{\theta}(t)^T \phi(i-1)]^2 = \frac{1}{t}\sum_{i=1}^{t}\varepsilon^2(i, \hat{\theta}(t)) \tag{34}$$

So that $\hat{\theta}$ is:

$$\hat{\theta}(t) = \left[\sum_{i=1}^{t}\phi(i-1)\phi(i-1)^T\right]^{-1}.\sum_{i=1}^{t}y(i).\phi(i-1) = F(t)\sum_{i=1}^{t}y(i).\phi(i-1) \tag{35}$$

with

$$F(t)^{-1} = \sum_{i=1}^{t}\phi(i-1)\phi(i-1)^T \tag{36}$$

The recursive form is obtained by using $\hat{\theta}(t+1)$ and manipulate

$$\hat{\theta}(t) = F(t)\sum_{i=1}^{t}y(i).\phi(i-1)$$
$$\hat{\theta}(t+1) = F(t+1)\sum_{i=1}^{t+1}y(i).\phi(i-1) \tag{37}$$

And $F(t+1)^{-1} = F(t)^{-1} + \phi(t)\phi(t)^T$, so that:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + F(t+1).\phi(t).\varepsilon^0(t+1) \tag{38}$$

The Schneider 'MCR_sc' object implements the RLS identification method (*Fig. 10*).

4.1.3 The Recursive Extended Last Square (RELS) method

The RELS method uses the last square criteria from RLS method but adapted for ARMAX plants [17]:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + C(q^{-1})e(t) \tag{39}$$

The parameter vector and the measures vector are defined by:

$$\phi(t)^T = [-y(t) \quad \dots \quad -y(t-n_A+1) \quad u(t-d) \quad \dots \quad u(t-d+n_B+1) \quad \varepsilon(t) \quad \dots \quad \varepsilon(t-n_C+1)]$$
$$\hat{\theta}(t)^T = [\hat{a}_1(t) \quad \dots \quad \hat{a}_{n_A}(t) \quad \hat{b}_1(t) \quad \dots \quad \hat{b}_{n_B}(t) \quad \hat{c}_1(t) \quad \dots \quad \hat{c}_{n_C}(t)] \tag{40}$$

The Schneider 'MCE_sc' object implements the online identification for the RELS method (*Fig. 10*).

4.1.4 Recursive Maximum likelihood (RLM) method

The RLM is an upgrade method of the RELS principle for ARMAX plants [17]. The aim is to reduce the correlation between the predictive error and the observation vector by filtering the measured vector with $1/\hat{C}(t,q^{-1})$, with $\hat{C}(t,q^{-1})$ the estimation of $C(t)$.

The parameter vector and the measures vector are then defined by:

$$\phi(t)^T = \frac{1}{\hat{C}(t,q^{-1})} *$$
$$[-y(t) \quad \dots \quad -y(t-n_A+1) \quad u(t-d) \quad \dots \quad u(t-d+n_B+1) \quad \varepsilon(t) \quad \dots \quad \varepsilon(t-n_C+1)]$$
$$\hat{\theta}(t)^T = [\hat{a}_1(t) \quad \dots \quad \hat{a}_{n_A}(t) \quad \hat{b}_1(t) \quad \dots \quad \hat{b}_{n_B}(t) \quad \hat{c}_1(t) \quad \dots \quad \hat{c}_{n_C}(t)] \tag{41}$$

The Schneider 'MVR_sc' object implements the online identification for the RELS method (*Fig. 10*).

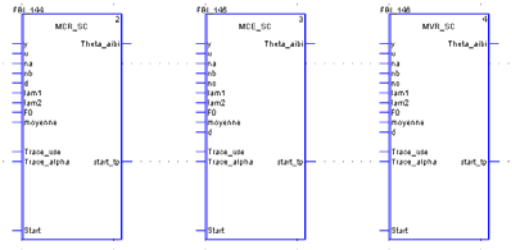

Fig. 10 – Online identification methods under Schneider PLC object

4.2 The model validation Schneider PLC objects

During the modeling phase the process is validated before any further corrections. In this objective, the 'Advanced Automation Tool Kit' provides three objects which helps user to choose the correct decision.

