Auto-tuned Predictive Control Based on Minimal Plant Information

G. Valencia-Palomo * J.A. Rossiter *

* *Department of Automatic Control and Systems Engineering, University of Sheffield, South Yorkshire, U.K. S1 3JD. (e-mail: g.valencia-palomo@shef.ac.uk, j.a.rossiter@shef.ac.uk).*

Abstract: This paper makes two key contributions. First there is a definition and implementation of a novel auto-tuned predictive controller. The key novelty is that the modelling is based on relatively crude but pragmatic plant information. Secondly, the paper tackles the issue of availability of predictive control for low level control loops. Hence the paper describes how the controller is embedded in an industrial Programmable Logic Controller (PLC) using the IEC 1131.1 programming standard. Laboratory experiment tests were carried out in two bench-scale laboratory systems to prove the effectiveness of the combined algorithm and hardware solution. For completeness, the results are compared with a commercial PID controller (also embedded in the PLC) using the most up to date auto-tuning rules.

Keywords: Predictive control, auto-tuning, programmable logic controller, IEC-1131.1.

1. INTRODUCTION

Control design methods based on the predictive control concept have found wide acceptance in industry and in academia, mainly because of the open formulation that allows the incorporation of different types of models of prediction and the capability of constraint handling in the signals of the system.

Model predictive control (MPC) has had a peculiar evolution. It was initially developed in industry where the need to operate systems at the limit to improve production requires controllers with capabilities beyond PID. Early predictive controllers were based in heuristic algorithms using simple models. Small improvements in performance led to large gains in profit. The research community has striven to give a theoretical support to the practical results achieved and thus the economic argument, predictive control has merited large expenditure on complex algorithms and the associated architecture and set up times. However, with the perhaps notable exception of Predictive Functional Control (PFC) (Richalet, 1993), there has been relatively little penetration into markets where PID strategies dominate, and this despite the fact that predictive control still has a lot to offer in the SISO domain because of its enhanced constraint handling abilities and the controller format being more flexible than PID. The major obstacles cost, complexity and the algorithm not being available in the off the shelf hardware most likely used for local loop control.

Some authors have improved the user-friendliness (complexity) of MPC software packages available for high level control purposes (Froisy, 2006; Zhu *et al.*, 2008). Nevertheless, they have the same implementation drawback in that the development platform is a stand-alone computer running under Windows® OS. Furthermore, these packages involve complex identification procedures which thus requires the control commissioning to be in the hands of a few skilled control engineers; ownership by non control experts is an impediment for more widespread utilization.

Some early industrial work (Richalet, 2007) has demonstrated that with the right promotion and support, technical staff are confident users of PFC where these are an alternative to PID on a standard PLC unit. Technical staff relate easily to the tuning parameters which are primarily the desired time constant and secondly a coincidence point which can be selected by a simple global search over horizons choices. Because PFC is based on a model, the controller structure can take systematic account of dead-times and other characteristics, which are not so straightforward with PID. Also constraint handling can be included to some extent by using predicted violations to trigger a temporary switch to a less aggressive strategy.

The vendors conjecture is that PFC was successfully adopted because of two key factors: first there is effective support in technician training programmes (get it on the syllabus) and second the algorithm is embedded in standard PLC hardware they encounter on the job, thus making it easily accessible (and cheap). However, despite its obvious success academia has shied away from the PFC algorithm because its mathematical foundations are not as systematic or rigorous as other approaches; the performance/stability analysis is primarily an a *posteriori* approach as opposed to the a *priori* one more popular in modern literature. So there is a challenge for the academic community to propose more rigorous but nevertheless intuitive and simple algorithms which could equally be embedded in cheap control units.

On the other hand, in recent specialized conferences authors are often focussing on the level of rigor required in the modelling and tuning procedure for different cases (Morari *et al.*, 2008). However, accessibility and useability in such a mass market may require different assumptions from those typically adopted in the literature; specifically much less rigor and more automation in the modelling will be essential.

Hence, the first objective of this paper is to develop an auto-tuned MPC controller based on minimal plant information which would be available from staff at technician level only who may be responsible for maintaining and tuning local loops. Secondly, the paper aims to demonstrate how an MPC algorithm, using this model information, can be embedded in a commercial PLC (Valencia-Palomo and Rossiter, 2008); this paper gives some extensions to that developments in (Valencia-Palomo *et al.*, 2008) and of particular interest to readers will be the incorporation of systematic constraint handling within the PLC unit. A final objective is to contrast the auto-tuned MPC with a commercial PID controller in order to show that the MPC is a practical (available and same cost) alternative to PID for local loops.

