Journal of Process Control 24 (2014) 239–249

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Model predictive control for batch processes: Ensuring validity of predictions

D. Laurí[a,*], B. Lennox[a], J. Camacho[b]

[a] Control Systems Group, School of Electrical and Electronic Engineering, The University of Manchester, Manchester, UK
[b] Signal Theory, Telematics and Communications Department CITIC, University of Granada, Granada, Spain

ARTICLE INFO

Article history:
Received 16 January 2013
Received in revised form 26 July 2013
Accepted 5 November 2013
Available online 25 December 2013

Keywords:
Batch optimization
Model predictive control
Latent variable
Validity of predictions

ABSTRACT

The intuitive and simple ideas that support model predictive control (MPC) along with its capabilities have been the key to its success both in industry and academia. The contribution this paper makes is to further enhance the capabilities of MPC by easing its application to industrial batch processes. Specifically, this paper addresses the problem of ensuring the validity of predictions when applying MPC to such processes. Validity of predictions can be ensured by constraining the decision space of the MPC problem. The performance of the MPC control strategy relies on the ability of the model to predict the behaviour of the process. Using the model in the region in which it is valid improves the resulting performance. In the proposed approach four validity indicators on predictions are defined: two of them consider all the variables in the model, and the other two consider the degrees of freedom of the controller. The validity indicators are defined from the latent variable model of the process. Further to this, these are incorporated as constraints in the MPC optimization problem to bound the decision space and ensure the proper use of the model. Finally, the MPC cost function is modified to enable fine case-specific tuning if desired. Provided the indicators are quadratic, the controller yields a quadratic constrained quadratic programming problem for which efficient solvers are commercially available. A fed-batch fermentation example shows how MPC ensuring validity of predictions improves performance and eases tuning of the controller. The target in the example provided is end-point control accounting for variations in the initial measurable conditions of the batch.

1. Introduction

Batch and fed-batch[1] processes have widespread applications in chemical and life science industries for the production of products with high added value (e.g. medicines, enzymes, high-performance polymers). The primary control objective in batch processes is to reach a specified product quality at the point of batch termination. Accurately controlling the final quality of a batch process is challenging in that physical measurements of the quality parameters that can be used to predict the final quality are often not available on-line [1,2]. Data-driven models provide an answer to this problem and are widely used for process monitoring [3–5]. The common approach to control the quality of a batch process is end-point based MPC [6]. In end-point based MPC the control sequence from the current point until the end of the batch is determined such that product quality at the end of the batch reaches a desired value. Depending on the strategy used, the trajectory is either determined at the start of the batch and on-line local controllers are used to track the trajectory (batch-to-batch) [7–10]; or the MPC problem is solved at each decision point within the batch and applied in a receding horizon policy (within-batch) [11,12]. The former is easier to deploy, whereas the later may be less affected by process noise and disturbances. Intermediate solutions can also be defined in which the period to recalculate the MPC trajectory is larger than that used in local controllers that track the calculated trajectory [13]. All these strategies have the commonality of solving an MPC problem, either off-line or on-line. The MPC problem calculates the control trajectory from the calculation instant to the end of the batch, using a model of the process. For the control results to be successful, the model should approximate the behavior of the process to an acceptable level, which for any non-linear process requires it to be used in the region in which it is valid. An important source of variability in predictions is the initial state, e.g. variability in the raw material. Consequently all measurable variables available at the decision points should be considered for inclusion in the model so that the accuracy of the predictions can be improved. All end-point MPC optimization techniques need to overcome the following **challenges** to ensure high quality control is achieved:

* Corresponding author. Tel.: +34 645834681.
E-mail addresses: david.lauripla@manchester.ac.uk, dalaupl@gmail.com, David.LauriPla@pfizer.com (D. Laurí).

[1] In the sequel batch refers to both batch and fed-batch processes.

0959-1524/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jprocont.2013.11.005

(I) Cope with disturbances: unmeasured variables that affect quality, or changes in measurable variables after the decision point, e.g. feed property change.
(II) Cope with faults: Measurement errors or faults in actuators or sensors.
(III) Acceptable computational complexity: Make sure the time needed to solve the MPC problem is smaller than the available time.
(IV) Account for process-model mismatch: The model used in MPC is an approximation of the real process, hence the model should be used in a region in which it is known to be valid.

There are two approaches to cope with disturbances, **challenge (I)**. One approach is to recompute the trajectory of the manipulated variables (MVs) to account for the disturbance using a moving horizon MPC strategy [12]. Another approach is to improve the predictor to actually account for the disturbance, e.g. by using batch-to-batch iterative learning control (ILC). In [10], an ILC approach based on moving window re-identification is proposed. There are also approaches that combine both, ILC and moving horizon MPC [14,15].