4.2.1 The model order validation

The model order validation test proposed by the 'Advanced Automation Tool Kit' uses the manipulation of the information matrix Q_m and $Q_m^{'}$ defined below [20]:

$$Q_m = \frac{1}{N}\sum_{k=1}^{N} \begin{bmatrix} u(k) \\ u(k+1) \\ u(k-1) \\ u(k+2) \\ .. \\ u(k-m+1) \\ u(k+m) \end{bmatrix} \begin{bmatrix} y(k+1) & u(k+1) & .. & y(k+m) & u(k+m) \end{bmatrix}$$

$$Q_m^{'} = \frac{1}{N}\sum_{k=1}^{N} \begin{bmatrix} u(k) \\ u(k+1) \\ u(k-1) \\ u(k+2) \\ .. \\ .. \\ u(k-m+1) \end{bmatrix} \begin{bmatrix} y(k+1) & u(k+1) & .. & .. & y(k+m) \end{bmatrix}$$

(42)

The method consists on building the determinant and doing the following calculus:

$$RDI(m) = \left| \frac{\det(Q_m)}{\det(Q_{m+1})} \right| \qquad RI(m) = \left| \frac{\det(Q_{m+1}^{'})}{\det(Q_m)} \right|$$

(43)

The system order *n* is found when RDI increases and when RI decreases (significantly for both).

The Schneider 'RDI_sc' performs this test. It is implemented up to the third order. This limitation is due to the PLC software which is sufficient for most industrial systems (*Fig. 11*).

4.2.2 The whiteness test

The whiteness test helps on the validation of the modeling [17]. It tries to estimate the predictive error whiteness. The method uses the following test:

$$RN(i) = \frac{R(i)}{R(0)} = \left. \frac{\frac{1}{N}\sum_{t=1}^{N} \varepsilon(t)\varepsilon(t-i)}{\frac{1}{N}\sum_{t=1}^{N} \varepsilon^2(t)} \right|_{i=1,2,...\max(n_a, n_b+d)}$$

(44)

with N the number of measurement.

A practical acceptable limit is usually $\left| RN(i) \le \frac{2.17}{\sqrt{N}} \right|$ [17].

The Schneider 'RNType1_sc' object build the whiteness test up to i=50. This whiteness test is dedicated to RLS, RELS and RLM identification methods (*Fig. 11*).

4.2.3 The simulation process

The 'Advanced Automation Tool Kit' provides the possibility to simulate a model by using an object named 'SimuSysteme_sc'. This PLC object is able to simulate ARMA and ARMAX systems (*Fig. 11*).

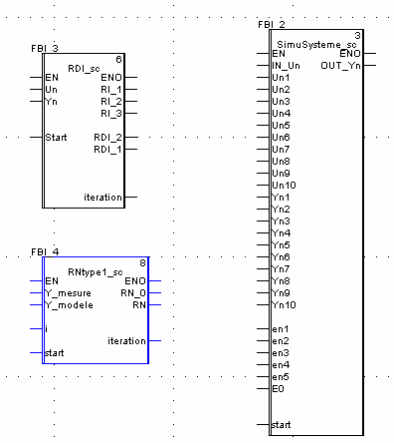

Fig. 11 – Model validation methods under Schneider PLC object

5 Advanced Control Implementation in PLC

The PLC programming concept is a cyclic execution process. The diversity of process control applications have also led to the introduction of the multi program cyclic principle for PLCs by means of four standard languages available through the IEC61131-3 norm [21]: the Instruction List (IL), the Structured Text (ST), the Ladder Diagram (LD) and the Functional Block Diagram (FBD). The IEC61131-3 norm proposes sharing the use of a program unit defined in one language by any of the others.

The advanced control algorithm implementation is not restricted by any of the four standard languages provided for PLCs. However the object programming development should take into account the cyclic nature of the PLC execution.

The advanced control algorithms are set up by using the cyclic execution as a sampling time reference. The

algorithms are then developed with emphasis on the sampling aspect commonly defined in the automation processes. The use of the ST language is a good compromise to deal with complex implementations such as loops and recursive mechanisms.

