



# Dynamic self-optimizing control for unconstrained batch processes

Lingjian Ye<sup>a,b</sup>, Sigurd Skogestad<sup>b,\*</sup>

<sup>a</sup> Ningbo Institute of Technology, Zhejiang University, Ningbo, Zhejiang, 315100, China

<sup>b</sup> Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, 7491, Norway



## ARTICLE INFO

### Article history:

Received 17 January 2018

Revised 25 June 2018

Accepted 29 June 2018

Available online 6 July 2018

### Keywords:

Self-optimizing control

Controlled variable

Dynamic optimization

Batch process

## ABSTRACT

In this paper, we consider near-optimal operation for a class of unconstrained batch processes using the self-optimizing control (SOC) methodology. The existing static SOC approach is extended to the dynamic case by means of a static reformulation of the dynamic optimization problem. However, the dynamic SOC problem is posed as a structure-constrained controlled variable (CV) selection problem, which is different from the static cases. A lower-block triangular structure is specified for the combination matrix,  $\mathbf{H}$ , to allow for optimal operation whilst respecting causality. A new result is that the structure-constrained SOC problem still results in a convex formulation, which has an analytic solution where the optimal CVs associated with discrete time instants are solved separately. In addition, the inputs are directly determined based on current CV functions for on-line utilization. A fed-batch reactor and a batch distillation column are used to demonstrate the usefulness of the proposed approach.

© 2018 Elsevier Ltd. All rights reserved.

## 1. Introduction

Self-optimizing control (SOC) (Jäschke et al., 2017; Skogestad, 2000) is a relatively new control methodology for real-time optimization (RTO) of industrial processes. A key feature is that near-optimal performance (with acceptable economic loss) can be achieved with simple implementations. The core concern in SOC is to identify controlled variables (CVs), such that when disturbances occur and the CVs are maintained at constant setpoints, the system is automatically operated near optimum without the need of re-optimization, or at least with only infrequent re-optimization. Since the implementation only requires simple feedback controllers, the optimizing speed of SOC is generally much faster than other RTO approaches (Francois et al., 2005; Jäschke and Skogestad, 2011; Ye et al., 2013a; 2014), which require on-line identification and optimization. Importantly, SOC is complementary to these other RTO approaches (Jäschke and Skogestad, 2011; Ye et al., 2017c), as the setpoints of the self-optimizing variables are still degrees of freedom to optimize. This reserves the possibility of handling unknown disturbances which are not considered in SOC.

In an SOC control system, the achievable optimizing performance depends on the CVs selected. Therefore, intensive research has been carried out on selecting optimal CVs to minimize the economic loss when disturbances and other uncertainties occur. Typ-

ically, linear combinations of the measurements,  $\mathbf{c} = \mathbf{H}\mathbf{y}$ , are considered where  $\mathbf{H}$  is the combination matrix to be optimized. Compared to the traditional cases where  $\mathbf{H}$  is a selection matrix with single measurements controlled, measurement combinations provide a more generalized option which allows the performance to be further improved. To this end, Halvorsen et al. (2003) proposed the so-called exact local method on the basis of a local analysis. The optimal CV selection problem was formulated as a nonlinear programming (NLP). Alstad and Skogestad (2007) considered a special case when there is no measurement noise and derived the null space method (Alstad and Skogestad, 2007), which has a simple explicit solution  $\mathbf{H}\mathbf{F} = \mathbf{0}$  where  $\mathbf{F} = \partial\mathbf{y}^{\text{opt}}/\partial\mathbf{d}$  is the sensitivity of optimal measurements with respect to the assumed disturbances,  $\mathbf{d}$ . However, Alstad et al. (2009) later showed that the NLP problem for the “exact local method” has an explicit analytical solution. Meanwhile, Kariwala (2007) and Kariwala et al. (2008) developed an eigenvalue decomposition approach for the exact local method, and the local worst-case and average losses were used as minimized criteria, respectively. The exact local method is based on a linearized local input-output model, hence satisfactory performance is only guaranteed within a small operating region around the nominal point. To overcome this drawback, approaches aimed at enlarging the self-optimizing operating region have recently been proposed (Ye et al., 2013a; 2015). For example, the global average loss over the entire nonlinear operating space was proposed to be minimized (Ye et al., 2015).

All the above approaches deal with static optimization problems, hence they are most suitable for continuous processes, where

\* Corresponding author.

E-mail address: [skoge@ntnu.no](mailto:skoge@ntnu.no) (S. Skogestad).

typically the economics are dominated by the steady states. For batch processes, which call for dynamic optimization, not so many systematic SOC methods have been reported. Previously, the maximum gain rule (Halvorsen et al., 2003) has been proposed for optimal operation of batch processes (Dahl-Olsen et al., 2008). However, the CV candidates were restricted to be single measurements, and the criterion was based on an inaccurate loss function. In another work, a local perturbation control approach was used to select CVs for dynamic systems (Hu et al., 2012a). Again, this method did not provide a solution to identify optimal measurement combinations. More recently, the null space method has been extended to dynamic processes (Oliveira et al., 2016). However, the application of the null space method is limited because of the inherent assumptions. In other works, the invariant of dynamic optimization, namely the Hamiltonian function, was proposed to be controlled for dynamic processes (Jaschke et al., 2011; Ye et al., 2013b). In these approaches, the Hamiltonian of dynamic optimization problems need to be analytically derived first, then they are inferred or numerically approximated with measurement combinations. Grema and Cao (2016) performed dynamic optimization of oil reservoir waterflooding processes under uncertainties. Their method was extended from the (data-based) NCO regression methods (Girei et al., 2014; Ye et al., 2013a). As pointed out in ref (Ye et al., 2014), an underlying drawback may be the over-regression for unnecessary non-optimal data points, and the lack of a rigorous way of dealing with measurement noise. Recently, a batch-to-batch SOC approach was proposed for batch processes, by utilizing the repetitive nature of batch processes (Ye et al., 2017b). However, a main disadvantage for the batch-to-batch optimization is that the input sequence is purely determined from historical measurements occurred at previous batches, hence the current uncertainties are not accounted for. In this paper, however, the within-batch optimization will be considered.

In general, the dynamic optimization techniques for batch process are classified into the analytic (indirect) approach and numerical (direct) approach (Srinivasan et al., 2003). The former is normally built upon the necessary conditions of optimality of Pontryagin's Minimum Principle and the Hamilton–Jacobi–Bellman principle. Lagrange multipliers (adjoint states) are typically introduced and solved in companion with the optimal inputs/states. Since it is symbolically difficult to eliminate the undesired adjoints from the equations of interest, the analytic approach is restricted to some small-scale processes (Srinivasan et al., 2003). Some of the SOC approaches (Jaschke et al., 2011; Ye et al., 2013b) mentioned above fall into this category. On the other hand, the numerical approach breaks the problem down into a sequence of decision steps over time, from which a static NLP problem is formulated (Biegler, 2007; Han et al., 2018). Although mathematically less rigorous, the performance loss due to numerical approximation turns out to be minor with a fine parameterization scheme. In practical cases, the numerical approach is easier to implement, and many optimization solvers/packages are available to solve large-scale dynamic problems efficiently, such as the NPSOL, SNOPT, IPOPT, etc (Biegler, 2007; Biegler and Zavala, 2009).

Following the perspective of the numerical approach, it is interesting to point out that existing static SOC approaches can be, in principle, adapted to dynamic systems on basis of the reformulated static NLP, which substantially reduces the gap between static and dynamic SOC problems. However, there are some potential obstacles in the context of dynamic problems, for example:

- The number of decision variables is large.
- For dynamic problems, causality issues need to be taken care of. More specifically, for a static NLP, the true optimal solution assumes full system information about the whole time horizon.

However, for operation, feedback controllers cannot make use of future measurements.

To cope with the causality problem, this paper specifies the coefficients associated with future measurements as zeros, thus resulting in a structure-constrained SOC problem. Among several possible structures of the combination matrix  $\mathbf{H}$ , the present study mainly considers a lower-block triangular structure, which admits the maximal attainable dynamic performance. In this regard, the exact local method is extended and a locally optimal dynamic SOC solution is obtained. An important new result is the derivation of a convex formulation with a structure-constrained  $\mathbf{H}$ , and an analytical solution for the optimal CV selection. The locally optimal CVs assigned to each time instant are solved separately along the time horizon. One may argue that the obtained SOC solution becomes complex, but this seems reasonable due to the dynamic nature of batch processes. Nonetheless, since the dynamic SOC design is carried out offline, the computational cost is not important.

The paper is organized as follows. In Section 2, the main existing results for static SOC methods are summarized. In particular, the exact local method is introduced, together with its convex formulation and analytic solution. In Section 3, the dynamic SOC problem is posed as a static one, however with the combination matrix structurally constrained. Solution methods are presented in this section. A fed-batch reactor and a batch distillation column are investigated in Section 4, and the final section concludes this article.

## 2. Self-optimizing control for static systems

### 2.1. Problem descriptions

Self-optimizing control (SOC) is concerned with a class of optimal operation problem in the presence of disturbances and uncertainties,  $\mathbf{d}$ . Consider a static NLP problem as follows

$$\begin{aligned} \min_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) \\ \text{s.t. } \mathbf{g}(\mathbf{u}, \mathbf{d}) \leq 0 \end{aligned} \quad (1)$$

with available measurements

$$\mathbf{y}_m = \mathbf{y} + \mathbf{n} = \mathbf{f}_y(\mathbf{u}, \mathbf{d}) + \mathbf{n} \quad (2)$$

where  $J$  is a cost function to be minimized, which is generally an economic index,  $\mathbf{u} \in \mathbb{R}^{n_u}$  and  $\mathbf{d} \in \mathbb{R}^{n_d}$  are the manipulated variables and disturbances, respectively. The disturbances may include also parameter changes.  $\mathbf{y}$ ,  $\mathbf{y}_m$ ,  $\mathbf{n} \in \mathbb{R}^{n_y}$  are the true, measured outputs and measurement error (noise).  $\mathbf{f}_y$  is the input-output mapping function and  $\mathbf{g}$  is the inequality constraints. Throughout this paper, the strong second-order sufficiency conditions (SOSCs) are assumed for an optimization problem, with guaranteed regularity and differentiability of all functions at the optimal solution.

Suppose that a subset of the constraints  $\mathbf{g}$  are active at the optimal point. In static SOC problems, these active constraints are supposed to be directly measured and controlled by assigning the same number of the degrees of freedom. We further assume that the active set is unchanged in the whole operating region. (Note that in the framework of SOC, the changing active set problem has also been considered elsewhere with different strategies (Cao, 2005; Hu et al., 2012b; Manum and Skogestad, 2012; Ye et al., 2017a).) In this case, one is able to formulate a reduced optimization problem using the remaining degrees of freedom (Ye et al., 2013a), provided with the linear independence constraint qualification (LICQ) and strict complementarity slackness. Since SOC is mainly concerned with the unconstrained part of a general NLP problem, to focus on the key methodology development, hereafter, we consider the following unconstrained optimization problem

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) \quad (3)$$

Note that for the sake of simplicity, we have used the same notation  $\mathbf{u}$  for the remaining degrees of freedoms, as in the original problem (1). To evaluate the optimality of the operation, the economic loss  $L$  is defined as the difference between the actual cost  $J(\mathbf{u}, \mathbf{d})$  and the optimal cost,  $J^{\text{opt}}(\mathbf{d}) = J(\mathbf{u}^{\text{opt}}(\mathbf{d}), \mathbf{d})$ , i.e.

$$L = J(\mathbf{u}, \mathbf{d}) - J^{\text{opt}}(\mathbf{d}) \quad (4)$$

The objective for SOC is to select measurement combinations as CVs,  $\mathbf{c} = \mathbf{H}\mathbf{y}$ , such that loss  $L$  is small or minimized, when  $\mathbf{c}$  are maintained at constant setpoints,  $\mathbf{c}_s$ . Here, the combination matrix  $\mathbf{H} \in \mathbb{R}^{n_c \times n_y}$  ( $n_c = n_u$ ) is on the form

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{n_c} \end{bmatrix} = \begin{bmatrix} h_{11} & \cdots & h_{1n_y} \\ \vdots & & \vdots \\ h_{n_c 1} & \cdots & h_{n_c n_y} \end{bmatrix}$$

where  $\mathbf{h}_i$  ( $i = 1, \dots, n_c$ ) is a row vector representing individual CVs.

To proceed, define an average (expected) loss over the entire uncertain operating region as follows (assuming that  $\mathbf{d}$  and  $\mathbf{n}$  are independent)

$$L_{\text{av}} = E[L] = \int_{\mathbf{d} \in \mathcal{D}, \mathbf{n} \in \mathcal{N}} \rho(\mathbf{d}) \rho(\mathbf{n}) L \, d\mathbf{n} d\mathbf{d} \quad (5)$$

where  $\mathcal{D}$  and  $\mathcal{N}$  is the variation region spanned by  $\mathbf{d}$  and  $\mathbf{n}$ , respectively,  $\rho(\cdot)$  is the probability density of a random variable. In this paper, we choose to use the average economic loss as the criterion. According to Kariwala et al. (2008), the  $\mathbf{H}$  that minimizes the average loss is “super-optimal”, because it simultaneously minimizes the worst-case loss hence guarantees a better economic performance.

## 2.2. Exact local method for static SOC

### 2.2.1. Loss evaluation

The exact local method (Halvorsen et al., 2003) was developed based on linearization of the plant model  $\mathbf{y} = \mathbf{f}_y(\mathbf{u}, \mathbf{d})$  around the optimal nominal point

$$\Delta \mathbf{y} = \mathbf{G}_y \Delta \mathbf{u} + \mathbf{G}_{yd} \Delta \mathbf{d} \quad (6)$$

where  $\mathbf{G}_y$  and  $\mathbf{G}_{yd}$  are gain matrices and the symbol “ $\Delta$ ” denotes a small deviation from the nominal value. For example,  $\Delta \mathbf{d} = \mathbf{d} - \mathbf{d}^*$  where  $\mathbf{d}$  is the actual disturbance and  $\mathbf{d}^*$  is the nominal disturbance. In the exact local method, the linear model is assumed to be a good approximation for the original nonlinear system  $\mathbf{f}_y$ .

A disturbance  $\Delta \mathbf{d}$  causes the optimal inputs  $\mathbf{u}_{\text{opt}}(\mathbf{d})$  to deviate from the nominal point,  $\mathbf{u}_{\text{opt}}(\mathbf{d}^*)$ . The change can be derived by linearizing the first order necessary conditions of optimality, i.e. the gradient  $\mathbf{J}_u$ , around the nominal point. The new optimal inputs are calculated as Halvorsen et al. (2003)

$$\Delta \mathbf{u}_{\text{opt}}(\mathbf{d}) = -\mathbf{J}_{uu}^{-1} \mathbf{J}_{ud} \Delta \mathbf{d} \quad (7)$$

where  $\mathbf{J}_{uu}$  and  $\mathbf{J}_{ud}$  are the second order sensitivities (Hessian) of  $J$ . Under the assumption of strong second-order sufficiency conditions,  $\mathbf{J}_{uu}$  is a symmetric positive definite matrix. Around this new optimum, the local economic loss is evaluated in a quadratic form in terms of input deviations, as

$$L = \frac{1}{2} \mathbf{e}_u^T \mathbf{J}_{uu} \mathbf{e}_u \quad (8)$$

where  $\mathbf{e}_u \triangleq \Delta \mathbf{u} - \Delta \mathbf{u}_{\text{opt}}(\mathbf{d})$  is the deviation between the actual input change  $\Delta \mathbf{u}$  and the optimal change  $\Delta \mathbf{u}_{\text{opt}}(\mathbf{d})$ . Given a combination matrix  $\mathbf{H}$ , the actual value of  $\mathbf{u}$  under closed-loop control of CVs ( $\mathbf{c} = \mathbf{H}\mathbf{y}$ ) is a function of the occurred disturbances and measurement noises (Halvorsen et al., 2003)

$$\Delta \mathbf{u} = -(\mathbf{H}\mathbf{G}_y)^{-1} \mathbf{H}\mathbf{G}_{yd} \Delta \mathbf{d} + (\mathbf{H}\mathbf{G}_y)^{-1} \mathbf{H}\mathbf{n} \quad (9)$$

Inserting (7) and (9) into (8) and with some rearrangements, the local loss is obtained as follows

$$L = \frac{1}{2} \|\mathbf{z}\|_2^2, \quad (10)$$

where

$$\mathbf{z} \triangleq \mathbf{V}(\mathbf{H}\mathbf{G}_y)^{-1} \mathbf{H} \underbrace{[\mathbf{F}\mathbf{W}_d \quad \mathbf{W}_n]}_{\tilde{\mathbf{F}}} \begin{bmatrix} \mathbf{d}' \\ \mathbf{n}' \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{d}' \\ \mathbf{n}' \end{bmatrix} \quad (11)$$

Here,  $\mathbf{V}$  is a matrix satisfying  $\mathbf{V}^T \mathbf{V} = \mathbf{J}_{uu}$  (Note that in existing literatures (Alstad et al., 2009; Halvorsen et al., 2003; Kariwala et al., 2008),  $\mathbf{V} = \mathbf{J}_{uu}^{1/2}$  is a typical practice),  $\mathbf{F} \triangleq \frac{\partial \mathbf{y}^{\text{opt}}}{\partial \mathbf{d}} = -\mathbf{G}_y \mathbf{J}_{uu}^{-1} \mathbf{J}_{ud} + \mathbf{G}_{yd}$  is the gain matrix of optimal measurements  $\mathbf{y}$  with respect to disturbances,  $\begin{bmatrix} \mathbf{d}' \\ \mathbf{n}' \end{bmatrix}$  is a scaled vector of the combined uncertainties  $\begin{bmatrix} \mathbf{d} \\ \mathbf{n} \end{bmatrix}$  such that all elements vary within a norm of 1, where  $\mathbf{W}_d$  and  $\mathbf{W}_n$  (diagonal) are their absolute magnitudes. The matrices  $\tilde{\mathbf{F}}$  and  $\mathbf{M}$  are defined as  $\tilde{\mathbf{F}} = [\mathbf{F}\mathbf{W}_d \quad \mathbf{W}_n]$  and  $\mathbf{M} = \mathbf{V}(\mathbf{H}\mathbf{G}_y)^{-1} \mathbf{H}\tilde{\mathbf{F}}$ .

