Korean J. Chem. Eng., **30**(2), 269-277 (2013)
DOI: 10.1007/s11814-012-0158-1

A quasi-sequential parameter estimation for nonlinear dynamic systems based on multiple data profiles

Chao Zhao*, Quoc Dong Vu, and Pu Li**,†**

*College of Chemistry and Chemical Engineering, FuZhou University, FuZhou 350108, P. R. China
**Simulation and Optimal Processes Group, Institute of Automation and Systems Engineering,
Ilmenau University of Technology, P. O. Box 100565, 98693 Ilmenau, Germany

(*Received 21 February 2012 • accepted 19 September 2012*)

Abstract−A three-stage computation framework for solving parameter estimation problems for dynamic systems with multiple data profiles is developed. The dynamic parameter estimation problem is transformed into a nonlinear programming (NLP) problem by using collocation on finite elements. The model parameters to be estimated are treated in the upper stage by solving an NLP problem. The middle stage consists of multiple NLP problems nested in the upper stage, representing the data reconciliation step for each data profile. We use the quasi-sequential dynamic optimization approach to solve these problems. In the lower stage, the state variables and their gradients are evaluated through integrating the model equations. Since the second-order derivatives are not required in the computation framework this proposed method will be efficient for solving nonlinear dynamic parameter estimation problems. The computational results obtained on a parameter estimation problem for two CSTR models demonstrate the effectiveness of the proposed approach.

Key words: Dynamic Parameter Estimation Problem, Nonlinear Programming, Quasi-sequential Dynamic Optimization, Multiple Data Sets, CSTR

INTRODUCTION

Developing rigorous dynamic process models with a highly predictive quality is essential for successful implementation of on-line optimization and advanced process control techniques since these applications heavily depend on model parameter values obtained from experimental data [1,2]. Parameter estimation is a critical step in the development and update of a rigorous process model. The nonlinear programming (NLP) formulation is the most popular methods used for parameter estimation of steady-state nonlinear models [3]. The resulting optimization problem can be solved with a standard NLP solver, such as sequential quadratic programming [3], interval analysis [5], Quasi-Newton algorithm [4], and heuristic optimization methods [16,17]. However, rigorous modeling of an industrial dynamic process usually leads to a nonlinear differential-algebraic equation (DAE) system with thousands of variables. For dynamic systems there may be time-dependent parameters to be estimated from available plant data. In addition, a number of measured dynamic profiles will be used, i.e., they have to be considered simultaneously for the parameter estimation. As a consequence, a complex DAE constrained optimization problem is to be solved for parameter estimation. Therefore, it is desirable to develop efficient estimation strategies and numerical algorithms which should be able to solve such challenging estimation problems, including multiple data profiles and large parameter sets.

Over the past years, many efforts have been made to solve different parameter estimation problems, and a number of decomposition algorithms have been proposed to improve the performance of estimation problems with error-in-variables (EIV) formulation. By employing a two-level strategy for estimation and simulation, Dovi and Paladino [6] presented a constrained variation approach to decouple the parameter estimation problem where the dependent variables are eliminated by solving model equations through a simulation step. Kim and Edgar [7] developed a slightly different approach using a two-stage NLP procedure to address data reconciliation and parameter estimation step separately. Tjoa and Biegler [8] proposed a similar approach based on successive quadratic programming (SQP) for parameter estimation problem with implicit models.

Due to the differential constraints usually present in the dynamic process model, nonlinear dynamic estimation problems with EIV formulation are more challenging. Albuquerque and Biegler [2] proposed an estimation approach based on an efficient decomposition strategy to estimate the process states and parameters simultaneously. In [9] a nonlinear trust-region SQP approach using a full discretization method was developed to the parameter estimation for a polymerization reactor. Zavala et al. [11] extended this simultaneous approach to solve the multiple data, large-scale, DAE constrained parameter estimation problems. Further development of this approach was presented in [13], in which the associated large-scale parameter estimation problem is solved using interior-point algorithm (IPOPT) and parallel computing strategy. Due to the nonlinear nature of the process models, the resulting parameter estimation optimization problem is nonconvex and may contain multiple local optima. To obtain global optimum of the parameter estimation problems, a global optimization procedure based on the deterministic branch and bound global optimization algorithm (αBB) was presented to solve the EIV formulation [15]. More recently, some heuristic optimization

†To whom correspondence should be addressed.
E-mail: pu.li@tu-ilmenau.de

methods, such as genetic algorithm (GA) [16] and particle swarm optimization (PSO) [17], have been used to solve complex parameter estimation and data reconciliation problems.

Although there has been considerable interest in parameter estimation in the process industry, very few studies have been made on exploiting the structure of the EIV formulation and multiple data profiles. All the aforementioned approaches need large mathematical manipulations to obtain the second-order derivatives of the model equations, which will become extremely expensive for the solution of large-scale DAE constrained parameter estimation problems and, therefore, have not been easily implemented with standard NLP software. Making use of the optimality condition of the sub-NLP problem, Faber et al. [10] proposed a three-stage framework for the estimation of nonlinear steady-state systems with multiple data-sets. In this study we extend the method of [10] to the parameter estimation of dynamic systems described by DAEs and derive a quasi-sequential algorithm for dynamic parameter estimation problems. It means that in this study a dynamic three-stage estimation framework is developed. Due to the decomposition of the optimization variables, the proposed approach can solve time-dependent parameter estimation problems with multiple data profiles by a standard NLP solver. The paper begins with a detailed discussion of the EIV formulation of the parameter estimation problem for dynamic systems. Based on the quasi-sequential optimization approach, a three-stage estimation computation framework is then derived for solving dynamic parameter estimation problems. The sensitivity computation strategy is also analyzed accordingly. Further, two practical examples are used to demonstrate the performance of this approach.

PROBLEM FORMULATION

1. Parameter Estimation for Dynamic Systems

We consider processes that can be modeled by a system of nonlinear differential-algebraic equations (DAEs):

$$\mathbf{f}(\dot{\mathbf{x}}(\mathrm{t}), \mathbf{x}(\mathrm{t}), \mathbf{u}(\mathrm{t}), \theta(\mathrm{t})) = 0 \tag{1}$$

$$\mathbf{g}(\mathbf{x}(\mathrm{t}), \mathbf{u}(\mathrm{t}), \theta(\mathrm{t})) = 0 \tag{2}$$

$$\mathbf{x}(\mathrm{t}_0) = \mathbf{x}_0 \tag{3}$$

$$\mathrm{t} \in [\mathrm{t}_0, \mathrm{t}_f] \tag{4}$$

In this formulation, $\mathbf{x}$(t) is a vector of state (dependent) variables and $\mathbf{u}$(t) is a vector of input (independent) variables, $\boldsymbol{\theta}$(t) is a vector of unknown parameters (time-dependent or time-independent) and will be determined by the parameter estimation. Eqs. (1) and (2) define differential-algebraic model equations arising from the material and energy balances. Eq. (3) represents the initial condition of the state variables.