As a practical example the recursive first order system (without dead-time) implementation in ST language is written as follow:

```
Te_in_Real := TIME_TO_REAL(Te_system);
TETA_in_Real := TIME_TO_REAL(TETA_POSR);
Ap := EXP_REAL((-Te_in_real)/(TETA_in_real));
Bp := 1.0 - Ap;
Output_FOWDT := (Ap* Output_FOWDT) +
                    (Bp*GAIN_POSR*Input_FOWDT);
```

While using the ST language for implementation, the FBD language is optimal for testing. By synchronizing the object to a pre-determined fixed sampling time, the process is able to work with sampling behavior in such a way that it satisfies the discrete implementation of the internal advanced algorithm implementations. The FBD language is a powerful validation test environment for the object integration phase. It allows online visualization of the pertinent variables and parameters (see the example *Fig. 12*).

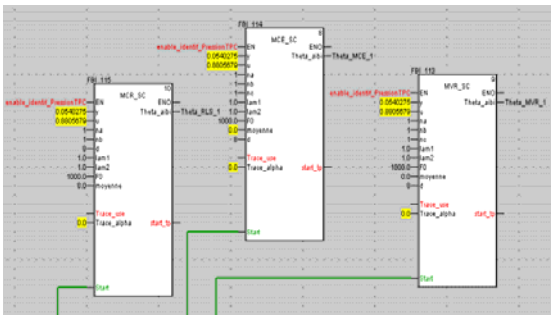

Fig. 12 – Online identification with Schneider DFB's of the 'Advanced Automation Tool Kit'

6 Experimental Result

6.1.1 Online identification with the 'Advanced Automation Tool Kit'

Tests have been done for a pressure regulation system so that it is an unstable system of order one with time delay of four seconds. The model to be found has the expression (ARMAX):

$$y(t+1)_{Model} = -a_1 y(t) + b_1 u(t-8) + e(t+1) + c_1 e(t)\big|_{T_e=0.5s} \tag{45}$$

The process identification is done trough two different experimental protocols:

- Under Matlab with the classical RELS and RLM algorithms (offline treatment)
- Under the Schneider PLC environment (online treatment).

We obtain the following models:

$$\begin{aligned} y(t+1)_{Matlab\_RELS} &= 0.9999\,y(t) + 0.0005476\,u(t-8) + e(t+1) - 0.2411e(t) \\ y(t+1)_{Matlab\_RLM} &= 0.9476y(t) + 0.0005676u(t-8) + e(t+1) - 0.0656e(t) \\ y(t+1)_{PLC\_RELS} &= 1.00031y(t) + 0.000567u(t-8) + e(t+1) - 0.256e(t) \\ y(t+1)_{PLC\_RLM} &= 0.95448y(t) + 0.00066u(t-8) + e(t+1) - 0.063e(t) \end{aligned} \tag{46}$$

The evolution of b_1 and c_1 are shown in *Fig. 13 to 16*:

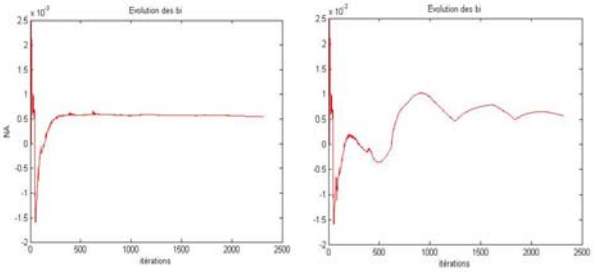

Fig. 13 – b_1 estimation along the Matlab identification process for RELS and RLM methods

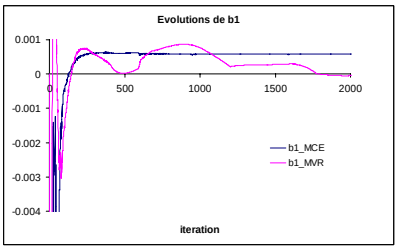

Fig. 14 – b_1 estimation along the PLC identification process with MCE_sc and MVR_sc objects

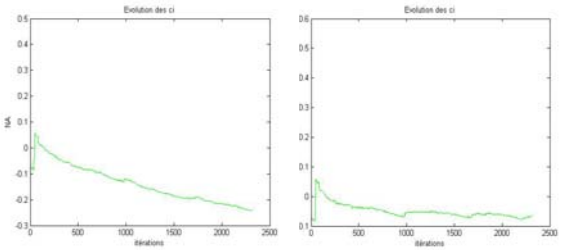

Fig. 15 – c_1 estimation along the Matlab identification process for RELS and RLM methods

