



Generalized predictive control with dual adaptation

Yong Kuen Ho^{a,*}, Farouq S. Mjalli^b, Hak Koon Yeoh^a

^a Chemical Engineering Department, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia

^b Petroleum & Chemical Engineering Department, Sultan Qaboos University, Muscat 123, Oman

HIGHLIGHTS

- RLS algorithm was used for both model adaptation and auto-tuning in the GPC.
- No *a priori* knowledge on move suppression weight is needed.
- Response to nonlinearity of the system was also enhanced by this new scheme.

ARTICLE INFO

Article history:

Received 18 May 2012

Received in revised form

10 August 2012

Accepted 23 August 2012

Available online 31 August 2012

Keywords:

Process control

Systems engineering

Parameter identification

Nonlinear dynamics

Recursive least squares

Generalized predictive control

ABSTRACT

In this work, the recursive least squares (RLS) algorithm, which traditionally was used in the generalized predictive controller (GPC) framework solely for model adaptation purposes, was extended to cater for auto-tuning of the controller. This new combination which eases the task of controller tuning, contains both model adaptation and auto-tuning capabilities within the same controller structure. Hereafter this scheme will be referred to as the adaptive-model based self-tuning generalized predictive control (AS-GPC). The variable forgetting factor recursive least squares (VFF-RLS) algorithm was selected to capture the dynamics of the process online for the purpose of model adaptation in the controller. Based on the evolution of the process dynamics given by the VFF-RLS algorithm in the form of first order plus dead time (FOPDT) model parameters, the move suppression weight for the AS-GPC was recalculated automatically at every time step based on existing single input single output (SISO) analytical tuning expressions originally used for offline tuning of constraint-free predictive controllers. Closed loop simulation on a validated transesterification reactor model, known for inherent nonlinearities, revealed the superiority of the proposed constrained control scheme in terms of servo and regulatory control as compared to the GPC with model adaptation only, the conventional GPC as well as the conventional PID controller. The tuning expressions used, although intended for constraint-free predictive controllers, yielded good results even in the constrained case.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Adaptive model predictive control (MPC) techniques are adept in handling dynamically challenging and nonlinear processes due to the ability to self-adjust model or tuning parameters in real time. In one implementation as the generalized predictive control (GPC) (Clarke et al., 1987a,b), traditionally it involves model adaptation alone (e.g., Corrêa et al., 2002; Diaz et al., 1995; Ho et al., 2010, 2012a, 2012b; Moon et al., 2005, 2006), in which the recursive least squares (RLS) algorithm (Seborg et al., 1986) is normally used to capture the dynamics of a slowly time-varying process in the form of linear time invariant (LTI) process model

parameters (Seborg et al., 2004) at every time step. In general, two key components within the structure of the controller can be made adaptive – the GPC internal model and the GPC tuning parameters. The need for both of these to be made concurrently adaptive is obvious: for a nonlinear and time-varying process, it is impossible to adopt a single LTI model to represent the dynamics of the process across all operational regions and at all times; with the change in the model a new set of tuning parameters should be adopted. This idea of updating the tuning parameters as the model parameters change was demonstrated by Dougherty and Cooper (2003a,b) in the multiple model adaptive control (MMAC) strategy for the dynamic matrix control (DMC), where a pre-calculated distinct set of tuning parameters was used for every single model in the model bank. Although adaptation of the controller was not automated in their proposed strategy, the work showed that the controller should be retuned according

* Corresponding author. Tel.: +60 12 6313536.

E-mail address: yongkuen.ho@gmail.com (Y.K. Ho).

to the process model employed. However, for the GPC, usually one of two such possibilities to make the controller adaptive is used.

Several authors have proposed different methods to auto-tune (or self-tune) the MPC controller online, while keeping the model parameters fixed. In the work of Al-Ghazzawi et al. (2001), the authors developed analytical sensitivity expressions relating the process outputs and the DMC tuning parameters (i.e., the weights on the output residuals and the move suppression coefficients) and utilized it for automatic online adjustments of the tuning parameters. The prediction and control horizons were set at predetermined values and only the weights on the output residuals and the move suppression weights were retuned in real time. Han et al. (2006) proposed a self-tuning strategy based on particle swarm optimization (PSO). A novel performance index was developed and PSO was used to search for the optimal tuning parameters while minimizing the performance index to cater to the worst operating conditions. This method was used to tune the DMC by Kawai et al. (2007). However, instead of computing the entire set of tuning parameters online, the optimization problem was restricted to computing the optimal values of the inputs and outputs weights only, hence reducing the computation load required. Ali and Al-Ghazzawi (2003) proposed a self-tuning scheme based on fuzzy logic. In this scheme, the prediction horizon, the move suppression weights, and the weights of the output residuals were determined from predefined fuzzy rules which formulate the general tuning guides reported in the literature, while the control horizon was fixed at a constant value. Similar fuzzy-based self-tuning schemes were also reported for the nonlinear MPC (Ali, 2003; Zarkogianni et al., 2011). In an attempt to auto-tune the constraint-free DMC, van der Lee et al. (2008) used a combination of the genetic algorithm (GA) and the multi-objective fuzzy decision making (MOFDM) algorithm to determine the optimal tuning parameters to be used in the controller. In this scheme, a few sets of initial tuning parameters were simulated and the objective function for each tuning parameter set was calculated. Following this, the worst set of tuning parameters was discarded based on MOFDM while the best two sets entered the next GA iteration. The optimal tuning parameters at the end of the GA cycle were then passed to the controller for implementation.