The loss in (10) is for a specific single disturbance  $\mathbf{d}'$  and noise  $\mathbf{n}'$ . The average loss  $L_{\text{av}}$  defined in (5) can be derived (Alstad et al., 2009; Kariwala et al., 2008) assuming that the scaled disturbances and noise are normally distributed,  $\begin{bmatrix} \mathbf{d}' \\ \mathbf{n}' \end{bmatrix} \sim N(\mathbf{0}, \mathbf{I})$ . The resulting average loss is  $L_{\text{av}} = \frac{1}{2} \|\mathbf{M}\|_F^2$  so the local SOC problem becomes

$$\min_{\mathbf{H}} L_{\text{av}} = \min_{\mathbf{H}} \frac{1}{2} \|\mathbf{M}\|_F^2 \quad (12)$$

where  $\|\cdot\|_F$  stands for Frobenius norm of a matrix. Notice that, this formulation also applies to other distributions for the random vector  $\begin{bmatrix} \mathbf{d}' \\ \mathbf{n}' \end{bmatrix}$ , for example a uniform distribution, because the loss function differs by a constant factor, which does not affect the optimal  $\mathbf{H}$  (Kariwala et al., 2008; Yelchuru and Skogestad, 2012).

### 2.2.2. Convex formulation for static SOC problems

It has been proved (Alstad et al., 2009) that a convex formulation to the local SOC problem (12) can be derived which yields an analytic optimal  $\mathbf{H}$ . To explain this point, we first note that the following lemma is true for a general SOC problem:

**Lemma 1.** Alstad et al. (2009). The average loss by using a transformed combination matrix,  $\mathbf{H}' = \mathbf{Q}\mathbf{H}$  is equivalent to  $\mathbf{H}$ , i.e.  $L_{\text{av}}(\mathbf{H}') = L_{\text{av}}(\mathbf{H})$ , where  $\mathbf{Q} \in \mathbb{R}^{n_u \times n_u}$  is any nonsingular matrix.

The correctness of this lemma can be easily confirmed by inserting  $\mathbf{H}' = \mathbf{Q}\mathbf{H}$  into  $L_{\text{av}}$  in equation (12) such that two terms  $\mathbf{Q}^{-1}$  and  $\mathbf{Q}$  cancel with each other, under the assumption that  $\mathbf{Q}$  is invertible.

Lemma 1 has some implications. First, the solution to (12) is not unique. To guarantee uniqueness, some constraints are required so that extra degrees of freedom in the decision variables  $\mathbf{H}$  are eliminated. Second, such constraints can be appropriately chosen such that (12) is significantly simplified, for example  $\mathbf{H}\mathbf{G}_y = \mathbf{V}$  ( $\mathbf{V} = \mathbf{J}_{uu}^{1/2}$ ) is typically utilized (Alstad et al., 2009; Ye et al., 2015). This way, (12) is reformulated into a constrained convex optimization problem. Based on these implications, the following two theorems outline the convex formulation of exact local method, together with an analytic optimal solution.

**Theorem 1.** The solution to the local SOC problem shown in (12) can be solved through the following reformulated constrained convex quadratic programming (QP) (Alstad et al., 2009)

$$\begin{aligned} \min_{\mathbf{H}} L_{\text{av}} &= \min_{\mathbf{H}} \frac{1}{2} \|\mathbf{H}\tilde{\mathbf{F}}\|_F^2 \\ \text{s.t.} \quad \mathbf{H}\mathbf{G}_y &= \mathbf{J}_{uu}^{1/2} \end{aligned} \quad (13)$$

**Theorem 2.** An analytic solution to the convex constrained QP (13) is given as (Alstad et al., 2009)

$$\mathbf{H}^T = (\tilde{\mathbf{F}}\tilde{\mathbf{F}}^T)^{-1} \mathbf{G}_y (\mathbf{G}_y^T (\tilde{\mathbf{F}}\tilde{\mathbf{F}}^T)^{-1} \mathbf{G}_y)^{-1} \mathbf{J}_{uu}^{1/2} \quad (14)$$

For the original optimization problem in (12), because of the non-uniqueness, the solution can be further simplified to (Rangaiah and Kariwala, 2012; Yelchuru and Skogestad, 2012))

$$\mathbf{H}^T = (\tilde{\mathbf{F}}\tilde{\mathbf{F}}^T)^{-1} \mathbf{G}_y \quad (15)$$

as the dropped part is a nonsingular square matrix.

The proofs of above theorems can be found in the cited references, hence they are not further presented for brevity.

### 3. Dynamic self-optimizing control for batch processes

In this paper, we extend the static SOC methods to the dynamic optimization problem of batch processes, such that near-optimal operation can be achieved via selecting optimal CVs. For methodology development, we deal with batch processes with the following assumptions:

**Assumption 1.** The batch process is represented in a discrete-time form with a fixed finite time horizon.

**Assumption 2.** Measurements are available at all discrete time points with no time delays.

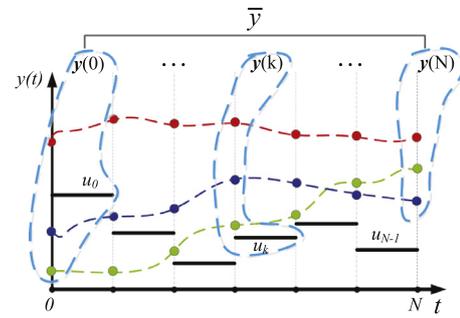
**Assumption 3.** The operation of the batch system is unconstrained, or can be transformed into an unconstrained operation problem.

**Assumption 1** admits a simple form with a static reformulation of the optimization problem. In general, the continuous physical system of batch process can be approximated with a discrete representation, for example, by appropriate control parameterizations. Variations of the process variables between the discrete time points are overlooked, which is reasonable provided that we use a sufficiently fine discretization. **Assumption 2** ensures that measurements can be used to constitute the CVs immediately after their occurrence. Otherwise, the structure of CV combination matrix will be additionally constrained. **Assumption 3** may seem restrictive as batch processes are typically operated under various constraints. However, the active constraints are commonly controlled variables in the first place. Furthermore, many constraints can be transformed into unconstrained formulations. For example, input saturation constraints can be eliminated as switching time between successive input arcs, and end point constraints can be absorbed in the cost function using the penalty approach; see the batch column case in Section 4 as an example. It should be mentioned that constraints in batch systems are difficult to handle in general. There are two main types of constraints, namely path constraints and terminal constraints. Path constraints may be handled using open-loop operation or on-line feedback control with associated input arcs (Francois et al., 2005), where the remaining input arcs are then parameterized to form a static NLP. Terminal constraints are generally difficult to handle within a single batch. In any case, systematic ways dealing with various constraints are beyond the scope of this paper; readers are referred to, e.g. (Srinivasan and Bonvin, 2007), for discussions regarding various constraints. The scope of this work is restricted to cases where **Assumption 3** holds.

#### 3.1. Discrete-time batch process

The discrete-time batch process is described as

$$\mathbf{x}(k+1) = \mathbf{f}^k(\mathbf{x}(k), \mathbf{u}(k), \tilde{\mathbf{d}}(k)) \quad (16)$$



**Fig. 1.** Measurements in the operation horizon (Solid circles and dashed lines in color: measurements at discrete and within grids; Solid bars: manipulated variables).

$$\mathbf{y}_m(k) = \underbrace{\mathbf{g}^k(\mathbf{x}(k), \mathbf{u}(k))}_{\mathbf{y}(k)} + \mathbf{n}(k)$$

with a fixed finite time horizon  $[0, N]$ .  $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ ,  $\mathbf{x}(k) \in \mathbb{R}^{n_x}$  and  $\mathbf{y}(k)/\mathbf{y}_m(k) \in \mathbb{R}^{n_y}$  denote the manipulated variables, state variables and true/measured output variables at time instant  $k$ , respectively.  $\mathbf{f}^k$  and  $\mathbf{g}^k$  are the time-varying state dynamic and measurement functions.  $\tilde{\mathbf{d}}$  and  $\mathbf{n}$  represent disturbances and measurement errors(noise). Furthermore, we may consider the initial system states, denoted as  $\mathbf{x}_0 \equiv \mathbf{x}(0)$ , are uncertain. Hence, both  $\mathbf{x}(0)$  and  $\tilde{\mathbf{d}}$  are jointly referred as the *disturbances*,  $\tilde{\mathbf{d}}^T \triangleq [\mathbf{x}_0^T \tilde{\mathbf{d}}^T]$ .

The objective of plant operation is to minimize the following general cost function

$$J = \phi(\mathbf{x}(N)) + \sum_{k=0}^{N-1} \psi^k(\mathbf{x}(k), \mathbf{u}(k)) \quad (17)$$

where  $\phi$  is a scalar cost associated with the final state  $\mathbf{x}(N)$ , and  $\psi^k$  is the contribution at time  $k$  to the integrated cost, which is allowed to be a time-varying function of states and inputs.

Based on the above, the dynamic optimization is formulated as

$$\min_{\tilde{\mathbf{u}}} J = \phi(\mathbf{x}(N)) + \sum_{k=0}^{N-1} \psi^k(\mathbf{x}(k), \mathbf{u}(k)) \quad (18)$$

s.t. *dynamic model* : (16)

where  $\tilde{\mathbf{u}}$  is a stacked input variable vector defined as

$$\tilde{\mathbf{u}} \triangleq [\mathbf{u}(0)^T \quad \dots \quad \mathbf{u}(N-1)^T]^T \quad (19)$$

with the dimension  $n_{\tilde{\mathbf{u}}} = Nn_u$ .

#### 3.2. Selection of candidate measurements

Similar to the stacked input variable in (19), the stacked states and measurements can also be defined along the time sequence

$$\begin{aligned} \tilde{\mathbf{x}} &\triangleq [\mathbf{x}(0)^T \quad \dots \quad \mathbf{x}(N)^T]^T \\ &= [\mathbf{x}(0) \quad \dots \quad \mathbf{f}^{N-1}(\mathbf{x}(N-1), \mathbf{u}(N-1), \tilde{\mathbf{d}}(N-1))]^T \end{aligned} \quad (20)$$

$$\begin{aligned} \tilde{\mathbf{y}} &\triangleq [\mathbf{y}(0)^T \quad \dots \quad \mathbf{y}(N)^T]^T \\ &= [\mathbf{g}^0(\mathbf{x}(0), \mathbf{u}(0)) \quad \dots \quad \mathbf{g}^N(\mathbf{x}(N), \mathbf{u}(N))]^T \end{aligned} \quad (21)$$

We will assume that  $\mathbf{u}(k)$  is included in the measurement set  $\mathbf{y}(k)$  and that  $\mathbf{u}(k)$  is kept constant between the discretized samples. See Fig. 1 for a visual illustration.

With full information about the system model and its parameters/disturbances, it is well known that the dynamic optimization

problem (18) can be solved through a static reformulation, by using for example dynamic programming (Biegler and Zavala, 2009). In general, the following static NLP problem can be considered

$$\begin{aligned} \min_{\mathbf{u}} J &= \phi(\mathbf{x}(N)) + \sum_{k=0}^{N-1} \psi^k(\mathbf{x}(k), \mathbf{u}(k)) \\ \text{s.t.} \quad \mathbf{x}(\mathbf{0}) &= \mathbf{x}_0 \\ \bar{\mathbf{x}} &= \mathbf{f}(\bar{\mathbf{u}}, \bar{\mathbf{d}}), \quad \bar{\mathbf{y}} = \mathbf{g}(\bar{\mathbf{u}}, \bar{\mathbf{d}}) \end{aligned} \quad (22)$$

where  $\bar{\mathbf{d}}$  is the stacked disturbances,  $\mathbf{f}$  and  $\mathbf{g}$  are the reformulated static mapping function of states and outputs, respectively. In the above dynamic optimization problem, the strong second-order sufficiency conditions for optimality and regularity conditions are also assumed.

**Objective:** Self-optimizing control (SOC) aims to identify a set of CVs which are to be controlled at constant setpoints, such that the economic loss is acceptable even when there are uncertainties. For batch processes, a distinct feature differing it from the static cases is that available measurements  $\bar{\mathbf{y}}$  along the time direction, reflect the system dynamics. Therefore, all measurements from the entire horizon,  $\bar{\mathbf{y}}$ , can be regarded as potential candidates for the purpose of SOC. The objective of this paper is to identify an optimal combination matrix  $\bar{\mathbf{H}}$ , such that economic loss is minimized by selecting self-optimizing CVs,  $\mathbf{c} = \bar{\mathbf{H}}\bar{\mathbf{y}}$ .

As in the static SOC, the problem formulation (22) integrates all uncertainties, namely the initial states, parametric process disturbances and measurement noise, into a unified framework, which later leads to a pure output feedback control. This is different from the formulation of classic optimal control problems (Bryson and Ho, 1975), where these uncertainties are handled in different fashions, thus leading to different online implementations. More detailed discussions are given in section 3.7.

### 3.3. Structure of combination matrix

In general, we want to select  $n_c = n_{\bar{u}}$  controlled variables,  $\mathbf{c} = \bar{\mathbf{H}}\bar{\mathbf{y}}$ , with  $\bar{\mathbf{H}}$  of dimension  $n_c \times n_{\bar{y}}$ , whose row vectors represent individual CVs. Regarding the structure of combination matrix  $\bar{\mathbf{H}}$ , which would be an important issue for the dynamic SOC problem, the following typical structures are discussed:

- (1) *Structure 1: Full  $\bar{\mathbf{H}}$*  (infeasible because of causality). This is the most general case, where all elements in  $\bar{\mathbf{H}}$  are allowed to vary and considered as decision variables for self-optimizing control. Interestingly, this case is equivalent to the static SOC problem where the structure of combination matrix is full. Since the static SOC has been proved to be convex with an analytic solution given in Section 2, a full  $\bar{\mathbf{H}}$  here is also solvable for the dynamic problem. Unfortunately, here a causality problem arises. For example, at time instant  $k$ , measurements corresponding to  $t > k$  are not available. However, in a full matrix  $\bar{\mathbf{H}}$ , those elements associated with future measurements are not necessarily zero, so in practice it is impossible to control the desired CVs within one batch.
- (2) *Structure 2: Lower-block triangular (LBT)  $\bar{\mathbf{H}}$* . In this case, the causality problem is avoided by restricting  $\bar{\mathbf{H}}$  to be a lower-block triangular (LBT) matrix

$$\begin{aligned} \bar{\mathbf{H}} &= \begin{bmatrix} \mathbf{H}_0(0) & 0 & 0 & \dots & 0 \\ \mathbf{H}_0(1) & \mathbf{H}_1(1) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_0(N-1) & \mathbf{H}_1(N-1) & \mathbf{H}_2(N-1) & \dots & \mathbf{H}_{N-1}(N-1) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{H}(0) & 0 & 0 & \dots & 0 \\ \mathbf{H}(1) & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ & & & & \mathbf{H}(N-1) \end{bmatrix} \\ &\text{written as } \text{lbt}[\mathbf{H}(0), \mathbf{H}(1), \dots, \mathbf{H}(N-1)] \end{aligned} \quad (23)$$

where  $\mathbf{H}_i(k)$  is of dimension  $n_u \times n_y$ , which represents the contribution of measurements at time  $i$  to CVs at time  $k$ . For brevity, we define  $\mathbf{H}(k) \triangleq [\mathbf{H}_0(k) \dots \mathbf{H}_k(k)]$  of dimension  $n_u \times (n_y k)$ , and furthermore write the big combination matrix as  $\bar{\mathbf{H}} = \text{lbt}[\mathbf{H}(0), \dots, \mathbf{H}(N-1)]$ . We will see below that we can derive an analytical optimal solution for this structure (Theorem 3).