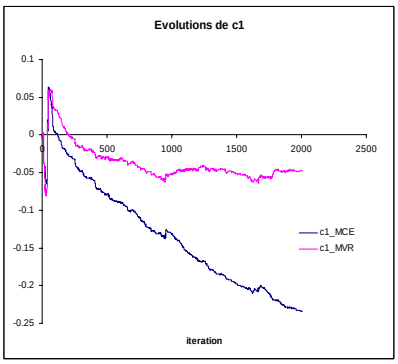

Fig. 16 – c_1 estimation along the PLC identification process with MCE_sc and MVR_sc objects

The PLC results highlight the power of the online identification introduced by the 'Advanced Automation Tool Kit'. The equations (46) show the similar results between Matlab algorithms and Schneider objects. In addition the estimated evolution of the model parameters (*Fig. 12 to 15*) follows the same behavior so that this experimental identification expresses clearly the validity of the PLC objects for the model identification.

6.1.2 MultiController application to a second order with dead time

The first object implementation of the MultiController into a Schneider PLC solution (Unity) has produced valuable results. The MultiController has introduced advanced control algorithms for the large scale UNICOS framework project. It offers to experimental plants a way to use new controllers.

The system is represented by this model:

$$G(z) = \frac{0.1269z^{-1} + 0.09614z^{-2}}{1 - 1.323z^{-1} + 0.4346z^{-2}} . z^{-3} \Big|_{T_e = 1s} \qquad (47)$$

Here are the settings of the MultiController algorithms:

PID	***Smith Predictor***	***PFC***	***GPC***
K=0.26 Ti=3.25s	T=4.25s G=2 Delay=3s	TS=1s, H=1 TRBF=5s, Km=2, Tm=4.25, Delay=3s	N1=4, N2=20 Nu=1, Lambda=150 Te=1s

Fig. 16 and 17 show the process output signal and the control signal in a system driven by (47) with several control algorithms given by the MultiController object.

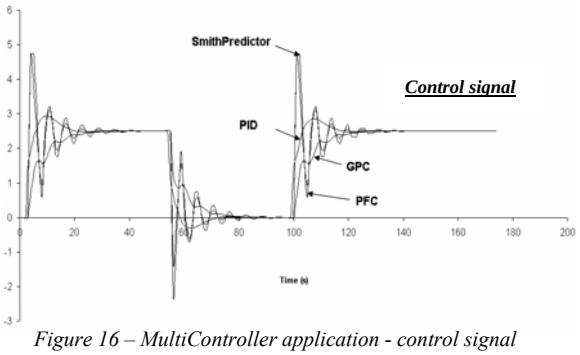

Figure 16 – MultiController application - control signal

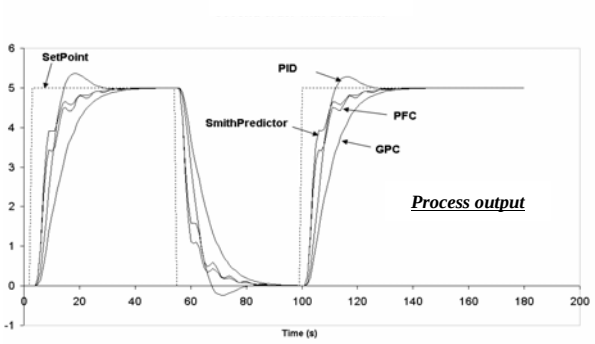

Figure 17 – MultiController application - process output

This application clearly shows the possibilities of the MultiController. The PID controller introduced an overshoot. On the contrary, the Smith Predictor and the PFC controller are both well adapted for speed process output response without the PID inconvenience. The control signals of those solutions do not converge gently but can be acceptable in some circumstances. Finally, the GPC controller produces a smooth process output response.

7 Conclusion

The MultiController object of the UNICOS framework is the combination of an efficient object programming process and advanced control features. By its robust design, this object is able to capture tuning parameters of all control algorithms through a single custom HMI.

