

A reinforced combinatorial particle swarm optimization based multimodel identification of nonlinear systems

AHMED A. ADENIRAN AND SAMI EL FERIK

Systems Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

(RECEIVED November 24, 2015; ACCEPTED June 1, 2016)

Abstract

Several industrial systems are characterized by high nonlinearities with wide operating ranges and large set point changes. Identification and representation of these systems represent a challenge, especially for control engineers. Multimodel technique is one effective approach that can be used to describe nonlinear systems through the combination of several submodels, where each is contributing to the output with a certain degree of validity. One major concern in this technique, especially for systems with unknown operating conditions, is the partitioning of the system's operating space and thus the identification of different submodels. This paper proposes a three-stage approach to obtain a multimodel representation of a nonlinear system. A reinforced combinatorial particle swarm optimization and hybrid K-means are used to determine the number of submodels and their respective parameters. The proposed method automatically optimizes the number of submodels with respect to the submodel complexity. This allows operating space partition and generation of a parsimonious number of submodels without prior knowledge. The application of this approach on several examples, including a continuous stirred tank reactor, demonstrates its effectiveness.

Keywords: K-Means; Multimodel; Nonlinear Systems; Particle Swarm Optimization; Systems Identification

1. INTRODUCTION

In the past years, great efforts have been devoted to modeling and identification of nonlinear systems due to the inherent nonlinearities in real-life industrial plants and processes. For such systems, assumption of linearity fails, and an accurate mathematical model based on the physics of the system is infeasible due to large number of parameters, some of which can be unknown (Ronen et al., 2002). Researchers have proposed several nonlinear identification methods (e.g., DeVeaux et al., 1993; Kashiwagi & Rong, 2002; Billings, 2013; de Costa Martins & de Araujo, 2015; Scarpiniti et al., 2015; Pan et al., 2016), and demonstrated that these methods are capable of identifying nonlinear systems with different degrees of performance. However, they have several drawbacks. An excellent review on these methods can be found in Yassin et al. (2013).

A multimodel-based alternative approach to modeling and identification of these systems has attracted many researchers recently. In contrast to the conventional modeling technique, a system is represented by a set of models that are combined,

with different degree of validity, to form the global model. Each model represents the system in a specific region of operation. Owing to its potential benefits, this effective field of research has received several contributions, and has gained lots of interest in many fields of application such as biochemical (Dunlop et al., 2007), process control (Xue & Li, 2005), communication (Abbas & Tufvesson, 2011), and power systems (Gan & Wang, 2011). Despite its benefits, the approach still faces several challenges.

One main challenge of the multimodel approach is the partitioning of the system's operating space to a number of subspaces. This further raises the question of how many submodels are required to adequately represent the entire operating region of the nonlinear system when combined within the multimodel framework. One solution to this problem is the design of identification experiments for known operating spaces (e.g., Dunlop et al., 2007; Gregorčič & Lightbody, 2007; Novak et al., 2009; Wernholt & Moberg, 2011). Data are collected for each operating space, and a submodel is identified for each *a priori* known region.

However, the knowledge involved in any industrial systems and processes, especially chemical process, is often incomplete. In addition, they may be subjected to unknown parameter variations and exhibit wide operating ranges.

Reprint requests to: Ahmed A. Adeniran, Systems Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia. E-mail: selferik@kfupm.edu.sa

Therefore, in such situations, partitioning of the operating space and identification of the submodels can be very challenging due to lack of prior information on the system's operating conditions.

The problem of identifying the parameters of the submodels is coupled with the data partition problem, whereby each data point needs to be associated with the most suitable submodel. In the partitioning process, optimization of number of submodels and their parameters is very crucial for the correct identification of the system. While too few submodels can deteriorate the systems, increasing the number of submodels does not necessarily improve the performances obtained (Ksouri-Lahmari et al., 2004).

In the literature, many interesting algorithms have been proposed to address this challenge. Ronen et al. (2002) used fuzzy clustering for operating space partition and utilized both cluster and model validities methods to find the optimal number of clusters. The approach repeats the procedure for a number of submodels until a satisfactory number is obtained. Venkat et al. (2003) proposed a loss function, linear model based reconstruction error, to determine the optimal number of clusters resulting in minimum model error. Samia et al. (2002, 2008) utilized a heuristic method based on the number of neurons in the output layer of a Kohonen network. In their approach, the network has to be trained repeatedly until a satisfactory number of clusters is obtained. Moreover, the training of the network is very slow and increases with the number of data samples. Ltaief et al. (2008), Mohamed et al. (2011), and Bedoui et al. (2012) utilized subtractive clustering that automatically determines the number of clusters. The accuracy of their algorithm was known to depend on proper selection of its parameters (Demirli et al., 2003).

In Gugaliya and Gudi (2010) and Novak et al. (2011) large number of clusters was initially assumed and later reduced by merging similar clusters. Teslic et al. (2011) proposed an iterative incremental partitioning algorithm using G-K clustering. Numbers of clusters are iteratively increased by splitting the worst modeled cluster if its standard deviation error is greater than a certain threshold. Elfelly et al. (2010) proposed the use of a separate clustering algorithm called the rival penalized competitive learning neural network. Adequate numbers of clusters are determined by visual consideration of only clusters' centers that are enclosed by the data distribution when the initial number of clusters is larger than the real number of operating clusters. In general, all the aforementioned methods partitioned the operating space and/or determined the number of adequate submodels based on data distribution only, which may not reflect the complexity of the system's behavior.

In this paper, an efficient three-stage algorithm is proposed to obtain a multimodel representation of nonlinear systems without prior knowledge of the operating conditions. In the first stage, estimation of the initial parameters and the number of submodels are both obtained through a reinforced combinatorial particle swarm optimization (RCPSO). To identify more efficient submodels, hybrid K-means is used to obtain

the final submodels in the second stage. Finally, based on a constrained Kalman filter, the identified submodels are interpolated to form the multimodel output that has the capability to approximate the real system. The main advantage of the proposed framework is its automatic optimization of the required number of submodels with respect to submodel complexity. This implies that the operating space of the system can be partitioned into a parsimonious number of submodels, and the structure of the submodels can be assumed without prior knowledge. Thus, the algorithm can automatically converge to a good compromise between the number of submodels and complexity. Another interesting advantage is that the partition and selection of number of submodels is based not only on data distribution but also on the linearity of the operating region and that of the assumed model structure. Benchmark simulation examples, including a continuous stirred tank reactor system, are provided to illustrate the effectiveness of the proposed method.

The rest of this paper is organized as follows: Section 2 describes the multimodel framework and problem formulation. In Section 3, the first stage of the proposed multimodel approach is discussed, followed by the second and third stages in Sections 4 and 5, respectively. Simulation examples are provided in Section 6, to demonstrate the effectiveness of the proposed method. Finally, a brief conclusion is given in Section 7.

2. MULTIMODEL FRAMEWORK AND PROBLEM FORMULATION

Multimodel representation of complex nonlinear system generally involves interpolation of a number of submodels to form the global system (see Fig. 1). Considering a nonlinear system of the form

$$\begin{aligned} y(k) &= F(y(k-1), y(k-2), \dots, y(k-na), \\ &u(k-1), u(k-2), \dots, u(k-nb)), \end{aligned} \quad (1)$$

where $u(k)$ and $y(k)$ are the input and output of the system respectively at instant k . The integers nb and na are the time lag of the input and output, respectively. The multimodel representation of the system can be described by

$$y(k) = \sum_{i=1}^m f_i(x(k))\phi_i(k), \quad (2)$$

where m is the number of submodels and $f_i(\cdot)$ and $\phi_i(k)$ are the i th submodel and its validity (weight), respectively. The validity describes the contribution of each submodel to the observed output and allows smooth transition between the local models when the system moves from one operating point to another. For easy interpretation, the validity satisfies the convexity property (Shorten et al., 1999; Orjuela et al., 2006):

$$\sum_{i=1}^m \phi_i(k) = 1, \quad \forall k; \quad (3)$$

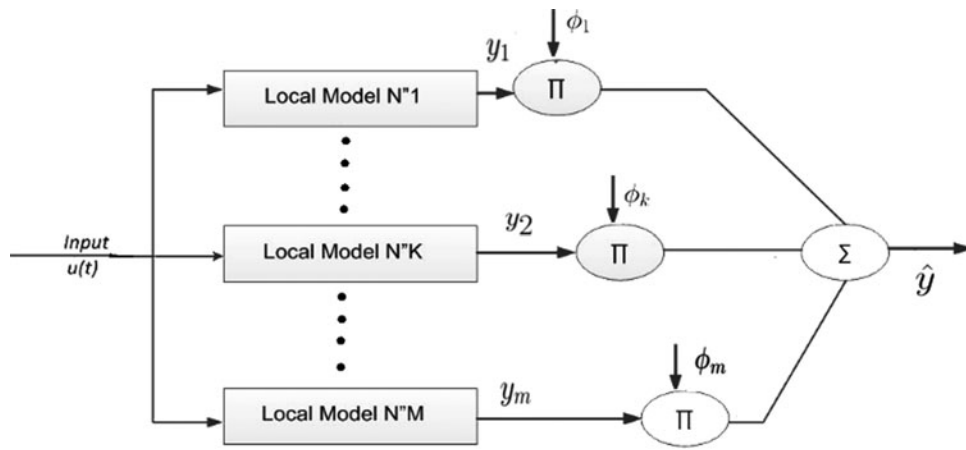


Fig. 1. Output blended multimodel identification structure.

$$0 \leq \varphi_i(k) \leq 1, \quad \forall k, \forall i \in 1, \dots, M. \tag{4}$$

Given a set of input–output data, the multimodel identification problem is to estimate the number of submodels (m), the parameters of each submodel ($f_i(\cdot)$), and the validity (φ_i) for each submodel.

Note that submodel $f_i(\cdot)$ can be linear, nonlinear, or a combination of the two submodels. In this study, linear submodels are considered. Therefore, the function $f_i(x(k))$ can be written as

$$f_i(x(k)) = x_i(k)\theta_i^T(k), \tag{5}$$

where θ_i is the vector of parameters of i th the submodel, which can be estimated from the data pairs:

$$z(k) = \{x(k), y(k) : k = 1, \dots, N\}, \tag{6}$$

where

$$x(k) = [y(k-1), y(k-2), \dots, y(k-na), u(k-1), u(k-2), \dots, u(k-nb)]$$

is the regressor vector.

The multimodel identification of complex nonlinear systems from a finite set of input–output data is quite involving as the submodels identification is depends on the data partition. In order to solve the aforementioned problem, a three-stage algorithm is proposed:

1. *Obtaining the number of submodels and the initial submodel:* This stage involves application of RCPSO to obtain m number of partitions and the representative data sets for each partition. The m number of clusters obtained from RCPSO is then used to estimate the initial submodels and initial cluster centers.
2. *Obtain the final submodels:* In this stage hybrid a K-means criterion is applied to the result of the previous stage to refine the submodels.

3. *Interpolation of the submodels:* Finally, the validity of each submodel is estimated to realize the multimodel output.

The details of these stages are provided in the subsequent sections.

3. STAGE 1: OBTAINING THE NUMBER OF SUBMODELS AND THE INITIAL PARTITION

The aim of this stage is to determine the number of partitions and to evolve a partition representing a possible grouping of the given data set. That is, given a data set $Z = [z_1, z_2, \dots, z_N]$ in \mathcal{R}^d , that is, N points each with d dimension ($d = na + nb$), we need to simultaneously find the number of partition (m) and divide Z into m exhaustive and mutually exclusive clusters $C = [c_1, c_2, \dots, c_m]$ with respect to a predefined criteria such that

1. $c_i \neq \emptyset, i = 1, \dots, m;$
2. $c_i \cap c_l = \emptyset, i = 1, \dots, m, i \neq l;$
3. $\cup_{i=1}^m c_i = Z.$

To achieve this objective with partition algorithm, a modification to combinatorial PSO (CPSO; Jarboui et al., 2007) is proposed. The modification is necessary because the CPSO algorithm required the number of clusters to be fixed a prior. In what follows, particle swarm optimization, CPSO, and its modification for the determination of the initial submodels are described.

3.1. PSO

PSO (Kennedy & Eberhart, 1995) is a metaheuristic search algorithm, mimicking the movement of organisms in a bird flock or fish school. Due to its simple concept and fast convergence, PSO has attracted much attention and wide applications in various fields, including systems identification problems (e.g., Modares et al., 2010; Tang et al., 2010; Majhi &

Panda, 2011; Cornoiu et al., 2013; Hu et al., 2014; Zeng et al., 2014; Kramar et al., 2015). PSO combines self and social experience for directing search to an optimal solution. The position of individuals referred to as particles is influenced by its best position ever attained (called Pbest), the position of the best in the swarm (Gbest), and the current velocity V_i that drives the particles. At each generation t , each particle i adjusts its velocity V_{ij}^t and position p_{ij}^t for each dimension j by referring to its personal best, Pbest, and the global best, Gbest. The following equations are used by the original PSO for velocity and particle updates:

$$V_{ij}^{t+1} = \omega V_{ij}^t + c_1 r_1 (Pbest_{ij}^t - p_{ij}^t) + c_2 r_2 (Gbest^t - p_{ij}^t), \quad (7)$$

$$p_{ij}^{t+1} = xp_{ij}^t + V_{ij}^{t+1}, \quad (8)$$

where V_{ij}^t and p_{ij}^t are the j th element of the i th velocity and particle vector, respectively, at generation t ; $Pbest_{ij}^t$ is the personal best position of the i th particle and $Gbest^t$ is the global best during iterations 1 to t ; ω is the inertia weight that controls the impact of the previous velocities on the current velocity; r_1 and r_2 are uniformly distributed random variables in range $[0, 1]$; and c_1 and c_2 are the acceleration constants.

3.2. CPSO-based partition

CPSO (Jarboui et al., 2007) is an extension to the original PSO to be able to cope with the clustering problem. It has a similar procedure as the original PSO except that it differs in two characteristics: particle and velocity representation. In CPSO, particles P_i are encoded with label-based representation, where $P_i = [p_{i1}, p_{i2}, \dots, p_{iN}]$ provides integer numbers representing the cluster's number of data points, such that $p_{ij} \in \{1, 2, \dots, m\}$, where m is the number of clusters. The velocity of each particle uses a dummy variable that permits transition from combinatorial to continuous state and vice versa. Thus, the velocity and particle are updated through the following equations:

$$v_{ij}^{t+1} = \omega v_{ij}^t + r_1 c_1 (-1 - y_{ij}^t) + r_2 c_2 (1 - y_{ij}^t), \quad (9)$$

where y_{ij}^t is a dummy variable defined by

$$y_{ij}^t = \begin{cases} 1 & \text{if } p_{ij}^t = Gbest_j^t \\ -1 & \text{if } p_{ij}^t = Pbest_j^t \\ -1 \text{ or } 1 & \text{randomly if } p_{ij}^t = Pbest_j^t = Gbest_j^t \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

After velocity update, the position of each particle is updated through the dummy variable according to the following equations:

$$\lambda_{ij}^{t+1} = y_{ij}^t + v_{ij}^{t+1}, \quad (11)$$

$$y_{ij}^t = \begin{cases} 1 & \text{if } \lambda_{ij}^t > \alpha \\ -1 & \text{if } \lambda_{ij}^t < -\alpha \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

$$p_{ij}^t = \begin{cases} Gbest_j^t & \text{if } y_{ij}^t = 1 \\ Pbest_j^t & \text{if } y_{ij}^t = -1, \\ \text{randomly selected} & \text{otherwise} \end{cases} \quad (13)$$

where α is a design parameter determined by the user.

3.3. RCPSO-based partition

This section introduces the RCPSO and how it is used to determine the number of partitions and the initial partition. As mentioned earlier, the CPSO algorithm is reinforced as, contrary to our case, the number of partitions in the algorithm is fixed *a priori*. In addition, the CPSO objective is not adequate for use in submodel identification, which is of paramount importance. The following four new features distinguish RCPSO from CPSO:

1. *Particles encoding*: Similar to CPSO, RCPSO uses the label-based integer encoding to represent each particle. However, instead of assigning the same number of clusters to all particles, in RCPSO each particle evolve with its own number of clusters at each iteration. Each particle position $P_i = [p_{i1}, p_{i2}, \dots, p_{iN}]$, characterized by N elements, where N is the number of data points, provides integer numbers representing the cluster number of each data point, such that $p_{ij} \in \{1, 2, \dots, m_i\}$ represents the cluster number of the j th data point in the i th particle and m_i is the number of clusters associated with the i th particle. In addition, m_i is assumed to lie in the range $[m_{\min}, m_{\max}]$, where m_{\min} is 2 by default and m_{\max} is manually specified. The particle and velocity updates follow that of CPSO [Eqs. (9)–(13)].
2. *Avoiding empty clusters*: In the label-based representation, it is possible to generate a solution with an empty cluster if the number of its clusters is smaller than the largest cluster number associated with the particle solution. To avoid this, new positions of particles are checked. At each generation, particles with an empty cluster are corrected by changing the largest cluster number to the smallest unused cluster number.
3. *Fitness function*: The fitness criterion used in CPSO are the variance ratio criterion (VRC) and sum of square error of the cluster. The sum of square error is not appropriate when the number of clusters is not known in advance. This is because the maximum number of clusters will always be favored because its value will decrease as the number of cluster increases. In contrast, VRC has been used when the number of clusters is not known. However, in RCPSO, the use of a fitness function based on cluster regression error fused in minimum descriptive length reflects the peculiar nature of the problem at hand. Cluster regression error criterion guides toward a linear submodel in each cluster, while the minimum descriptive length dis-

courage large number of clusters. Thus, unlike VRC, the peculiarity of the problem at hand resides in our objective to use a linear model for each cluster while minimizing the number of unknown clusters.