By specifying the elements associated with future measurements as zero, we only use measurements available up to time  $k$  (denoted as  $\bar{\mathbf{y}}(k)$ ) for SOC purpose. In addition, it would be a natural choice to control the CVs associated with time  $t = k$ ,

$$\mathbf{c}(k) = \mathbf{H}(k)\bar{\mathbf{y}}(k), \quad k = 0, \dots, N-1 \quad (24)$$

in sequence as the operation proceeds.

- (3) *Structure 3: diagonal  $\bar{\mathbf{H}}$  (time-varying)*. A further simplified case is when  $\bar{\mathbf{H}}$  is a block-diagonal matrix

$$\begin{aligned} \bar{\mathbf{H}} &= \begin{bmatrix} \mathbf{H}(0) & 0 & 0 & \dots & 0 \\ 0 & \mathbf{H}(1) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{H}(N-1) \end{bmatrix} \\ &\text{written as } \text{diag}[\mathbf{H}(0), \dots, \mathbf{H}(N-1)] \end{aligned} \quad (25)$$

Here, the sub-matrices  $\mathbf{H}(k)$  are of dimension  $n_u \times n_y$  and different from the ones defined in (23). Then, individual CVs are decomposed as

$$\mathbf{c}(k) = \mathbf{H}(k)\mathbf{y}(k), \quad k = 0, \dots, N-1 \quad (26)$$

which only makes use of the current measurements  $\mathbf{y}(k)$ . We will see below that in the noise-free case and sufficient number of measurements we can use the nullspace method to derive an optimal solution for this structure (section 3.7.1). However, more generally, Structure 3 must be obtained numerically.

- (4) *Structure 4: diagonal  $\bar{\mathbf{H}}$  (time-invariant)*. To further simplify  $\bar{\mathbf{H}}$ , one may enforce all elements  $\mathbf{H}(k)$  in eq (25) to be the same,

$$\begin{aligned} \bar{\mathbf{H}} &= \begin{bmatrix} \mathbf{H} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{H} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{H} \end{bmatrix} \text{written as } \text{diag}(\mathbf{H}) \end{aligned} \quad (27)$$

where  $\mathbf{H}$  is a time-invariant sub-matrix. This time, the individual CVs are

$$\mathbf{c}(k) = \mathbf{H}\mathbf{y}(k), \quad k = 0, \dots, N-1 \quad (28)$$

It is evident that as one proceeds from Structure 1 to 4, the sub-optimality (loss) increases because the latter structure is a special case of the previous one. However, the latter cases are easier to implement in operation. On the other hand, for the last two cases, the mathematical solutions to find the optimal  $\bar{\mathbf{H}}$  are

generally difficult (Yelchuru and Skogestad, 2012). Since Structure 1 is non-causal and thus not feasible, we will select Structure 2 as the starting point to analyze the SOC performance. The limitation of the combination matrix  $\bar{\mathbf{H}}$  to have a particular form, as in Structures 2–4, is known as the structure-constrained SOC problem (Yelchuru and Skogestad, 2012). Such problems are generally believed to be non-convex problems and no closed-form solutions were derived (Yelchuru, 2012; Yelchuru and Skogestad, 2012). In this paper, however, we will show that a convex constrained QP can be formulated for Structure 2, and then an analytic solution follows. Therefore, the dynamic self-optimizing control problem for batch processes can be efficiently solved. However, for Structures 3 and 4, the problems remain non-convex. It should be highlighted that when the assumption of no measurement delay is not true, more elements in Structure 2 should be zeros. For example, if the measurements are delayed one sample, the block diagonal elements in  $\bar{\mathbf{H}}$  should be zeros, because they are associated with the current measurements.

### 3.4. Linear model and sensitivity matrices

To proceed in the same way as in the static case, let us assume that at the nominal operating conditions with zero uncertainties, an optimal input sequence,  $\mathbf{u}_{\text{opt}}^*(k)$  is found. Then, the nonlinear plant (16) is linearized along the obtained nominal trajectory, which gives the following locally linear dynamic model in terms of deviation variables

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{B}_{\bar{d}}(k)\bar{\mathbf{d}}(k) \\ \mathbf{y}_m(k) &= \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k) + \mathbf{n}(k) \end{aligned} \quad (29)$$

where  $\mathbf{A}(k) = \partial \mathbf{f}^k / \partial \mathbf{x}(k)$ ,  $\mathbf{B}(k) = \partial \mathbf{f}^k / \partial \mathbf{u}(k)$ ,  $\mathbf{B}_{\bar{d}}(k) = \partial \mathbf{f}^k / \partial \bar{\mathbf{d}}(k)$ ,  $\mathbf{C}(k) = \partial \mathbf{g}^k / \partial \mathbf{x}(k)$  and  $\mathbf{D}(k) = \partial \mathbf{g}^k / \partial \mathbf{u}(k)$ , respectively.

By iteratively applying the dynamic linear model (29), the input-output gain matrix between the stacked process variables  $\bar{\mathbf{y}}$  and  $\bar{\mathbf{u}}$ , denoted as  $\bar{\mathbf{G}}_y$ , becomes

$$\bar{\mathbf{G}}_y \triangleq \frac{\partial \bar{\mathbf{y}}}{\partial \bar{\mathbf{u}}} = \begin{bmatrix} \mathbf{D}(0) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}(1)\mathbf{B}(0) & \mathbf{D}(1) & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}(N-1)\mathbf{A}(N-2)\dots\mathbf{A}(1)\mathbf{B}(0) & \mathbf{C}(N-1)\mathbf{A}(N-2)\dots\mathbf{A}(2)\mathbf{B}(1) & \dots & \mathbf{D}(N-1) \end{bmatrix} \quad (30)$$

In the same way, the disturbance-output gain matrix can be obtained. First, the gain matrix of  $\bar{\mathbf{y}}$  associated with the initial states  $\mathbf{x}_0$  becomes

$$\bar{\mathbf{G}}_{y\mathbf{x}0} \triangleq \frac{\partial \bar{\mathbf{y}}}{\partial \mathbf{x}_0} = \begin{bmatrix} \mathbf{C}(0)\mathbf{A}(0) \\ \mathbf{C}(1)\mathbf{A}(1)\mathbf{A}(0) \\ \vdots \\ \mathbf{C}(N-1)\mathbf{A}(N-2)\dots\mathbf{A}(0) \end{bmatrix} \quad (31)$$

In a general case, where we allow the parametric disturbance,  $\bar{\mathbf{d}}$ , to be time varying, the gain matrix of  $\bar{\mathbf{y}}$  in terms of the time stacked disturbance  $\bar{\mathbf{d}} \triangleq [\bar{\mathbf{d}}(0)^T \dots \bar{\mathbf{d}}(N-1)^T]^T$ ,  $\bar{\mathbf{G}}_{y\bar{d}}$  becomes

$$\bar{\mathbf{G}}_{y\bar{d}} \triangleq \frac{\partial \bar{\mathbf{y}}}{\partial \bar{\mathbf{d}}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}(1)\mathbf{B}_{\bar{d}}(0) & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots \\ \mathbf{C}(N-1)\mathbf{A}(N-2)\dots\mathbf{A}(1)\mathbf{B}_{\bar{d}}(0) & \mathbf{C}(N-1)\mathbf{A}(N-2)\dots\mathbf{A}(2)\mathbf{B}_{\bar{d}}(1) & \dots & \mathbf{0} \end{bmatrix} \quad (32)$$

Finally, the overall disturbance-output sensitivity matrix  $\mathbf{G}_{yd}$  readily follows as  $\bar{\mathbf{G}}_{yd} = [\bar{\mathbf{G}}_{y\mathbf{x}0} \quad \bar{\mathbf{G}}_{y\bar{d}}]$ . Notice that, in some cases, we may only be interested in time-invariant disturbance  $\bar{\mathbf{d}}$  ( $\bar{\mathbf{d}}(k) = \bar{\mathbf{d}}(0)$ ), that is, disturbances that occur initially remain unchanged in a single batch. It can then be easily shown that the gain matrix  $\bar{\mathbf{G}}_{y\bar{d}} \triangleq \frac{\partial \bar{\mathbf{y}}}{\partial \bar{\mathbf{d}}(0)}$  is equal to the sum of (32) along the row direction.

Based on above sensitivity matrices, the local input-output linear model is

$$\bar{\mathbf{y}} = \bar{\mathbf{G}}_y \bar{\mathbf{u}} + \bar{\mathbf{G}}_{y\bar{d}} \bar{\mathbf{d}} \quad (33)$$

where  $\bar{\mathbf{d}} \triangleq [\mathbf{x}_0^T \bar{\mathbf{d}}^T]^T$ . One notes that above relationship is in the same form as in the static case, except that the process variables are defined as stacked ones.

In the same fashion, the other matrices, for example  $\bar{\mathbf{J}}_{uu}$ ,  $\bar{\mathbf{F}}$ ,  $\bar{\bar{\mathbf{F}}}$ ,  $\bar{\mathbf{W}}_d$ ,  $\bar{\mathbf{W}}_n$ , can also be readily extended to the dynamic case. For example,

$$\begin{aligned} \bar{\mathbf{J}}_{uu} &\triangleq \frac{d^2 J}{d\bar{\mathbf{u}}^2} \\ \bar{\mathbf{F}} &\triangleq -\bar{\mathbf{C}}_y \bar{\mathbf{J}}_{uu}^{-1} \bar{\mathbf{J}}_{ud} + \bar{\mathbf{C}}_{yd} \end{aligned}$$

and so on; see (5)–(10) for their definitions in the static case. Note that  $\bar{\mathbf{J}}_{uu}$  is defined as the total derivative, which follows by eliminating state variables in the dynamic process model. Under the assumption of strong second order sufficient condition,  $\bar{\mathbf{J}}_{uu}$  is positive definite. By utilizing the available results in the static case, we are allowed to formulate the following dynamic SOC problem:

$$\min_{\bar{\mathbf{H}}} L_{av} = \min_{\bar{\mathbf{H}}} \frac{1}{2} \|\mathbf{V}(\bar{\mathbf{H}}\bar{\mathbf{C}}_y)^{-1}\bar{\mathbf{H}}\bar{\bar{\mathbf{F}}}\|_F^2 \quad (34)$$

s.t.  $\bar{\mathbf{H}}$  on the form of a particular structure

where  $\mathbf{V}$  is a matrix satisfying  $\mathbf{V}^T \mathbf{V} = \bar{\mathbf{J}}_{uu}$ .

### 3.5. Main result: Optimal LBT-structured $\bar{\mathbf{H}}$

Based on the reformulation of the static NLP, the static SOC results can be extended to dynamic systems with notations adaptations. However, the solution given in Theorem 2 is a “full” combination matrix (Structure 1). As has been explained, Structure 1 suffers from causality problem for dynamic systems and cannot be used here. The causal Structures 2–4 impose structural con-

straints. In general, structural constraints cannot be reformulated as convex QP problems (at least for static processes) (Yelchuru, 2012; Yelchuru and Skogestad, 2012). Instead, it was proposed in Yelchuru (2012) to solve these structured problems with: (1) brute-force search using a nonlinear optimization solver, which is computationally inefficient and furthermore suffers from local optima; (2) convex approximation by dropping the nonlinear term, in particular the term  $(\mathbf{H}\mathbf{G}_y)^{-1}$ , which leads to suboptimal solutions.

In this paper, we will show that a convex QP can be formulated for the case of a lower-block triangular (LBT) structured  $\bar{\mathbf{H}}$  (Structure 2). Furthermore, the reformulated constrained QP can be

solved with an analytical solution. The main results are presented as follows.

**Lemma 2.** For a LBT-structured combination matrix  $\bar{\mathbf{H}}$  in (23), a nonsingular transformation  $\bar{\mathbf{H}}' = \mathbf{Q}\bar{\mathbf{H}}$  is also LBT, if the nonsingular  $\mathbf{Q}$  is LBT.

The correctness of Lemma 2 can be established very easily. Using this result and Lemma 1, we further know that the losses of both LBT-structured  $\tilde{\mathbf{H}}'$  and  $\tilde{\mathbf{H}}$  are equivalent, because Lemma 1 does not impose extra structural requirements on  $\mathbf{Q}$ .

**Theorem 3.** (Convex formulation for dynamic SOC problem). For the LBT-structured  $\tilde{\mathbf{H}}$  in (23), the optimal solution of (34) is equivalent to the following convex constrained QP problem

$$\begin{aligned} \min_{\tilde{\mathbf{H}}} L_{av} &= \min_{\tilde{\mathbf{H}}} \frac{1}{2} \|\tilde{\mathbf{H}}\tilde{\mathbf{F}}\|_F^2 \\ \text{s.t.} \quad &\tilde{\mathbf{H}}\tilde{\mathbf{G}}_y = \mathbf{V} \\ &\tilde{\mathbf{H}} \text{ on form of } \tilde{\mathbf{H}} = \text{lbt}[\mathbf{H}(0), \dots, \mathbf{H}(N-1)] \end{aligned} \quad (35)$$

where  $\mathbf{V}$  is a lower triangular matrix satisfying  $\mathbf{V}^T\mathbf{V} = \tilde{\mathbf{J}}_{uu}$ .

**Proof.** First, we will prove that there is no loss in generality by enforcing the equality constraint of  $\tilde{\mathbf{H}}\tilde{\mathbf{G}}_y = \mathbf{V}$  when solving  $\tilde{\mathbf{H}} = \text{lbt}[\mathbf{H}(0), \dots, \mathbf{H}(N-1)]$ , on the condition that  $\mathbf{V}$  is lower triangular. This can be proved by recognizing the following facts:

1. For an arbitrary LBT-structured  $\tilde{\mathbf{H}}_{unc}$ , which may not satisfy the equality constraint, we can always find a  $\mathbf{Q} = \mathbf{V}(\tilde{\mathbf{H}}_{unc}\tilde{\mathbf{G}}_y)^{-1}$ , such that  $\tilde{\mathbf{H}}_{con} = \mathbf{Q}\tilde{\mathbf{H}}_{unc}$  satisfying the constraint  $\tilde{\mathbf{H}}_{con}\tilde{\mathbf{G}}_y = \mathbf{V}$ .
2. Since  $\mathbf{Q} = \mathbf{V}(\tilde{\mathbf{H}}_{unc}\tilde{\mathbf{G}}_y)^{-1}$  is LBT (note that  $\tilde{\mathbf{G}}_y$  is LBT, see eq (30)) and  $\tilde{\mathbf{H}}_{unc}$  is LBT, then from Lemma 2,  $\tilde{\mathbf{H}}_{con}$  is also LBT.
3. The losses of  $\tilde{\mathbf{H}}_{con}$  and  $\tilde{\mathbf{H}}_{unc}$  are equivalent according to Lemma 1. Therefore, the introduced equality constraint does not lose generality in searching for an optimal  $\tilde{\mathbf{H}}$ .