In the error-in-variables approach, the objective is to minimize the weighted squared error between the measurements and the predicted values by the model, and all of measured variables are included in the objective function. To obtain more reliable parameter estimates, usually a series of measured data profiles from different operating scenarios will be used for the estimation. Using the definition of the model presented in Eqs. (1) and (2), the resulting parameter estimation problem can be formulated as a constrained dynamic optimization problem:

$$\min_{\theta} \mathrm{F} = \sum_{j=1}^{NS} \mathrm{F_j} = \sum_{j=1}^{NS}\sum_{i=1}^{NK} [\mathbf{y}_j(\mathrm{t}_i) - \mathbf{y}_{j,i}^M]^T \mathbf{V}_y^{-1} \\ [\mathbf{y}_j(\mathrm{t}_i) - \mathbf{y}_{j,i}^M] + [\mathbf{u}_j(\mathrm{t}_i) - \mathbf{u}_{j,i}^M]^T \mathbf{V}_u^{-1} [\mathbf{u}_j(\mathrm{t}_i) - \mathbf{u}_{j,i}^M] \tag{5}$$

s.t.

$$\mathbf{f}_j(\dot{\mathbf{x}}_j(\mathrm{t}), \dot{\mathbf{y}}_j(\mathrm{t}), \mathbf{x}_j(\mathrm{t}), \mathbf{y}_j(\mathrm{t}), \mathbf{u}_j(\mathrm{t}), \theta(\mathrm{t})) = 0, \tag{6}$$

$$\mathbf{g}_j(\mathbf{x}_j(\mathrm{t}), \mathbf{y}_j(\mathrm{t}), \mathbf{u}_j(\mathrm{t}), \theta(\mathrm{t})) = 0, \tag{7}$$

$$\mathbf{h}_j(\mathbf{x}_j(\mathrm{t}), \mathbf{y}_j(\mathrm{t}), \mathbf{u}_j(\mathrm{t}), \theta(\mathrm{t})) \geq 0, \tag{8}$$

$$\boldsymbol{\theta}^L \leq \boldsymbol{\theta}(\mathrm{t}) \leq \boldsymbol{\theta}^U \tag{9}$$

$$\mathrm{x}_j(\mathrm{t}_0) = \mathrm{x}_{j,0} \qquad \mathrm{j}=1, 2, \ldots, \mathrm{NS} \tag{10}$$

where $\mathbf{x} \in \mathbf{X} \subseteq \Re^n$, $\mathbf{y} \in \mathbf{Y} \subseteq \Re^m$, $\mathbf{u} \in \mathbf{U} \subseteq \Re^q$, $\boldsymbol{\theta} \in \Re^p$, $\mathbf{f} \cup \mathbf{g} \subseteq \Re^{n+m}$, $\mathbf{h} \subseteq \Re^k$. $\mathbf{y}$(t) denotes the vector of measured state variables, $\mathbf{V}_y$ and $\mathbf{V}_u$ in the objective function are the known covariance matrixes of the measurement errors of the dependent and independent variables, respectively. It is assumed that measured data have been received at the given time points t_i, i=1, …, NK, yielding the measured profiles from the output $\mathbf{y}_{j,i}^M$ and input $\mathbf{u}_{j,i}^M$ at the j th experiment, both of which are subject to errors. NS is the number of data profiles available. The inequality constraints $\mathbf{h}$ are formulated to satisfy process or unit operation constraints such as a maximum pressure or particularly required product purity.

It is noted that the number of equality and inequality constraints is NS times of the number of the original constraints, since all profiles are considered simultaneously. This leads to a dynamic optimization problem with a large number of constraints. Therefore, a decomposition strategy is desired to reduce the size of the problem. In this study, a three-stage framework combined with the quasi-sequential approach is derived to efficiently solve such parameter estimation problems for dynamic systems.

2. Collocation Method

At first we use the collocation method to discretize the dynamic system, so as to transform the DAEs into nonlinear algebraic equation systems. In comparison to other discretization methods, collocation on finite elements has the advantage of a higher accuracy of the polynomial approximation [13]. As shown in Fig. 1, the time horizon will be divided into time intervals (elements) and in each interval the state variables will be represented with a linear combination of a set of Lagrange polynomials:

$$\mathbf{x}_l(\mathrm{t}) = \sum_{j=0}^{NC} \mathrm{P}_j(\mathrm{t}) \mathbf{x}_{l,j} \tag{11}$$

$$= \sum_{j=0}^{NC} \left[\prod_{\substack{i=0 \\ i \neq j}}^{NC} \frac{\mathrm{t} - \mathrm{t}_i}{\mathrm{t}_j - \mathrm{t}_i} \right] \mathbf{x}_{i,j} \quad l = 1, \ldots, \mathrm{NL} \tag{12}$$

where NC is the number of collocation points in the time interval and NL the number of time intervals in the time period, respectively. With this representation the dependent variables and differential terms in the model equations on the collocation points of a time interval will be

$$\mathbf{x}_l(\mathrm{t}_i) = \mathrm{P}_j(\mathrm{t}) \mathbf{x}_{l,j} = \mathbf{x}_{l,i} \quad \mathrm{i} = 0, \ldots, \mathrm{NC} \tag{13}$$

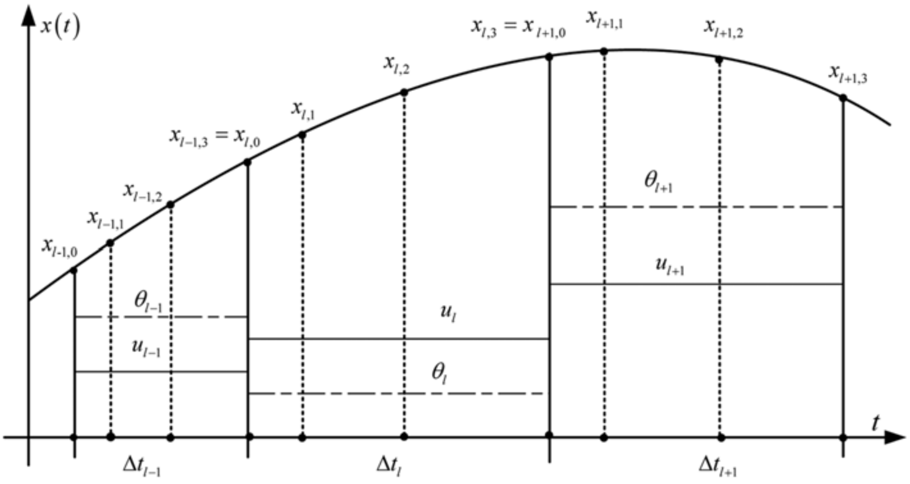

Fig. 1. Radau collocation on finite elements (NC=3).

$$\frac{\mathbf{x}_l(\mathrm{t}_i)}{\mathrm{dt}}=\sum_{j=0}^{NC}\frac{\mathrm{dP}_j(\mathrm{t}_i)}{\mathrm{dt}}\mathbf{x}_{l,j}\quad \mathrm{i}=0,\ldots,\mathrm{NC} \tag{14}$$

To keep the continuity of the state variables, we use the Radau collocation on finite elements, i.e., the last collocation point of an interval will be the initial point of the next interval. The control variables and parameters are represented as piecewise constants in each element in this study. Using this formulation it is relatively easy to address time-dependent parameter estimation problems. With the above discretization method the constrained dynamic optimization can now be transformed into an NLP problem:

$$\min_{\boldsymbol{x}_{j,l,i},\boldsymbol{u}_{j,l,i},\theta_{l,i}}\ \mathrm{F}(\mathbf{x}_{j,l,i},\mathbf{y}_{j,l,i},\mathbf{u}_{j,l,i},\theta_{l,i}) \tag{15}$$

$$\tilde{\mathbf{g}}_j(\mathbf{x}_{j,l,i},\mathbf{y}_{j,l,i},\mathbf{u}_{j,l,i},\theta_{l,i})=0, \tag{16}$$

$$\mathbf{h}_j(\mathbf{x}_{j,l,i},\mathbf{y}_{j,l,i},\mathbf{u}_{j,l,i},\theta_{l,i})\geq 0, \tag{17}$$

$$\boldsymbol{\theta}^L\leq\boldsymbol{\theta}_{l,i}\leq\boldsymbol{\theta}^U \tag{18}$$

$$\mathrm{i}=1\ldots,\ \mathrm{NC},\ \mathrm{l}=1\ldots,\ \mathrm{NL} \tag{19}$$

where $\tilde{\mathbf{g}}_j=[\mathbf{f}_j,\mathbf{g}_j]$. The dimension of this NLP problem is NS×(NL×NC×n+NL×m)+NL×p. For large-scale process models the dimension of this optimization problem can be so large that it cannot be efficiently solved with standard optimization software. Thus, it is desirable to develop efficient solution strategies and numerical algorithms to handle such computationally intensive problems.