Given the data set Z defined in Eq. (6), the cluster regression error (CRE) is defined by

$$CRE = \sum_{i=1}^{m'_{\max}} SE, \tag{14}$$

where

$$SE = \sqrt{\left(\frac{1}{n_i} \sum_{j=1}^{n_i} (y_j - x_j \theta_i^T)^2\right)}. \tag{15}$$

In Eqs. (14) and (15), m'_{\max} is the maximum number of clusters assigned to a solution, n_i is the number of data points in the i th cluster, and θ_i is the parameter of the linear model associated with the i th cluster. This can be ob-

tained using the least square technique as follows:

$$\theta_i = \left[\sum_{j=1}^{n_i} x_j x_j^T \right]^{-1} \left[\sum_{j=1}^{n_i} y_j x_j \right]. \tag{16}$$

Finally, the fitness function is defined by

$$\text{fitness} = \frac{m'_{\max} \log(N) + N \log(CRE^2)}{2}, \tag{17}$$

where N is the total number of data points. The smaller the fitness value, the better the clustering solution.

4. *Avoiding small size data:* A situation may occur where the number of data points assigned to a cluster is too small. On the one hand, if the number of data points n_i is less than the dimension of the data, $d = na + nb$, then the model obtained from the cluster will be singular. On the other hand, if $n_i \geq d$ but $n_i < td$, where td signifies a reasonable minimum number of data points (i.e., 5% of data points), then the model obtained may not be well defined. In order to avoid these two situa-

```

1: Initialize  $c_i = \tilde{c}_i, \theta_i = \tilde{\theta}_i, i = 1, \dots, m, l = 1$ 
2: repeat
3:   Assign data to a cluster such that

$$u_{ij} = \begin{cases} 1 & \text{if } (y_j - \tilde{x}_j \theta_i^T)^2 + \lambda(\tilde{x}_j - c_i)^2 \\ & \leq (y_j - \tilde{x}_j \theta_k^T)^2 + \lambda(\tilde{x}_j - c_k)^2, \quad i \neq k \\ 0 & \text{otherwise} \end{cases}$$

4:   compute center  $\theta_i$  and  $c_i$ 

$$\theta_i = \left[ \sum_{x_j \in u_i} x_j x_j^T \right]^{-1} \left[ \sum_{x_j \in u_i} x_j y_j \right], \quad i = 1, \dots, m$$


$$c_i = \frac{1}{|U_i|} \sum_{x_j \in u_i} x_j, \quad i = 1, \dots, m, \text{ and } |U_i| = \sum_{j=1}^{n_i} u_{ij}$$

5:   Remove undefined cluster as follows
6:   for  $i = 1 \dots m$  do
7:     if  $n_i < na + nb$  then
8:        $c_i = \emptyset$ 
9:        $\theta_i = \emptyset$ 
10:       $m = m - 1$ 
11:      break
12:    end if
13:  end for
14:  Compute  $J^l$ 
15:   $l = l + 1$ 
16: until  $\|J^{l-1} - J^l\| \leq \epsilon$ 

```

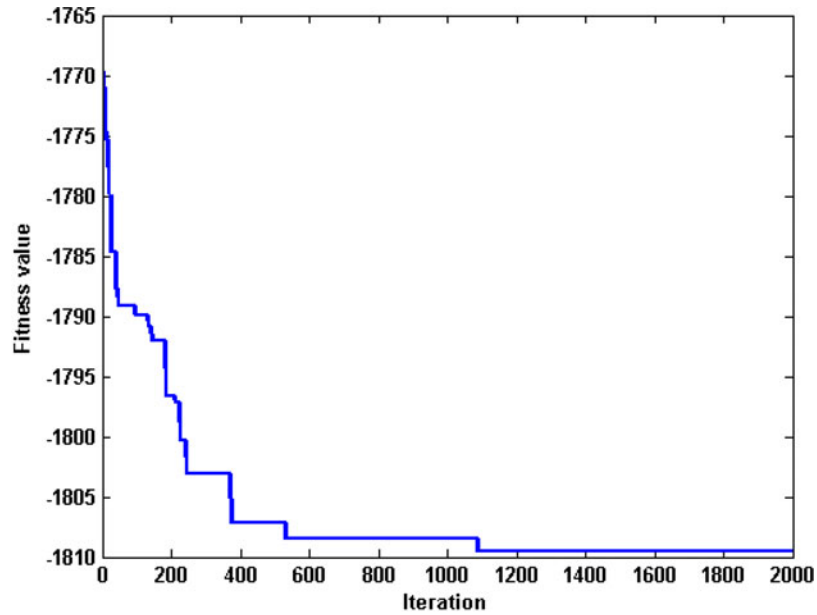
Fig. 2. Stage 2: estimating final submodels.

Table 1. Reinforced combinatorial particle swarm optimization parameter settings

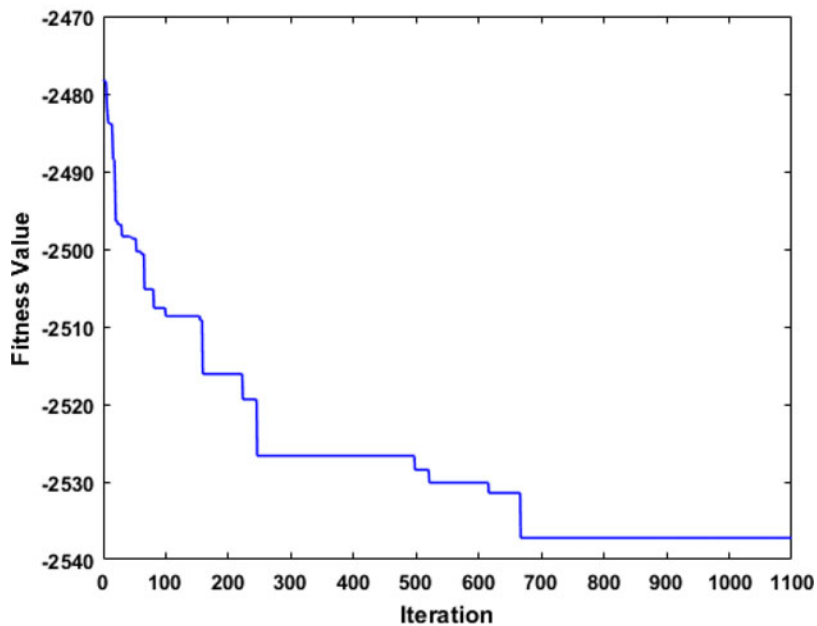
Parameters	Values
Swarm size	20
Max. iterations	2000
w, α	0.4, 0.35
$c1, c2$	2, 2
$mmin, mmax$	2, 20

tions, the SE in Eq. (15) is not calculated from the data when $n_i < 0.05N$; instead, a penalty value is assigned to the SE to discourage having a small number of data points in a cluster. This penalty value can be an arbitrary value higher than the SE value for an acceptable number of data points. As such, a penalty value of 1000 has been chosen in all simulation examples.

After the encoding of the particles as discussed above, the execution of RCPSO to obtain the number of clusters and the initial partition is done according to the following steps:



(a) Two-parameter structure



(b) Four-parameter structure

Fig. 3. Reinforced combinatorial particle swarm optimization objective function convergence plot.

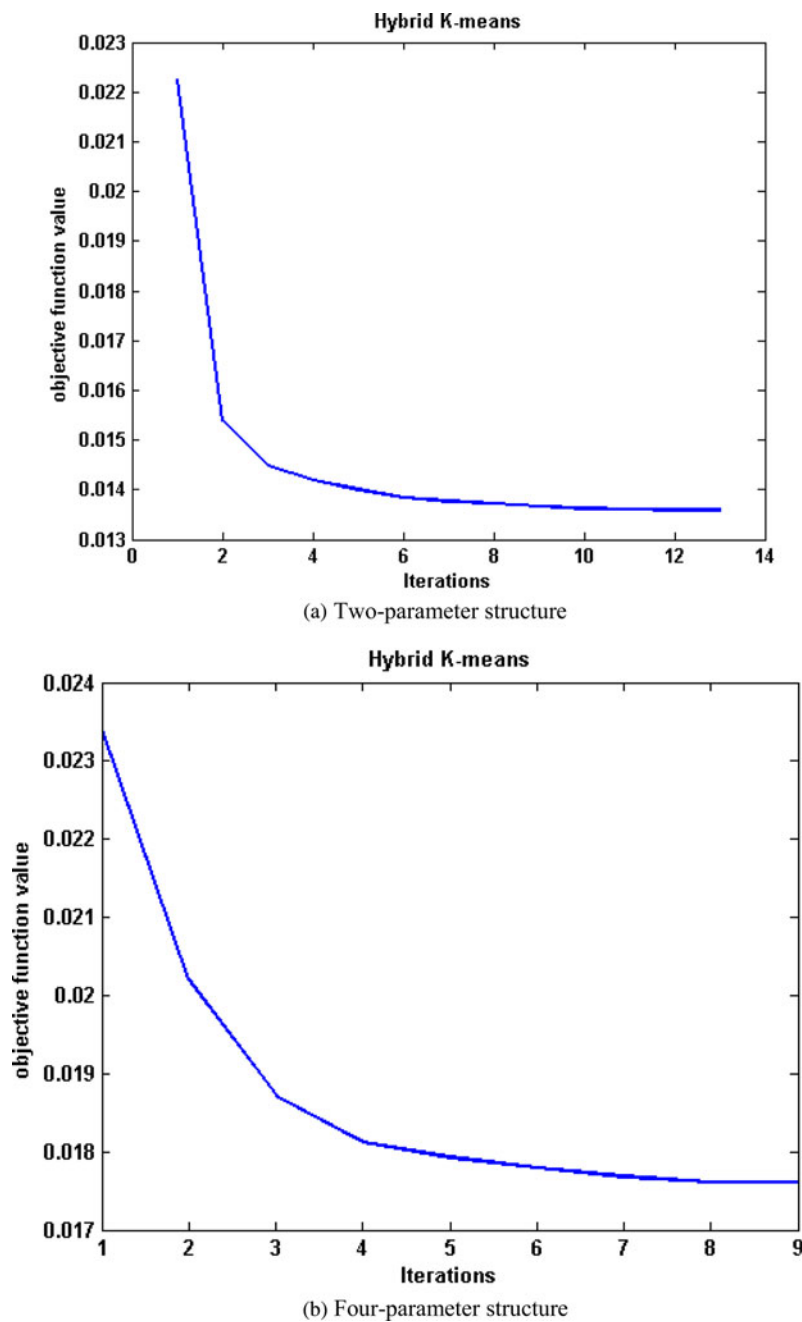


Fig. 4. Hybrid K-means objective function convergence plot.

STEP 1. Initialize particle position vector X and associated velocity V in the population randomly. For each particle i , the number of clusters m_i is randomly generated in the range $[m_{\min}, m_{\max}]$, then each data point is randomly assigned to one cluster.

STEP 2. Evaluate the fitness function for each particle using Eq. (17).

STEP 3. Compare the fitness value of each particle with its previous best solution (Pbest) fitness and update Pbest with the current solution if it is better than the previous value (Pbest).

STEP 4. Compare the fitness value with the overall previous best (Gbest) fitness. Update Gbest to the current solution if its fitness value is better than Gbest fitness value.

STEP 5. Update positions and velocities of particles using Eqs. (9) to (13).

STEP 6. Check for empty clusters in all particle solutions and correct if they exist.

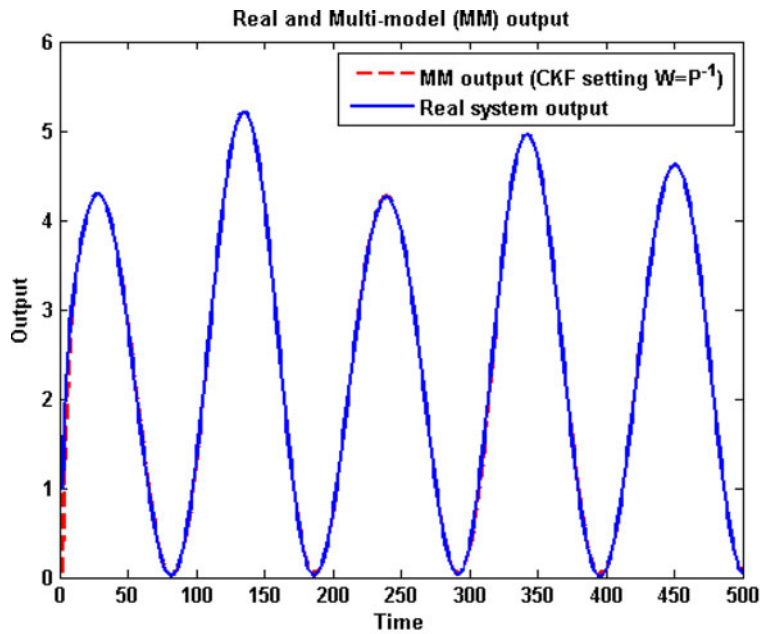
STEP 7. Repeat Step 2 to Step 6 until the maximum number of iterations is completed.

Table 2. Performance of proposed multimodel based on the initial submodels

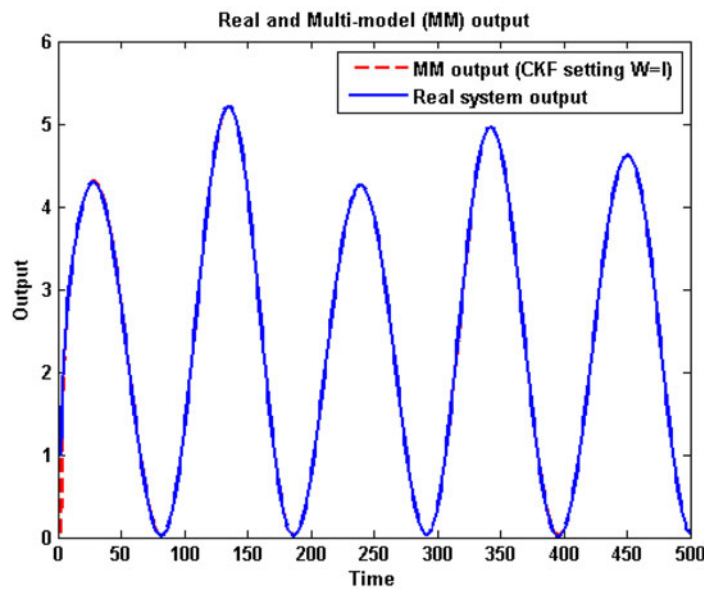
CKF Setting	Two-Parameter Structures		Four-Parameter Structures	
	MSE	VAF	MSE	VAF
$W = P^{-1}(k)$	0.0645	97.67%	0.0500	98.20%
$W = I$	0.0347	98.78%	0.0499	98.20%

Table 3. Performance of proposed multimodel based on final submodels

CKF Setting	Two-Parameter Structures		Four-Parameter Structures	
	MSE	VAF	MSE	VAF
$W = P^{-1}(k)$	0.0075	99.73%	0.0150	99.46%
$W = I$	0.0053	99.81%	0.0151	99.46%

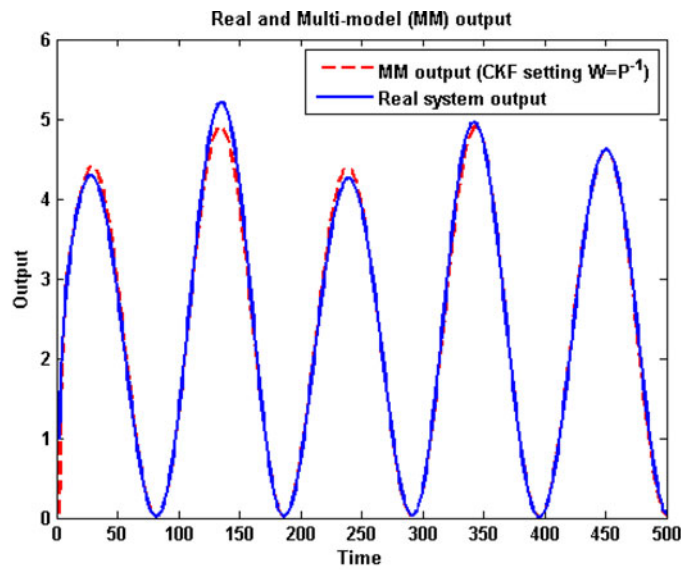


(a) Two-parameter structure with CKF $W = P^{-1}(k)$

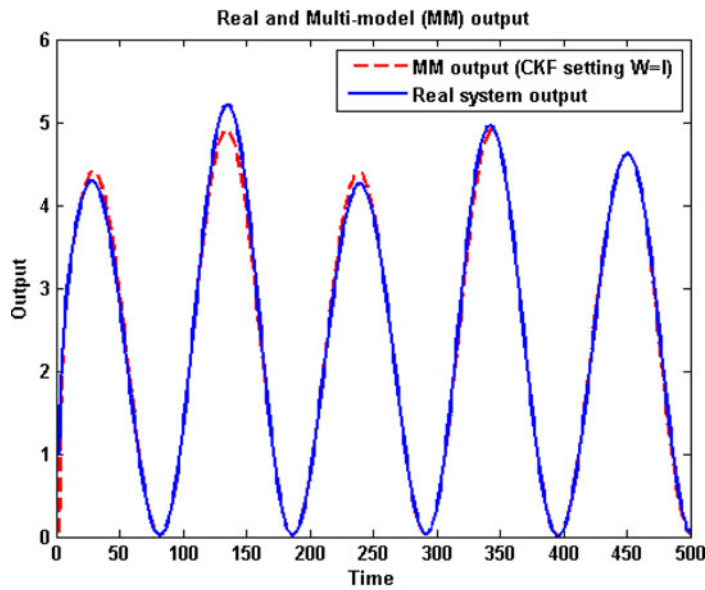


(b) Two-parameter structure with CKF $W = I$

Fig. 5. Multimodel identification outputs using validation data.