Second, there exists a lower-triangular  $\mathbf{V}$  such that  $\mathbf{V}^T\mathbf{V} = \tilde{\mathbf{J}}_{uu}$ . This can be confirmed by recognizing the existence of Cholesky decomposition of a symmetric, positive definite matrix  $\tilde{\mathbf{J}}_{uu}$ . However, the normal Cholesky decomposition is defined as a multiplication of a lower triangular matrix and its transpose, which is a reversed order in our problem. This can be addressed by rotating  $\tilde{\mathbf{J}}_{uu}$  by 180-degree and performing a normal Cholesky decomposition and then rotating the decomposed matrix back.

Third, by inserting the enforced relationship  $\tilde{\mathbf{H}}\tilde{\mathbf{G}}_y = \mathbf{V}$  into the objective function of  $L_{av} = \frac{1}{2} \|\mathbf{V}(\tilde{\mathbf{H}}\tilde{\mathbf{G}}_y)^{-1}\tilde{\mathbf{H}}\tilde{\mathbf{F}}\|_F^2$ , then  $L_{av} = \frac{1}{2} \|\tilde{\mathbf{H}}\tilde{\mathbf{F}}\|_F^2$  follows.

The above facts leads to the proposed convex constrained QP problem. □

*Comments:* The above convex formulation is similar to the static case in the sense that both of them have utilized the trick of a nonsingular transformation (Lemma 1). However, the main difference is that due to causality issue in the dynamic process, the combination matrix must be LBT, which requires a different equality constraint. Here, it is critical to choose a slightly different lower triangular matrix  $\mathbf{V}$ , rather than a full  $\tilde{\mathbf{J}}_{uu}^{1/2}$ . It is also worth mentioning that the convex formulation applies to the LBT-structured  $\tilde{\mathbf{H}}$  but not to the diagonal-structured one (Structures 3 and 4). This is because in point 1 of the proof, where the transformation  $\tilde{\mathbf{H}}_{con} = \mathbf{Q}\tilde{\mathbf{H}}_{unc}$  is used,  $\tilde{\mathbf{H}}_{con}$  and  $\tilde{\mathbf{H}}_{unc}$  are no longer structurally consistent, because both  $\tilde{\mathbf{G}}_y$  and  $\mathbf{V}$  are not block diagonal. In this case, enforcing the proposed constraint loses generality and the derivations no longer hold.

For the convex constrained QP, an analytic solution also follows, as given in the following theorem.

**Theorem 4.** (Analytic solution of the dynamic SOC problem in Theorem 3) The optimal solution to the convex constrained QP (35) can be solved by determining submatrices  $\mathbf{H}(i)$  separately, as

$$\mathbf{H}(i)^T = (\tilde{\mathbf{F}}_i\tilde{\mathbf{F}}_i^T)^{-1}\mathbf{G}_{yi}\left(\mathbf{G}_{yi}^T(\tilde{\mathbf{F}}_i\tilde{\mathbf{F}}_i^T)^{-1}\mathbf{G}_{yi}\right)^{-1}\mathbf{V}_i^T \quad (36)$$

where  $\tilde{\mathbf{F}}_i$  is a submatrix of  $\tilde{\mathbf{F}}$  intersected by its first  $n_y(i+1)$  rows and  $(n_d + n_y(i+1))$  columns,  $\mathbf{G}_{yi}$  is a submatrix of  $\mathbf{G}_y$  intersected by

its first  $n_y(i+1)$  rows and first  $n_u(i+1)$  columns,  $\mathbf{V}_i$  denotes the  $i$ th partition of  $\mathbf{V}$ .

**Proof.** Based on the convex constrained QP (35), decomposing the overall loss row-wise in terms of  $\tilde{\mathbf{H}}(i)$ , we have

$$\begin{aligned} L_{av} &= \frac{1}{2} \|\tilde{\mathbf{H}}\tilde{\mathbf{F}}\|_F^2 = \frac{1}{2} \left\| \begin{bmatrix} \mathbf{H}(0) & 0 & 0 & \dots & 0 \\ & \mathbf{H}(1) & 0 & \dots & 0 \\ & & \vdots & \ddots & \vdots \\ & & & & \mathbf{H}(N-1) \end{bmatrix} \right. \\ &\quad \times \left. \begin{bmatrix} \tilde{\mathbf{F}}_0 & & & & \\ & \tilde{\mathbf{F}}_1 & & & \\ & & \ddots & & \\ & & & & \tilde{\mathbf{F}}_{N-1} \end{bmatrix} \right\|_F^2 \\ &= \frac{1}{2} \left\| \begin{bmatrix} \mathbf{H}(0)\tilde{\mathbf{F}}_0 \\ \vdots \\ \mathbf{H}(N-1)\tilde{\mathbf{F}}_{N-1} \end{bmatrix} \right\|_F^2 = \sum_{i=1}^{N-1} \|\mathbf{H}(i)\tilde{\mathbf{F}}_i\|_F^2 \end{aligned} \quad (37)$$

On the other hand, the constraint  $\tilde{\mathbf{H}}\tilde{\mathbf{G}}_y = \mathbf{V}$  is also decomposed as

$$\begin{aligned} \tilde{\mathbf{H}}\tilde{\mathbf{G}}_y &= \mathbf{V} \\ \Leftrightarrow &\begin{bmatrix} \mathbf{H}(0) & 0 & 0 & \dots & 0 \\ & \mathbf{H}(1) & 0 & \dots & 0 \\ & & \vdots & \ddots & \vdots \\ & & & & \mathbf{H}(N-1) \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{G}}_{y0} & & & & \\ & \tilde{\mathbf{G}}_{y1} & & & \\ & & \ddots & & \\ & & & & \tilde{\mathbf{G}}_{y(N-1)} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{V}_0 & 0 & 0 & \dots & 0 \\ & \mathbf{V}_1 & 0 & \dots & 0 \\ & & \vdots & \ddots & \vdots \\ & & & & \mathbf{V}_{N-1} \end{bmatrix} \\ \Leftrightarrow &\mathbf{H}(i)\tilde{\mathbf{G}}_{yi} = \mathbf{V}_i, \quad \forall i = 0, \dots, N-1 \end{aligned} \quad (38)$$

Since both the objective function and equality constraint have been decomposed onto individual  $\mathbf{H}(i)$ , it is equivalent to solving the following  $N$  constrained optimization problems separately

$$\begin{aligned} &\begin{cases} \min_{\tilde{\mathbf{H}}} L_{av} = \min_{\tilde{\mathbf{H}}} \frac{1}{2} \|\tilde{\mathbf{H}}\tilde{\mathbf{F}}\|_F^2 \\ \text{s.t.} \quad \tilde{\mathbf{H}}\tilde{\mathbf{G}}_y = \mathbf{V}, \quad \tilde{\mathbf{H}} \text{ on form of } \tilde{\mathbf{H}} = \text{lbt}[\mathbf{H}(0), \dots, \mathbf{H}(N-1)] \end{cases} \\ \Leftrightarrow &\begin{cases} \min_{\{\mathbf{H}(0), \dots, \mathbf{H}(N-1)\}} \sum_{i=1}^{N-1} \|\mathbf{H}(i)\tilde{\mathbf{F}}_i\|_F^2 \\ \text{s.t.} \quad \mathbf{H}(i)\tilde{\mathbf{G}}_{yi} = \mathbf{V}_i, \quad \forall i = 0, \dots, N-1 \end{cases} \\ \Leftrightarrow &\begin{cases} \min_{\mathbf{H}(i)} \frac{1}{2} \|\mathbf{H}(i)\tilde{\mathbf{F}}_i\|_F^2, \quad \forall i = 0, \dots, N-1 \\ \text{s.t.} \quad \mathbf{H}(i)\tilde{\mathbf{G}}_{yi} = \mathbf{V}_i \end{cases} \end{aligned} \quad (39)$$

All subproblems in (39) are convex constrained QP problems, which can be solved based on the KKT-conditions of augmented Lagrange functions. Actually, by making an analogy between the subproblem in (39) and the static SOC problem (13), an analytic solution can be derived as follows

$$\mathbf{H}(i)^T = (\tilde{\mathbf{F}}_i\tilde{\mathbf{F}}_i^T)^{-1}\mathbf{G}_{yi}\left(\mathbf{G}_{yi}^T(\tilde{\mathbf{F}}_i\tilde{\mathbf{F}}_i^T)^{-1}\mathbf{G}_{yi}\right)^{-1}\mathbf{V}_i^T \quad (40)$$

where the proof is similar to the one in ref (Alstad et al., 2009) hence not further provided. □

**Theorem 4** implies that for the dynamic SOC problem, self-optimizing CVs associated with different time instants can be obtained separately, which is made possible by enforcing the introduced equality constraint such that the effects of non-optimal operations along the time direction are not cross-interacting. Such decoupling is similar to the static case, however, no time issue is involved in the latter case. Another point is that in the static case, the right part of  $\mathbf{H}^T$  can be simply dropped (see eqs (15) and (14)) because the dropped part  $(\mathbf{G}_y^T(\tilde{\mathbf{F}}\tilde{\mathbf{F}}^T)^{-1}\mathbf{G}_y)^{-1/2}\mathbf{J}_{uu}^{1/2}$  is a nonsingular square matrix. However, here the term  $(\mathbf{G}_{y_i}^T(\tilde{\mathbf{F}}_i\tilde{\mathbf{F}}_i^T)^{-1}\mathbf{G}_{y_i})^{-1}\mathbf{V}_i^T$  cannot be dropped because it is of dimension  $i \times n_u$ . The nonsingular transformation property (left multiplying  $\mathbf{Q}$ ) can nonetheless be utilized for the overall matrix  $\tilde{\mathbf{H}}$ .

To better explain how to obtain the CVs with the proposed method, a numerical example is provided.

*Illustrative example:* A  $2 \times 2$  linear time-invariant system is described as

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) + \mathbf{B}_d\tilde{\mathbf{d}} \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}u(k) + \mathbf{n}(k) \end{aligned} \tag{41}$$

where

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0.6 & 0.4 \\ 0.3 & 0.5 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.3 \\ 0.07 \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} -0.7 \\ 0.2 \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \end{aligned}$$

with a time horizon of  $N = 3$  and a nominal initial condition  $\mathbf{x}(0) = \mathbf{0}$ . In this case, the three measurements in  $\mathbf{y}$  include all the states and inputs, but in general there is no requirement of full state information (see the case study section).

The cost function is

$$J = \frac{1}{2}\mathbf{x}(3)^T\mathbf{S}\mathbf{x}(3) + \frac{1}{2}\sum_{i=0}^2 [\mathbf{x}(i)^T\mathbf{Q}\mathbf{x}(i) + ru(i)^2] \tag{42}$$

where the weight matrices are set as

$$\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}, \quad r = 0.1$$

The disturbances are  $\mathbf{d} = [\mathbf{x}_1(0) \ \mathbf{x}_2(0) \ \tilde{\mathbf{d}}]^T$  with magnitudes as 0.3, 0.5, 1, respectively. Here, we assume that  $\tilde{\mathbf{d}}$  is time-invariant for simplicity. The magnitudes of measurement errors are taken as 0.02 for the two states and 0.01 for the input  $u$ , respectively.

Around the nominal trajectory, the Hessian and  $\mathbf{V}$  were obtained from a static NLP solver,

$$\begin{aligned} \tilde{\mathbf{J}}_{uu} &= \begin{bmatrix} 0.223050 & 0.087546 & 0.061183 \\ 0.087546 & 0.206339 & 0.071150 \\ 0.061183 & 0.071150 & 0.194900 \end{bmatrix}, \\ \mathbf{V} &= \begin{bmatrix} 0.424578 & 0 & 0 \\ 0.153547 & 0.424694 & 0 \\ 0.138588 & 0.161164 & 0.441475 \end{bmatrix} \end{aligned} \tag{43}$$

The gain matrix  $\tilde{\mathbf{G}}_y$  and sensitivity matrix  $\tilde{\mathbf{F}}$  become

$$\tilde{\mathbf{G}}_y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0.3 & 0 & 0 \\ 0.07 & 0 & 0 \\ 0 & 1 & 0 \\ 0.208 & 0.3 & 0 \\ 0.125 & 0.07 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \tilde{\mathbf{F}} = \begin{bmatrix} 0.3 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.3 & -0.437 & 1.152 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.09 & 0.069 & -0.354 & 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 & 0 \\ 0.069 & 0.219 & 0.281 & 0 & 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 \\ -0.15 & -0.261 & 1.171 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0.036 & 0.051 & -0.449 & 0 & 0 & 0 & 0 & 0 & 0 & 0.02 & 0 & 0 \\ 0.051 & 0.112 & 0.316 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.02 & 0 \\ -0.078 & -0.142 & 1.217 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix} \tag{44}$$

Based on **Theorem 4**, the combination matrix associated with  $t = 0$  is

$$\mathbf{H}(0)^T = (\tilde{\mathbf{F}}_0\tilde{\mathbf{F}}_0^T)^{-1}\mathbf{G}_{y0}(\mathbf{G}_{y0}^T(\tilde{\mathbf{F}}_0\tilde{\mathbf{F}}_0^T)^{-1}\mathbf{G}_{y0})^{-1}\mathbf{V}_0^T$$

where  $\mathbf{G}_{y0} = [0 \ 0 \ 1]^T$ ,

$$\tilde{\mathbf{F}}_0 = \begin{bmatrix} 0.3 & 0 & 0 & 0.02 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0.02 & 0 \\ -0.3 & -0.437 & 1.152 & 0 & 0 & 0.01 \end{bmatrix}, \quad \mathbf{V}_0 = 0.4246, \text{ and we obtain}$$

$$\mathbf{H}(0) = [0.4231 \ 0.3706 \ 0.4246]$$

In the same way,  $\mathbf{H}(1)$  and  $\mathbf{H}(2)$  are readily obtained by using the other partitioned matrices in (44),

$$\mathbf{H}(1) = \begin{bmatrix} -0.1 & 0.115 & -0.098 & 0.895 & -0.236 & 0.4247 \end{bmatrix}$$

$$\mathbf{H}(2) = \begin{bmatrix} -0.094 & 0.027 & -0.071 & 0.362 & -0.134 & -0.006 & 0.576 & -0.076 & 0.4414 \end{bmatrix}$$

The example continues below.

### 3.6. On-line implementation

Note that an input adaptation law (controller) can be derived based on the obtained CV functions, provided  $\mathbf{u}$  is included in the measurements. Suppose that the CVs associated with time instant  $t = k$  are

$$\mathbf{c}(k) = \mathbf{H}(k)\bar{\mathbf{y}}(k) \tag{45}$$

with setpoints  $\mathbf{c}_s(k)$ . Denote a measurement subset as  $\bar{\mathbf{y}}'(k)$  by excluding  $\mathbf{u}(k)$ , then a simple rearrangement of the CVs gives

$$\mathbf{c}_s(k) = \underbrace{\begin{bmatrix} \mathbf{H}^v(k) & \mathbf{H}^u(k) \end{bmatrix}}_{\mathbf{H}(k)} \underbrace{\begin{bmatrix} \bar{\mathbf{y}}'(k) \\ \mathbf{u}(k) \end{bmatrix}}_{\bar{\mathbf{y}}(k)}$$

$$\Rightarrow \mathbf{u}(k) = \mathbf{H}^u(k)^{-1} [\mathbf{c}_s(k) - \mathbf{H}^v(k)\bar{\mathbf{y}}'(k)] \tag{46}$$

where  $\mathbf{H}^v(k)$  and  $\mathbf{H}^u(k)$  are submatrices of  $\mathbf{H}(k)$  associated with  $\bar{\mathbf{y}}'(k)$  and current inputs  $\mathbf{u}(k)$ , respectively. Above relationship requires an invertible  $\mathbf{H}^u(k)$ , which is ensured by the following corollary.