THREE-STAGE APPROACH TO PARAMETER ESTIMATION FOR DYNAMIC SYSTEMS

For the parameter estimation of nonlinear steady-state models, Faber et al. [10] proposed a sequential approach to solving large-scale parameter estimation problems with multiple data-sets, where a nested three-stage computation was presented to decompose the problem. We extend this idea to develop a new decomposition approach for dynamic parameter estimation problems. Based on the collocation method and the quasi-sequential dynamic optimization approach, a novel three-stage computation framework for parameter estimation problem to dynamic systems is derived in this section. In particular, estimation problems with time-dependent parameters can also be addressed with this method. The scheme of proposed estimation framework is shown in Fig. 2.

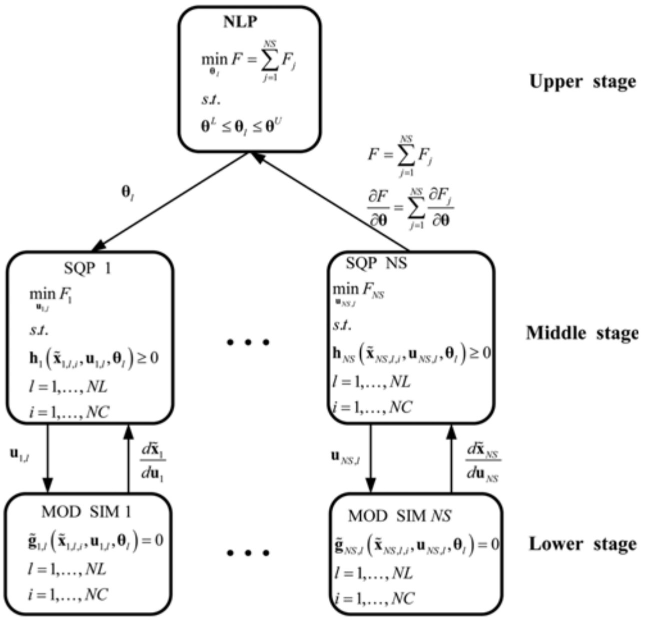
NLP
$\min_{\theta_l} F=\sum_{j=1}^{NS}F_j$
$s.t.$
$\theta^L\leq\theta_l\leq\theta^U$
Upper stage
θ_l
$F=\sum_{j=1}^{NS}F_j$
$\frac{\partial F}{\partial\theta}=\sum_{j=1}^{NS}\frac{\partial F_j}{\partial\theta}$
SQP 1
$\min_{u_{1,l}} F_1$
$s.t.$
$h_1(\tilde{x}_{1,l,i},u_{1,l},\theta_l)\geq 0$
$l=1,\ldots,NL$
$i=1,\ldots,NC$
SQP NS
$\min_{u_{NS,l}} F_{NS}$
$s.t.$
$h_{NS}(\tilde{x}_{NS,l,i},u_{NS,l},\theta_l)\geq 0$
$l=1,\ldots,NL$
$i=1,\ldots,NC$
Middle stage
$u_{1,l}$
$\frac{d\tilde{x}_1}{du_1}$
$u_{NS,l}$
$\frac{d\tilde{x}_{NS}}{du_{NS}}$
MOD SIM 1
$\tilde{g}_{1,l}(\tilde{x}_{1,l,i},u_{1,l},\theta_l)=0$
$l=1,\ldots,NL$
$i=1,\ldots,NC$
MOD SIM NS
$\tilde{g}_{NS,l}(\tilde{x}_{NS,l,i},u_{NS,l},\theta_l)=0$
$l=1,\ldots,NL$
$i=1,\ldots,NC$
Lower stage

Fig. 2. Three stage parameter estimation strategy.

According to Eqs. (5)-(10), the dynamic parameter estimation problem with multiple data profiles can be rewritten as the following:

$$\begin{aligned}\min_{\boldsymbol{\theta}_l,\boldsymbol{u}_{j,l},\boldsymbol{x}_{j,l,i},\boldsymbol{y}_{j,l,i}}\mathrm{F}&=\sum_{j=1}^{NS}\mathrm{F}_j=\sum_{j=1}^{NS}\sum_{l=1}^{NL}\sum_{i=1}^{NC}[\mathbf{y}_{j,l,i}-\mathbf{y}_{j,l,i}^M]^T\mathbf{V}_y^{-1}[\mathbf{y}_{j,l,i}-\mathbf{y}_{j,l,i}^M]\\&+[\mathbf{u}_{j,l}-\mathbf{u}_{j,l}^M]^T\mathbf{V}_{\boldsymbol{u}}^{-1}[\mathbf{u}_{j,l}-\mathbf{u}_{j,l}^M]\end{aligned} \tag{20}$$

s.t.

$$\tilde{\mathbf{g}}_j(\mathbf{x}_{j,l,i},\mathbf{y}_{j,l,i},\mathbf{u}_{j,l,i},\theta_l)=0 \tag{21}$$

$$\mathbf{h}_j(\mathbf{x}_{j,l,i},\mathbf{y}_{j,l,i},\mathbf{u}_{j,l,i},\theta_l)\geq 0 \tag{22}$$

$$\boldsymbol{\theta}^L\leq\boldsymbol{\theta}_l\leq\boldsymbol{\theta}^U \tag{23}$$

In this formulation we assume that the measurement points coincide with element positions, which means NK=NL. The upper stage as shown in Fig. 2 solves the parameter estimation problem in which the variables **u** and **y** are considered as functions of θ_l

$$\min_{\theta_l} \mathrm{F} = \sum_{j=1}^{NS} \mathrm{F}_j = \sum_{j=1}^{NS}\sum_{l=1}^{NL}\sum_{i=1}^{NC} [\mathbf{y}_{j,l,i}(\theta_l) - \mathbf{y}_{j,l,i}^M]^T \mathbf{V}_y^{-1} [\mathbf{y}_{j,l,i}(\theta_l) - \mathbf{y}_{j,l,i}^M] + [\mathbf{u}_{j,l}(\theta_l) - \mathbf{u}_{j,l}^M]^T \mathbf{V}_{\boldsymbol{u}}^{-1} [\mathbf{u}_{j,l}(\theta_l) - \mathbf{u}_{j,l}^M] \tag{24}$$

s.t.

$$\boldsymbol{\theta}^L \leq \boldsymbol{\theta}_l \leq \boldsymbol{\theta}^U \tag{25}$$

Since only θ_L are treated as optimization variables, the size of this optimization problem is NL×p. As shown in Fig. 2, the middle stage consists of multiple NLPs nested in the upper stage, representing a data reconciliation step for each data profile, which can be considered as a dynamic optimization problem. With given values of the parameters from the upper stage, the sub-NLP problem for each data profile has the following form:

$$\min_{\boldsymbol{u}_{j,l}, \boldsymbol{x}_{j,l,i}, \boldsymbol{y}_{j,l,i}} \mathrm{F}_j = \sum_{l=1}^{NL}\sum_{i=1}^{NC} [\mathbf{y}_{j,l,i} - \mathbf{y}_{j,l,i}^M]^T \mathbf{V}_y^{-1} [\mathbf{y}_{j,l,i} - \mathbf{y}_{j,l,i}^M] + [\mathbf{u}_{j,l} - \mathbf{u}_{j,l}^M]^T \mathbf{V}_{\boldsymbol{u}}^{-1} [\mathbf{u}_{j,l} - \mathbf{u}_{j,l}^M] \tag{26}$$

s.t.

$$\tilde{\mathbf{g}}_j(\mathbf{x}_{j,l,i}, \mathbf{y}_{j,l,i}, \mathbf{u}_{j,l}, \theta_l) = 0 \tag{27}$$

$$\mathbf{h}_j(\mathbf{x}_{j,l,i}, \mathbf{y}_{j,l,i}, \mathbf{u}_{j,l}, \theta_l) \geq 0 \tag{28}$$