(c) Four-parameter structure with CKF $W = P^{-1}(k)$



(d) Four-parameter structure with CKF $W = I$

Fig. 5 (cont.)

3.4. Estimation of the initial submodels

Given a set of cluster representatives

$$\tilde{z} = \{\tilde{z}_i = (\tilde{x}_i, \tilde{y}_i), \quad i = 1, \dots, m\},$$

from the RCPSO algorithm, where m is the number of clusters, the next task is to estimate the initial submodels. For this purpose, a least square estimation is applied to the data set in each cluster to find the initial submodel. The coefficients vector θ_i for each submodel is computed through the formula

$$\tilde{\theta}_i = (\tilde{\Phi}_i^T \tilde{\Phi}_i)^{-1} \tilde{\Phi}_i^T \tilde{y}_i, \tag{18}$$

where $\tilde{\Phi}_i = [\tilde{x}_i(1), \dots, \tilde{x}_i(n_i)]^T$ and \tilde{y}_i are the regression matrix and output vector belonging to the i th cluster, respectively and n_i is the number data in the i th partition.

In addition, the centers of the data are calculated by finding the mean of the data in each cluster produced by the previous stage. The center of each cluster is given as

$$\tilde{c}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \tilde{x}_{ij} \quad i = 1, 2, \dots, m. \tag{19}$$

4. STAGE 2: OBTAIN THE FINAL SUBMODELS

This stage involves a refinement to the submodels produced in the previous stage. In order to achieve this objective, a hy-

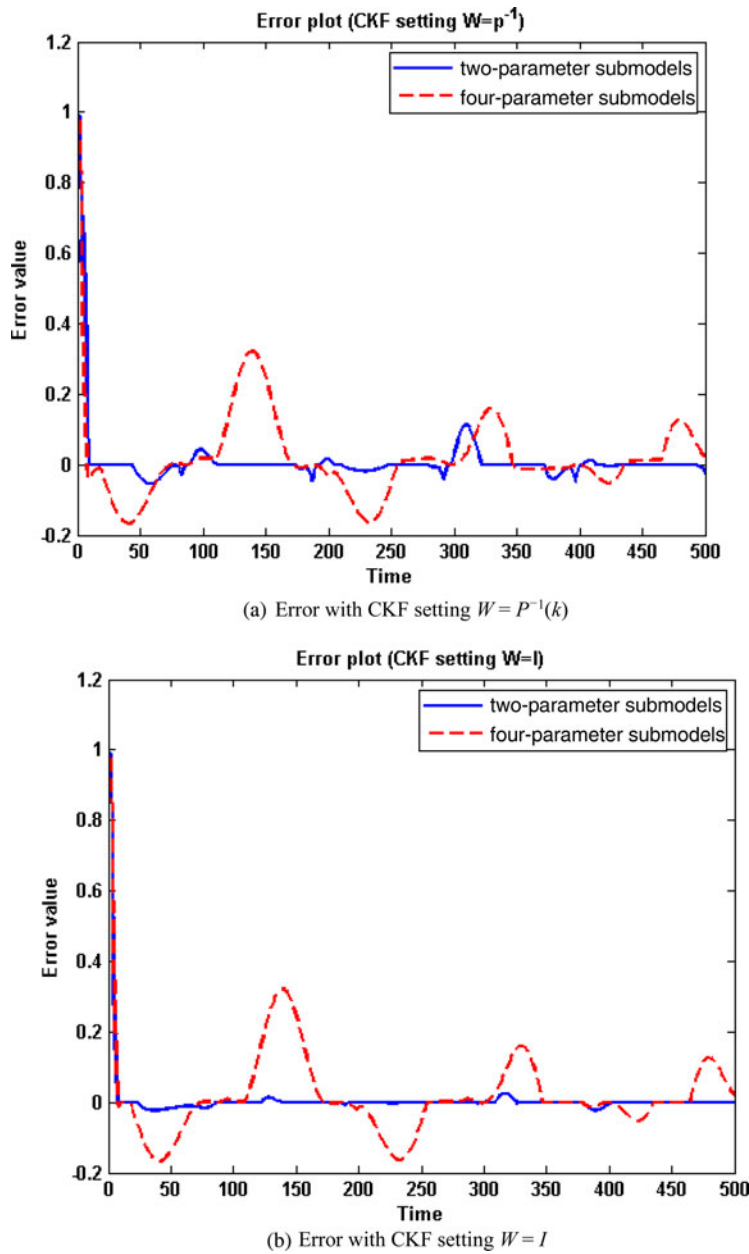


Fig. 6. Multimodel identification error using validation data.

brid K-means criterion (Stanforth et al., 2007) is adopted. Given a data set $Z = [z_1, z_2, \dots, z_N]$ in \mathcal{R}^d , K-means algorithm group the data Z into k clusters such that an objective function is minimized. The K-means objective function is defined as the sum of the square error between each data point and the corresponding cluster center:

$$J_1(c) = \sum_{i=1}^m \sum_{z_j \in u_i} (z_j - c_i)^2, \quad (20)$$

where z_i and c_i are the data point and cluster center, respectively. The objective function is minimized using an alternat-

ing optimization procedure. It starts with an arbitrary k centers and assigns each data point to the nearest center. The assignment of each data point is defined by a binary membership matrix U , such that

$$u_{ij} = \begin{cases} 1 & \text{if } (z_j - c_i)^2 \leq (z_j - c_k)^2, \quad i \neq k \\ 0 & \text{otherwise} \end{cases}. \quad (21)$$

Each center is updated as the mean of all points assigned to it. These two steps are carried out iteratively until a predefined termination criterion is met, which occurs when with RCPSO there is no change in the objective function.

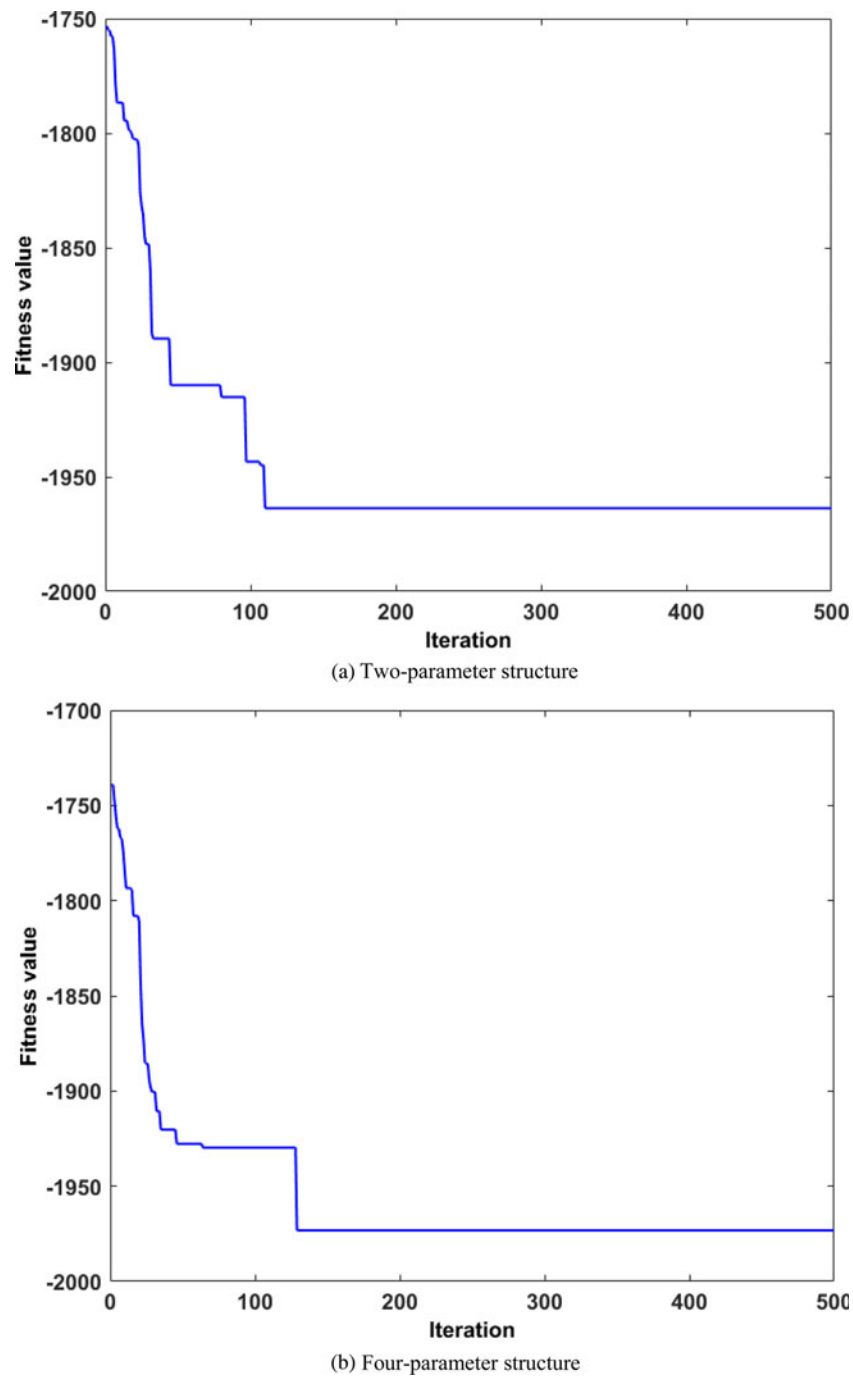


Fig. 7. Reinforced combinatorial particle swarm optimization objective function convergence plot.

In the same spirit of K-mean algorithm, the following linear regression loss function can also be formulated:

$$J_2(\theta) = \sum_{i=1}^m \sum_{\tilde{x}_j \in u_i} (y_j - \tilde{x}_j \theta_j^T)^2. \quad (22)$$

Thus, instead of minimizing the cluster error objective function, a linear regression objective function is minimized. Combining the two objectives J_1 and J_2 , the hybrid K-means

objective can be written as

$$J(\theta, c) = \lambda J_1(c) + J_2(\theta), \quad (23)$$

where $\lambda \in [0, 1]$ is a constant term to be defined by the user to specify a relative weight of the objective function. This formulation allows not only partitioning of the data set but also associating a submodel to each partition. In addition, the partitions formed are guided toward linear regions. This fits perfectly into

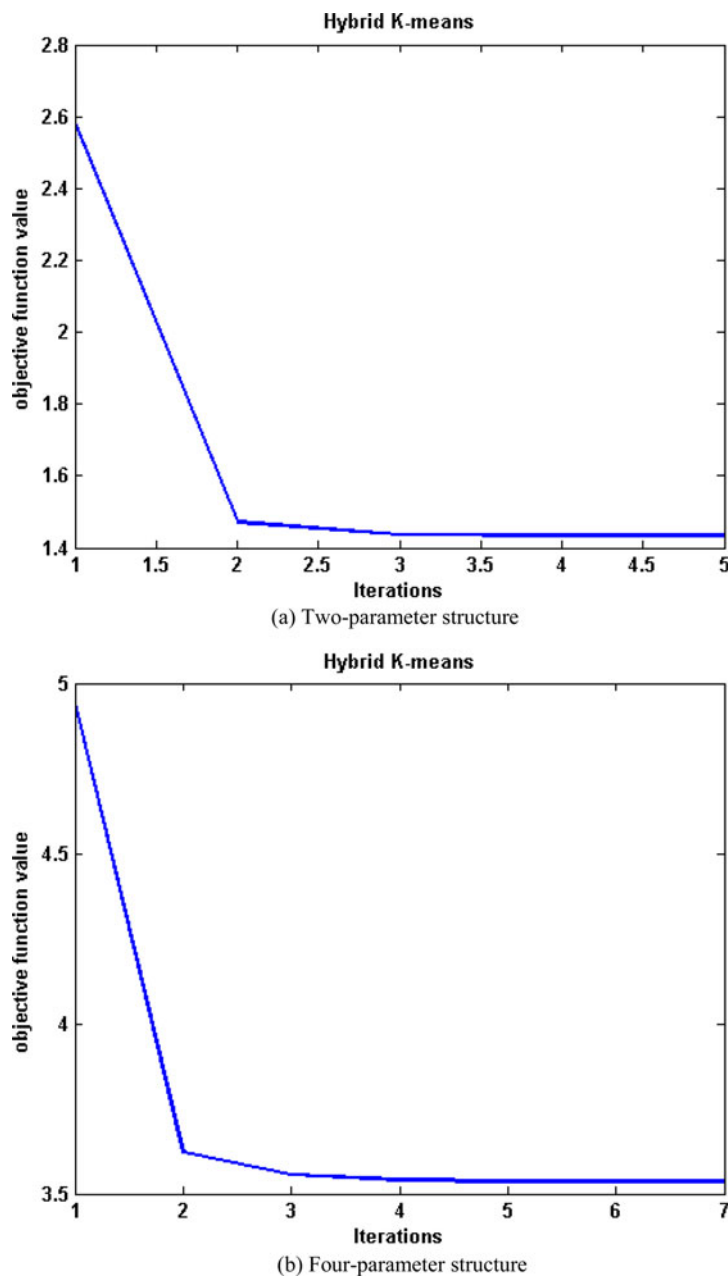


Fig. 8. Hybrid K-means objective function convergence plot.

the problem definition, as the aim of partitioning the input space is to form a linear submodel for each partition. Furthermore, the inclusion of J_1 will allow us to assign new data to a partition in a situation where the submodels need to be updated online.

The description of how this stage utilizes the hybrid K-means algorithm is shown in Figure 2. It begins with using the previously estimated cluster centers and associated model parameters as initialization of the algorithm. This eliminates the burden of the determination of the number of clusters, diminishes the effect of initialization, and increases the convergence rate of the K-means algorithm. Line 2 starts a loop that repeats itself for as long as there is a significant change in the objective func-

tion. It begins by determining the membership matrix U by Eq. (21), which assign each data point to a cluster. Line 4 estimates the parameter vector θ_i and the center c_i for each cluster. Next, from Lines 5 to 13, undefined clusters are detected and removed from subsequent updates. An undefined cluster is characterized by singular cluster, which may result when the size of the cluster falls below the number of regressors.

