Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-8 (2015) 531–538

Model Predictive Control in Industry: Challenges and Opportunities

Michael G. Forbes* Rohit S. Patwardhan Hamza Hamadah** R. Bhushan Gopaluni*****

**Honeywell Process Solutions, North Vancouver, BC V7J 3S4, Canada*
***Process & Control Systems Department, Saudi Aramco, Dhahran, 31311, Saudi Arabia*
**** Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC V6T 1Z3, Canada*

Abstract: With decades of successful application of model predictive control (MPC) to industrial processes, practitioners are now focused on ease of commissioning, monitoring, and automation of maintenance. Many industries do not necessarily need better algorithms, but rather improved usability of existing technologies to allow a limited workforce of varying expertise to easily commission, use, and maintain these valued applications. Continuous performance monitoring, and automated model re-identification are being used as vendors work to deliver automated adaptive MPC. This paper examines industrial practice and emerging research trends towards providing sustained MPC performance.

Keywords: industrial control, process control, model-based control, predictive control, adaptive control, performance monitoring, control applications, human factors.

1. INTRODUCTION

Model predictive control (MPC) is an industry accepted technology for advanced control of many processes. Recall that DMC (dynamic matrix control) was introduced around 1980 (Cutler and Ramaker, 1980); by 1997 a number of commercial MPC software packages were available (see, for example, Qin and Badgwell (1997)). Industrial expectations for MPC have increased from providing superior control for multivariable systems to doing so with minimum set-up effort and ease of maintenance. In today's process industries, MPC is often considered a required solution for many applications. At the same time, resources of expert practitioners to commission, monitor, and maintain MPC are increasingly limited. For this reason, both vendors and customers are looking for ways to sustain MPC performance with minimum manual intervention. In this paper, some established and emerging trends in the industrial application of MPC for sustained performance are discussed. Section 2 gives considerations in commissioning MPC for long term success. Section 3 examines ease of operation as a contributing factor to successful applications. Section 4 discusses industrial MPC maintenance practice, with a focus on performance monitoring and adaptive control. Section 5 reviews emerging research trends for industrial MPC.

2. COMMISSIONING AN MPC APPLICATION

Sustained performance of an MPC depends on many important decisions made during commissioning. An MPC that is easy to configure, operate, and maintain has a good chance of long term success. While examples are given in the following subsections, the main themes of this section are:

- MPC structure and use of features affect maintainability,
- Difficult MPC set-up may cause less robust tuning and model mismatch due to software use errors. Chances of sustainable performance are immediately reduced.

2.1 MPC Structure

One benefit of MPC is that it determines the optimal actions to take for large multi-input, multi-output (MIMO) systems. MPC simultaneously adjusts all inputs to control all outputs while accounting for all process interactions. As a result, MPC often takes actions that improve plant performance beyond what a skilled and experienced operator can achieve.

However, there are also drawbacks to the use of a single MPC to control an entire MIMO system, which may inhibit the success of an application. One obvious alternative to putting all variables into a single MPC is to break up the problem into a number of smaller systems which have limited interactions. In the following points, some of the potential drawbacks to including all process variables in a single MPC are listed, along with the comparative advantage possible if the MPC is broken into several smaller systems:

- A single MPC can optimize an entire process, but it may also be difficult to understand and monitor performance of a large application due to the large number of interactions between variables. Splitting the MPC into smaller systems, may make it easier to judge the behaviour of each MPC.
- When a large application performs poorly due to model mismatch, it may be difficult to determine which submodel(s) need updating. Often, plant experimentation and identification for all models is time consuming, or introduces unnecessary variability to the process. With multiple small MPCs, when one of the small MPCs is not performing, there are fewer models to evaluate.
- If the controller cannot be used for some reason (a set-up error for example), then no controlled variables (CVs) are controlled. With multiple small systems, one MPC can be turned off independently of the other MPCs, leaving most CVs under control. Some commercial packages have

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.09.022

features to handle this issue; practitioners should determine how to best utilize such functionality.

- If some CVs and/or manipulated variables (MVs) are dropped from the controller, the likelihood of unexpected MV movements, as the MPC re-optimizes using the remaining process variables, is relatively high for a large system. (These different movements may be correct, but they also may be undesired consequences of an unusual operating mode. Even if correct, different MV movements may be questioned by an operator.) If CVs and/or MVs are dropped from a smaller MPC, it may be easier to anticipate how the MPC will re-optimize, and there will be less MVs to monitor for unanticipated movements.