Although the studies mentioned were mostly implemented in the framework of DMC, these methods were in principle also applicable for the GPC. For auto-tuning specific to the GPC controller, two algorithms were proposed in the work of Liu and Wang (2000) to carry out a multi-objective online optimization to obtain the tuning parameters and the control moves. In another work, Li and Du (2002) adopted a simulation-optimization approach to tune the move suppression weight for the GPC according to some fuzzy criteria. Given *a priori* knowledge on the requirements of controller smoothness for their bioreactors, Diaz et al. (1995) proposed a decreasing law for the move suppression weight. In a recent work for the GPC implemented through the programmable logic controller (PLC), Valencia-Palomoa and Rossiter (2010) proposed that the GPC be auto-tuned based on standard second order characteristics (i.e., rise time, settling time, overshoot, process gain, dead time and sampling time of the process) of the process.

In the above methods, with few exceptions (e.g., Diaz et al., 1995; Valencia-Palomoa and Rossiter, 2010), optimization routines were used not only for the purpose of producing optimized control moves, but also they were used directly/indirectly to compute the optimal set of controller tuning parameters online. Although these methods do not require much knowledge about the process from the control engineer to initiate the tuning procedure – thus alleviating the difficulties that control engineers

faced in tuning the predictive controller – these optimization approaches were computationally demanding and mathematically involved (Garriga and Soroush, 2010). Moreover, these studies utilized a static internal model, and no attempts were reported to account for the nonlinearities and time-varying dynamics of the process in the model itself. While the procedures involved in the studies done by Diaz et al. (1995) as well as Valencia-Palomoa and Rossiter (2010) are computationally less demanding and relatively simple to implement, the former is process specific while the latter requires user input in the internal model of the controller. This raises the question whether a simpler but more general tuning algorithm could be developed. Going further, instead of having either adaptive tuning parameters or model parameters, will making both adaptive give a better controller? We are not aware of studies that explored this possibility.

This study addresses the gap in the feasibility of having a GPC controller where *both* the model parameters and the tuning parameters are adapted in real-time, and with simplicity. For ease of reference, the proposed control algorithm, which incorporates both the model adaptation and self-tuning strategies in a single controller, is referred to as the adaptive-model based self-tuning generalized predictive control (AS-GPC) while the GPC with model adaptation only (which was included for comparison purposes) is termed adaptive-model based generalized predictive control (A-GPC). The brief structure of this paper is as follows. Section 2 addresses the basic structure of the AS-GPC scheme. In Section 3, the implementation and analysis of the constrained AS-GPC scheme on a selected nonlinear transesterification reactor model will be discussed. In the same section also, the performance of the AS-GPC scheme will be benchmarked against that of the A-GPC, GPC and conventional PID schemes. Section 4 concludes this study.

2. Basic structure of the AS-GPC scheme

The general idea of the proposed strategy is illustrated in Fig. 1. As opposed to the conventional GPC strategy, where users are required to re-determine the model parameters and retune the controller manually when unsatisfactory controller performance arises, the newly proposed scheme requires the users to only input a few components, viz. the RLS design parameters, a reduced amount of GPC tuning parameters (determined offline and held unchanged throughout the entire course of implementation), and setpoints, while the re-modelling of the process and the retuning of the move suppression weight are taken care of by the controller itself. In practice, a normal user (e.g., operators and technicians) needs only be concerned about the values of the setpoints and the remaining tuning parameters, while the manipulation of the RLS design parameters (which normally are determined one-off) shall be reserved for expert users (e.g., engineers) only. The move suppression weight was chosen in this work as the single parameter for auto-tuning implementation due to its efficacy in affecting the closed loop performance of the GPC (Diaz et al., 1995; Li and Du, 2002; McIntosh et al., 1991; Shridhar and Cooper, 1997a,b). Instead of performing an optimization layer, auto-tuning of the move suppression weight was achieved by formulating the parameter estimation problem such that the output of the RLS estimation concurs with the easy-to-use FOPDT (Seborg et al., 2004) tuning correlations as proposed by Shridhar and Cooper (1997b). In doing so, our primary concern is “simplicity” and not “optimality”. More importantly, with little additional computational cost, the proposed control algorithm relieves a normal user from the challenging efforts involved in retuning the GPC as well as re-modeling the process offline to

Download English Version:

<https://daneshyari.com/en/article/6592335>

Download Persian Version:

<https://daneshyari.com/article/6592335>

[Daneshyari.com](https://daneshyari.com)