The PLC object implementation follows the IEC61131-3 norm by means of coding the advanced algorithm in ST language. It also takes into account the cyclic nature of a PLC execution process through the program.

The 'Advanced Automation Tool Kit' is a set of advanced control features developed in addition to the MultiController object. It allows the use of a PLC environment to do online identification and online validation steps. This paper demonstrated the validity of the package proposed for Schneider PLC's under the Unity software.

The MultiController object implementation gives alternative solutions to standard PID controllers and increases the available control solutions to solve non-negligible complex problems. The Smith Predictor solution is able to solve dead time problems. The RST controller can be used to obtain robust closed loop responses. The MultiController also shows the powerful use of predictive algorithms for control loop system under PLC.

The advanced algorithms proposed by the MultiController object provide control loop solutions that enable the process control engineer to have access to more expert automation tools in a PLC-SCADA based environment.

ISBN 978-3-901608-32-2

REFERENCES

[1] P. Gayet et al., "UNICOS a framework to build industry-like control systems, Principles and Methodology", ICALEPCS'2005, Geneva, Switzerland, October 2005.

[2] P. Gayet, C.H Sicard, "Deployment and integration of industrial controls: the case of the LHC cryogenics controls", 2003.

[3] Michael Mattsson, "Evolution and Composition of Object-Oriented Frameworks", University of Karskrona/Ronneby, 2000.

[4] Ivar Jacobson, "Object-Oriented Software Engineering – A Use Case Driven Approach", 1992, ACM press, Addson-Wesley.

[5] O.J. Smith, "A controller to overcome dead band time", ISA J., vol. 6, no. 2 pp.28-33, February 1959.

[6] M.R. Matausek and A.D. Micic, "A Modified Smith Predictor for Controlling a Process with an Integrator and Long Dead-Time", IEEE Transactions on Automatic Control, Vol. 41, no. 8, August 1996.

[7] D.W. Clarke, C. Mohtadi, P.S. Tuffs, "Generalized Predictive Control – Part I. The Basic Algorithm", Automatica, Vol.23, No2, pp.137-148, 1987.

[8] D.W. Clarke, C. Mohtadi, P.S. Tuffs, "Generalized Predictive Control – Part II. Extensions and Interpretations", Automatica, Vol.23, No2, pp.149-160, 1987.

[9] G. Dreyfus, J. Richalet, G. Lavielle, J. Mallet, "La commande prédictive, Mise en œuvre et applications industrielles", Eyrolles, 2004.

[10] E.F. Camacho, "Constrained Generalized Predictive Control", IEEE Transactions on Automatic Control, Vol. 38, No.2, 1993

[11] C. Bordons, E.F. Camacho, "A Generalized Predictive Controller for a Wide Class of Industrial Processes".

[12] J. Richalet, "Pratique de la commande prédictive", ADERSA, Hermès, Paris, 1993.

[13] P. Boucher, D. Dumur, "La Commande Prédictive", Méthodes et Pratiques de l'Ingénieur, Editions Technip, 1996.

[14] A.W. Pike, M.J. Grimble, M.A. Johnson, A.W. Ordys, S. Shakoor, "Predictive Control ("The Control Handbook", Section 51)".

[15] P. Borne, G. Dauphin-Tanguy, J.-P. Richard, F. Rotella, I. Zambettakis, "Analyse et régulation des processus industriels (Tome 2 - Régulation Numérique)", Méthodes et Pratiques de l'Ingénieur, Editions Technip, 1993.

[16] L. Ljung, "System Identification - Theory for the user", Prentice Hall PTR, 1987.

[17] I.D. Landau, "Commande des systèmes – conception, identification et mise en oeuvre", Hermès Science Publication, Paris, 2002.

[18] D. Graupe, "Identification of Systems - 2nd edition", Robert E. Krieger Publishing Company, New York, 1976.

[19] The Mathworks company, http://www.mathworks.com/

[20] P. Borne, R. Ben Abdennour, M. Ksouri, F. M'sahli, "Identification et commande numérique des procédés industriels", Editions Technip, 2001.

[21] F.F. Jimenez, thesis "Conception sure des automatismes industriels : modélisation synchrone de langage d'automates programmables de la norme CEI-61131-3", Université Rennes 1, 2001.