**Corollary 1.**  $\mathbf{H}^u(k)$  equals to the  $k$ th diagonal submatrix in  $\mathbf{V}$ .

**Proof.** When measurements include the inputs, it is easy to show that  $\mathbf{D}(k) = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}$  in the linear dynamic model (29). Since the diagonal block of the gain matrix  $\bar{\mathbf{G}}_y$  defined in (30) is  $\mathbf{D}(k)$ , the enforced relationship  $\mathbf{H}(k)\bar{\mathbf{G}}_{yk} = \mathbf{V}_k$  is expanded as

$$\mathbf{H}(k)\bar{\mathbf{G}}_{yk} = \mathbf{V}_k \Rightarrow \begin{bmatrix} \mathbf{H}^v(k) & \mathbf{H}^u(k) \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \times & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \times & \mathbf{V}_{kk} \end{bmatrix} \Rightarrow \mathbf{H}^u(k) = \mathbf{V}_{kk} \tag{47}$$

where “ $\times$ ” represents nonzero elements, and  $\mathbf{V}_{kk}$  is the  $k$ th diagonal submatrix in  $\mathbf{V}$ , which is invertible due to the positive definite  $\bar{\mathbf{J}}_{uu}$ .  $\square$

Note that  $\mathbf{H}^v$ ,  $\mathbf{H}^u$  and  $\mathbf{c}_s$  for all  $k$  can be calculated off-line, so one just needs to measure  $\bar{\mathbf{y}}'(k)$  in the on-line phase and use (46) to update the control inputs, which is very easy to implement. The perfect control of CVs in one sampling period without any overshoot is similar to the *deadbeat response* in the digital control field (Westphal, 1995), however, a main difference in our problem is that the setpoints of CVs are time-varying and old measurements are included in the control law.

*Illustrative example (continued):* Since the nominally optimal trajectory of this example is the origin, all setpoints are exactly 0. On basis of the CVs obtained previously, the input control law becomes

$$u(0) = -\frac{1}{0.4246} (0.4231x_1(0) + 0.3706x_2(0))$$

$$= \begin{bmatrix} -0.9965 & -0.8730 \end{bmatrix} \mathbf{x}(0)$$

$$u(1) = \begin{bmatrix} 0.2342 & -0.2714 & 0.2319 & -2.1078 & 0.5549 \end{bmatrix} \begin{bmatrix} \mathbf{y}(0) \\ \mathbf{x}(1) \end{bmatrix}$$

$$u(2) = \begin{bmatrix} 0.212 & -0.061 & 0.161 & -0.820 & 0.303 & 0.014 & -1.304 & 0.173 \end{bmatrix} \begin{bmatrix} \mathbf{y}(0) \\ \mathbf{y}(1) \\ \mathbf{x}(2) \end{bmatrix} \tag{48}$$

where  $\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$ ,  $\mathbf{y}(0) = \begin{bmatrix} \mathbf{x}(0) \\ u(0) \end{bmatrix}$  and  $\mathbf{y}(1) = \begin{bmatrix} \mathbf{x}(1) \\ u(1) \end{bmatrix}$ .

### 3.7. Discussion of proposed method

**3.7.1. Null space method (Oliveira et al., 2016):** A special case with no measurement noise and sufficient number of measurements ( $n_y \geq n_d + n_u$ )

For this special case, zero loss can be obtained with  $\bar{\mathbf{H}}$  on the form  $\bar{\mathbf{H}} = \text{diag}(\mathbf{H}(1), \dots, \mathbf{H}(N))$  given in (25) (Structure 3). To see this, note that without noise we have  $\bar{\mathbf{F}} = \bar{\mathbf{F}}\bar{\mathbf{W}}_d$ , hence

$$L = \frac{1}{2} \|\mathbf{z}\|_2^2, \quad \mathbf{z} = \mathbf{V}(\bar{\mathbf{H}}\bar{\mathbf{G}}_y)^{-1} \bar{\mathbf{H}}\bar{\mathbf{F}}\bar{\mathbf{W}}_d \tag{49}$$

where the middle term is

$$\begin{aligned} \bar{\mathbf{H}}\bar{\mathbf{F}} &= \begin{bmatrix} \mathbf{H}(0) & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{H}(1) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{H}(N-1) \end{bmatrix} \begin{bmatrix} \mathbf{F}(0) \\ \mathbf{F}(1) \\ \vdots \\ \mathbf{F}(N-1) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{H}(0)\mathbf{F}(0) \\ \mathbf{H}(1)\mathbf{F}(1) \\ \vdots \\ \mathbf{H}(N-1)\mathbf{F}(N-1) \end{bmatrix} \end{aligned} \quad (50)$$

Here  $\mathbf{F}(k)$  is the optimal sensitivity matrix of  $\mathbf{y}^{\text{opt}}(k)$  partitioned from  $\bar{\mathbf{F}}$ . Since  $\mathbf{F}(k)$  is of dimension  $n_y \times n_d$  and  $n_y > n_d + n_u$ , it is possible to select  $\mathbf{H}(k)$  in the null space of  $\mathbf{F}(k)$ , such that  $\mathbf{H}(k)\mathbf{F}(k) = 0$  holds. This leads to the big matrix  $\bar{\mathbf{H}}\bar{\mathbf{F}} = 0$ , hence the loss  $L = \frac{1}{2} \|\mathbf{V}(\bar{\mathbf{H}}\bar{\mathbf{C}}_y)^{-1} \bar{\mathbf{H}}\bar{\mathbf{F}}\bar{\mathbf{W}}\bar{\mathbf{d}}\|^2 = 0$  for any disturbance. The feedback solution obtained from the nullspace method is the same as the optimal state feedback obtained from the classical optimal control theory (Manum et al., 2008).

*Illustrative example (continued):* Since the null space method can only be applied when  $n_y \geq n_d + n_u$ , we now restrict  $\mathbf{d} = \mathbf{x}(0)^T$  and let  $\bar{\mathbf{d}}$  be fixed. In this case, the optimal sensitivity matrices are calculated as follows

$$\mathbf{F}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1.001 & -0.874 \end{bmatrix}, \quad \mathbf{F}(1) = \begin{bmatrix} 0.300 & 0.138 \\ 0.230 & 0.439 \\ -0.501 & -0.521 \end{bmatrix},$$

$$\mathbf{F}(2) = \begin{bmatrix} 0.122 & 0.102 \\ 0.170 & 0.224 \\ -0.260 & -0.283 \end{bmatrix}$$

Then  $\mathbf{H}(k)$  in (50) is selected as the null space of  $\mathbf{F}(k)$

$$\begin{aligned} \mathbf{H}(0) &= [0.6018 \quad 0.5257 \quad 0.6012], \\ \mathbf{H}(1) &= [0.6020 \quad 0.5255 \quad 0.6012], \\ \mathbf{H}(2) &= [0.6281 \quad 0.4843 \quad 0.6090] \end{aligned}$$

which gives the following input control law

$$\begin{aligned} u(0) &= [-1.0010 \quad -0.8744] \mathbf{x}(0), \\ u(1) &= [-1.0012 \quad -0.8739] \mathbf{x}(1), \\ u(2) &= [-1.0313 \quad -0.7953] \mathbf{x}(2) \end{aligned}$$

which is a time-varying feedback law  $u(k) = \mathbf{K}(k)\mathbf{x}(k)$ . Similar to the classic optimal control theory, the present state measurements  $\mathbf{x}(k)$  contain all information because there is no noise.

### 3.7.2. Different structures of $\bar{\mathbf{H}}$ and the time horizon

The proposed dynamic SOC solution gives a time-varying lower-block triangular structured  $\mathbf{H}$  (Structure 2), which gives time-varying CVs and also setpoints along the trajectory. This may seem complex, but the most appealing merit of SOC is still preserved. That is, we get an on-line invariant control strategy to cope with disturbances, where the CVs and their setpoints are not necessarily re-optimized for changing disturbances, which is done in many other real-time optimization approaches. A limitation of the proposed method is, nonetheless, that when the time horizon  $N$  is large, the combination matrix  $\bar{\mathbf{H}}$  is also very large. However, for a finite batch system, it is generally possible to approximate the real process with a small value of  $N$ . An alternative is to consider using Structure 3 or 4 to get a suboptimal, but simpler solution. In addition, other structure may be considered. For example, an interesting topic would be to select CVs using only a few recent measurements, namely,  $\mathbf{c}(k)$  is a linear combination of  $\mathbf{y}(k-m), \dots, \mathbf{y}(k)$ . This way, one can further pursue the trade-off between the economic performance and CV complexity, as it gives rise to a new

sparse structure of  $\bar{\mathbf{H}}$ , lying between structures 2 and 3/4. Using the most recent measurements was previously considered in associated regression approach (Grema and Cao, 2016). However, we have been unable to formulate such considerations into convex optimization problems, from which analytic solutions can be obtained. This is because there are not enough degrees of freedom using the nonsingular transformation of  $\bar{\mathbf{H}}$  to eliminate the nonlinear term in the loss function. Actually, the constant CV case (Structure 4) reduces to the output feedback problem for linear dynamic systems, which has been proven to be NP-hard (Blondel and Tsitsiklis, 1997).

Another limitation related to a large  $N$  is the off-line computational cost for obtaining  $\bar{\mathbf{H}}$ . We note that the algorithm contains several steps: (1) numerical optimization at the nominal condition; (2) obtain sensitivity matrices along the optimal trajectory; (3) calculate  $\bar{\mathbf{H}}$  based on Theorem 4. The computational cost for Step 1 depends on the used dynamic programming solver. The sensitivity matrices in Step 2 can be available along with the dynamic programming solver, except that a Cholesky decomposition for the rotated  $\mathbf{J}_{uu}$  is additionally needed ( $n^3/3$  flops for the cost). The full  $\bar{\mathbf{H}}$  is obtained using Theorem 4, which is a closed-form solution. A computational experiment shows that the algorithm finishes within 100 s when  $N = 100$ ,  $n_y = 20$  (in a notebook with Intel i5 CPU), which most likely fulfills performance requirements for batch processes.

### 3.7.3. Comparisons with the optimal control method

The presented SOC approach implements a simple output feedback control policy, where the uncertain initial states, parametric disturbances and measurement noise, are systematically handled in an integrated manner. This is quite different from the classic optimal control theory (Bryson and Ho, 1975) and related methods, such as the neighbouring-extremal control (NEC) (Gros et al., 2009), which are based on full state feedback. In the case of partially measured states and/or with measurement errors, the methods require online state observers such as the Kalman filter and its variants. The situation becomes even more difficult when there are unknown disturbances or parameters in the process model (Gros et al., 2009). In this case, one needs to extend the estimators to handle the disturbances (Lee and Bryson, 1989), which is not a trivial task. In contrast, the SOC solution does not involve separated online estimations, hence it is easier to implement. An additional advantage of SOC is that it provides a quantified index (the loss) indicating the influence of various uncertainties.

## 4. Case studies

### 4.1. Fed-batch reactor

#### 4.1.1. Process description

The fed-batch reactor involves two reactions (Gros et al., 2009; Oliveira et al., 2016),  $A + B \rightarrow C$  and  $2B \rightarrow D$ , where  $A$  and  $B$  are the reactants,  $C$  is the desired product, and  $D$  is the byproduct produced by the side reaction. The first-principle nonlinear dynamic model of this reactor is

$$\frac{dc_A}{dt} = -k_1 c_A c_B - c_A u / V, \quad c_A(0) = c_{A0} \quad (51)$$

$$\frac{dc_B}{dt} = -k_1 c_A c_B - 2k_2 c_B^2 - (c_B - c_B^m) u / V, \quad c_B(0) = c_{B0} \quad (52)$$

$$\frac{dV}{dt} = u, \quad V(0) = V_0 \quad (53)$$

**Table 1**  
Parameter values and variation magnitudes.

Variable	Description	Nominal Value	Variation Magnitude	Unit
$c_{A0}$	initial concentration (A)	0.72	$\pm 20\%$	mol/l
$c_{B0}$	initial concentration (B)	0.0614	$\pm 20\%$	mol/l
$V_0$	initial volume	1	$\pm 20\%$	l
$k_1$	kinetic coefficient (main)	0.053	$\pm 40\%$ (Case 2)	l/(mol · min)
$k_2$	kinetic coefficient (side)	0.128	$\pm 40\%$ (Case 2)	l/(mol · min)
$c_B^{\text{in}}$	inlet concentration of B	5	fixed	mol/l
$t_f$	batch duration	250	fixed	min

$$c_C = (c_{A0}V_0 - c_A V)/V \quad (54)$$

$$c_D = ((c_A + c_{B\text{in}} - c_B)V - (c_{A0} + c_{B\text{in}} - c_{B0}))/2V \quad (55)$$

where  $c_X$  and  $c_{X0}$  represents the concentration of component  $X$  and its initial value, respectively;  $V$  is the reactor volume with initial value  $V_0$ ;  $k_1$  and  $k_2$  are the kinetic constants for the two reactions;  $u$  is the feed rate of  $B$  (manipulated variable) and  $c_B^{\text{in}}$  is the concentration of  $B$  in the feed.

The operational objective is to maximize the amount of product  $C$  whilst minimizing the byproduct  $D$  at the final batch time  $t_f$ , by manipulating the feed rate of reactant  $B$ ,  $u(t)$ , which is constrained within the bound  $0 \leq u(t) \leq 0.001$  [l/min]. The batch time is fixed at  $t_f = 250$  min. Therefore, a dynamic optimization problem is formulated as follows

$$\begin{aligned} \min_{u(t)} J &= [c_D(t_f) - c_C(t_f)]V(t_f) \\ \text{s.t.} \quad &\text{process model (51) – (55)} \\ &0 \leq u(t) \leq 0.001 \text{ l/min} \end{aligned} \quad (56)$$

The process parameters, together with their nominal values and expected variations, are given in Table 1. On basis of the given nominal condition, a numerical optimization was first performed using a dynamic optimization solver. The open-source tool CasADi (Andersson, 2013) (Version 3.1.0) with IPOPT (Biegler and Zavala, 2009) as the NLP solver is used. The numerical optimization was performed by dividing the time horizon  $[0, t_f]$  equally into  $N = 20$  grids with piecewise constant inputs (control vector parameterization). Numerical experiments showed that the choice  $N = 20$  is sufficient enough to get a good optimizing accuracy, because the performance improvement in  $J$  is less than  $10^{-8}$  mol when  $N$  is further increased. On the other hand, it is noted that the analytical optimal input arc for this batch reactor is singular unconstrained (Gros et al., 2009; Srinivasan et al., 2003), thus  $\bar{\mathbf{J}}_{uu}$  is asymptotically singular in terms of  $N$ . In this case, the strong second-order sufficiency conditions for the static NLP formulation resulting from the control parameterization have to be checked first. To this end, the condition number of the Hessian,  $\bar{\mathbf{J}}_{uu}$ , is calculated as 60.4, which is reasonably small hence the condition is considered satisfied. In contrast, the condition number is large (1494.7) when  $N = 100$ . The differences between the analytical and numerical approaches were also discussed in, e.g. ref (Podmajerský et al., 2013). Fig. 2 shows the optimal input trajectory  $u(t)$  and system states for  $N = 20$ . We see that  $u(t)$  is unconstrained over the entire time horizon. The reason is that increasing the feed of reactant  $B$ ,  $u(t)$ , is on one hand helpful to produce product  $C$ , but on the other hand, it gives more undesired byproduct  $D$ .

#### 4.1.2. Disturbances and measurements

To explore the proposed dynamic SOC scheme, two cases with different uncertainties are considered:

**Case 1.** The initial values of the 3 states are uncertain, i.e.  $\mathbf{d} = [c_{A0} \ c_{B0} \ V_0]^T$ . In this case, we compare the proposed method and the null space method. Note that measurement noise is included, which is not explicitly accounted for by the null space method.

**Case 2.** In addition to uncertain initial states, the reaction kinetic parameters  $k_1$  and  $k_2$ , are included as uncertainties. Therefore, the disturbance variable is  $\mathbf{d} = [c_{A0} \ c_{B0} \ V_0 \ k_1 \ k_2]^T$ . In this case, we compare three different structures of the combination matrix  $\mathbf{H}$  (Structures 2, 3, 4).

The variations for the initial states are  $\pm 20\%$  of their nominal values and  $\pm 40\%$  for  $k_1$  and  $k_2$ , see Table 1. The process measurements considered are the 3 system states and the input variable  $u(t)$ , i.e.

$$\mathbf{y} = [c_A \ c_B \ V \ u]^T$$

Note that we assume  $c_C$  and  $c_D$  are only measured at end of the batch to compute the cost, hence they cannot be adopted for within-batch usage. All measurements (except for noise-free case in Case 1) are assumed to have zero-mean gaussian noise, with standard deviations of 0.03 mol/l for concentrations and 0.1 l for volume, respectively. To be more realistic, a small noise (0.025 ml/min) for  $u(t)$  is also included to represent possible implementation error in the manipulated variable.