The paper is organized as follows: Section 2 outlines the controllers and the auto-tuning rules, Section 3 describes the implementation of the controllers in the target hardware, Section 4 presents the simulation results on real hardware and finally in Section 5 are the conclusions and future work.

2. THE CONTROLLERS

This section outlines the auto-tuning rules and modelling assumptions for the MPC and PID strategies adopted. We note that the auto-tuning rules are only applicable to stable systems so discussion of unstable systems is deferred for future work.

2.1 Modelling assumptions

If anything, this paper is more generous with the auto-tuned PID than the MPC because it allows the PID algorithm a large quantity of measurement data and the ability to dither the input substantially during tuning to extract the required information. Moreover, the complexity of this algorithm means that the modelling is done offline. This decision was taken to give a stiff test for the auto-modelled/tuned MPC algorithms.

For MPC we provide crude modelling information only, for instance as could be provided by a technician or plant operator but specifically avoiding the use of a rigorous least squares model estimator which could be expensive if required for large numbers of loops and impractical to put on the PLC unit. The technician should provide estimates of behaviour as compared to standard second order characteristics: rise-time, settling time, overshoot, steady-state gain and dead-time. From this data an approximate second order model with dead-time is determined [1].

2.2 Design point, auto-tuning and constraint handling for PID

A novel auto-tuned PID controller as described in (Clarke, 2006; Gyöngy and Clarke, 2006) is used. A schematic

[1] We accept that for more complex dynamics a slightly more involved procedure may be required.

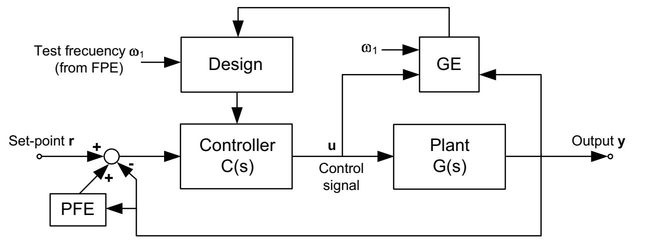

Fig. 1. Schematic diagram of the auto-tuning PID.

diagram of the system is shown in Fig. 1. The objective is to adapt the controller so as to achieve a carefully chosen design point on the Nyquist diagram.

The key components are phase/frequency and plant gain estimators (PFE, GE), described in detail in (Clarke, 2002). In essence a PFE injects a test sinewave into a system and continuously adapts its frequency ω_1 until its phase shift attains a desired value θ_d (in this case the design point). Also forming important part of the tuner, but not shown in Fig. 1, are variable band-pass filters (VBPF) at the inputs of the PFE and GE. These are second-order filters centered on the current value of the test frequency. They are used to isolate the probing signal from the other signals circulating on the loop (such as noise, set-point changes and load disturbances).

The algorithm is initialized using a first-order/dead-time (FODT) approximation $G_a(s)$ for the plant, obtained from a simple step test. The initialization involves the computation of suitable values for the parameters associated with the GE, PFE and the controller.

The controller is based on a design point in the Nyquist diagram. This design point is chosen to obtain the desired closed loop behavior, i.e. rise time, damping value, settling time. In this case, the desired damping value of 0.5 for all the systems is chosen. From this desired damping value, the variables for all the auto-tuning process are obtained as is shown in (Clarke, 2006; Gyöngy and Clarke, 2006).

The PID design does not take explicit account of constraints and thus ad hoc mechanisms are required. Typically input saturation with some form of anti-windup will be used but state constraints are not considered; this is a weakness.

2.3 Basic assumptions for MPC

For the purpose of this paper almost any conventional MPC algorithm can be deployed as the main distinguishing characteristic, **with sensible tuning**, is the model. Hence, assume that the MPC law can be reduced to minimising a GPC [2] cost function of the form:

$$J = \sum_{j=1}^{H_P} \left\| \hat{\mathbf{y}}\left(k+j|k\right) - \mathbf{w}\left(k+j|k\right) \right\|^2 + \sum_{j=1}^{H_C} \left\| \Delta\mathbf{u}\left(k+j|k\right) \right\|_{\lambda}^2 \tag{1}$$

where the second term in the eq. (1) is the control effort and λ is the weighting sequence factor. The reference trajectory $\mathbf{w}(k)$, is the desired output in closed loop of the system and is given by:

[2] To simplify some algebra compared to dual-mode approaches, e.g. (Rossiter *et al.*, 1998).