There are two approaches to cope with faults, **challenge (II)**. In the active approach the fault is detected and actions are taken mainly by controller reconfiguration [16,17]. In the passive approach the end-point-based MPC formulation explicitly accounts for model uncertainty. One solution is to use min-max based robust MPC in which the goal is to maximize the performance of the MPC by minimizing the worst-case tracking error (the largest difference between the prediction and the actual measurement). However, according to [18], these approaches are often computationally prohibitive. Alternatively, [18] proposes a robust reverse-time reachability region based MPC approach to ensure states can be driven inside a desired end-point neighborhood if the fault is repaired sufficiently fast.

There are two branches of solutions to attain an acceptable computational complexity, **challenge (III)**. One branch of solutions increases the available time for computation. This can be achieved for batch processes by either solving the MPC at the beginning of the batch in a batch-to-batch optimization policy, or by reducing the number of decision points during the batch to recompute the trajectory [19]. The other branch of solutions is trying to solve the problem faster. The common approach to solving the cost function faster is to use a linear or set of linear models instead of a non-linear model [20,21], or use a linear MPC controller over an input-output feedback linearized process [22]. Additionally, it is common practice to reduce the d.o.f. (degrees of freedom) to reduce computation time. Move blocking strategies reduce the d.o.f. by fixing the input or its derivatives to be constant over several timesteps. A survey of various move blocking strategies is presented in [23]. An alternative approach is to use Laguerre functions to approximate the control sequence in a large control window, but using a reduced number of d.o.f. [24]. Another solution is to reduce complexity by using latent variable methods in the identification stage and perform the minimization in the space of the latent variables [11,25].

This paper focuses on **challenge (IV)**, accounting for process-model mismatch. The different approaches to account for process-model mismatch include:

- Minimize process-model mismatch. A more complex model or an adaptive approximation can be used to minimize process-model mismatch.
- Weight changes on the control sequence from the nominal trajectory in the MPC cost function by means of λ_u. This is the simplest and most common approach in which λ_u is a design parameter that weights the control effort. Small values of λ_u give freedom to the controller to propose control trajectories far from the nominal conditions, which assuming the model is a linear approximation of the process around a fixed trajectory, can lead to poor predictions and reduced control performance. Large values of λ_u provide biased control as the deviation from the target term in the MPC cost function loses importance versus the movement suppression term. The most common solution is to tune λ_u by trial and error until the desired performance is obtained.
- Robust MPC: model uncertainty is considered in the MPC cost function and solved using a min–max optimization problem. Robust MPC is case dependent and can be challenging to apply it successfully [26]. In [18] a reverse-time reachability region based non-linear MPC for batch processes is presented to cope with model uncertainty. The disadvantage in robust approaches is computational complexity.
- Constrain the MPC problem to ensure the model is used in the region in which it is valid. Receding horizon MPC with hard validity constraints has been defined for continuous processes in [27], and for quality by design in [28–30]. In [31], validity indicators are weighted in the MPC cost function for batch processes, but they are not set as hard constraints. This paper is an extension of previous work of the authors [27] that implements hard validity constraints for batch processes. The novelty in this paper is it defines a systematic approach for inclusion of validity constraints that ensures feasibility. The validity indicators are included as two hard constraints and two hard constraints relaxed with slack variables to ensure feasibility. Additionally, the validity indicators are weighted in the cost function to enable further optional fine tuning of the controller.

Summing up, this paper formulates an MPC framework for batch processes that ensures validity of predictions by constraining the decision space. The performance of the MPC control strategy relies on the ability of the model to predict the behavior of the process. Hence, using the model only in the region in which it is valid, will improve the resulting control performance. The novelty proposed in this paper is the methodology that is used to implement hard and softened constraints to ensure the validity of predictions. The controller proposed in this paper could be formulated in the latent variable space as in [27], where the same control results would be obtained, but with reduced computational complexity. However, in this paper it is assumed that the computational complexity of the optimization problem is acceptable and the controller is formulated in the original space of the MVs for the sake of readability. Both the model and the validity indicators are obtained in the latent variable space and then formulated in the original space of the MVs to be included in the controller. Although the solution proposed in this paper focuses on challenge (IV), current solutions for the other three challenges presented could be combined with the methodology proposed in this paper to cope with challenges (I)–(III).

The structure of this paper is as follows: The traditional MPC methodology applied for end-point control of batch processes is briefly summarized in Section 2. The indices on validity of predictions considered in this paper are introduced in Section 3. A solution for further improvement of end-point control results is provided in Section 4. In Section 5, a fed-batch fermentation example shows how ensuring validity of predictions can improve end-product quality while simplifying deployment. The paper ends with concluding remarks in Section 6.

2. MPC for end-point control in batch processes

This section briefly describes the MPC methodology with one decision point for the batch, also known as the mid-course correction methodology. The decision point can be set towards the