#### 4.1.3. Results and simulations

To apply the proposed method, the continuous reactor system in  $[0, t_f]$  is discretized into  $N = 20$  finite horizons. A time-varying discrete linear model is obtained numerically by linearizing the process around the nominally optimal trajectory. On the basis of this linearized model, the sensitivity matrices ( $\bar{\mathbf{G}}_y$ ,  $\bar{\mathbf{G}}_{y_d}$ , and so on) are calculated as given in (30)–(32). The Hessian  $\bar{\mathbf{J}}_{uu}$  is directly obtained from the dynamic programming solver used for the nominal point optimization.

**Case 1.** Uncertain initial states.

**Null space method.** The null space method can be applied by ignoring the measurement noise. The combination matrix  $\bar{\mathbf{H}}$  is on the diagonal form in eq (25) (Structure 3), with the diagonal submatrices selected as the null space of  $\mathbf{F}(k)$  at each time instant  $k$ . The following result is obtained

$$\bar{\mathbf{H}} = \text{diag}(\mathbf{H}(0), \dots, \mathbf{H}(N-1)) \quad (57)$$

where

$$\begin{bmatrix} \mathbf{H}(0) \\ \mathbf{H}(1) \\ \vdots \\ \mathbf{H}(19) \end{bmatrix} = \begin{bmatrix} -0.0020 & 0.0165 & -6.59 \times 10^{-4} & 1 \\ -0.0021 & 0.0167 & -6.32 \times 10^{-4} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -0.0071 & 0.0238 & -3.68 \times 10^{-4} & 0.997 \end{bmatrix}$$

A plot of the time-varying coefficients associated with  $c_A$ ,  $c_B$  and  $V$  (normalized with their mean nominal values along the trajectories) is shown in Fig. 3(a). One notes that the normalized coefficient associated with  $c_A$  changes with time, whereas those associated with  $c_B$  and  $V$  are relatively constant.

**Exact local method.** Next, consider the exact local method which also takes into account measurement noise. A LBT-structured combination matrix  $\bar{\mathbf{H}}$  is obtained using

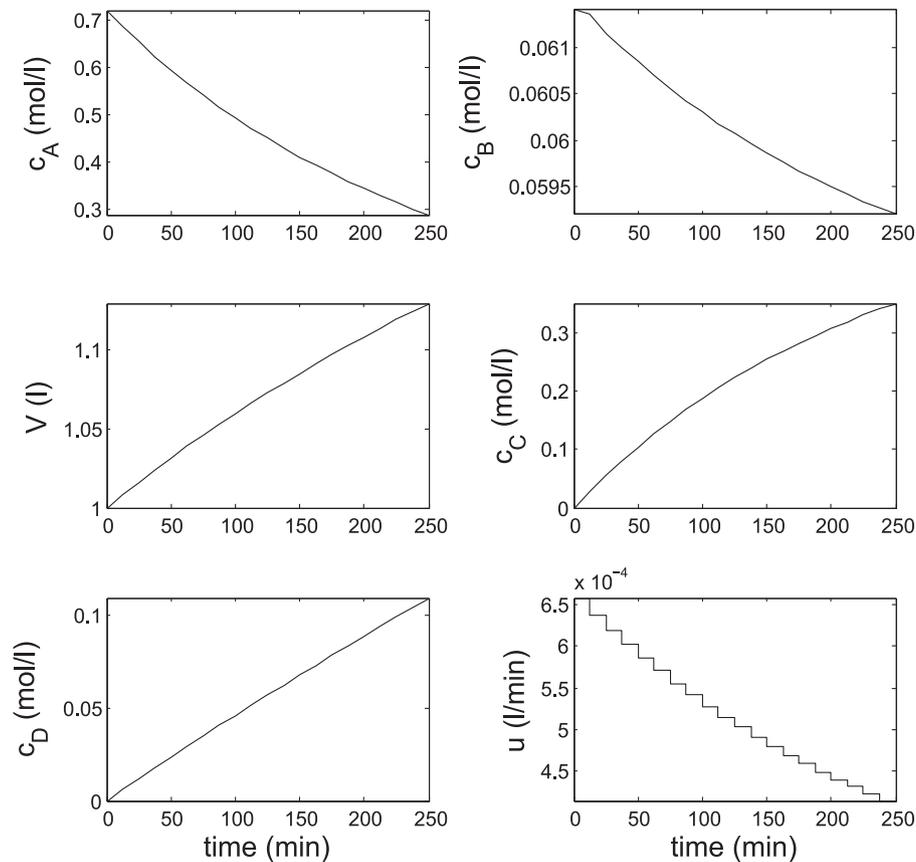


Fig. 2. Nominal optimal trajectories for fed-batch reactor.

**Theorem 4** (Structure 2). The results are as follows

$$\bar{\mathbf{H}} = \text{lb}(\mathbf{H}(0), \dots, \mathbf{H}(N-1)) \quad (58)$$

where

$$\mathbf{H}(0) = [-0.117 \quad 0.146 \quad -0.033 \quad 61.81]$$

$$\mathbf{H}(1) = [-0.065 \quad 0.063 \quad -0.030 \quad 41.87 \quad -0.086 \quad 0.039 \quad -0.03 \quad 62.1]$$

⋮

To better visualize the coefficients in  $\bar{\mathbf{H}}$ , we plot in Fig. 3 (b) the first 3 columns of  $\bar{\mathbf{H}}$ , which are associated with the initial measurements at  $k=0$ . The magnitudes of these coefficients show that the contributions of older measurements decay with time, which is reasonable from a physical point of view. However, older measurements still contain useful information because they reduce the effect of measurement noise. In the more general case with fewer measurements than disturbances (see Case 2), older measurements additionally help to better reconstruct the disturbances that occurred (see Case 2). Another interesting implication from Fig. 3 is the possibility of using approximations, for example, low order polynomials, to reduce the complexity of  $\bar{\mathbf{H}}$ .

**Comparison in terms of optimality.** The local average loss for the disturbances and noise is 0.00030 [mol] with the exact local method in (58). In contrast, if we substitute the combination matrix obtained from null space method in (57) into the loss function used when deriving the exact local method, the loss is 0.0128 [mol], which is about 4.8 % of the nominal cost and two orders of magnitude larger than the exact local method.

To validate the above results, the original nonlinear reactor model was simulated for 4 typical disturbance scenarios, where

all 3 disturbances in initial states are simultaneously perturbed by  $\pm 50\%$  and  $\pm 100\%$  of their maximal allowable uncertain magnitudes (Fig. 4). To include the effect of noisy measurements, 100 realizations were repeated to calculate the average loss. The re-

sults are summarized in Table 2. The numerical results support the following facts: (1) The null space method is indeed excellent in the noise-free case, since all losses are very small ( $2.4 \times 10^{-5}$  at the maximum) for the 4 investigated disturbance scenarios. This illustrates that nonlinearity is not a serious problem for this case. (2) In the more realistic case with measurement noise, the exact local method performs much better than the null space method. In the nonlinear model evaluations, the exact local method gives very small average losses (0.0002–0.0008), whereas the null space method gives much larger losses of about 0.0060 (more than 2 %

**Table 2**  
Average loss [mol] evaluated from nonlinear dynamic model (Case 1).

disturbance magnitudes	noise-free case	noisy case	
	null space method	null space method	exact local method
-100 %	$2.2 \times 10^{-5}$	0.0069	0.00083
-50 %	$1.4 \times 10^{-6}$	0.0060	0.00030
+50 %	$1.5 \times 10^{-6}$	0.0056	0.00023
+100 %	$2.4 \times 10^{-5}$	0.0059	0.00038

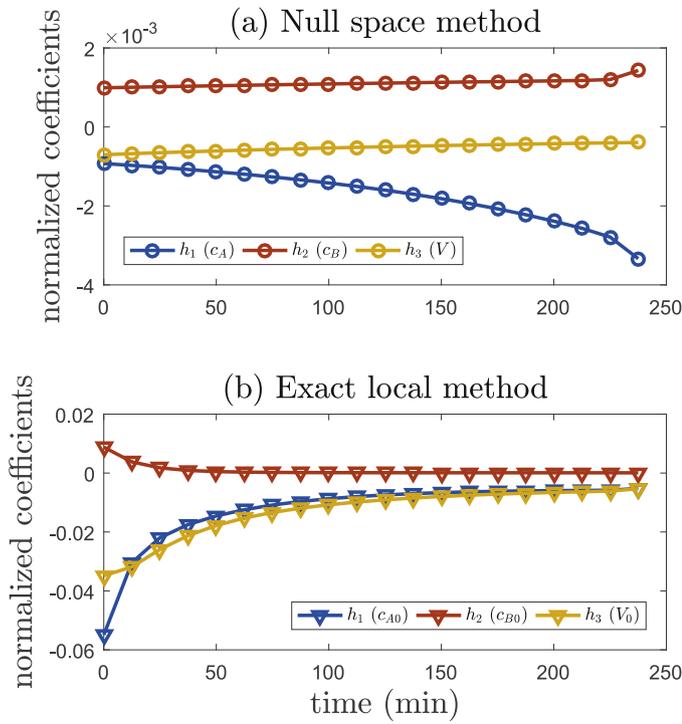


Fig. 3. Coefficients in combination matrix: (a) null space method (b) exact local method (the first 5 columns of  $\mathbf{H} = lbt(\mathbf{H}(0), \dots, \mathbf{H}(N-1))$ ).

of the nominal cost), which is more than 20 times larger than the exact local method.

Fig. 4 shows the input response  $u(t)$  and economic loss under different situations. Again, we note that null space method is very good in the noise-free case, where the responded inputs

(green dash-dot line) perfectly follow the optimal arcs (red line), which leads to negligible losses in all cases. However, the null space method is very sensitive to measurement noise (blue dotted line). We see that the input has severe fluctuations, which almost seem random at first glance! However, on average, the approach still captures the basic trend of the optimal feedrates. To confirm this, one sees that as the optimal input arc moves upward from Fig. 4 (a) to (d), the input arc hits the low bound more frequently in (a) and hits the upper bound more frequently in (d). Anyway, the use of null space method is restrictive in a noisy environment, and can lead to large economic losses. On the other hand, the exact local method effectively rejects the measurement noise, and the input arcs (black solid line) follow the desired optimal operations closely for all disturbance cases and the losses are very small. Note that losses are unavoidable in this case because of measurement noise.

**Case 2.** With uncertain  $k_1$  and  $k_2$ .

In this case, the disturbance variable is  $\mathbf{d} = [c_{A0} \ c_{B0} \ V_0 \ k_1 \ k_2]^T$ . The inclusion of more disturbances makes the null space inapplicable because the number of disturbances is greater than the number of measurements. Thus, for this case, in addition to the time-varying LBT structure from Theorem 4 (Structure 2), we consider the time-varying diagonal structure (Structure 3) and the time-invariant diagonal structure (Structure 4). Since no analytical solutions for Structure 3 and 4 are available, numerical optimizations were performed. During this process, we found that the solutions are sensitive to initial guesses of the combination matrix, due to the nonconvex formulations. Hence, we performed repeated trial and error optimizations with different starting points. In all cases, we also explicitly include  $u$  such that a feedback law can be derived based on obtained CVs.

We considered all possible measurement subsets when the number of measurements,  $n_y$ , varies from 2 to 4. For all cases, we obtained optimal combination matrices and further the input con-

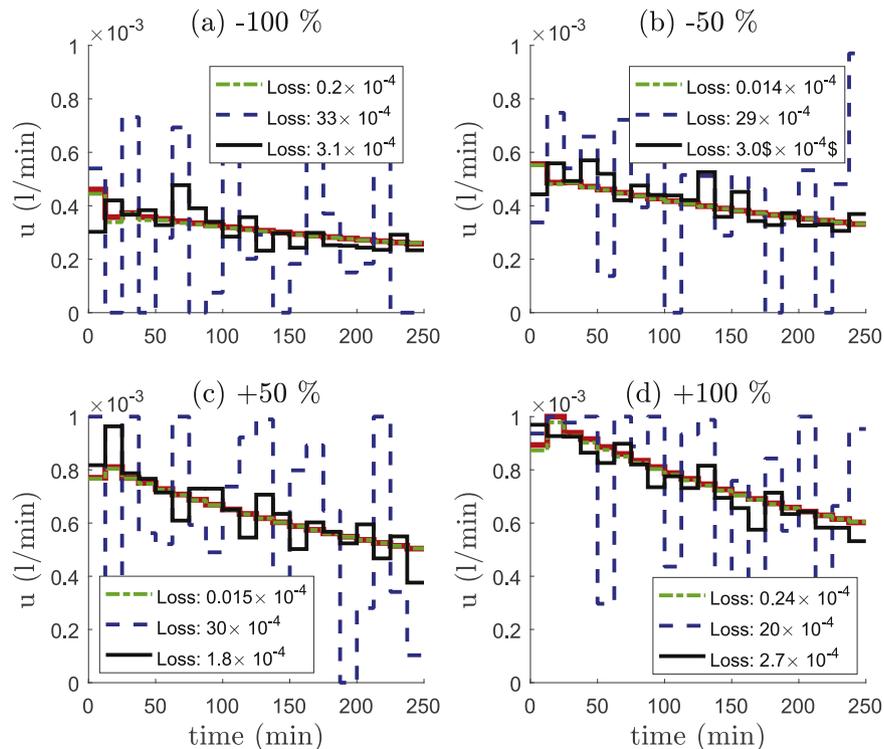
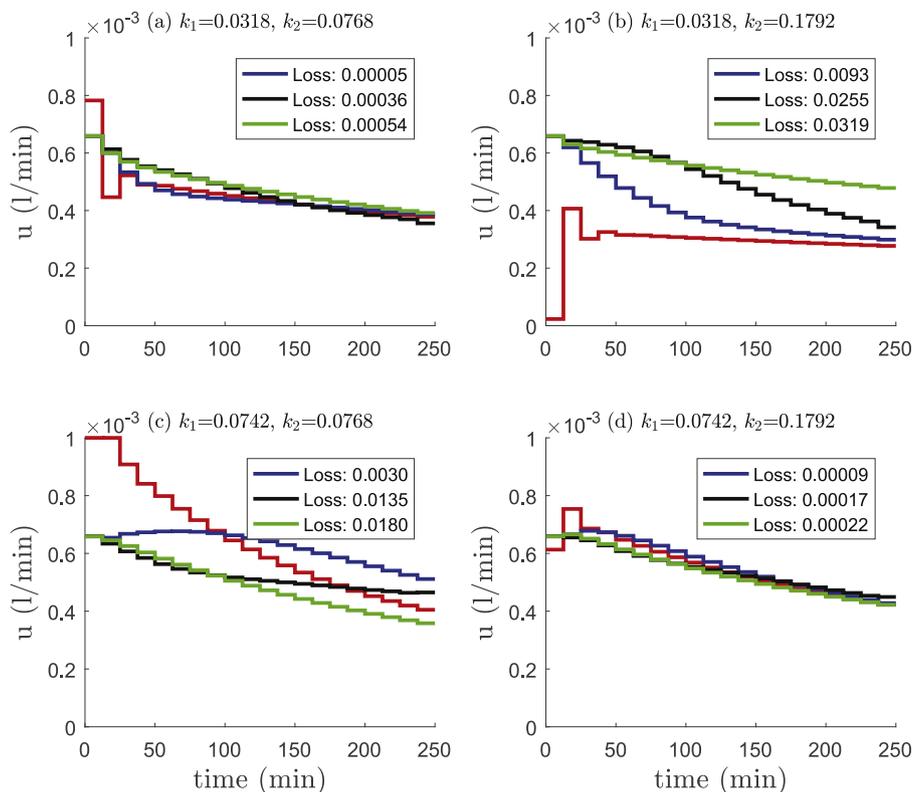


Fig. 4. Input responses and losses under different magnitude disturbances. (green: null space method applied to noise-free case; blue: null space method applied to noisy case; red: optimal input arc). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 5.** Input responses and losses under different kinetic scenarios. (blue: Structure 2; black: Structure 3; green: Structure 4; red: optimal input arc). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Table 3**

Local losses of structure 2–4 for measurement subsets (Case 2).