Lines 8 to 13 remove a cluster from the pool if the number of data points in the cluster is less than the number of regressors. In addition, the number of cluster is reduced by 1. The break statement in Line 12 ensure only one cluster is removed at each iteration when an undefined cluster is detected. This is

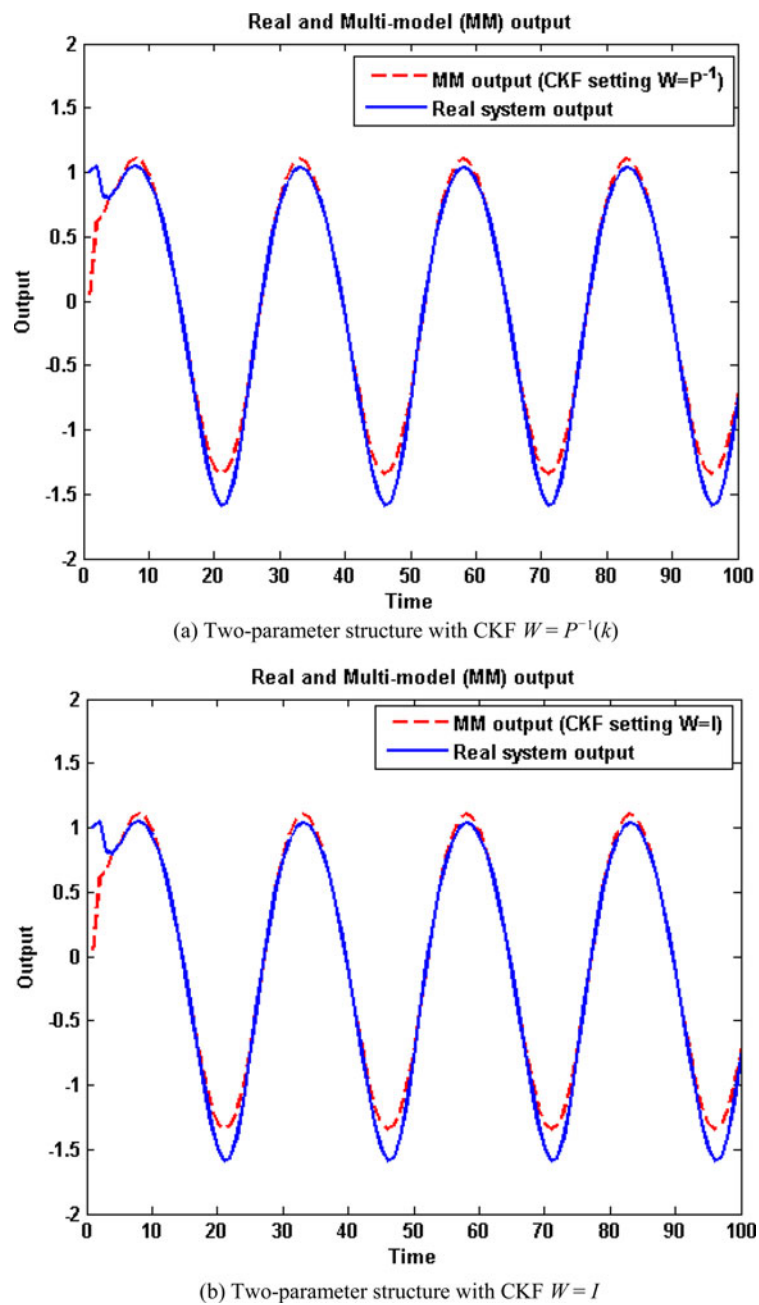


Fig. 9. Multimodal identification outputs using validation data.

to allows other undefined clusters, if they exist, to readjust during the next iteration and probably to circumvent undefined status. Line 15 computes the objective function J according to Eq. (23) while Line 16 increments the number of iteration. Next, the loop goes back to Line 3 to repeat the procedure.

Once the algorithm is completed, the final parameters θ of each submodel are obtained along with their associated centroid c_i . The submodels are now ready for interpolation in the next stage to obtain the final global model that will represent the system under consideration.

5. STAGE 3: INTERPOLATION OF THE SUBMODELS

This is the final stage where the multimodel output is realized by combining the estimated final submodels obtained in the previous stage (Fig. 2). To achieve this, the validity of each submodel has to be estimated. In the literature, different realizations exist to combine the submodels along with their validity. In this study, a weighted sum of the output of the submodels is used to form the output of the multimodel, where the weight signifies the validity of each submodel.

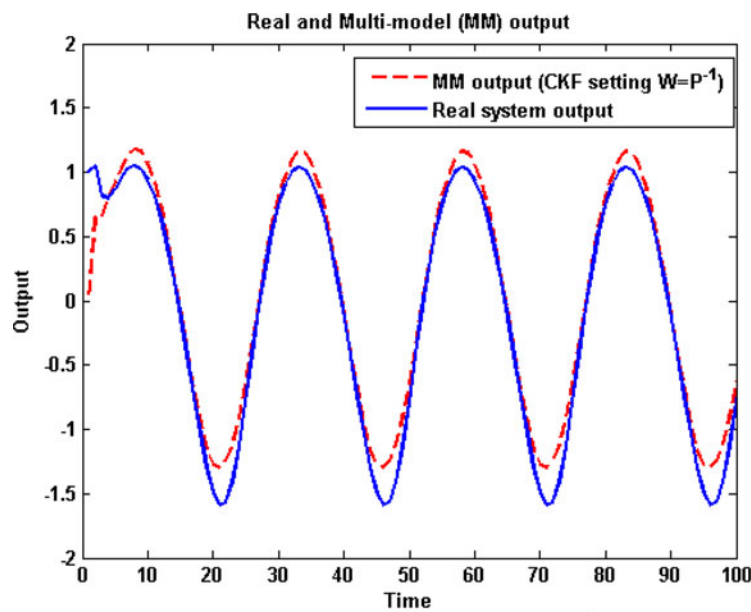
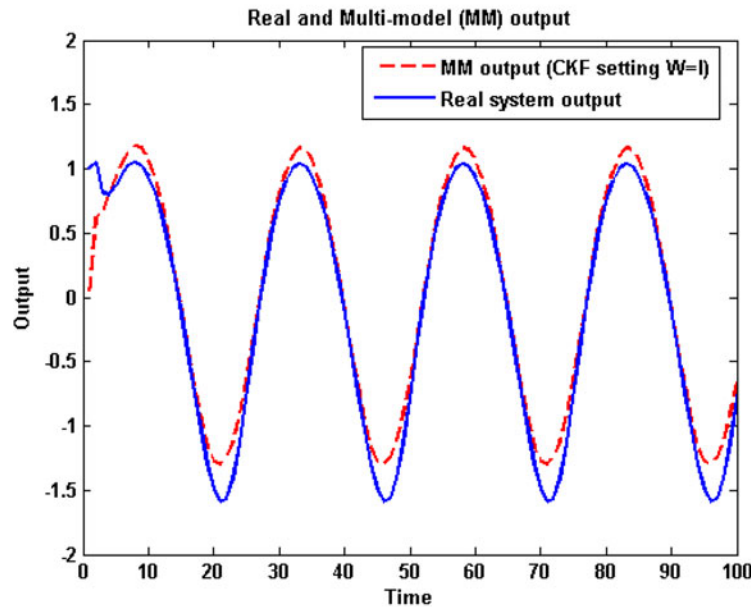
(c) Four-parameter structure with CKF $W = P^{-1}(k)$ (d) Four-parameter structure with CKF $W = I$

Fig. 9 (cont.)

Among many possible candidates for validity estimation, constrained Kalman filter (CKF) validity estimation is an effective technique introduced in our earlier work (Adeniran & Elferik, 2014). The CKF estimates the validity of each submodel at every time instant while satisfying the convexity property in Eqs. (3) and (4). The final output of the system is computed using Eq. (2). Detail description of the CKF algorithm can be found in Appendix A.

6. SIMULATIONS

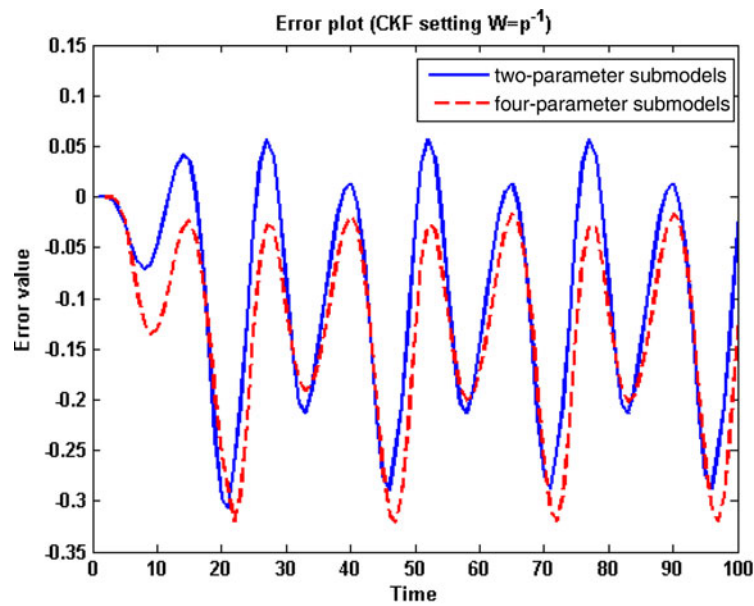
The effectiveness of the proposed multimodel approach is demonstrated in this section. Four simulation examples

were carried out. To illustrate flexibility of the proposed method on the number of parameters assumed for the submodels, two- and four-parameter submodel structures are examined. The two- and four-parameter submodel structures, subsequently refer to as two-parameter and four-parameter structures, respectively, are described by

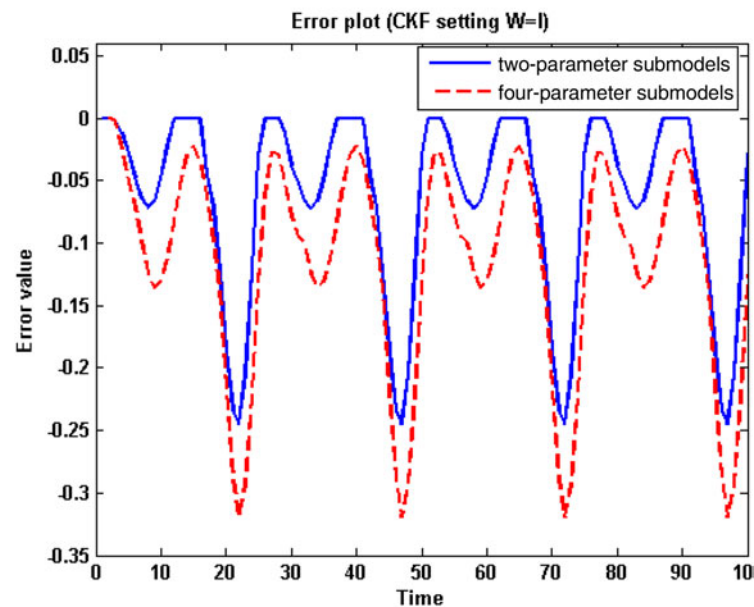
$$y_i(k) = a_{i1}y_i(k-1) + b_{i1}u(k-1), \quad (24)$$

$$y_i(k) = a_{i1}y_i(k-1) + a_{i2}y_i(k-2) + b_{i1}u(k-1) + b_{i2}u(k-2), \quad (25)$$

where a_{i1} , a_{i2} , b_{i1} , b_{i2} are the i th submodel scalar parameters to be identified by Stage 1 and Stage 2. Unless it is stated



(a) Error with CKF setting $W = P^{-1}(k)$



(b) Error with CKF setting $W = I$

Fig. 10. Multimodal identification error using validation data.

Table 4. Performance of proposed multimodel based on final submodels

CKF Setting	Two-Parameter Structures		Four-Parameter Structures	
	MSE	VAF	MSE	VAF
$W = P^{-1}(k)$	0.0122	98.92%	0.0279	98.99%
$W = I$	0.0100	99.32%	0.0210	99.14%

Table 5. Performance of proposed multimodel based on initial submodels

CKF Setting	Two-Parameter Structures		Four-Parameter Structures	
	MSE	VAF	MSE	VAF
$W = P^{-1}(k)$	0.0317	98.63%	0.0315	98.58%
$W = I$	0.0300	98.58%	0.0295	98.61%

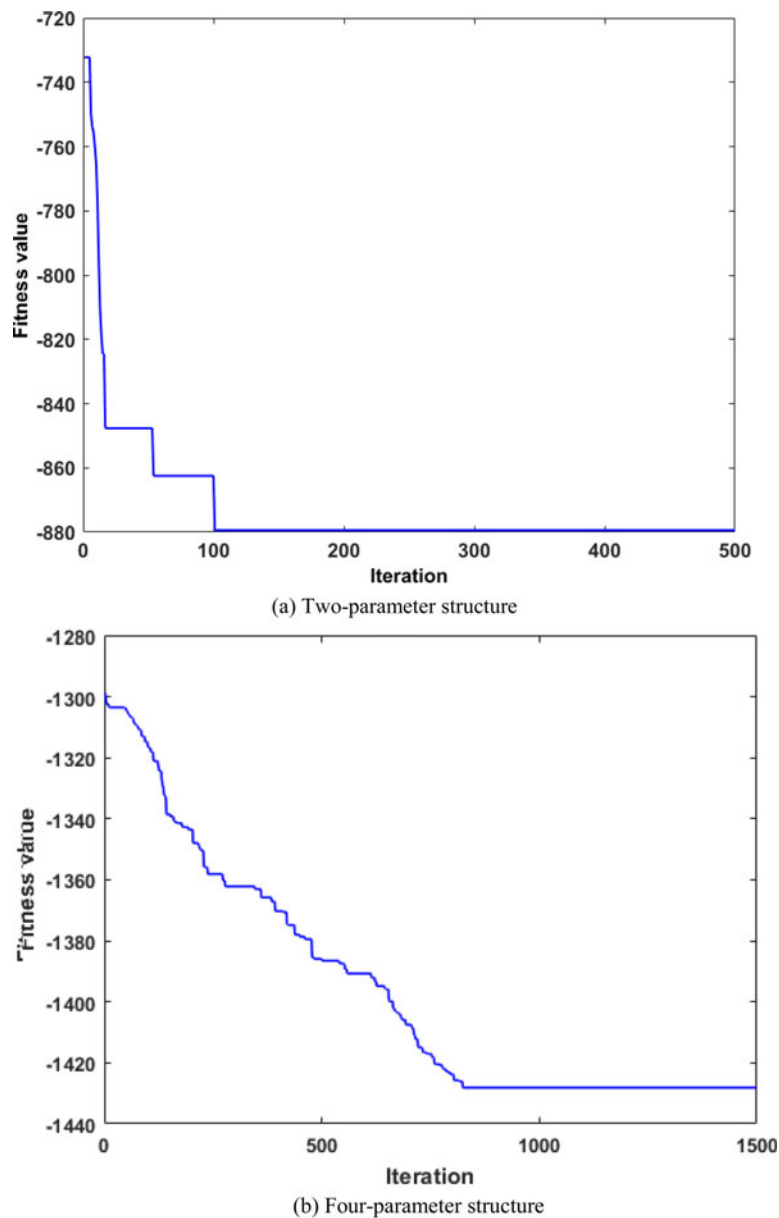


Fig. 11. Reinforced combinatorial particle swarm optimization objective function convergence plot.

otherwise, the parameter settings of RCPSO used are given in Table 1. Throughout the simulations, $\lambda = 0.01$ is used in Stage 2. In Stage 3, estimation of the validity using the CKF algorithm, setting $W = P^{-1}(k)$ and $W = I$, are both utilized, with the matrix P initialized as a diagonal matrix $100I$. The validity of each submodel, ϕ , is initialized randomly while satisfying the convexity property. The obtained multimodel is evaluated based on the validation data using the mean square error (MSE) and variance accounted for (VAF) performance measures:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y(i) - \hat{y}(i)), \quad (26)$$

$$\text{VAF} = \max\left(100 \times \left(1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)}\right), 0\right), \quad (27)$$

where y is the real system output and \hat{y} is the multimodel estimated output. All simulations are performed using MATLAB 2012b on a 2.4-GHz i3 64-bit Windows machine with 4 G of RAM.