The dimension of this problem is (NL×NC×n+NL×m). This number may be still too large for a standard NLP solver to deal with. To solve this large-scale NLP problem with available software, we use an efficient quasi-sequential optimization method proposed in [12]. The quasi-sequential approach uses a two-layer optimization strategy for solving dynamic optimization problems, as shown in Fig. 3. In the optimization layer, only independent variables and inequality constraints are included in the SQP formulation, corresponding to the middle stage in three-stage framework. For each sub-NLP problem with given θ_l in the optimization layer shown in Fig. 3 the following problem will be addressed,

$$\min_{\boldsymbol{u}_{j,l}} \mathrm{F}_j = \sum_{l=1}^{NL}\sum_{i=1}^{NC} [\mathbf{y}_{j,l,i} - \mathbf{y}_{j,l,i}^M]^T \mathbf{V}_y^{-1} [\mathbf{y}_{j,l,i} - \mathbf{y}_{j,l,i}^M] + [\mathbf{u}_{j,l} - \mathbf{u}_{j,l}^M]^T \mathbf{V}_{\boldsymbol{u}}^{-1} [\mathbf{u}_{j,l} - \mathbf{u}_{j,l}^M] \tag{29}$$

$$\mathbf{h}_j(\mathbf{x}_{j,l,i}, \mathbf{y}_{j,l,i}, \mathbf{u}_{j,l}, \theta_l) \geq 0 \tag{30}$$

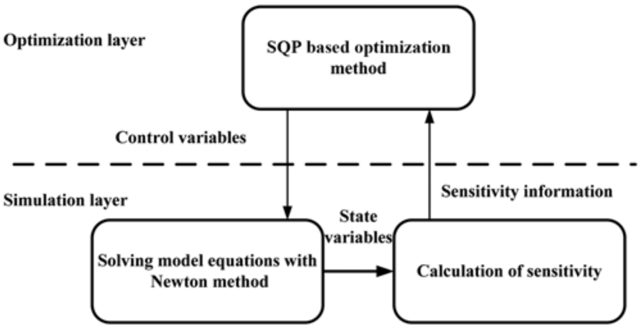
Optimization layer
SQP based optimization method
Control variables
Simulation layer
Sensitivity information
State variables
Solving model equations with Newton method
Calculation of sensitivity

Fig. 3. The structure of two layer optimization.

where only the independent variables $\mathrm{u}_{j,l}$ are treated as optimization variables; thus the size of this problem is (NL×m). The lower stage in Fig. 2 corresponds to the simulation layer in Fig. 3, which is nested in the sub-NLP solver. In this layer the dependent variables $\mathrm{x}_{j,l,i}$, representing the largest part of the whole variable space NL×NC×n, will be computed by solving the model equations with the Newton methods:

$$\tilde{\mathbf{g}}_{j,l}(\tilde{\mathbf{x}}_{j,l,i}, \mathbf{u}_{j,l}, \theta_l) = 0 \tag{31}$$

$$l=1, \ldots, \mathrm{NL};\ \mathrm{i}=1, \ldots, \mathrm{NC} \tag{32}$$

where $\tilde{\mathbf{x}}_j=[\mathbf{x}_j, \mathbf{y}_j]$. We omit *l*, i from the model equations in this section for notational simplicity. In the lower stage, it performs a simulation step with given $\mathbf{u}_{j,l}$ and θ_l. Since the degree of freedom is limited to the number of parameters in the upper stage and the number of independent variables in the middle stage, any standard NLP solver can be used to solve the related NLP problem.

SENSITIVITY COMPUTATION

The sensitivity of the dependent variables to the independent variables in each time element, required from the middle stage, can be transmitted through the continuity relation from element to element. On the basis of optimality condition in the middle stage, only the gradients of the dependent variables to the parameters are required in the upper stage, which can be computed by using the Jacobians of the model equations at the collocation points to the dependent variables and to the parameters, respectively. Considering the discretized equation system in the *k*th iteration of NLP and in the *l*th element, the model equations at the collocation point can be described as

$$\tilde{\mathbf{g}}_{j,l}(\tilde{\mathbf{x}}_{j,l,0}^k, \tilde{\mathbf{x}}_{j,l}^k, \mathbf{u}_{j,l}^k, \theta_l) = 0, \quad l=1,\ldots,\mathrm{NL} \tag{33}$$

where $\mathbf{x}_{j,l,0}^k$ is the initial value of $\mathbf{x}_{j,l}^k$ in the element. Through the first order Taylor expansion of Eq. (33) we obtain

$$\nabla_{\tilde{x}_{l,0}} \tilde{\mathbf{g}}_{j,l}^T (\tilde{\mathbf{x}}_{j,l,0} - \tilde{\mathbf{x}}_{j,l,0}^k) + \nabla_{\tilde{x}_l} \tilde{\mathbf{g}}_{j,l}^T (\tilde{\mathbf{x}}_{j,l} - \tilde{\mathbf{x}}_{j,l}^k) + \nabla_{\boldsymbol{u}_l} \tilde{\mathbf{g}}_{j,l}^T (\mathbf{u}_{j,l} - \mathbf{u}_{j,l}^k) = 0 \tag{34}$$

or

$$\mathbf{C}_{j,l,0}^k (\tilde{\mathbf{x}}_{j,l,0} - \tilde{\mathbf{x}}_{j,l,0}^k) + \mathbf{C}_{j,l}^k (\tilde{\mathbf{x}}_{j,l} - \tilde{\mathbf{x}}_{j,l}^k) + \mathbf{D}_{j,l}^k (\mathbf{u}_{j,l} - \mathbf{u}_{j,l}^k) = 0 \tag{35}$$

From Eq. (35) we can determine the sensitivity of the dependent variables with respect to initial states and independent variables of each element, respectively:

$$\frac{\partial \tilde{\mathbf{x}}_{j,l}}{\partial \mathbf{u}_{j,l}} = \nabla_{\tilde{x}_{l,0}}^{-T} \tilde{\mathbf{g}}_{j,l}^T \nabla_{\boldsymbol{u}_l} \tilde{\mathbf{g}}_{j,l}^T = -(\mathbf{C}_{j,l}^k)^{-1} \mathbf{D}_{j,l}^k \tag{36}$$

$$\frac{\partial \tilde{\mathbf{x}}_{j,l}}{\partial \tilde{\mathbf{x}}_{j,l,0}} = \nabla_{\tilde{x}_{l,0}}^{-T} \tilde{\mathbf{g}}_{j,l}^T \nabla_{\tilde{x}_{l,0}} \tilde{\mathbf{g}}_{j,l}^T = -(\mathbf{C}_{j,l}^k)^{-1} \mathbf{C}_{j,l,0}^k \tag{37}$$

For the continuity of the state variables between two intervals, we employ the last collocation point as the starting point of the state variables for the next interval rather than the extrapolation of the polynomial, as shown in Fig. 1. According to the chain rule, we can obtain

$$\frac{\partial \tilde{\mathbf{x}}_{j,l}}{\partial \mathbf{u}_{j,l-1}} = \frac{\partial \tilde{\mathbf{x}}_{j,l}}{\partial \tilde{\mathbf{x}}_{j,l,0}} \frac{\partial \tilde{\mathbf{x}}_{j,l-1}}{\partial \mathbf{u}_{j,l-1}} \tag{38}$$

and

$$\left[\frac{\mathrm{d}\tilde{\mathbf{x}}}{\mathrm{d}\mathbf{u}}\right]_j^T=\begin{bmatrix} s_1 & & & & & \\ s_{2,1} & s_2 & & & & \\ \vdots & \ddots & \ddots & & & \\ s_{l,1} & & & s_l & & \\ \vdots & & s_{i,j} & \ddots & \ddots & \\ s_{NL,1} & s_{NL,2} & \cdots & s_{NL,l} & \cdots & s_{NL} \end{bmatrix} \tag{39}$$

where

$$s_l=\frac{\partial\tilde{\mathbf{x}}_l}{\partial\mathbf{u}_l}\qquad s_{i,j}=\frac{\partial\tilde{\mathbf{x}}_i}{\partial\mathbf{u}_j} \tag{40}$$