	Measurements	Structure 2	Structure 3	Structure 4
$n_y = 2$	$c_A, u$	0.0152	0.0191	0.0196
	$c_B, u$	0.0161	0.0191	0.0193
	$V, u$	0.0193	0.0193	0.0194
$n_y = 3$	$c_A, c_B, u$	0.0105	0.0158	0.0167
	$c_A, V, u$	0.0049	0.0138	0.0158
	$c_B, V, u$	0.0148	0.0164	0.0167
$n_y = 4$	$c_A, c_B, V, u$	0.0047	0.0127	0.0144

trol laws, given in Table 3. For example, when  $n_y = 2$  and the measurements are  $[c_A, u]$ , the obtained control laws are as follows:

$$\begin{aligned} \text{Structure 2 : } u(0) &= -0.00071 + 0.0019c_A(0) \\ u(1) &= -0.00055 + 0.0027c_A(0) - 0.73u(0) \\ &\quad - 0.0004c_A(1) \dots \end{aligned} \quad (59)$$

$$\begin{aligned} \text{Structure 3 : } u(0) &= -0.0013 + 0.0027c_A(0), \\ u(1) &= -0.00019 + 0.0012c_A(1), \dots \end{aligned} \quad (60)$$

$$\begin{aligned} \text{Structure 4 : } u(0) &= -0.000026 + 0.00095c_A(0), \\ u(1) &= 0.000015 + 0.00095c_A(1), \dots \end{aligned} \quad (61)$$

The losses for the various measurement subsets are given in Table 3. We make some observations: (1) For all cases, the ranking in terms of the loss is Structure 2 < Structure 3 < Structure 4, which is reasonable due to more and more restricted combination matrix for Structure 3 and 4. For example, when all measurements are included ( $n_y = 4$ ), the losses for Structure 2, 3 and 4 are 0.0047, 0.0127 and 0.0144, respectively. (2) Controlling measurement combinations improves the economic performance. For example, with

Structure 2 and using only the measurements  $[c_A, u]$  ( $n_y = 2$ ), the local average loss (Structure 2) is 0.0152 mol, whilst it is reduced to 0.0047 mol by using all the measurements ( $n_y = 4$ ). This fact holds for Structures 3 and 4 as well.

In the following dynamic simulations, we consider the case with all measurements ( $n_y = 4$ ). Four kinetic scenarios for  $[k_1, k_2]$  are considered as  $[0.0318, 0.0768]$ ,  $[0.0318, 0.1792]$ ,  $[0.0742, 0.0768]$  and  $[0.0742, 0.1792]$ . These values are chosen such that they are either +100% or -100% magnitude of their uncertain ranges. Meanwhile, the initial system states are all taken as the nominal condition for brevity. For all kinetic scenarios, we show in Fig. 5 their closed-loop responses without measurement noise for Structures 2, 3, 4. Noise is excluded to make the comparisons clearer, otherwise their differences will be buried in the random noise. A general observation is that the dynamic self-optimizing control method (Structure 2, blue curve) works reasonably well. In most cases, the inputs (blue curve) follow the optimal input arcs (red curve) quite tightly, especially for the first and fourth kinetic scenarios. The worst case is the scenario (b) with  $[k_1, k_2] = [0.0318, 0.1792]$ . Here the deviations from the optimal input are relatively large, hence so are the economic losses. This may be caused by the fact that the kinetics are rather extreme (small main reaction rate and large side reaction rate) such that true optimal input arc is significantly different from the nominal case. Note that the change in the kinetic parameters cannot be well detected until enough measurements have been collected. For example, at  $t = 0$  we have only initial measurements of the states which contain no information about  $k_1$  and  $k_2$ . Therefore, the optimizing performance is likely to be poor at the early stage of the dynamic operation. This can indeed be observed from subfigures (b) and (c). As time evolves, the operation is driven towards the optimum as more measurements become available. Regarding Structures 2, 3 and 4, it is evident that the simplified Structure 4 (green line) gives the worst results for all kinetic scenarios, where its input arcs being most distant from the optimal

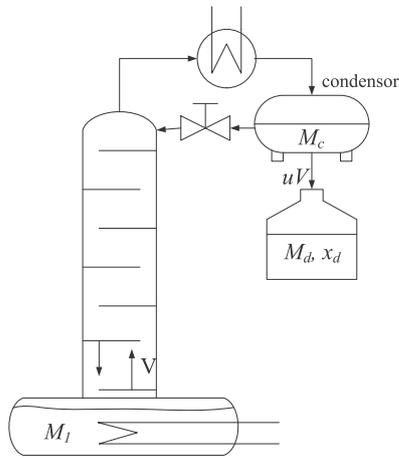


Fig. 6. Batch distillation column.

arcs. In all cases, the losses are Structure 2 < Structure 3 < Structure 4, as shown in boxes in Fig. 5.

## 4.2. Batch distillation column

### 4.2.1. Process description

In this subsection, we study optimal operation problem of the batch distillation column in Fig. 6 (Welz et al., 2002). This is a binary system, where the valuable product is the light component which is collected at the top of column. The process model is based on the following assumptions: (1) constant molar flows (same vapor flow on stages), (2) constant relative volatility, (3) equilibrium stages, (4) no vapor holdup, (5) constant liquid holdup on stages and in condenser and, (6) total condenser.

Based on the above assumptions and material balance relationships, the process model is:

$$\text{Reboiler: } \frac{dM_1}{dt} = -uV \quad (62)$$

$$\frac{dx_1}{dt} = \frac{V}{M_1} (x_1 - y_1 + (1 - u)x_2) \quad (63)$$

$$\text{Stages: } \frac{dx_i}{dt} = \frac{V}{M_i} (y_{i-1} - y_i + (1 - u)(x_{i+1} - x_i)) \quad (64)$$

$$\text{Condenser: } \frac{dx_c}{dt} = \frac{V}{M_c} (y_p - x_c) \quad (65)$$

with  $i = 2, \dots, p$ . The notation is as follows.  $p$ : number of stages;  $M_i$ : liquid holdup on stage  $i$  (counting from the bottom and stage 1 is the reboiler);  $x_i$  and  $y_i$ : molar fraction in liquid and vapor on stage  $i$ ;  $x_c$ : liquid molar fraction in condenser;  $V$ : vapor flow;  $u$ : distillate ratio  $u = D/V$ ,  $0 \leq u \leq 1$ . With a total condenser,  $x_c = y_p$  follows. The vapor-liquid equilibrium relationship holds with constant relative volatility  $\alpha$  gives on all stages:

$$y_i = \frac{\alpha x_i}{1 + (\alpha - 1)x_i} \quad (66)$$

The accumulated distillate,  $M_d$ , and its composition,  $x_d$ , are

$$M_d(t) = \int_0^t uV dt = M_1(0) - M_1(t) \quad (67)$$

$$x_d(t) = \frac{\sum_{i=1}^p x_i(0)M_i(0) - x_i(t)M_i(t)}{M_1(0) - M_1(t)} \quad (68)$$

Table 4  
Parameter values of batch distillation column.

Parameter	Value	Unit	Parameter	Value	Unit
$p$	10		$V$	$15 \pm 2$	kmol/h
$t_f$	10	h	$x_d^{des}$	0.9	
$\alpha$	$1.5 \pm 0.1$		$M_1(0)$	100	kmol
$M_i$	0.2	kmol	$x_i(0)$	0.5	
$M_c$	2	kmol	$x_c(0)$	0.5	

The operation objective is to maximize the amount of distillate valuable product  $M_d x_d$  for a fixed batch duration  $[0, t_f]$ , while satisfying a desired product quality. The following optimization problem is formulated:

$$\max_{u(t)} J = M_d(t_f) x_d(t_f) \quad (69)$$

$$\text{s.t. } x_d(t_f) \geq x_d^{des}$$

$$0 \leq u(t) \leq 1$$

$$\text{dynamic process model: (62) – (68)}$$

where  $x_d^{des}$  is the minimal allowable quality. The nominal model parameters are given in Table 4, where  $\alpha$  and  $V$  are the uncertain disturbances, whose variation ranges are  $1.4 \leq \alpha \leq 1.6$  and  $13 \leq V \leq 17$  kmol/h, respectively.

### 4.2.2. Nominal optimization and problem reformulation

The optimal trajectory of the distillate ratio  $u(t)$  consists of three arcs (Welz et al., 2002; 2008). The first arc is zero distillate ( $u(t) = 0$ , active input constraint) to initially separate the components. The second arc maintains  $u(t)$  at unconstrained values. The third arc is  $u(t) = 1$  to empty the volume of light component product before terminating the operation. However, the effect of the third arc is negligible and can be absorbed into the second one. Therefore, for optimization purpose the arc of  $u(t)$  is parameterized as  $\mathbf{u} = [t_s, u_0, \dots, u_{N-1}]^T$ , where  $0 < t_s < t_f$  is the switching time between the first two arcs, then the remaining time is equally divided into  $N$  grids, each of which implements a constant input,  $u_k$ .

In the following, we select  $N = 40$ . This choice is reasonable as it gives a nominal product of  $J = 20.37$  kmol, which corresponds to a loss of about 0.09 kmol (less than 0.5%) compared to the case with  $N = 100$ ; see Fig. 7 which shows the trajectories of  $x_d$ ,  $x_{10}$  (composition on top stage) and the objective function  $J$ . However, a problem here is that the end product composition is optimally constrained at  $x_d(t_f) = x_d^{des}$ , and this is also the case for all other operating conditions. Since the basis for all our results is that we have unconstrained operation, we propose to consider the following modified cost function which adds the end point constraint as a penalty

$$-J' = -M_d(t_f) x_d(t_f) + \omega (x_d(t_f) - x_d^{des})^2 \quad (70)$$

where  $\omega$  is the penalty factor which is set as  $10^5$  afterwards. Even with a large value for the weight  $\omega$ , the penalty approach may still result in a small constraint violation. We therefore introduce an extra safety margin of 0.005, for the desired product quality, which is realized by using  $x_d^{des} = 0.905$  in (70) in the following.

With the above input parameterization without the third arc and the modified cost function, a new static NLP is reformulated, which is nominally unconstrained. In the new formulation, the nominally optimal  $t_s$  is 1.03 h and  $J' = 19.82$ . With this, the true collected product is  $J = 19.85$  kmol with a purity of  $x_d = 0.904$ . Note that the reduced product compared to the nominal case is mainly caused by the backoff of  $x_d$ .

### 4.2.3. Self-optimizing control and results

We want to find a control policy so that operation remains close to optimal when there are uncertain disturbances. For self-

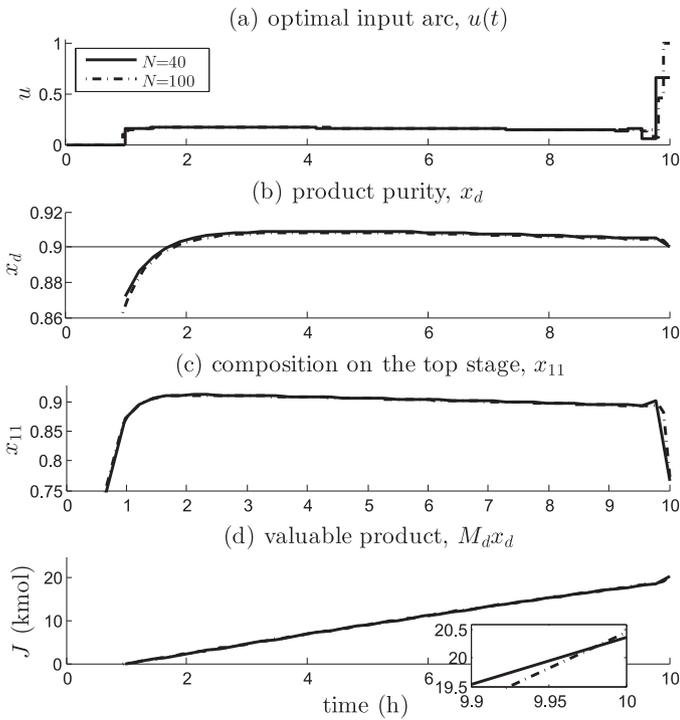


Fig. 7. Nominally optimal trajectories for the batch column ( $N=40$  and  $N=100$ ).

optimizing control, we consider the following two measurement sets:

$$\mathbf{y}_1 = [x_d \quad u]^T, \quad \mathbf{y}_2 = [x_2 \quad x_6 \quad x_9 \quad u]^T$$

where the terminal value of  $x_d$  in  $\mathbf{y}_1$  is the constraint itself, which seems to be an obvious controlled variable in practice.  $\mathbf{y}_2$  contains three in-column compositions, which is advantageous because the in-column compositions have stronger connections with other easy-to-measure variables, e.g. the tray temperatures, hence it would be easier to build an inferential model first when the compositions are unmeasured in a practical application. In both cases, the magnitudes of measurement noise are taken as 0.05 for the compositions and 0.01 for  $u$ .

For both measurements sets  $\mathbf{y}_1$  and  $\mathbf{y}_2$ , we consider the optimal time-varying LBT structure (Structure 2), the simplified time-varying diagonal structure (Structure 3) and the further simplified time-invariant diagonal structure (Structure 4). We obtain the input laws for Structure 2 analytically (Theorem 4), and for Structure 3 and 4 numerically. For example, the solution for measurement set  $\mathbf{y}_1$  becomes (following the initial startup period),

$$\begin{aligned} \text{Structure 2 : } u_0 &= -1.95 + 2.40x_d(t_s), \\ u_1 &= -2.222 + 1.397x_d(t_s) - 0.672u_0 \\ &\quad + 1.40x_d(t_s + \Delta t), \\ &\dots \end{aligned} \quad (71)$$

$$\begin{aligned} \text{Structure 3 : } u_0 &= -0.948 + 1.249x_d(t_s), \\ u_1 &= -0.557 + 0.82x_d(t_s + \Delta t) \\ &\dots \end{aligned} \quad (72)$$

$$\begin{aligned} \text{Structure 4 : } u_0 &= -2.334 - 2.826x_d(t_s), \\ u_1 &= -2.367 - 2.826x_d(t_s + \Delta t) \\ &\dots \end{aligned} \quad (73)$$

Table 5  
Local average losses for the batch column.

subset	Structure of $\bar{\mathbf{H}}$	average loss
$\mathbf{y}_1$	Structure 2	11.22
	Structure 3	21.53
	Structure 4	23.48
$\mathbf{y}_2$	Structure 2	2.31
	Structure 3	11.23
	Structure 4	16.37

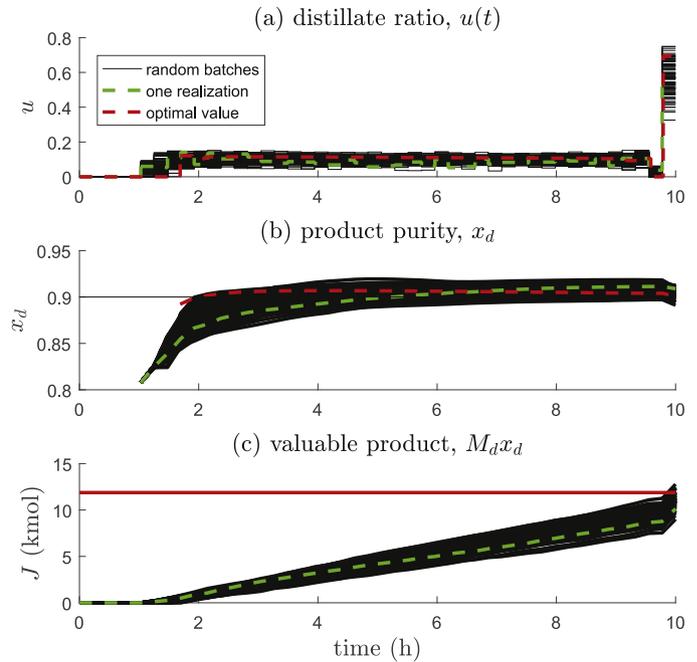


Fig. 8. Dynamic simulations for  $\mathbf{y}_1$  (Structure 2),  $[\alpha, V] = [1.4, 13]$ .

where  $\Delta t = (t_f - t_s)/N$ . Notice that, in all cases, the derived results also contain an input law for the switching time,  $t_s$ , as a function of initial conditions,  $t_s = k_0 + k_1x_d(0)$ . However, since there is in this case no uncertainties assumed in the initial states, it is equivalent to fixing  $t_s$  at the nominal value. The local average losses for different schemes are given in Table 5. It can be observed that for measurement set  $\mathbf{y}_1$  (Structure 2) gives a loss of 11.22, and as expected the loss is larger when using Structures 3 and 4 (21.53 and 23.48, respectively). With measurement set  $\mathbf{y}_2$ , the losses are smaller especially for Structure 2 which gives a loss of 2.31.