6.1. Example 1

In the first example, a discrete-time system from Elfelly et al. (2010) is considered. The system is described by

$$y(k) = a_1(k)y(k-1) + a_2(k)y(k-2) + b_1(k)u(k-1) + b_2u(k-2).$$

The variation laws of different parameters of the process is

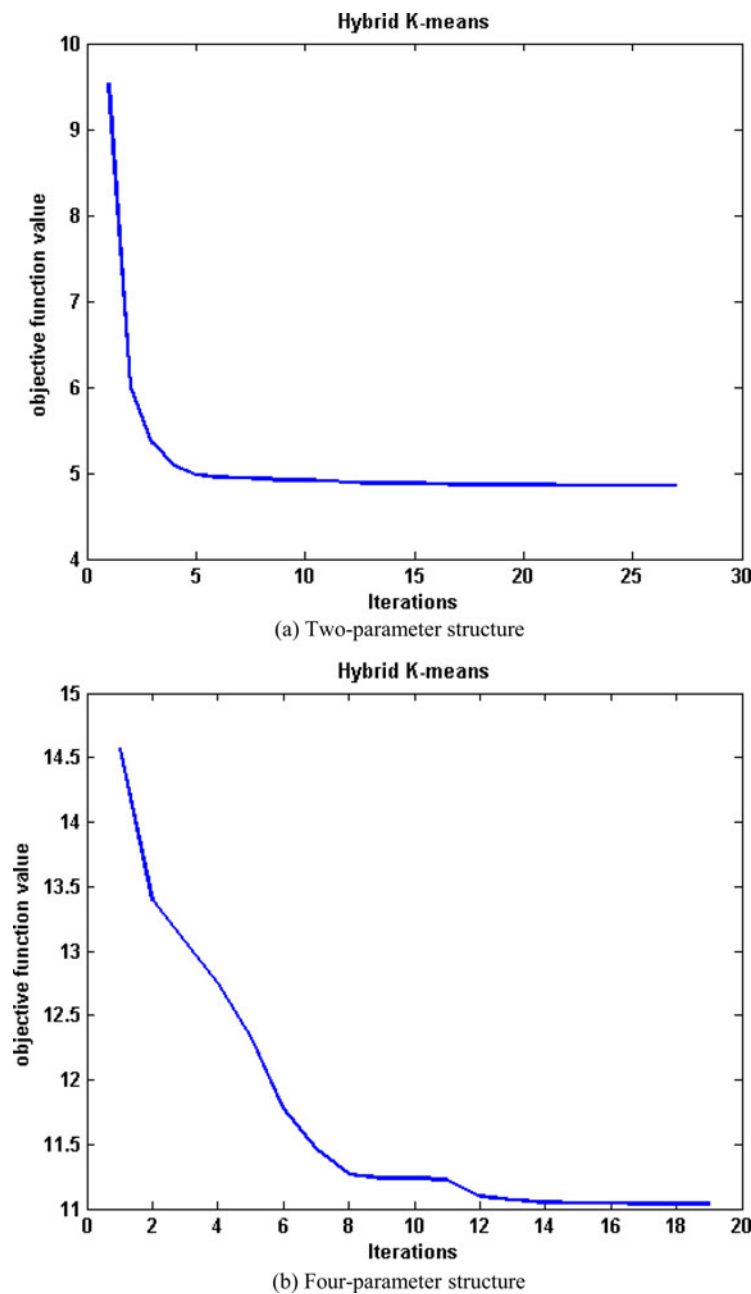


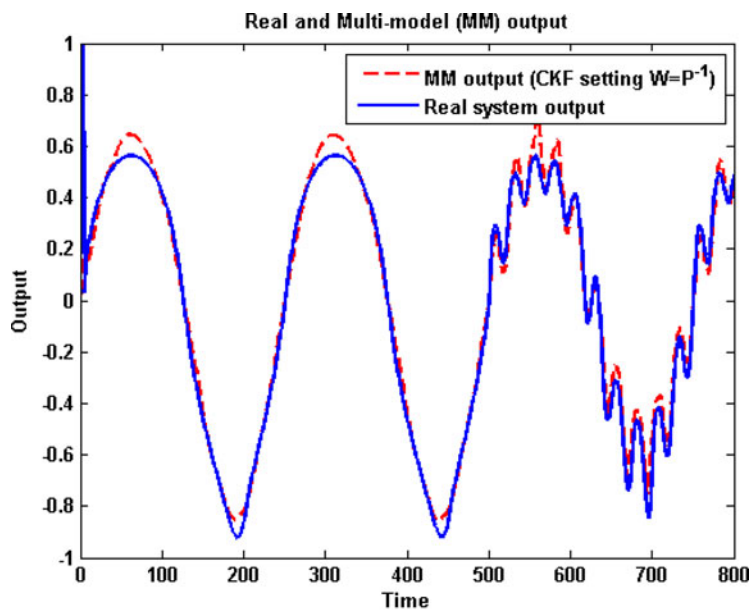
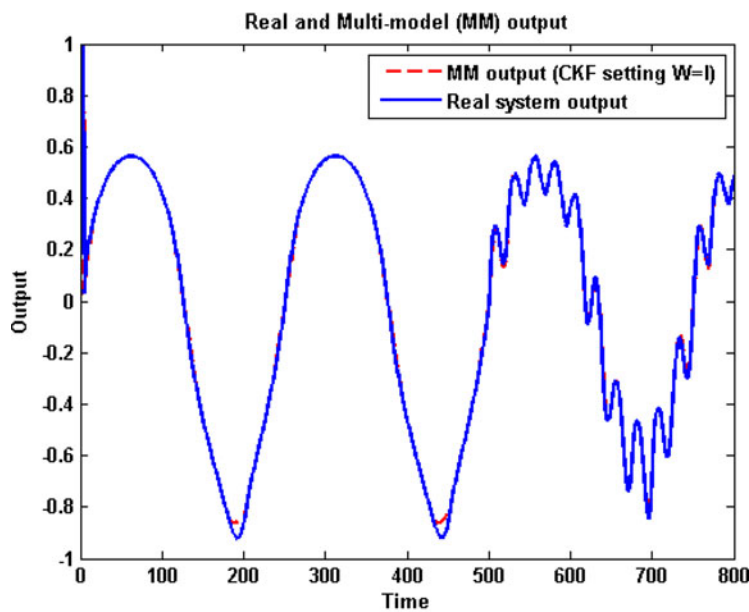
Fig. 12. Hybrid K-means objective function convergence plot.

given by

$$\begin{aligned}
 a_1(k) &= 0.04 \sin(0.035k) + 0.8, \\
 a_2(k) &= 0.005 \sin(0.03k) + 0.1, \\
 b_1(k) &= 0.02 \sin(0.03k) + 0.5, \\
 b_2(k) &= 0.01 \sin(0.035k) + 0.2.
 \end{aligned}$$

The system was excited with uniform random signal $u(k)$ on the range $[-1, 1]$. A total of 600 data points were generated for the two submodel structures above. The initial and final submodels obtained in Stage 1 and Stage 2 are shown in Table B.1 and Table B.2 in Appendix B, respectively. It

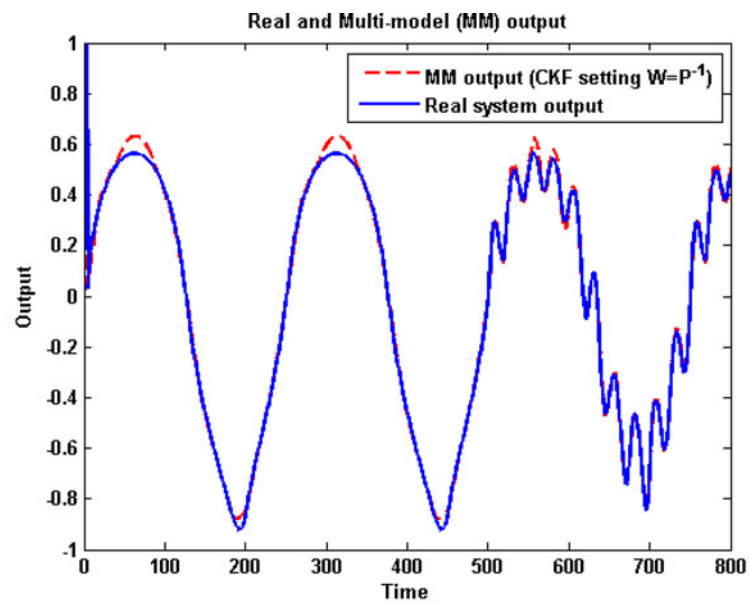
can be observed from the tables that four and two submodels were identified in Stage 1 for two-parameter and four-parameter structures, respectively. The convergence paths of the objective function in the developed RCPSO are shown in Figure 3. Although the figures show faster convergence for four-parameter structures compared to two-parameter structures, this is not always the case. It is possible that the algorithm started with particle positions closer to the final solution in the four-parameter structures than the two-parameter structure because of random initialization of the particle positions. The convergence of the hybrid K-means in Stage 2 is shown in Figure 4. Faster convergence can also be noticed

(a) Two-parameter structure with CKF $W = P^{-1}(k)$ (b) Two-parameter structure with CKF $W = I$ **Fig. 13.** Outputs of multimodel identification using validation data.

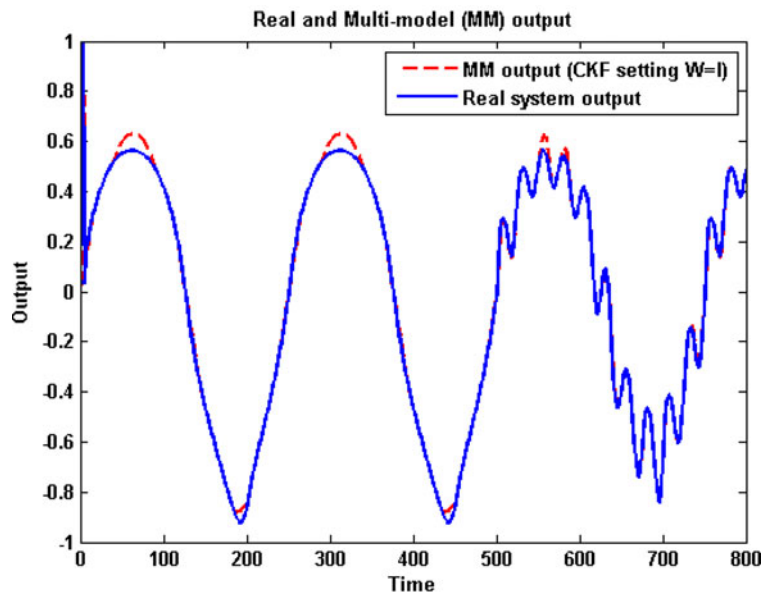
in the four-parameter structures than in the two-parameter structures.

In the validation stage, a different input signal [$u(k) = 1 + \sin(0.06k)$] was injected into the real systems and the identified submodels. The submodels were interpolated with CKF validity estimation as described in Stage 3 to form the multimodel output. The real system's output and the multimodel output for the two submodels' structures are compared using the MSE and VAF. To evaluate the significance of the refinement in Stage 2, the multimodel output is estimated based on the initial and final submodels obtained in Stage 1 and Stage 2, respectively.

The multimodel identification results are shown in Tables 2 and 3 for the combined final submodels and initial submodels, respectively. It can be observed that the performance of the proposed algorithm is better when all the final submodels are combined, and can well approximate the real system either with two-parameter or four-parameter structure using both CKF settings. However, the two-parameter structure shows better performance than the four-parameter structure. In addition, the CKF with settings $W = I$ has slightly better performance values in the two-parameter submodel structure. Furthermore, it can be pointed out that fewer parameters (8) have been achieved in comparison to the multimodel approach



(c) Four-parameter structure with CKF $W = P^{-1}(k)$



(d) Four-parameter structure with CKF $W = I$

Fig. 13 (cont.)

adopted in Elfelly et al. (2010; 12 parameters). Figures 5 and 6 show the comparison between the real system output and the multimodel output for the three-stage algorithm.

6.2. Example 2

A nonlinear dynamical system taken from Orjuela et al. (2013) is considered for multimodel identification:

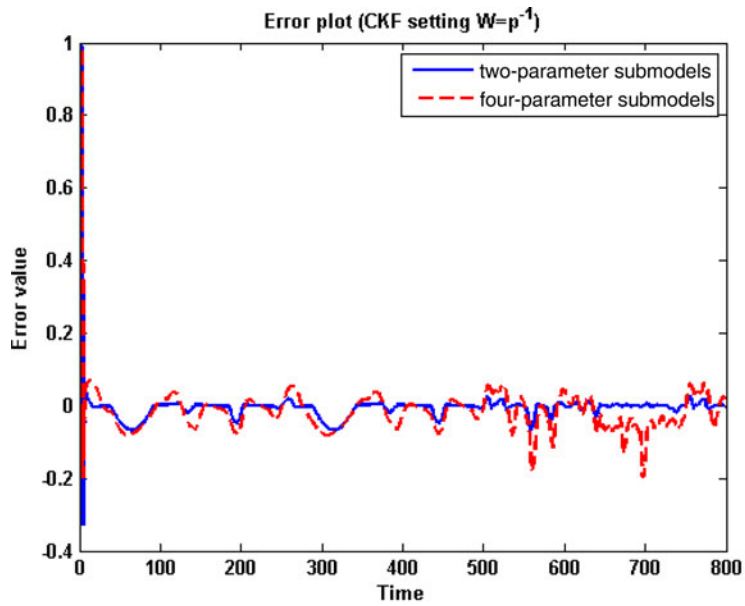
$$y(k + 1) = (0.6 - 0.1a(k))y(k) + a(k)u(k), \tag{28}$$

$$a(k) = \frac{6 - 0.006y(k)}{1 + 0.2y(k)}, \tag{29}$$

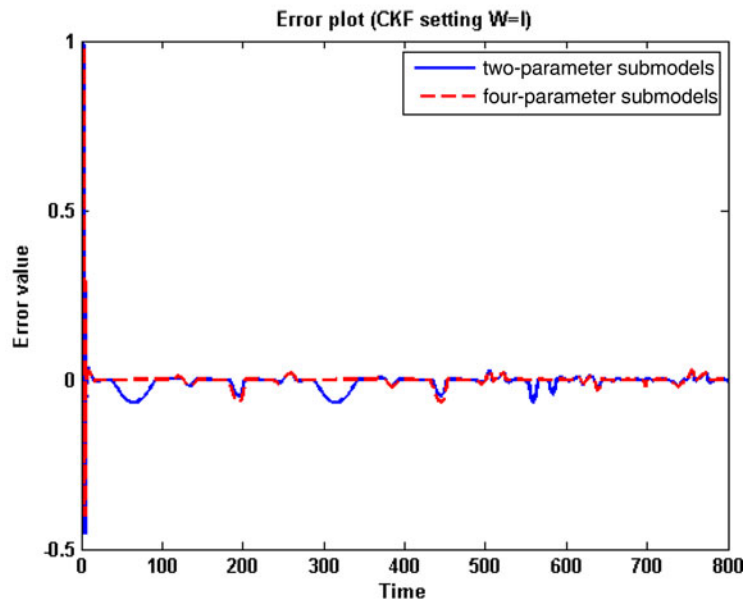
The identification was carried out using the proposed approach with a data set of 600 samples of uniform random signal within the range of $[-0.9, 0.9]$. Tables B.3 and B.4 in Appendix B show the initial and final submodels parameters with their associated centers.

The proposed multimodel approach identified two submodels for both two-parameter and four-parameter structures. Figures 7 and 8 show the convergence of the objective function in Stages 1 and 2, respectively.

To verify the accuracy of the proposed approach, the output of the nonlinear system was compared with the output of the proposed multimodel approach using the validation data un-



(a) Error with CKF setting $W = P^{-1}(k)$



(b) Error with CKF setting $W = I$

Fig. 14. Multimodel identification error using validation data.

Table 6. Performance of proposed multimodel based on final submodels

CKF Setting	Two-Parameter Structures		Four-Parameter Structures	
	MSE	VAF	MSE	VAF
$W = P^{-1}(k)$	0.0047	98.54	0.0012	99.55
$W = I$	0.0002	99.93	0.0005	99.84

Table 7. Performance of proposed multimodel based on initial submodels

CKF Setting	Two-Parameter Structures		Four-Parameter Structures	
	MSE	VAF	MSE	VAF
$W = P^{-1}(k)$	0.0061	98.27	0.0059	98.17
$W = I$	0.0034	98.92	0.0056	98.27

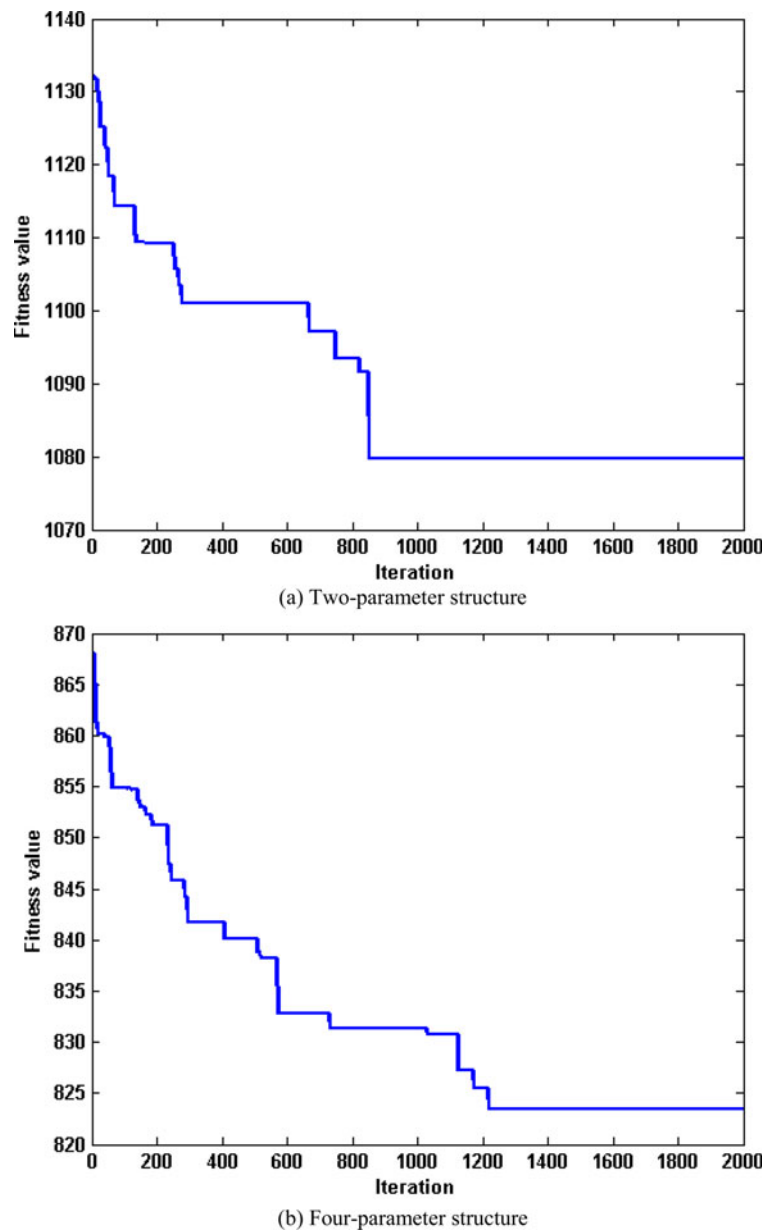


Fig. 15. Reinforced combinatorial particle swarm optimization objective function convergence plot.

der the following input:

$$u(k) = \sin\left(\frac{2\pi}{25}k\right).$$

Figures 9 and 10 show the output of the multimodel approach and the error obtained, respectively, when the final submodels are combined during the validation, while Table 4 shows their performance measure. It can be observed that both submodels' structures can approximate the real system. In addition, both CKF settings show the same performance. Table 5 shows the performance of the proposed technique using the initial submodels (without Stage 2). Although the signifi-

cance of Stage 2 seems not pronounced in this example, its inclusion does provide slightly better performance as seen in Table 4.