The gradient of the objective function in the upper stage can be formulated as

$$\frac{\mathrm{dF}}{\mathrm{d}\theta}=\sum_{j=1}^{NS}\left[\frac{\partial \mathrm{F}_j}{\partial\mathbf{u}_j}\frac{\partial\mathbf{u}_j}{\partial\theta}+\frac{\partial \mathrm{F}_j}{\partial\mathbf{y}_j}\left(\frac{\partial\mathbf{y}_j}{\partial\mathbf{u}_j}\frac{\partial\mathbf{u}_j}{\partial\theta}+\frac{\partial\mathbf{y}_j}{\partial\theta}\right)\right] \tag{41}$$

Rearranging yields

$$\frac{\mathrm{dF}}{\mathrm{d}\theta}=\sum_{j=1}^{NS}\left[\left(\frac{\partial \mathrm{F}_j}{\partial\mathbf{u}_j}+\frac{\partial \mathrm{F}_j}{\partial\mathbf{y}_j}\frac{\partial\mathbf{y}_j}{\partial\mathbf{u}_j}\right)\frac{\partial\mathbf{u}_j}{\partial\theta}+\frac{\partial \mathrm{F}_j}{\partial\mathbf{y}_j}\frac{\partial\mathbf{y}_j}{\partial\theta}\right] \tag{42}$$

To compute these sensitivities of both the independent variables $\mathbf{u}_j$ and the dependent variables $\mathbf{y}_j$ to the parameters θ, the optimality condition of each sub-NLP in the middle stage can be utilized; at its convergence point there is

$$\phi_j(\mathbf{u}_j^*,\theta)=\frac{\partial \mathrm{F}_j}{\partial\mathbf{u}_j}+\frac{\partial \mathrm{F}_j}{\partial\mathbf{y}_j}\frac{\partial\mathbf{y}_j}{\partial\mathbf{u}_j}=0 \tag{43}$$

According to the optimality condition (43), the first term in (42) is equal to zero, and then the gradients required will result in

$$\frac{\mathrm{dF}}{\mathrm{d}\boldsymbol{\theta}}=\sum_{j=1}^{NS}\left[\frac{\partial \mathrm{F}_j}{\partial\mathbf{y}_j}\frac{\partial\mathbf{y}_j}{\partial\boldsymbol{\theta}}\right] \tag{44}$$

In this way, the second-order derivatives of the model equations are avoided, and thus the computation expense is significantly reduced.

To compute these derivatives of dependent variables $\mathbf{y}_j$ to the parameters θ, the first order Taylor expansion of Eq. (33) can be used again for the following equation:

$$\mathbf{g}_{j,l}(\tilde{\mathbf{x}}_{j,l,0}^*,\tilde{\mathbf{x}}_{j,l}^*,\mathbf{u}_{j,l}^*,\theta_l)=0,\ \ l=1,\ldots,\mathrm{NL} \tag{45}$$

where $\tilde{\mathbf{x}}_{j,l}^*$ $\mathbf{u}_{j,l}^*$ are the optimal values under the optimality condition of each sub-NLP in the middle stage. From Eq. (45) we can determine the sensitivity of the dependent variables with respect to parameters on each element,

$$\nabla_{\tilde{x}_{l,0}}\tilde{\mathbf{g}}_{j,l}^T\Delta\tilde{\mathbf{x}}_{j,l,0}+\nabla_{\tilde{x}_l}\tilde{\mathbf{g}}_{j,l}^T\Delta\tilde{\mathbf{x}}_{j,l}+\nabla_{\theta_l}\tilde{\mathbf{g}}_{j,l}^T\Delta\theta_l=0 \tag{46}$$

or

$$\mathbf{C}_{j,l,0}\Delta\tilde{\mathbf{x}}_{j,l,0}+\mathbf{C}_{j,l}\Delta\tilde{\mathbf{x}}_{j,l}+\mathbf{E}_{j,l}\Delta\theta_l=0 \tag{47}$$

According to the chain rule, we can have

$$\frac{\partial\tilde{\mathbf{x}}_{j,l}}{\partial\theta_l}=-(\mathbf{C}_{j,l}^k)^{-1}\mathbf{E}_{j,l} \tag{48}$$

$$\frac{\partial\tilde{\mathbf{x}}_{j,l}}{\partial\boldsymbol{\theta}_{l-1}}=\frac{\partial\tilde{\mathbf{x}}_{j,l}}{\partial\tilde{\mathbf{x}}_{j,l,0}}\frac{\partial\tilde{\mathbf{x}}_{j,l-1}}{\partial\boldsymbol{\theta}_{l-1}} \tag{49}$$

and then

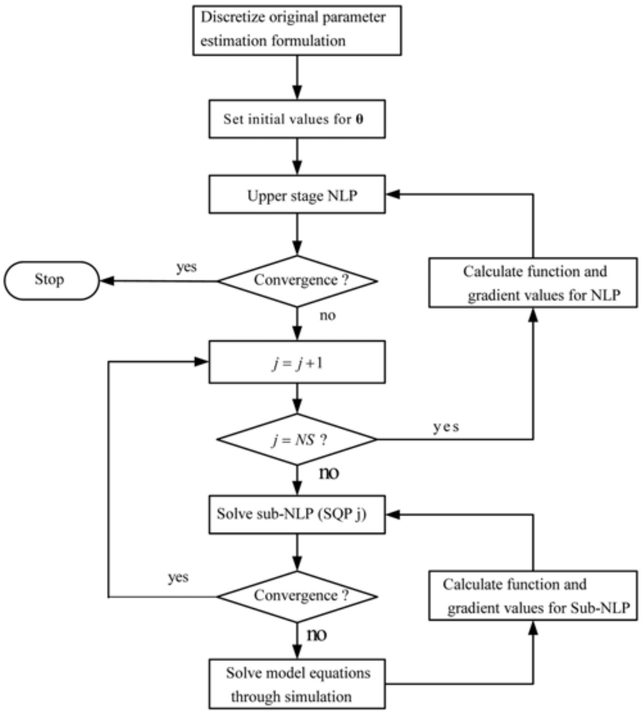
Discretize original parameter estimation formulation
Set initial values for θ
Upper stage NLP
Stop
yes
Convergence ?
no
Calculate function and gradient values for NLP
j = j+1
j = NS ?
yes
no
Solve sub-NLP (SQP j)
yes
Convergence ?
no
Calculate function and gradient values for Sub-NLP
Solve model equations through simulation

Fig. 4. Flow diagram of the quasi-sequential dynamic parameter estimation approach.

$$\left[\frac{\mathrm{d}\tilde{\mathbf{x}}}{\mathrm{d}\boldsymbol{\theta}}\right]_j^T=\begin{bmatrix} r_1 & & & & & \\ r_{2,1} & r_2 & & & & \\ \vdots & \ddots & \ddots & & & \\ r_{l,1} & & & r_l & & \\ \vdots & & r_{i,j} & \ddots & \ddots & \\ r_{NL,1} & r_{NL,2} & \cdots & r_{NL,l} & \cdots & r_{NL} \end{bmatrix} \tag{50}$$

where

$$r_l=\frac{\partial\tilde{\mathbf{x}}_l}{\partial\theta_l}\qquad r_{i,j}=\frac{\partial\tilde{\mathbf{x}}_i}{\partial\theta_j} \tag{51}$$

NUMERICAL IMPLEMENTATION

A step-by-step summary of the quasi-sequential dynamic parameter estimation method is given below:

- Given a fixed number of collocation points in each subinterval, discretize the parameters and independent variables with piecewise constant, and dependent variables with collocation method.
- Set the convergence tolerances and the bounds for parameters.
- Choose a starting point for $\boldsymbol{\theta}^0$ for all elements.
- Set the initial counter for the NLP in the upper stage k=0.
- At each iteration k:

1. Evaluate the objective function

$$\mathrm{F}=\sum_{j=1}^{NS}\mathrm{F}_j \tag{52}$$

and gradients

$$\frac{\mathrm{dF}}{\mathrm{d}\boldsymbol{\theta}}=\sum_{j=1}^{NS}\left[\frac{\partial \mathrm{F}_j}{\partial\mathbf{x}_j}\frac{\partial\mathbf{x}_j}{\partial\boldsymbol{\theta}}\right] \tag{53}$$

by successively calling the sub-NLPs in the middle stage (optimization layer):

(a) Provide an initial value for $\mathbf{u}_j^0$ and bounds for all variables.