2.2 Process Modeling

Success of model-based controllers, such as MPC, depends on having reasonably accurate process models. Often a designed experiment is run to generate the data containing sufficient process excitation needed to accurately identify models. A common problem with this approach is that the type of plant experiments that yield the best data are also likely to perturb the process beyond current operating limits. This issue is well known, and sometimes is mitigated by clever experimental designs. In other cases plants accept some small short term deviations in production or quality in exchange for the long term benefits of a successful MPC application. However, there is another aspect of process modeling that can impact the long term sustainability of MPC performance, which may not be as widely considered: ease of identifying the model.

Most industrial MPC packages include model identification software. This software helps the user to take plant data and develop the models needed for MPC. The ease with which the software can be used can have a big effect on how well MPC is maintained. Identification software can suffer from:

- Poor workflow, requiring many steps, menu selections, button clicks, and so forth, to go from raw data to a final model. Each step is the opportunity to make an error.
- High complexity, which allows for a great deal of flexibility in the model building process, but which may overwhelm the occasional, inexpert user, again offering opportunities for mistakes to be made.

These challenges may not be a problem during the commissioning process where often an expert user performs the model identification. However, maintenance of the MPC, including re-identifying models, often falls to a non-expert. The difficulty of the identification task may then prevent MPC performance from being sustained because:

- Non-intuitive identification software hinders user confidence and willingness to update process models as often as needed,
- Incorrect use of complex identification software leads to poor model selection,
- Most users will not be able to judge, by inspection, if higher-order model parameters are valid.

To help overcome some of these issues, it is common (but not universal) for MPC practitioners to use first-order plus deadtime models unless there is strong evidence that a higher order model is required. The advantage of using these simple models, even at the expense of some model-plant mismatch, is that someone who is not a controls expert can 'sanity check' these models, and judge if the gain, time delay, and time constant are plausible. Additionally, if a review of the model predictions versus data reveals poor identification of model parameters, most users can manually adjust gain, time delay, and time constant. The use of this simple model form comes with the expense of providing a very good but not optimal process model, but reduces the risk of large model errors due to user unfamiliarity with higher-order models.

Another approach to process model maintenance is to use an adaptive algorithm to automatically detect controller performance degradation due to model mismatch, generate data with a new plant experiment, and identify and deploy a new model. Adaptive control is discussed later in this paper.

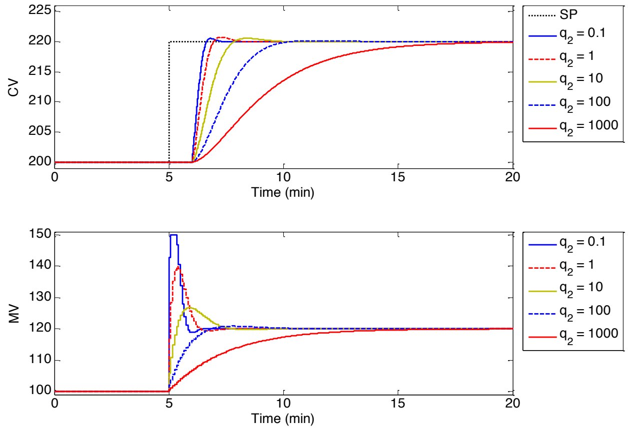

Fig. 1. Setpoint change for SISO MPC for different MV movement cost weight (q2)

2.3 Controller Tuning

For basic MPC there are many tuning parameters: prediction horizon, control horizon, setpoint tracking cost weights, and input movement cost weights. More advanced MPC may have additional tuning parameters relating to reference trajectories, output funnelling, blocking, etc. While there is often a clear explanation of what these tuning parameters are meant to influence in the MPC formulation, it is not always easy to find the parameter values that achieve a desired controller performance. Fig. 1 gives an example of a basic single-input single-output (SISO) MPC executing a step setpoint change for different choices of the input movement cost weight, q_2. The five example responses each have a different character, but to achieve these different behaviours it is necessary to change the input movement cost weight by an order of magnitude. Even if the two most aggressive tunings are rejected as extreme, it is still necessary to pick a value for this tuning parameter from the range of 10 to 1000. This is a big challenge, resulting in much trial and error, for tuning. In the more general case where there are multiple input movement cost weights, plus other parameters to select,