6.3. Example 3

In the third example, the following highly nonlinear dynamical system is considered for identification:

$$y(k+1) = \frac{u(k)}{1 + y^2(k-1) + y^2(k-2)} + \frac{y(k)y(k-1)y(k-2)u(k-1)(y(k-2)-1)}{1 + y^2(k-1) + y^2(k-2)}. \quad (30)$$

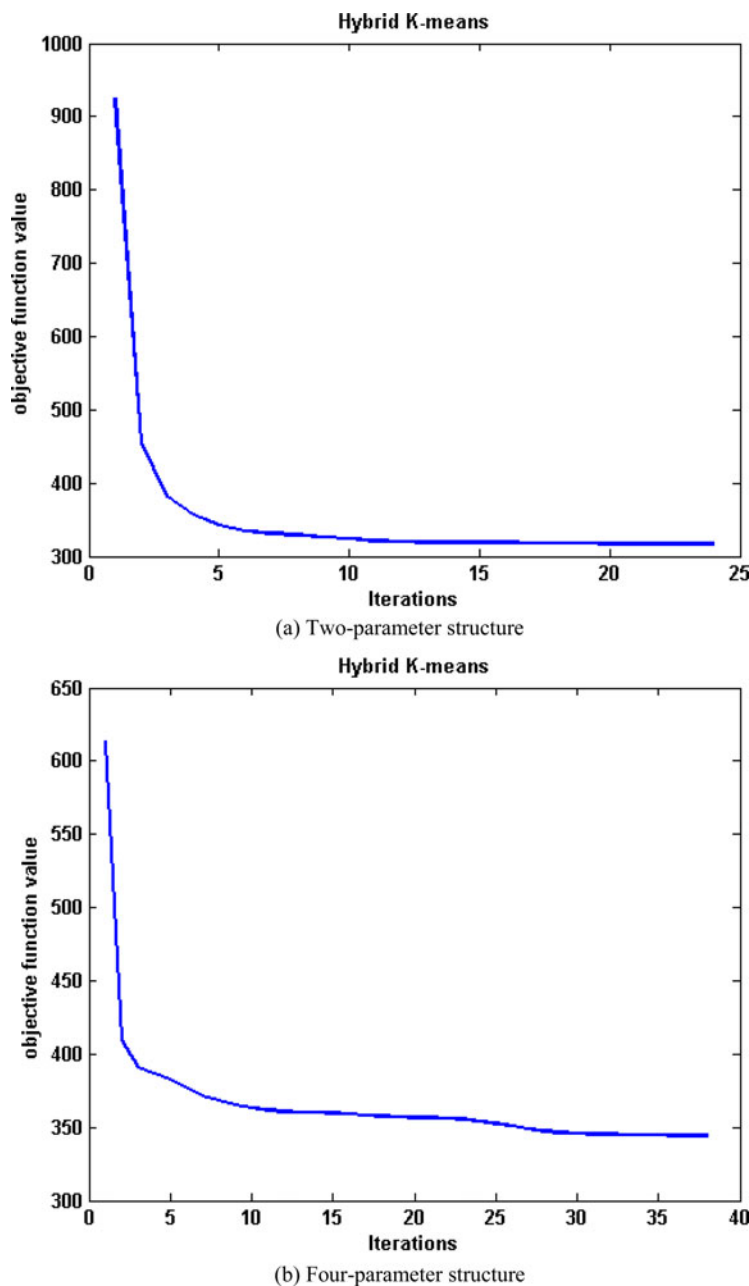


Fig. 16. Hybrid K-means objective function convergence plot.

It is a benchmark system proposed in (Narendra & Parthasarathy, 1990) and subsequently has been used in (Chen & Khalil, 1995; Verdult et al., 2002; Wen et al., 2007; Orjuela et al., 2013). The system was excited by uniformly distributed random signal in the interval $[-1, 1]$. The identification was carried out with a data set of 800 samples.

Using the proposed multimodel approach, four and two submodels were identified for two-parameter and four-parameter structures, respectively. The convergence of the objective functions in Stages 1 and 2 for both structures are shown in Figures 11 and 12, respectively. The initial and final submodel parameters with their associated centers

for the two stages are shown in Tables B.5 and B.6 in Appendix B.

Validation of the multimodel identification was done with a second data set of 800 samples generated by an input signal given by

$$u(k) = \begin{cases} \sin\left(\frac{2\pi}{250}k\right) & \text{if } k \leq 500 \\ 0.8 \sin\left(\frac{2\pi}{250}k\right) + 0.2 \sin\left(\frac{2\pi}{25}k\right) & \text{if } k > 500 \end{cases}$$

Simulation results obtained from Figures 13 and 14 show that the model outputs closely agree with the system output

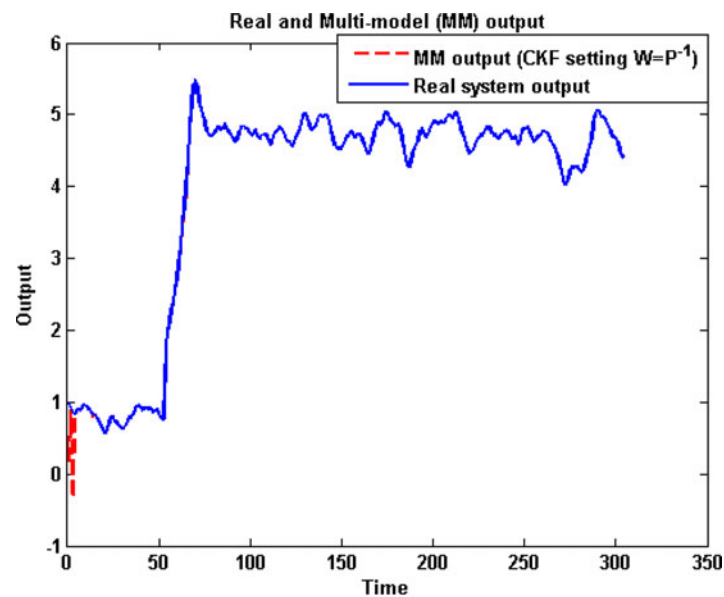
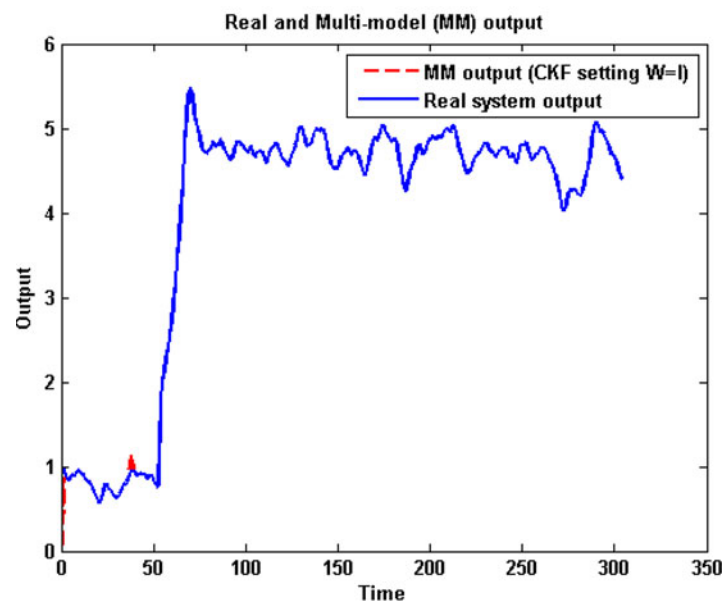
(a) Two-parameter structure with CKF $W = P^{-1}(k)$ (b) Two-parameter structure with CKF $W = I$

Fig. 17. Outputs of multimodel identification using validation data.

for both submodels' structures when the final submodels were combined. In addition, it can be concluded from Table 6 that the two-parameter structure with CKF setting $W = I$ shows the best performance. Furthermore, a slightly degraded performance can be observed in Table 7 that shows the results obtained when the initial submodels were combined instead of the final submodels.

As mentioned earlier, this system has been identified with other multimodel approaches in the literature. Using the same validation data set, Verdult et al. (2002) achieved MSE of 0.0002 and VAF of 99.9% with four third-order models (24 parameters), Wen et al. (2007) achieved MSE of 0.112 and

VAF of 97.9% with 10 BPWA functions (306 parameters), and Orjuela et al. (2013) achieved MSE of 0.00067 and VAF of 99.7% with 16 parameters. It is quite noteworthy to point out that the proposed multimodel approach does yield close performances to other techniques in the literature but with fewer parameters (8).

6.4. Example 4

In this example, a benchmark continuous stirred tank reactor nonlinear chemical system is considered. The system is described by the following equation in which all variables are

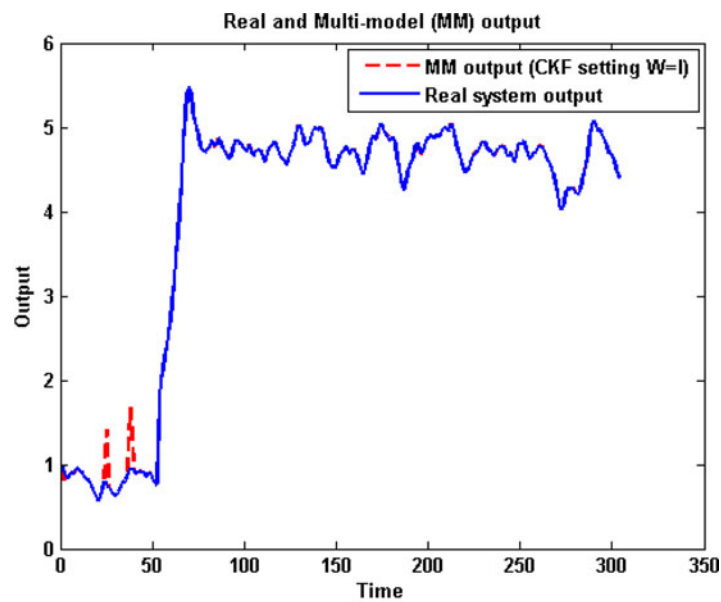
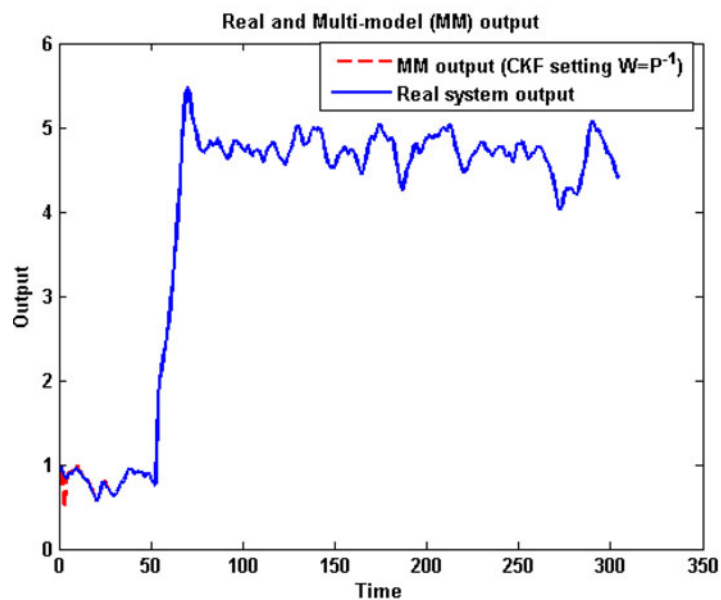
(c) Four-parameter structure with CKF $W = P^{-1}(k)$ (d) Four-parameter structure with CKF $W = I$

Fig. 17 (cont.)

dimensionless (Galan et al., 2003; Du et al., 2009):

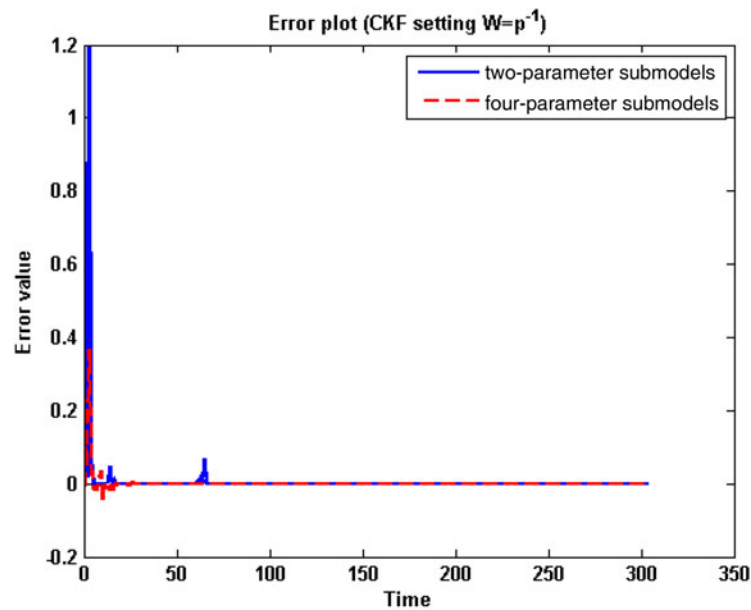
$$\begin{aligned} \dot{x}_1 &= -x_1 + D_a(1 - x_1)\exp\left(\frac{x_2}{1 + \frac{x_2}{\gamma}}\right), \\ \dot{x}_2 &= -x_2 + BD_a(1 - x_1)\exp\left(\frac{x_2}{1 + \frac{x_2}{\gamma}}\right) + \beta(u - x_2), \\ y &= x_2, \end{aligned} \quad (31)$$

where x_1 is the reagent conversion, x_2 is the reactor temperature, and u is the coolant temperature. In addition, u and x_2 are the

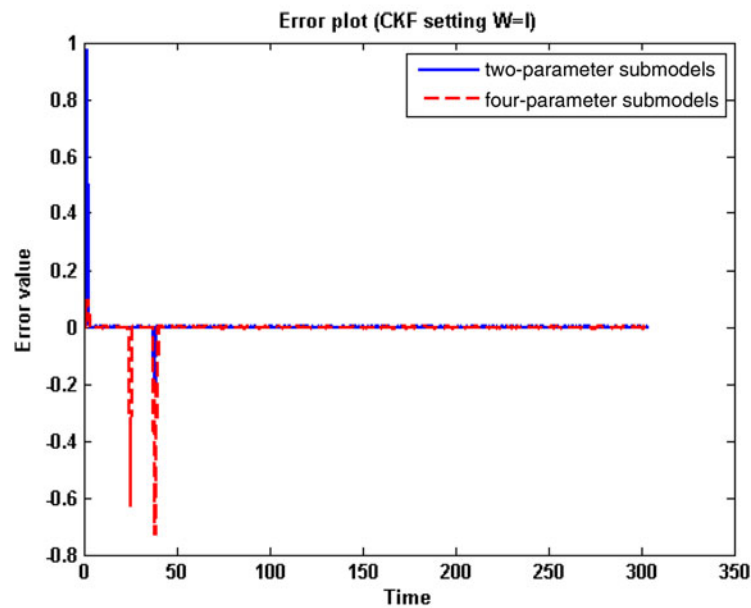
input and output of the system, respectively. The nominal values for the constants are $D_a = 0.072$, $\gamma = 20$, $B = 8$, and $\beta = 0.3$.