(b) Set the iteration counter for the sub-NLP j as i=0.

(c) At each iteration i with $\boldsymbol{\theta}^k$ and $\mathbf{u}_j^i$:

i. Evaluate the objective function F_j and gradients

$$\frac{dF_j}{d\mathbf{u}_j}=\frac{\partial F_j}{\partial \mathbf{u}_j}+\frac{\partial F_j}{\partial \mathbf{y}_j}\frac{d\mathbf{y}_j}{d\mathbf{u}_j} \tag{54}$$

as well as

$$\frac{d\mathbf{h}_j}{d\mathbf{u}_j}=\frac{\partial \mathbf{h}_j}{\partial \mathbf{u}_j}+\frac{\partial \mathbf{h}_j}{\partial \tilde{\mathbf{x}}_j}\frac{d\tilde{\mathbf{x}}_j}{d\mathbf{u}_j} \tag{55}$$

by calling the lower stage (simulation layer):

A. Solve the equality constraints (model equations) with Newton methods to get dependent variables $\tilde{\mathbf{x}}_j$.

B. Computer the sensitivity matrix ($\partial\tilde{\mathbf{x}}_j/\partial\mathbf{u}_j$) using Eq. (39).

ii. Call the NLP solver to update $\boldsymbol{\theta}_j^i$.

iii. Update iteration counter i=i+1.

(d) Compute the gradients ($\partial\mathbf{y}_j/\partial\theta$) using Eq. (50) at convergence of all sub-NLPs.

(e) Return to upper stage.

2. Call a constrained NLP solver to update $\boldsymbol{\theta}^k$.

3. k=k+1.

- Stop, at convergence of the upper stage NLP.

The implementation of the computation framework is shown in Fig. 4. Note that when the original dynamic parameter estimation problem is transformed into an NLP problem by using the collocation method, we only need a standard NLP solver for solving the upper NLP problem (Eq. (24)) as well as the middle stage NLP problem (Eq. (29)) and a simulation step for the lower stage. Both the dependent and the independent variables in the upper stage are functions of the parameters, while the dependent variables in the middle stage are the functions of the independent variables. In the lower stage, the dependent variables are functions of both the independent variables and the parameters in computing sensitivities for the upper stage.

CASE STUDIES

To test the performance of the proposed approach, two practical estimation problems from the literature are considered. All computations were done using the Visual C++ compiler on an HP E2180 PC with 2 GHz of CPU and 1 GB of RAM. To solve the NLP problems from the upper and middle stages, a standard SQP solver from the IMSL library was used. In both examples, measurements are generated by adding Gaussian noise to true values.

1. Example 1

A CSTR is considered as shown in Fig. 4. An exothermic CSTR with an irreversible, first-order reaction A→B takes place in the liquid phase and the temperature is regulated with external cooling. This example is taken from [12] with the assumption that the liquid level is not constant. Mass and energy balances lead to the following nonlinear state model:

$$\frac{dh}{dt}=\frac{F_0-F}{\pi r^2}$$

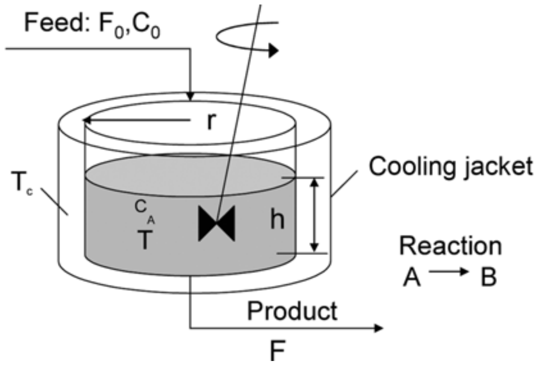

Fig. 5. Diagram of CSTR.

$$\frac{dc}{dt}=\frac{F_0(c_0-c)}{\pi r^2 h}-k_1\cdot c \tag{56}$$

$$\frac{dT}{dt}=\frac{F_0(T_0-T)}{\pi r^2 h}-\frac{-\Delta H}{\rho C_p}k_1\cdot c+\frac{2\cdot U}{r\rho C_p}(T_c-T)$$

where h, c, F, T are the level of the tank, product concentration, outlet flow rate and reactor temperature; F_0, c_0, T_0 are the inlet flow, inlet concentration and coolant liquid temperature. In this example, there are three dependent variables (h, c, T) and two independent variables (F, T_c). Five measurements of process variables were gained from simulated data with added noise.

The reaction rate constant is expressed as

$$k_1=c_1\exp\left(\frac{-E}{RT}\right) \tag{57}$$

where c_1 and E are the Arrhenius constants. These parameters will be estimated based on measurement data profiles. Following [7] a parameter transformation is used, resulting in the following rate equations:

$$k_1=\theta_1\exp\left[-\theta_2\left(\frac{T_r}{T}-1\right)\right] \tag{58}$$

with θ_1=$c_1$exp(−E/RT_r) and θ_2=−E/RT_r. T_r is reference temperature (350 K).

Experimental data sets were generated by adding random noises to five data sets simulated at a steady-state operating point of h=0.659 m, c=0.887 mol/L, T=324.5 K, F=100 L/min, and T_c=300 K. In this simulation case, the overall time horizon is 50 minutes and sampled time is set to 1 minute; thus each data contains 50 subintervals (measured points). The measured values of the variables are obtained by

$$[\hat{\mathbf{u}}_i,\hat{\mathbf{y}}_i]^T=[\hat{F}_i,\hat{T}_{c,i},\hat{h}_i,\hat{c}_i,\hat{T}_i]^T+\text{rand}\,\sigma_i\quad i=1,\ldots,NS \tag{59}$$

where rand is a random number generated on the interval [−1, 1] and the noise level σ_i is set to 0.01. A measurement error with the standard deviations of 5% of the corresponding reference value (steady-state value) was considered. Since the proposed nested three-stage computation framework is able to solve data reconciliation and parameter estimation problems simultaneously, thus the reconciliation of all measured variables was also carried out. The initial guess values for the parameters are set to θ_1^0=0.1 and θ_2^0=10 and the independent variables are initialized at their measurement values. The constraints enforced on the input and output variables are expressed as follows:

$$\begin{aligned} &0.5 \le h \le 3.50 \text{ m} \\ &0.8 \le c \le 1.0 \text{ mol/L} \\ &85 \le F \le 115 \text{ L/min} \\ &299 \le T_c \le 302 \text{ K} \end{aligned} \tag{60}$$

And boundaries on the parameters are set to

$$\begin{aligned} &0.0 \le \theta_1 \le 2.0 \\ &10 \le \theta_2 \le 30 \end{aligned} \tag{61}$$

Here we discretize the NLP with the 3-point collocation in the each time interval; thus the parameter estimation problem is formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{\theta}_l, \boldsymbol{u}_{j,l}, \boldsymbol{x}_{j,l,i}, \boldsymbol{y}_{j,l,i}} \mathrm{F} &= \sum_{j=1}^{5} \mathrm{F}_j = \sum_{j=1}^{5}\sum_{l=1}^{50}\sum_{i=1}^{3} [\mathbf{y}_{j,l,i} - \mathbf{y}_{j,l,i}^M]^T \mathbf{V}_y^{-1} [\mathbf{y}_{j,l,i} - \mathbf{y}_{j,l,i}^M] \\ &+ [\mathbf{u}_{j,l} - \mathbf{u}_{j,l}^M]^T \mathbf{V}_{\boldsymbol{u}}^{-1} [\mathbf{u}_{j,l} - \mathbf{u}_{j,l}^M] \end{aligned} \tag{62}$$

s.t.

$$\begin{aligned} &\text{model equations (56)} \\ &\text{process constraints (60)} \\ &\text{parameter constraints (61)} \end{aligned} \tag{63}$$

where $\boldsymbol{\theta}=[\theta_1, \theta_2]^T$ $\mathbf{x}=[\mathrm{T}]^T$ $\mathbf{u}=[\mathrm{F}, \mathrm{T}_c]^T$ $\mathbf{y}=[\mathrm{h}, \mathrm{c}]^T$.