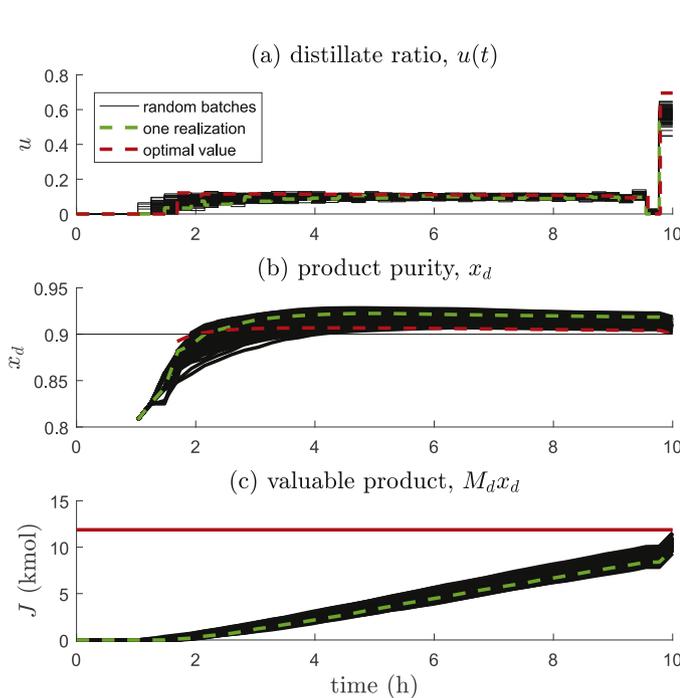
In the following, we investigate the performances of the two measurements subsets  $\mathbf{y}_1$  and  $\mathbf{y}_2$  for Structure 2. Dynamic simulations for two extreme scenarios,  $[\alpha, V] = [1.4, 13]$  and  $[1.6, 17]$ , are carried out. In the first case, the two disturbances are small and give less product. In the second case, the situation is opposite.

The results with  $[\alpha, V] = [1.4, 13]$  are showed in Fig. 8 and 9 for the two measurement sets, respectively, for 1000 batches with random measurement noise. It turns out that in the case of  $\mathbf{y}_1$ , 83.9 % batches satisfy the desired terminal quality,  $x_d(t_f) \geq 0.9$ , while for  $\mathbf{y}_2$  all batches satisfy this condition, as tabulated in Table 6. Regarding the economic index, the measurement set  $\mathbf{y}_1$  gives on average 11.01 kmol product, while  $\mathbf{y}_2$  gives 10.43 kmol. The better objective function for  $\mathbf{y}_1$  is caused by the under-purified product. For example, if we increase the safety margin by setting  $x_d^{des} = 0.912$  in (70), in which case the constraint satisfactions can be about 99 %, the collected product decreases to 10.14 kmol, which is as expected worse than  $\mathbf{y}_2$ .

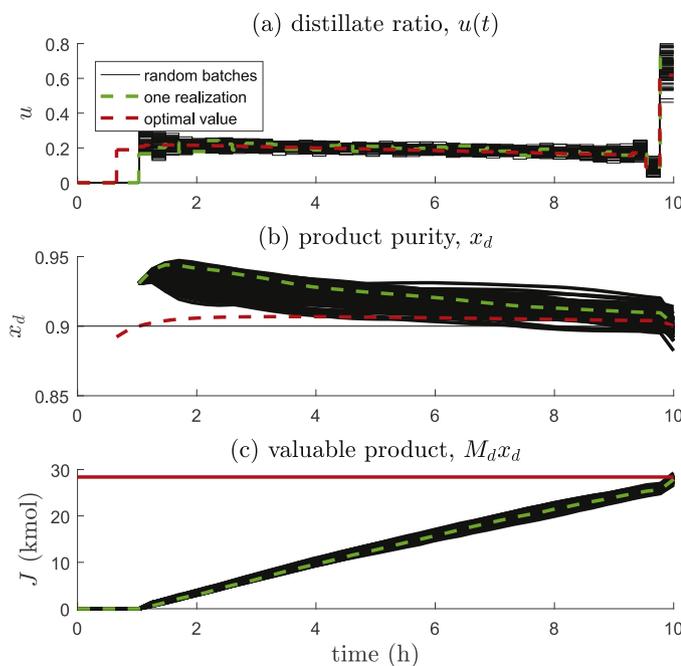
The results for the scenario with  $[\alpha, V] = [1.6, 17]$  are shown in Figs. 10 and 11. This time, measurement set  $\mathbf{y}_1$  performs even

**Table 6**  
Dynamic simulation performances with 1000 random batches.

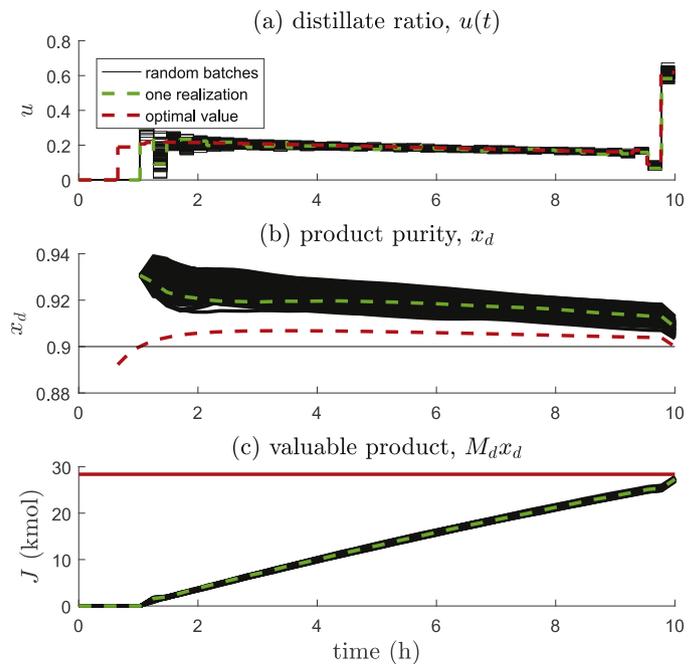
$[\alpha, V]$	subset	$x_d(t_f)$			average $J(\text{kmol})$
		average	s.t.d.	$\geq 0.9$	
[1.4, 13]	$\mathbf{y}_1$ (Structure 2)	0.904	0.0045	83.9 %	11.01
	$\mathbf{y}_2$ (Structure 2)	0.910	0.0034	100.0 %	10.43
[1.6, 17]	$\mathbf{y}_1$ (Structure 2)	0.900	0.0058	51.6 %	28.02
	$\mathbf{y}_2$ (Structure 2)	0.908	0.0021	100.0 %	27.31



**Fig. 9.** Dynamic simulations for  $\mathbf{y}_2$  (Structure 2),  $[\alpha, V] = [1.4, 13]$ .



**Fig. 10.** Dynamic simulations for  $\mathbf{y}_1$  (Structure 2),  $[\alpha, V] = [1.6, 17]$ .



**Fig. 11.** Dynamic simulations for  $\mathbf{y}_2$  (Structure 2),  $[\alpha, V] = [1.6, 17]$ .

worse regarding the product purity (51.6% constraint satisfaction). On the other hand, the control scheme using measurement set  $\mathbf{y}_2$  still gives 100 % constraint satisfaction and a smaller standard deviation for  $x_d$ . As shown in Table 6, the average objective function for  $\mathbf{y}_1$  is 28.02 kmol, and for  $\mathbf{y}_2$  it is  $J = 27.01$  kmol. However, again the results with  $\mathbf{y}_1$  are not really relevant because many batches do not satisfy the product constraint. Therefore, the scheme using  $\mathbf{y}_2$  is clearly the best. Note that the objective function values are very close to the exact true optimum, see Fig. 11(c).

## 5. Conclusions

In this paper, the self-optimizing control (SOC) methodology for static process operation was extended to dynamic optimal operation of unconstrained batch processes. The link between the static and dynamic SOC problems was based on a static reformulation of the dynamic optimization problem. In particular, the exact local method was extended to the dynamic case. For the dynamic case, the requirement of causality problem results in a structure-constrained SOC problem. For the lower-block triangular (LBT) structured, a convex constrained QP was formulated together with an analytical solution. On-line implementation of the proposed SOC solution is very simple, as the inputs can be directly calculated from the CV functions. A fed-batch reactor and a batch column were studied to illustrate the usefulness of the proposed method.

## Acknowledgement

L. Ye gratefully acknowledges the support from the National Natural Science Foundation of China (NSFC) (61673349, 61304081), the Talent project of Zhejiang Association for Science and Technology (2017YCGC014) and the China Scholarship Council (No. 201508330751). Parts of the work were performed during a study leave at NTNU in Norway, the support from NTNU is acknowledged. The open-source tool CasADi as well as the IPOPT solver for dynamic optimization are also acknowledged.

## References

- Alstad, V., Skogestad, S., 2007. Null space method for selecting optimal measurement combinations as controlled variables. *Ind. Eng. Chem. Res.* 46 (3), 846–853.
- Alstad, V., Skogestad, S., Hori, E.S., 2009. Optimal measurement combinations as controlled variables. *J. Process Control* 19 (1), 138–148.
- Andersson, J., 2013. A General Purpose Software Framework for Dynamic Optimization. Ph.D. thesis. Arenberg Doctoral School, KU Leuven, Leuven, Belgium.
- Biegler, L.T., 2007. An overview of simultaneous strategies for dynamic optimization. *Chem. Eng. Process.* 46 (11), 1043–1053.
- Biegler, L.T., Zavala, V.M., 2009. Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization. *Comput. Chem. Eng.* 33 (3), 575–582.
- Blondel, V., Tsitsiklis, J.N., 1997. NP-hardness of some linear control design problems. *SIAM J. Control Optim.* 35 (6), 2118–2127.
- Bryson, A.E., Ho, Y.C., 1975. *Applied optimal control: optimization, estimation and control*. CRC Press.
- Cao, Y., 2005. Direct and indirect gradient control for static optimisation. *Int. J. Autom. Comput.* 2 (1), 13–19.
- Dahl-Olsen, H., Narasimhan, S., Skogestad, S., 2008. Optimal output selection for control of batch processes. In: *Proc. of American Control Conference*.
- Francois, G., Srinivasan, B., Bonvin, D., 2005. Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *J. Process Control* 15 (6), 701–712.
- Girei, S.A., Cao, Y., Grema, A.S., Ye, L., Kariwala, V., 2014. Data-driven self-optimizing control. In: *Proceedings of the 24th European Symposium on Computer Aided Process Engineering*. Budapest, Hungary.
- Grema, A.S., Cao, Y., 2016. Optimal feedback control of oil reservoir waterflooding processes. *Int. J. Automation Comput.* 13 (1), 73–80.
- Gros, S., Srinivasan, B., Chachuat, B., Bonvin, D., 2009. Neighbouring-extremal control for singular dynamic optimisation problems. Part I: single-input systems. *Int. J. Control* 82 (6), 1099–1112.
- Halvorsen, I.J., Skogestad, S., Morud, J.C., Alstad, V., 2003. Optimal selection of controlled variables. *Ind. Eng. Chem. Res.* 42 (14), 3273–3284.
- Han, C., Jia, L., Peng, D., 2018. Model predictive control of batch processes based on two-dimensional integration frame. *Nonlinear Anal.* 28, 75–86. doi:10.1016/j.nahs.2017.11.002.
- Hu, W., Mao, J., Xiao, G., Kariwala, V., 2012. Selection of self-optimizing controlled variables for dynamic processes. In: *8th Intl. Symposium on ADCHEM*, pp. 774–779.
- Hu, W., Umar, L.M., Xiao, G., Kariwala, V., 2012. Local self-optimizing control of constrained processes. *J. Process Control* 22 (2), 488–493.
- Jäschke, J., Cao, Y., Kariwala, V., 2017. Self-optimizing control—a survey. *Annual Rev. Control* 43, 199–223.
- Jäschke, J., Fikar, M., Skogestad, S., 2011. Self-optimizing invariants in dynamic optimization. In: *CDC-ECE*, pp. 7753–7758.
- Jäschke, J., Skogestad, S., 2011. NCO tracking and self-optimizing control in the context of real-time optimization. *J. Process Control* 21 (10), 1407–1416.
- Kariwala, V., 2007. Optimal measurement combination for local self-optimizing control. *Ind. Eng. Chem. Res.* 46 (11), 3629–3634.
- Kariwala, V., Cao, Y., Janardhanan, S., 2008. Local self-optimizing control with average loss minimization. *Ind. Eng. Chem. Res.* 47 (4), 1150–1158.
- Lee, A.Y., Bryson, A.E., 1989. Neighbouring extremals of dynamic optimization problems with parameter variations. *Optimal Control Appl. Methods* 10 (1), 39–52.
- Manum, H., Narasimhan, S., Skogestad, S., 2008. A new approach to explicit mpc using self-optimizing control. In: *American Control Conference, 2008. IEEE*, pp. 435–440.
- Manum, H., Skogestad, S., 2012. Self-optimizing control with active set changes. *J. Process Control* 22 (5), 873–883.
- Oliveira, V., Jäschke, J., Skogestad, S., 2016. Null-space method for optimal operation of transient processes. In: *11th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS-CAB 2016)*, pp. 418–423. Trondheim, Norway.
- Podmajerský, M., Fikar, M., Chachuat, B., 2013. Measurement-based optimization of batch and repetitive processes using an integrated two-layer architecture. *Journal of Process Control* 23 (7), 943–955.
- Rangaiah, G.P., Kariwala, V., 2012. *Plantwide control: Recent developments and applications*. John Wiley & Sons.
- Skogestad, S., 2000. Plantwide control: The search for the self-optimizing control structure. *J. Process Control* 10 (5), 487–507.
- Srinivasan, B., Bonvin, D., 2007. Real-time optimization of batch processes by tracking the necessary conditions of optimality. *Industr. Eng. Chem. Res.* 46 (2), 492–504.
- Srinivasan, B., Palanki, S., Bonvin, D., 2003. Dynamic optimization of batch processes: I. characterization of the nominal solution. *Comput. Chem. Eng.* 27 (1), 1–26.
- Welz, C., Srinivasan, B., Bonvin, D., 2002. Evaluation of measurement-based optimization schemes for batch distillation. Barcelona, Spain, pp. 1586–1601.
- Welz, C., Srinivasan, B., Bonvin, D., 2008. Measurement-based optimization of batch processes: Meeting terminal constraints on-line via trajectory following. *J. Process Control* 18 (3), 375–382.
- Westphal, L.C., 1995. A special control law: deadbeat control. In: *Handbook of Control Systems Engineering*. Springer, pp. 493–504.
- Ye, L., Cao, Y., Li, Y., Song, Z., 2013. Approximating necessary conditions of optimality as controlled variables. *Ind. Eng. Chem. Res.* 52 (2), 798–808.
- Ye, L., Cao, Y., Ma, X., Song, Z., 2014. A novel hierarchical control structure with controlled variable adaptation. *Ind. Eng. Chem. Res.* 53 (38), 14695–14711.
- Ye, L., Cao, Y., Skogestad, S., 2017. Global self-optimizing control for uncertain constrained process systems. The 20th World Congress of the International Federation of Automatic Control. Toulouse, France.
- Ye, L., Cao, Y., Xiao, Y., 2015. Global approximation of self-optimizing controlled variables with average loss minimization. *Ind. Eng. Chem. Res.* 54 (48), 12040–12053.
- Ye, L., Guan, H., Yuan, X., Ma, X., 2017. Run-to-run optimization of batch processes with self-optimizing control strategy. *Canadian J. Chem. Eng.* 95 (4), 724–736. doi:10.1002/cjce.22692.
- Ye, L., Kariwala, V., Cao, Y., 2013. Dynamic optimization for batch processes with uncertainties via approximating invariant. In: *the 8th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1786–1791.
- Ye, L., Miao, A., Zhang, H., 2017. Real-time optimization of gold cyanidation leaching process in a two-layer control architecture integrating self-optimizing control and modifier adaptation. *Ind. Eng. Chem. Res.* 4002–4016.
- Yelchuru, R., 2012. Quantitative methods for controlled variables selection. Norwegian University of Science and Technology (NTNU) Ph.D. thesis.
- Yelchuru, R., Skogestad, S., 2012. Convex formulations for optimal selection of controlled variables and measurements using mixed integer quadratic programming. *J. Proc. Control* 22, 995–1007.