To test the ability of the proposed multimodel identification method, a random white noise step signal between $[-1.5, 1.5]$ is used as input to the system. The system is simulated with a sampling time of 0.2 min, and 700 pairs of input–output data were collected for the identification process and another 300 pairs for validation.

Based on the proposed method, four and three submodels were identified for the two-parameter and four-parameter structures, respectively. The initial and final submodel parameters with their associated centers are shown in Appendix B (Tables



(a) Error with CKF setting $W = P^{-1}(k)$



(b) Error with CKF setting $W = I$

Fig. 18. Multimodel identification error using validation data.

Table 8. Performance of proposed multimodel based on final submodels

CKF Setting	Two-Parameter Structures		Four-Parameter Structures	
	MSE	VAF	MSE	VAF
$W = P^{-1}(k)$	0.0064	99.42	0.0007	99.84
$W = I$	0.0053	99.75	0.0040	99.52

B.7 and B.8). However, the roots of the poles (*a* parameters) of the identified final submodels show that submodels 3 and 1 are unstable for two-parameter and four-parameter structures, respectively. The convergence of the objective functions in Stages 1 and 2 are shown in Figures 15 and 16, respectively.

The validation results using the second data pairs are shown in Figures 17 and 18 and Table 8 for both structures. It can be observed that the estimated output matched the real output.

Contrary to the notion that all the submodels must be stable for a decoupled multimodel to be stable (Orjuela, 2007), it

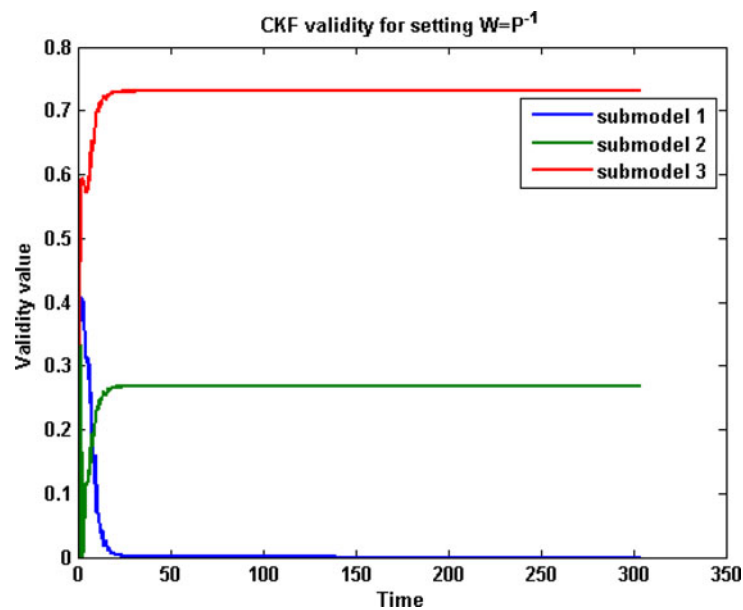


Fig. 19. Validity of four-parameter structure with constrained Kalman filter setting $W = P^{-1}(k)$.

Table 9. Performance of proposed multimodel based on initial submodels

CKF Setting	Two-Parameter Structures		Four-Parameter Structures	
	MSE	VAF	MSE	VAF
$W = P^{-1}(k)$	17.94	1.76	16.16	0
$W = I$	17.98	0.79	15.87	0

can be observed from Figure 17 that the multimodel output is stable. This is because the validity estimated by the CKF algorithm is able to filter out the unstable submodel during the blending operation. Figure 19 shows the validity values of the submodels for the four-parameter structure. It can be observed that the validity of submodel 1 is zero at instant 25. It shows that the identified decoupled multimodel acts like a Takagi–Sugeno multiple model where its stability depends not only on the stability of the submodels but also on the validity values. However, further study on this issue would be necessary to the conditions for this behavior. Furthermore, comparison of the results in Tables 8 and 9 indicates that significant improvement was obtained when the final submodels were blended together. This emphasized the significance of including Stage 2 into the methodology to refine the initial submodels for better approximation.

7. CONCLUSION

This paper presents a novel multimodel approach for the identification of nonlinear systems having several possible models depending on the operating region. In the proposed

approach, the number of the submodels are not known *a priori*. The proposed method consists of three stages. In the first stage, a reinforced combinatorial particle swarm optimization was developed to obtain the number of submodels and initial estimate of their parameters. In the second stage, hybrid K-means is used to refine the submodels in the first stage. The submodels are interpolated using a constrained Kalman filter in the third stage. Four simulated nonlinear systems examples that had been studied previously in the literature are presented to illustrate the performance of the method. The examples show that the proposed multimodel approach is effective in the identification of nonlinear systems. Although the proposed algorithm is designed for single-input/single-output systems, an extension to multiple-input/multiple-output systems should be easy by using a multiple-input/single-output formulation. Contrary to the notion that all the submodels must be stable for a decoupled multimodel to be stable, the CKF algorithm for the interpolation is shown to be able to produce a stable system when one of the submodels is unstable. In future research, we would like to further study the conditions necessary for the occurrence of this behavior and develop a control algorithm based on the identified submodels with the CKF algorithm.

ACKNOWLEDGMENTS

The authors acknowledge the support of King Fahd University of Petroleum and Minerals under Project IN141048.

REFERENCES

- Abbas, T., & Tufvesson, F. (2011). System identification in gsm/edge receivers using a multi-model approach. *ACEEE International Journal on Control System and Instrumentation 3(1)*, 41–46.

- Adeniran, A.A., & Elferik, S. (2014). Validity estimation for multi-model identification using constrained Kalman filter. *Proc IASTED on Modeling Identification and Control (MIC)*, (Hamza, M., Ed.), Innsbruck, Austria, February 17–19.
- Bedoui, S., Ltaief, M., & Abderrahim, K. (2012). New method for systematic determination of the model's base of time varying delay system. *International Journal of Computer Applications* 46(1), 13–19.
- Billings, S.A. (2013). *Models for Linear and Nonlinear Systems*. Hoboken, NJ: Wiley Online Library.
- Chen, F.-C., & Khalil, H. (1995). Adaptive control of a class of nonlinear discrete-time systems using neural networks. *IEEE Transactions on Automatic Control* 40(5), 791–801.
- Comoiu, M., Bara, C., & Popescu, D. (2013). Metaheuristic approach in nonlinear systems identification. *UPB Scientific Bulletin, Series A* 75(3), 91–104.
- da Costa Martins, J.K.E., & de Araujo, F.M.U. (2015). Nonlinear system identification based on modified anfis. *Proc. 12th Int. Conf. Informatics in Control, Automation and Robotics (ICINCO)*, 2015, Vol. 1, pp. 588–595, Colmar, Alsace, France, November 21–23.
- Demirli, K., Chengs, S.X., & Muthukumar, P. (2003). Subtractive clustering based on modeling of job sequencing with parametric search. *Fuzzy Sets and Systems* 137(2), 235–270.
- DeVeaux, R., Psychogios, D., & Ungar, L.H. (1993). A comparison of two nonparametric estimation schemes: Mars and neural networks. *Computers in Chemical Engineering* 17(8), 819–837.
- Du, J., Song, C., & Li, P. (2009). Application of gap metric to model bank determination in multilinear model approach. *Journal of Process Control* 19(2), 231–240.
- Dunlop, M.J., Franco, E., & Murray, R.M. (2007). Multi-model approach to identification of biosynthetic pathways. *Proc. 2007 American Control Conf.*, New York, July 11–13.
- Elfely, N., Dieulot, J., Benrejeb, M., & Borne, P. (2010). A new approach for multi-model identification of complex systems based on both neural and fuzzy clustering algorithms. *Engineering Applications of Artificial Intelligence* 23(7), 1064–1071.
- Galan, O., Romagnoli, J.A., Palazoğlu, A., & Arkun, Y. (2003). Gap metric concept and implications for multilinear model-based controller design. *Industrial & Engineering Chemistry Research* 42(10), 2189–2197.
- Gan, L., & Wang, L. (2011). A multi-model approach to predictive control of induction motor. *Proc. IECON 2011—37th Annual Conf. IEEE Industrial Electronics Society*, pp. 1704–1709, Melbourne, Australia, November 4–7.
- Gregorčič, G., & Lightbody, G. (2007). Local model network identification with Gaussian processes. *IEEE Transactions on Neural Networks* 18(9), 1404–1423.
- Gugaliya, J.K., & Gudi, R.D. (2010). Multimodel decomposition of nonlinear dynamics using fuzzy classification and gap metric analysis. *Proc. 9th Int. Symp. Dynamics and Control of Process Systems*. New York: Elsevier.
- Hu, Y.-J., Wang, Y., Wang, Z.-L., Wang, Y.-Q., & Zhang, B.C. (2014). Machining scheme selection based on a new discrete particle swarm optimization and analytic hierarchy process. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 28(2), 71–82.
- Jarboui, B., Cheikh, M., Siarry, P., & Rebai, A. (2007). Combinatorial particle swarm optimization (cpso) for partitional clustering problem. *Applied Mathematics and Computation* 192(2), 337–345.
- Kashiwagi, H., & Rong, L. (2002). Identification of volterra kernels of nonlinear van de Vusse reactor. *Transactions on Control, Automation, and Systems Engineering* 4(2), 109–113.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proc. IEEE Int. Conf. Neural Networks, 1995*, Vol. 4, pp. 1942–1948. Perth, Western Australia, November 27–December 1.
- Kramar, D., Cica, D., Sredanovic, B., & Kopac, J. (2015). Design of fuzzy expert system for predicting of surface roughness in high-pressure jet assisted turning using bioinspired algorithms. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*. Advance online publication.
- Ksouri-Lahmari, M., Borne, P., & Benrejeb, M. (2004). Multimodel: the construction of model bases. *Studies in Informatics and Control* 13(3), 199–210.
- Ljung, L. (Ed.). (1999). *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice Hall.
- Ltaief, M., Abderrahim, K., & Ben Abdennour, R. (2008). Contributions to the multimodel approach: systematic determination of a models' base and validities estimation. *Journal of Automation & Systems Engineering* 2(3).
- Majhi, B., & Panda, G. (2011). Robust identification of nonlinear complex systems using low complexity {ANN} and particle swarm optimization technique. *Expert Systems With Applications* 38(1), 321–333.
- Modares, H., Alfi, A., & Fateh, M.-M. (2010). Parameter identification of chaotic dynamic systems through an improved particle swarm optimization. *Expert Systems With Applications* 37(5), 3714–3720.
- Mohamed, R.B., Ben Nasr, H., & Sahli, F.M. (2011). A multi-model approach for a nonlinear system based on neural network validity. *International Journal of Intelligent Computing and Cybernetics* 4(3), 331–352.
- Narendra, K., & Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks* 1(1), 4–27.
- Nihan, N.L., & Davis, G.A. (1987). Recursive estimation of origin-destination matrices from input/output counts. *Transportation Research Part B: Methodological* 21(2), 149–163.
- Novak, J., Chalupa, P., & Bobal, V. (2009). Local model networks for modelling and predictive control of nonlinear systems. *Proc. 23rd European Conf. Modelling and Simulation*, Madrid, June 9–12.
- Novak, J., Chalupa, P., & Bobal, V. (2011). Modelling and predictive control of nonlinear systems using local model networks. *Proc. 18th IFAC Congr.*, Milan, Italy, August 28–September 2.
- Orjuela, R., Maquin, D., & Ragot, J. (2006). Nonlinear system identification using uncoupled state multiple model approach. *Workshop on Advanced Control and Diagnosis, ACD'2006*, Nancy, France, November 16–17.
- Orjuela, R., Marx, B., & Ragot, J., & Maquin, D. (2007). State estimation of nonlinear discrete-time systems based on the decoupled multiple model approach. *Proc. 4th Int. Conf. Informatics in Control, Automation and Robotics*, Angers, France, May 9–12.
- Orjuela, R., Marx, B., Ragot, J., & Maquin, D. (2013). Nonlinear system identification using heterogeneous multiple models. *International Journal of Applied Mathematics and Computer Science* 23(1), 103–115.
- Pan, W., Yuan, Y., Goncalves, J., & Stan, G. (2016). A sparse Bayesian approach to the identification of nonlinear state-space systems. *IEEE Transactions on Automatic Control* 61(1), 182–187.
- Ronen, M., Shabtai, Y., & Guterman, H. (2002). Hybrid modeling building methodology using unsupervised fuzzy clustering and supervised neural networks. *Biotechnology and Bioengineering* 77(4), 420–429.
- Samia, T., Ben Abdennour, R., Kamel, A., & Borne, P. (2002). A systematic determination approach of a models' base for uncertain systems: experimental validation. *Proc. IEEE International Conf. Systems Man and Cybernetics*, Vol. 6, pp. 73–81. New York: IEEE.
- Samia, T., Kamel, A., Ridha, B.A., & Mekki, K. (2008). Multimodel approach using neural networks for complex systems modeling and identification. *Nonlinear Dynamics and Systems Theory* 8(3), 299–316.
- Scarpiniti, M., Comminello, D., Parisi, R., & Uncini, A. (2015). Novel cascade spline architectures for the identification of nonlinear systems. *IEEE Transaction on Circuits and Systems I: Regular Papers* 62(7), 1825–1835.
- Shorten, R., Murry-Smith, R., Bjorgan, R., & Gollee, H. (1999). On the interpretation of local models in blended multiple structures. *International Journal of Control* 72(2), 620–628.
- Simon, D. (2006). *Kalman Filter Generalizations*. Hoboken, NJ: Wiley.
- Simon, D. (2010). Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *Control Theory Applications IET* 4(8), 1303–1318.
- Simon, D., & Chia, T.L. (2002). Kalman filtering with state equality constraints. *IEEE Transactions on Aerospace and Electronic Systems* 38(1), 128–136.
- Stanforth, R., Kolossov, E., & Mirkin, B. (2007). Hybrid k-means: combining regressionwise and centroid-based criteria for qsr. In *Selected Contributions in Data Analysis and Classification: Studies in Classification, Data Analysis, and Knowledge Organization* (Brito, P., Cucumel, G., Bertrand, P., & Carvalho, F., Eds.), pp. 225–233. Berlin: Springer.
- Tang, Y., Qiao, L., & Guan, X. (2010). Identification of wiener model using step signals and particle swarm optimization. *Expert Systems With Applications* 37(4), 3398–3404.
- Teslic, L., Hartmann, B., Nelles, O., & Škrjanc, I. (2011). Nonlinear system identification by Gustafson-Kessel fuzzy clustering and supervised local model network learning for the drug absorption spectra process. *IEEE Transactions on Neural Networks* 22(12), 1941–1951.
- Venkat, A.N., Vijaysai, P., & Gudi, R.D. (2003). Identification of complex nonlinear process based on fuzzy decomposition of the steady state space. *Journal of Process Control* 13(6), 473–488.
- Verdult, V., Ljung, L., & Verhaegen, M. (2002). Identification of composite local linear statespace models using a projected gradient search. *International Journal of Control* 75(16), 1385–1398.

- Wen, C., Wang, S., Jin, X., & Ma, X. (2007). Identification of dynamic systems using piecewise-affine basis function models. *Automatica* 43(10), 1824–1831.
- Wernholt, E., & Moberg, S. (2011). Nonlinear gray-box identification using local models applied to industrial robots. *Automatica* 47(4), 650–660.
- Xue, Z.-K., & Li, S.-Y. (2005). A multi-model identification algorithm based on weighted cost function and application in thermal process. *ACTA Automatica, SINICA* 31(3), 470–474.
- Yassin, I.M., Taib, M.N., & Adnan, R. (2013). Recent advancements and methodologies in system identification: a review. *Scientific Research Journal* 1(1), 14–33.
- Zeng, K., Tan, Z., Dong, M., & Yang, P. (2014). Probability increment based swarm optimization for combinatorial optimization with application to printed circuit board assembly. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 28(11), 429–437.

Ahmed A. Adeniran received his BS degree (with honors) in electrical and electronics engineering and his MS degree in computer science from the University of Ibadan. He also received MS and PhD degrees in systems engineering (control option) from King Fahd University of Petroleum and Minerals. Dr. Adeniran's research interests include dynamic system identification and control, fault diagnosis, and artificial intelligence, focusing on the development of data mining and machine learning methods for a variety of tasks and their applications to practical problems in science and engineering.