In this test, the proposed algorithm took six iterations of the upper stage NLP to converge to a solution with an objective value of F= 0.0071 and parameter values of [θ_1=0.993371, θ_2=24.98827], which correspond well to the real values [θ_1=0.999932, θ_2=25.0]. The CPU time was 68.4s.

The results of the data reconciliation for level h, concentration c, outlet flow F and reactor temperature T, are shown in Fig. 6 to Fig. 9. In these figures the red dotted lines correspond to the measured values with noise, the black dotted lines to the estimated values of measurements through application of proposed optimizer, the green dotted lines to the simulations with estimated parameters, and the blue dotted lines correspond to the simulations with true parameters. In terms of data reconciliation, the estimated values of the process variables contained far less noise than the simulated measurements, as are shown in Fig. 6 and Fig. 7. Fig. 8 shows the estimate of the independent variable of outlet flowrate F, so the proposed algorithm was able to track the true measurements very closely. The results clearly show that the proposed method gives satisfactory parameter estimation and data reconciliation performance.

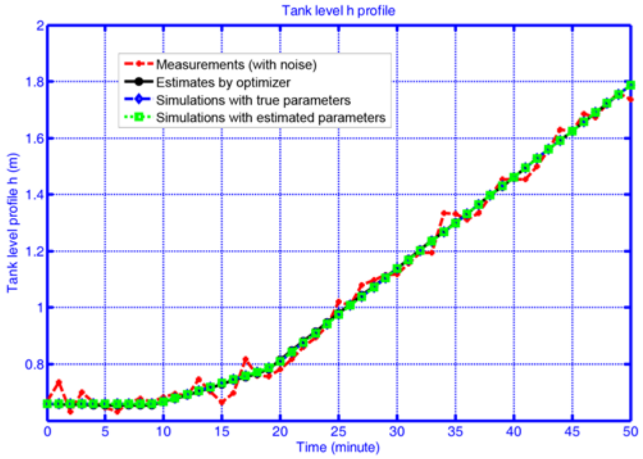

Fig. 6. Results of data reconciliation for level.

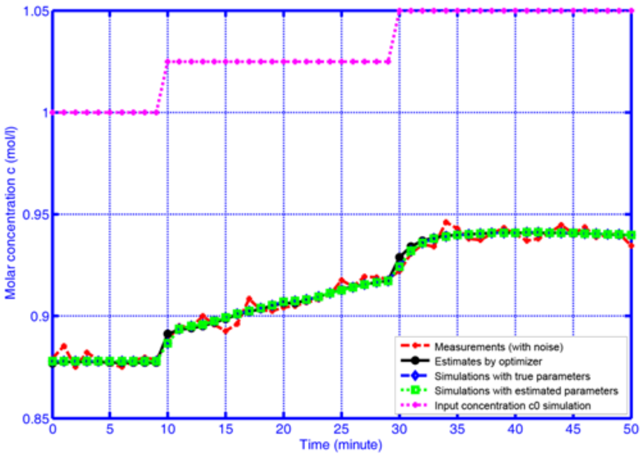

Fig. 7. Results of data reconciliation for concentration.

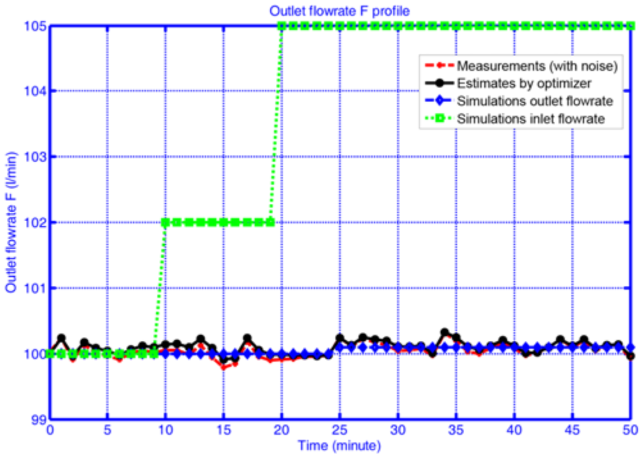

Fig. 8. Results of data reconciliation for outlet flowrate.

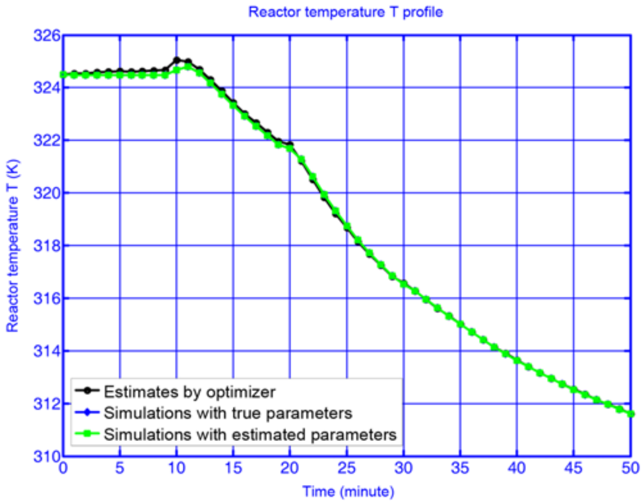

Fig. 9. Results of data reconciliation for reaction temperature.

2. Example 2

In this section, we present the results of comparison between proposed quasi-sequential nonlinear dynamic estimation (QSNDE) approach and nonlinear dynamic error-in-variables method (NDEVM) [14] based on collocation strategy to a parameter estimation problem. This example is taken from Kim [7], which was also used by Liebman [14] to demonstrate an algorithm for nonlinear dynamic data reconciliation. The dynamic model for the irreversible reaction system is

$$\begin{aligned} \frac{dA}{dt} &= \frac{1}{\tau}(A_0 - A) - kA \\ \frac{dB}{dt} &= \frac{1}{\tau}(B_0 - B) + kA \qquad (64) \\ \frac{dT}{dt} &= \frac{1}{\tau}(T_0 - T) + \frac{-\Delta H_r}{\rho C_p}(kA) \end{aligned}$$

The reader is referred to Kim [7] for details of the physical property data. Again referring to the Eq. (58) in example 1, parameters $\boldsymbol{\theta}$=[θ_1, θ_2](θ_1=c$_1$exp(–E/RT$_r$), θ_2=exp(E/RT$_r$)) have to be estimated based on measurement data. Measurements for three dependent variables (outlet temperature T, outlet concentrations A and B) and independent variables (inlet temperature T$_0$ and inlet concentration A$_0$) are generated by simulation around the parameters of $\boldsymbol{\theta}$=[0.017, 12.58] and adding random noises to 10 data profiles. Each data profile contains 50 measured points, namely the measured values of the variables are obtained by

$$[\hat{\mathbf{y}}_i, \hat{\mathbf{u}}_i]^T = [\hat{A}_i, \hat{B}_i, \hat{T}_i, \hat{A}_{0,i}, \hat{T}_{0,i}]^T + \text{rand}\,\sigma_i \qquad (61)$$

In this example, the noise level σ_i is set to 0.05 and the standard deviations of measurement errors in process variables are taken as 3% of the measured value. The initial values for the parameters are set to θ_1^0=0.1 and θ_2^0=10. The boundary on all parameters was set at $\boldsymbol{\theta}$=[0, 15] and those on the process variables were set at the measured value ±3 times their standard deviation. In this example, the NDEVM method was applied with history horizons of both five and ten time steps to solve data reconciliation and parameter estimation problems, respectively.