Sami El Ferik is an Associate Professor in control and instrumentation in the Department of Systems Engineering at King Fahd University of Petroleum and Minerals. He obtained his BS in electrical engineering from Laval University, Quebec, and his MS and PhD both in electrical and computer engineering from Ecole Polytechnique, University of Montreal. Dr. El Ferik's PhD work on flexible manufacturing systems modeling and control was cosupervised with mechanical engineering. After completion of his PhD and postdoctoral positions, he worked with Pratt and Whitney Canada as a Staff Control Analyst at the Research and Development Center of Systems, Controls, and Accessories. His research interests are in sensing, monitoring, and control with strong multidisciplinary research and applications. His research contributions are in control of drug administration, process control and control loop performance monitoring, control of systems with delays, modeling and control of stochastic systems, analysis of network stability, condition monitoring, and conditions.

APPENDIX A

The CKF algorithm for validity estimation

The CKF algorithm considered the multimodel representation in Eq. (2) as a parameter estimation problem as follows:

Given the multimodel output in Eq. (2), that is,

$$y(k) = \sum_{i=1}^M f_i(\varphi_i(k), \theta_i)\phi_i(k).$$

the vector form, Eq. (2), can be written as

$$y(k) = \tilde{y}(k)\Phi(k), \quad (\text{A.1})$$

where $\tilde{y}(k) = [y_1, y_2, \dots, y_M]$ is the known vector of local model outputs and $\Phi(k) = [\phi_1, \phi_2, \dots, \phi_M]^T$ is the unknown vector of weights for the local models.

Casting the parameter estimation to a state estimation problem, we have the following state equation:

$$\begin{aligned} \Phi(k+1) &= \Phi(k), \\ y(k) &= \tilde{y}(k)\Phi(k) + v(k), \end{aligned} \quad (\text{A.2})$$

where Φ is the vector of unknown parameter (validity) to be estimated, and $v(k)$ is the measurement noise with covariance $E[v(k)v(k)^T] = R(k)$. In the case where $v(k)$ is a white and Gaussian, the Kalman filter theory says that the posterior distribution of $\Phi(k)$, given all the observation up to $k-1$, is Gaussian mean value $\hat{\Phi}(k)$ and error covariance matrix P . One should note that an artificial noise $w(k)$ with variance $E[w(k)w(k)^T] = Q$ could be added to $\Phi(k+1)$ in the case of time-varying parameters and also to ensure persistence excitation and avoid ill conditioned numerical computation.

Furthermore, in order for the validity computation (Φ) to satisfy the partition of unity, the equality constraint in Eq. (3) needs to be added. In addition, as at any time instant it is possible for any local model to fully contribute or not to contribute to the system's output, there is a need to impose an inequality constraint [Eq. (4)]. Therefore, these two constraints need to be included in the estimation of (Φ) to give the following full state estimation equation:

$$\begin{aligned} \Phi(k+1) &= \Phi(k) + \omega(k), \\ y(k) &= \tilde{y}(k)\Phi(k) + v(k), \end{aligned} \quad (\text{A.3a})$$

such that

$$\beta\Phi(k) = 1, \quad 0 \leq \phi_i(k) \leq 1, \quad (\text{A.3b})$$

where β is a row vector of $[1, 1, \dots, 1, 1]$. The problem is thus formulated as the following: given state Eq. (A.3a), minimize the minimum mean square error estimate of the state $\Phi(k)$ [Eq. (A.4)] subject to constraints in Eq. (A.3b).

$$\min_{\Phi} E[(\Phi(k) - \hat{\Phi}(k))^2], \quad (\text{A.4})$$

where $E[\cdot]$ is the expectation operation, β is a row vector of $[1, 1, \dots, 1, 1]$, Φ is the unknown parameter, and $\hat{\Phi}$ is the estimated one.

The above problems are solved in two steps. In the first step, the equality constraint is solved using the projection techniques (Simon & Chia, 2002; Simon, 2006), where the unconstrained estimate $\hat{\Phi}(k)$ is projected onto the constraint space. The equality constrained optimization problem can be written as

$$\min_{\Phi} E[(\Phi(k) - \hat{\Phi}(k))^T W(\Phi(k) - \hat{\Phi}(k))] \quad (\text{A.5})$$

subject to

$$\beta\Phi(k) = 1, \quad (\text{A.6})$$

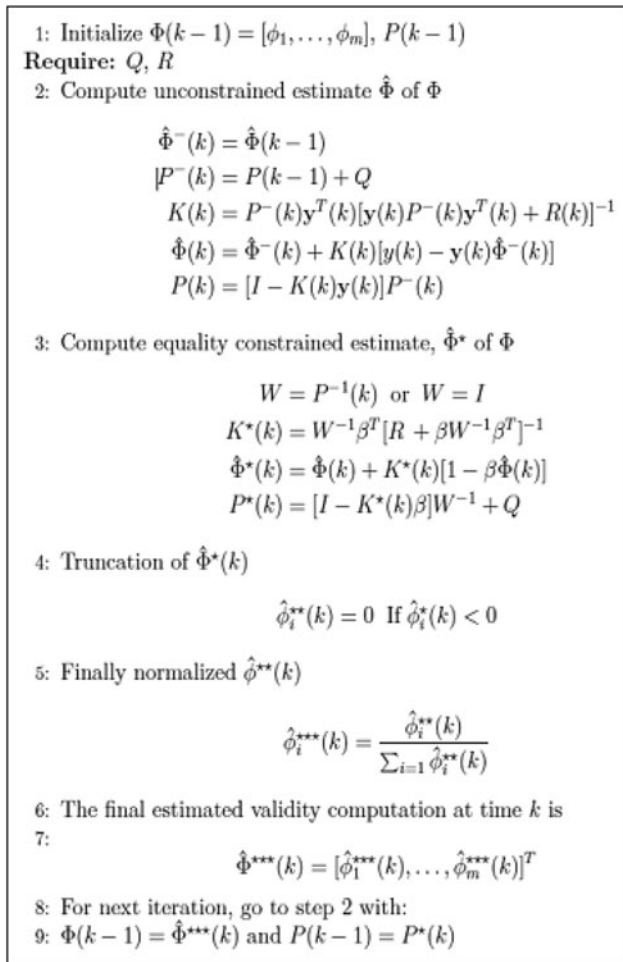


Fig. A.1. Stage 3: CFK for validity estimation.

where W is a positive define matrix. The solution to this problem is given as

$$\begin{aligned} \hat{\Phi}^*(k) &= \hat{\Phi}(k) + K^*(k) [1 - \beta\hat{\Phi}(k)], \\ K^*(k) &= W^{-1} \beta^T [R + \beta W^{-1} \beta^T]^{-1}, \\ P^*(k) &= [I - K^*(k)\beta]W^{-1} + Q, \end{aligned} \tag{A.7}$$

where $\hat{\Phi}$ is the unconstrained estimate, $\hat{\Phi}^*$ is the updated equality constrained estimate that satisfy Eq. (3), and W is a positive definite matrix weight. Setting $W = P^{-1}(k)$ in Eq. (A.7) results in minimum variance estimate, and setting $W = I$ gives least square estimate of $\Phi(k)$ (Simon, 2010). Both settings are implemented in this study. This implies that the unconstrained problem is first solved with the standard solution of the Kalman filter after which the obtained unconstrained estimate $\hat{\Phi}$ is used to update the constrained estimate in Eq. (A.7). Given observations $y(k), y(k-1), \dots, y(1)$ and local model outputs $\hat{y}(k), \hat{y}(k-1), \dots, \hat{y}(1)$, the optimal unconstrained estimate $\hat{\Phi}$ can be computed using the following Kalman filter equation

$$\begin{aligned} \hat{\Phi}^-(k) &= \hat{\Phi}(k-1), \\ P^-(k) &= P(k-1) + Q(k-1) \end{aligned}$$

$$\begin{aligned} \hat{\Phi}(k) &= \hat{\Phi}^-(k) + K(k)[y(k) - \hat{y}(k)\hat{\Phi}^-(k)], \\ K(k) &= P^-(k)\hat{y}(k)^T [\hat{y}(k)P^-(k)\hat{y}(k)^T + R(k)]^{-1}, \\ P(k) &= [I - K(k)\hat{y}(k)]P^-(k), \end{aligned} \tag{A.8}$$

where $P^-(k), \hat{\Phi}^-(k)$ are the *a priori* estimate of the error covariance and the validity estimate, respectively; $P(k), \hat{\Phi}(k)$ are the *a posteriori* estimate of the error covariance and the validity estimate respectively; and $K(k)$ is the Kalman gain.

In the second step truncation and normalization (Nihan & Davis, 1987) are adopted for the inequality constraints. The truncation is used to readjust each element of $\hat{\Phi}^*(k)$ in order not to violate the inequality constraint in Eq. (A.4) as follows:

$$\hat{\Phi}_i^{**}(k) = 0, \text{ if } \hat{\Phi}_i^*(k) < 0.$$

Finally, $\hat{\Phi}_i^{**}(k)$ is normalized since the truncation can violate the equality constraint in Eq. (3) and to satisfy the other part of the inequality constraint.

$$\hat{\Phi}_i^{***}(k) = \frac{\hat{\Phi}_i^{**}(k)}{\sum_{i=1} \hat{\Phi}_i^{**}(k)}, \tag{A.9}$$

where $\hat{\Phi}_i^{***}(k) = [\hat{\phi}_1^{***}(k), \hat{\phi}_2^{***}(k), \dots, \hat{\phi}_M^{***}(k)]^T$ is the final estimated validity computation at instant k . Note that the $\hat{\Phi}_i^{***}(k)$ and $P^*(k)$ are now the new *a posteriori* estimate of the error covariance and the validity estimate respectively at instant k . The summary of the CKF algorithm is shown in Figure A.1.

One important feature of the proposed CKF algorithm is the intrinsic ability to handle modeling error and measure noise through the covariance matrices Q and R . In the context of this work, the covariance R represents the measurements noise and the noise resulting from model mismatch ($R = \sigma_r + \sigma_b$). The measurement noise σ_r can be estimated by taken the variance of the output noise while the model mismatch σ_b can be estimated from the error obtained from the submodels.

In contrast, the covariance matrix Q is considered as the parameter that incorporate only the noise resulting from model mismatch $Q \equiv \sigma_b$. Because the submodels were estimated from the least square criterion, the model mismatch noise σ_b can be estimated from data by computing the empirical covariance from the classical result in least square theory (Ljung, 1999) for each submodel:

$$\sigma_{bi} = \left\| \frac{1}{n_i - (d + 1)} \sum_{j=1}^{n_i} (y_{ij} - \hat{y}_{ij}) \sum_{j=1}^{n_i} [\varphi_{ij} \varphi_{ij}^T]^\wedge \{-1\} \right\|, \tag{A.10}$$

where φ_i are the vectors of the regressors for i th submodel, and n_i is the number of data points used for estimating the i th submodel. In addition, d is the number of parameters, and y_i and \hat{y}_i are the actual and predicted output, respectively, for i th submodel. Hence, Q (a diagonal matrix) and R are computed as

$$Q = \begin{bmatrix} \sigma_{b1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_{bm} \end{bmatrix}, \tag{A.11}$$

$$R = \sigma_r + \|Q\|. \tag{A.12}$$

APPENDIX B

Submodel coefficients and centers

Table B.1. Initial submodels for Example 1

Initial Submodels	Two-Parameter Structures		Four-Parameter Structures			
	a_1	b_1	a_1	a_2	b_1	b_2
1	0.80021	0.46683	0.90904	-0.17929	0.50286	0.13472
2	0.79705	0.50171	1.041831	-0.2762	0.50508	0.05015
3	0.81370	0.42337	—	—	—	—
4	0.75039	0.61168	—	—	—	—
Initial Centers						
1	0.11698	0.05276	0.1198	0.11971	0.05152	0.05091
2	0.11998	0.05119	0.11543	-0.11181	0.04611	-0.05522
3	0.11966	0.052531	—	—	—	—
4	0.121169	0.04929	—	—	—	—

Table B.2. Final submodels for Example 1

Final Submodels	Two-Parameter Structures		Four-Parameter Structures			
	a_1	b_1	a_1	a_2	b_1	b_2
1	0.79812	0.48362	0.93806	-0.19068	0.51385	0.10845
2	0.82082	0.55976	0.80392	-0.12192	0.52162	0.18125
3	0.75896	0.42027	—	—	—	—
4	0.64245	0.84146	—	—	—	—
Final Centers						
1	0.13079	0.07222	0.14561	0.14268	0.05090	0.06226
2	0.12940	0.04096	0.09846	0.10036	0.05154	0.04208
3	0.10844	0.05039	—	—	—	—
4	0.09359	0.03171	—	—	—	—

Table B.3. Initial submodels for Example 2

Initial Submodels	Two-Parameter Structures		Four-Parameter Structures			
	a_1	b_1	a_1	a_2	b_1	b_2
1	0.54650	0.60674	0.47613	0.03870	0.60722	0.04359
2	0.56813	0.58940	0.61421	-0.04158	0.57869	-0.040053
Initial Centers						
1	-0.01412	-0.00080	-0.01639	-0.01316	-0.01488	-0.01593
2	0.07727	-0.16727	0.12038	0.05947	0.10028	0.13306

Table B.4. Final submodels for Example 2

Final Submodels	Two-Parameter Structures		Four-Parameter Structures			
	a_1	b_1	a_1	a_2	b_1	b_2
1	0.48004	0.60620	0.46668	0.04022	0.64528	0.04504
2	0.59707	0.60308	0.49261	0.03074	0.56777	0.03306
Final Centers						
1	0.01067	0.45786	-0.34414	-0.09591	-0.00413	-0.48350
2	-0.02951	-0.46991	0.29115	0.06810	-0.01360	0.41844

Table B.5. Initial submodels for Example 3

Initial Submodels	Two-Parameter Structures		Four-Parameter Structures			
	a_1	b_1	a_1	a_2	b_1	b_2
1	-0.00935	0.74062	-0.11662	0.03551	0.74416	0.08480
2	-0.00839	0.72822	0.40609	0.01631	0.74891	-0.25511
3	-0.03470	0.79774	—	—	—	—
4	0.05112	0.80693	—	—	—	—
Initial Centers						
1	-0.01530	-0.00774	-0.00143	-0.00626	-0.00842	0.00377
2	-0.00240	-0.00645	-0.14715	-0.06505	0.01370	-0.24046
3	-0.00191	0.00308	—	—	—	—
4	0.06674	-0.01483	—	—	—	—

Table B.6. Final submodels for Example 3

Final Submodels	Two-Parameter Structures		Four-Parameter Structures			
	a_1	b_1	a_1	a_2	b_1	b_2
1	-0.01637	0.65325	-0.05385	0.03735	0.86452	0.02908
2	0.04630	0.47067	-0.09456	0.04253	0.59664	0.06760
3	-0.03507	0.87663	—	—	—	—
4	0.00081	0.87200	—	—	—	—
Final Centers						
1	-0.19101	-0.05831	0.13269	0.02336	-0.00636	0.18525
2	0.20190	0.13531	-0.17305	-0.04705	-0.00841	-0.23354
3	0.09209	0.59731	—	—	—	—
4	0.02161	-0.59013	—	—	—	—

Table B.7. Initial submodels for Example 4

Initial Submodels	Two-Parameter Structures		Four-Parameter Structures			
	a_1	b_1	a_1	a_2	b_1	b_2
1	0.53667	0.01069	0.41435	0.28499	-0.13038	-0.45585
2	0.46132	0.18517	0.38424	0.15863	0.25568	0.25428
3	0.54070	-0.02090	0.33236	0.48919	0.05200	-0.11514
4	0.55891	0.11715	—	—	—	—
Initial Centers						
1	1.04448	-0.00451	0.88647	0.97696	0.00436	-0.18552
2	0.94996	0.11237	1.02903	0.99540	0.03286	0.09124
3	0.92811	-0.00836	0.99887	0.97750	0.04890	0.10564
4	1.01892	0.00898	—	—	—	—

Table B.8. Final submodels for Example 4

Final Submodels	Two-Parameter Structures		Four-Parameter Structures			
	a_1	b_1	a_1	a_2	b_1	b_2
1	0.51875	-0.25904	0.52332	0.60225	-0.79173	-1.45343
2	-0.16452	1.23775	0.37017	0.05227	1.00263	1.22693
3	1.02176	-2.10171	0.30684	0.37524	-0.15508	0.10070
4	0.91022	1.75512	—	—	—	—
Final Centers						
1	1.21339	-0.31725	0.84523	1.00344	0.02873	-0.07783
2	0.86379	0.64175	1.10918	0.90721	0.14359	0.18742
3	0.62801	-0.18007	0.98968	1.03175	-0.05245	-0.00450
4	1.07504	0.09944	—	—	—	—

Table B.9. Convergence time

	Model Structure	RCPSO Time (s)	Hybrid K-Means Time (s)
Example 1	Two parameter	474.80	1.32
	Four parameter	292.03	0.231
Example 2	Two parameter	615.11	0.16
	Four parameter	530.39	0.21
Example 3	Two parameter	655.98	3.52
	Four parameter	619.4	0.76
Example 4	Two parameter	548.21	2.72
	Four parameter	509.61	2.65

Note: RCPSO, reinforced combinatorial particle swarm optimization.