The estimates obtained using the three algorithms are given in Table 1, along with the corresponding objective function, the SSRES (sum of the squares of the residuals) and the computation time. As observed from the first row of Table 1, the NDEVM method (H=10) converged to a local minimum corresponding to a solution of 11.66 with the parameter values of $\boldsymbol{\theta}$=[0.012, 11.21] while QSNDE obtained the global solution of 2.06 with the parameter values of $\boldsymbol{\theta}$=[0.018, 12.72], which correspond well with the true parameter values of $\boldsymbol{\theta}$=[0.017, 12.58]. From the last row of Table 1, it can be observed that QSNDE method requires the least computing time, which is about 40% lower than that given by NDEVM method with H=5 and about 55% lower than that obtained using NDEVM method with H=10. This is mainly because in the QSNDE method the simulation step eliminates the largest part of the variables as well as the equality constraints, which results in a smaller optimization problem (both in terms of variables and constraints) that has to be solved. Also, because the second-order derivatives of the model equations are avoided in the QSNDE method, the computation expense is significantly reduced.

Table 1. Comparison of parameter estimates for CSTR model

Method	Estimates	Objective function	SSRES	CPU time (s)
NDEVM (H=5)	(0.01, 10.54)	11.66	24.1	409
NDEVM (H=10)	(0.012, 11.21)	9.25	15.4	551
QSNDE	(0.018, 12.72)	2.06	4.52	248

SSRES: sum of the squares of the residuals

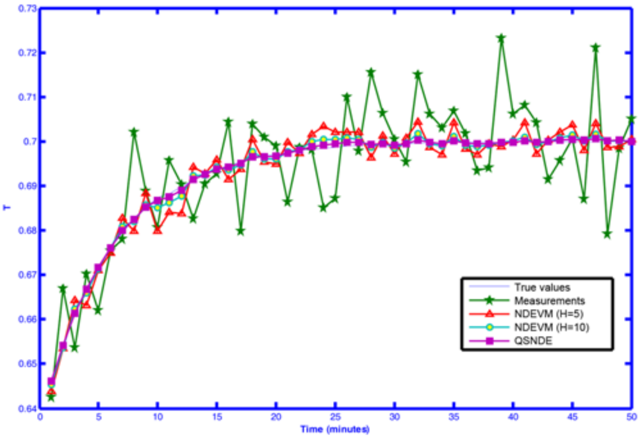

Fig. 10. Data reconciliation of by NDEVM and QSNDE.

The standard deviations of the noise in the simulated measurements and the noise in the reconciled estimates for all variables are shown in Table 2. With exception of the estimate error for the inlet concentration, significant reductions in the measurement error were achieved through the application of the three algorithms. This is because both NDEVM and QSNDE methods take the EVM parameter estimation formulation in which the measurement errors in all variables are treated in the optimization of objective function and

Table 2. Comparison of noise reduction

Variable	ME	NDEVM (H=5)		NDEVM (H=10)		QSNDE	
		EE	ER (%)	EE	ER (%)	EE	ER (%)
A	0.0594	0.0192	68	0.0163	73	0.0125	79
B	0.0537	0.0251	53	0.0192	64	0.0142	74
T	0.0485	0.0213	54	0.0185	61	0.0157	67
A$_0$	0.0559	0.0495	11	0.0670	--	0.0484	17
T$_0$	0.0479	0.0170	65	0.0165	63	0.0068	84

ME: Measurement error; EE: Estimated error; ER: Error reduction both parameter estimates and reconciled data estimates are provided. However, it can be observed from Table 2 that the QSNDE method can provide better agreement between measured and estimated values when compared to NDEVM methods, which can also be illustrated by Fig. 8 as well as the SSRES displayed in Table 2. The comparison results indicate that the presented approach outperformed NDEVM methods in terms of the robustness and computational efficiency.

CONCLUSIONS

Parameter estimation for nonlinear dynamic systems remains as a challenging task both methodologically and computationally. In this study, we present a three-stage computation framework for solving parameter estimation problems for dynamic systems based on multiple data profiles. First, the dynamic parameter estimation problem is transformed to an NLP problem by using collocation on finite elements. The model parameters to be estimated are treated in the upper stage by solving an NLP problem. The middle stage consists of multiple NLPs nested in the upper stage, representing the data reconciliation step for each data profile. The dynamic optimization problems in the middle stage are solved by an efficient quasi-sequential dynamic optimization method. Since the second-order derivatives of the model equations are avoided, the computation expense is significantly reduced. The computational results obtained from parameter estimation for two CSTR models demonstrate the effectiveness of the proposed approach.

ACKNOWLEDGEMENTS

Financial support from the China Scholarship Council and German Academic Exchange Service (DAAD) through the grants PPP-20073086, PPP-20096037 is gratefully acknowledged. This work is also supported by National Nature Science Foundation under Grant 60804027 and FZU-022335, 600338, 600567. The PhD scholarship program of the Vietnamese government is also acknowledged.

NOMENCLATURE

f, **g**, **h** : functions
x, **y**, **u** : vector of dependent, measured state and independent variables
$\mathbf{y}^M_{j,\,i}$, $\mathbf{u}^M_{j,i}$: measured output and input variable
$\boldsymbol{\theta}$: parameters
$\mathbf{V}_y$, $\mathbf{V}_u$: known covariance matrixes of the dependent and independent variables
A : outlet concentration
B : outlet concentrations
T : outlet temperature [K]
A_0 : inlet concentration
B_0 : inlet concentration
h : level of the tank
c : product concentration
c_0 : inlet concentration
F : outlet flow rate
F_0 : inlet flow
T_0 : inlet temperature
T_c : coolant liquid temperature
T_r : reference temperature, 350 K
H : history horizon
NS : number of data profiles
NC : number of collocation points
NL : number of time intervals
NK : number of time points in the experiment
DAEs : differential-algebraic equations
EIV : error in variables
NDEVM : dynamic error-in-variables method
QSNDE : quasi-sequential nonlinear dynamic estimation

Subscripts

i : index of collocation points
j : index of collocation points
l : index of time intervals

Superscripts

k : index of iteration
M : measurement

REFERENCES

1. M. J. Liebmann, T. F. Edgar and L. S. Lasdon, *Comput. Chem. Eng.*, **16**, 963 (1992).
2. J. S. Albuquerque and L. T. Biegler, *Comput. Chem. Eng.*, **19**, 1031 (1995).
3. I. B. Tjoa and L. T. Biegler, *Conput. Chem. Eng.*, **15**, 679 (1991).
4. H. I. Britt and R. H. Luecke, *Technometrics*, **15**, 233 (1973).
5. C. Y. Gau and M. A. Stadtherr, *Comput. Chem. Eng.*, **24**, 631 (2000).
6. V. G. Dovi and O. Paladino, *Comput. Chem. Eng.*, **13**, 731 (1989).
7. I. W. Kim, M. J. Liebmann and T. F. Edgar, *AIChE J.*, **36**, 985 (1990).
8. L. T. Biegler and I. B. Tjoa, *Annals of Operation Research*, **42**, 1 (1993).
9. A. Arora and L. T. Biegler, *Ind. Eng. Chem. Res.*, **43**, 3616 (2004).
10. R. Faber, P. Li and G. Wonzy, *Ind. Eng. Chem. Res.*, **42**, 5850 (2003).
11. V. M. Zavala, L. T. Biegler and C. D. Lairda, *Chem. Eng. Sci.*, **63**, 4834 (2008).
12. W. R. Hong, S. Q. Wang, P. Li, G. Wonzy and L. T. Biegler, *AIChE J.*, **52**, 255 (2006).
13. V. M. Zavala and L. T. Biegler, *Ind. Eng. Chem. Res.*, **45**, 7867 (2006).
14. I. W. Kim, T. F. Edgar and M. J. Liebman *Comput. Chem. Eng.*, **15**, 663 (1991).
15. W. R. Esposito and C. A. Floudas, *Ind. Eng. Chem. Res.*, **39**, 1291 (2000).
16. W. Wongrat, T. Srinophakun and P. Srinophakun, *Comput. Chem. Eng.*, **29**, 1059 (2005).
17. D. M. Prata, E. L. Lima and J. C. Pinto, *Chem. Eng. Sci.*, **65**, 4943 (2010).