SCIENCE CHINA
Information Sciences

• Supplementary File •

Robust identification of multi-model systems with a synergistic reweighted expectation algorithm based on Krylov subspace

Ronghuan Li[1], Yujie Ma[3], Junxia Ma[1,2*] & Weili Xiong[1,2]

[1]*School of Internet of Things Engineering, Jiangnan University, Wuxi* 214122*, Jiangsu, China*
[2]*Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi* 214122*, Jiangsu, China*
[3]*Department of Electrical and Electronic Engineering, The University of Manchester, Manchester* M13 9PL*, United Kingdom*

Appendix A

Appendix A.1 Summary of variable notations

A summary table of variable notations is illustrated in Table A1.

Appendix A.2 The definition of LPV systems

Consider a multi-model system represented by an LPV model, consisting of M Auto-Regressive with eXogenous (ARX) input sub-models

$$y_k = \sum_{m=1}^{M} \zeta_{m,w_k} y_{m,k} + v_k,$$

where v_k represents noise, y_k denotes the system output, ζ_{m,w_k} represents the weighting function that combines all sub-models, $y_{m,k}$ is the output of the m-th sub-model, w_k is the scheduling variable. The sub-model can be expressed as

$$y_{m,k} = \boldsymbol{\phi}_k^{\mathrm{T}} \boldsymbol{\theta}_m.$$

The model orders n_{a} and n_{b} are known. The information vector $\boldsymbol{\phi}_k$ and parameter vector $\boldsymbol{\theta}_m$ are presented as

$$\begin{aligned} \boldsymbol{\phi}_k &:= [-y_{k-1}, -y_{k-2}, \cdots, -y_{k-n_{\mathrm{a}}}, u_{k-1}, u_{k-2}, \cdots, u_{k-n_{\mathrm{b}}}]^{\mathrm{T}} \in \mathbb{R}^{(n_{\mathrm{a}}+n_{\mathrm{b}})\times 1}, \\ \boldsymbol{\theta}_m &:= [a_{m,1}, a_{m,2}, \cdots, a_{m,n_{\mathrm{a}}}, b_{m,1}, b_{m,2}, \cdots, b_{m,n_{\mathrm{b}}}]^{\mathrm{T}} \in \mathbb{R}^{(n_{\mathrm{a}}+n_{\mathrm{b}})\times 1}, \end{aligned}$$

where $\boldsymbol{\theta}_m$ are parameters of the m-th sub-model respectively. To ensure smooth transitions between operating points, the normalized exponential function is chosen as the weighting function, defined as

$$\zeta_{m,w_k} = \frac{\exp(-\frac{(w_k-\mathcal{H}_m)^2}{2(o_m)^2})}{\sum_{m=1}^{M} \exp(-\frac{(w_k-\mathcal{H}_m)^2}{2(o_m)^2})},$$

where $\mathcal{H} = \{\mathcal{H}_m\}_{m=1,\cdots,M}$ denotes the set of predesigned operating points, and $o_m, \{m = 1, \cdots, M\}$ represents the effective width of sub-models, each taking a value within the interval $[o_{min}, o_{max}]$. The effective width o_m indicates the effective operating range of the m-th sub-model.

Appendix A.3 The details of Q-function

To estimate both the sub-model identity (SMI) and model parameters, the expectation-maximization (EM) algorithm is employed, which iteratively maximizes a $\mathbb{Q}$-function. The $\mathbb{Q}$-function is constructed as follow

$$Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^s) = E_{C_{mis}|C_{obs},\hat{\boldsymbol{\theta}}^s}\{\ln p(C_{obs}, C_{mis}|\boldsymbol{\theta})\}. \tag{A1}$$

* Corresponding author (email: jxma@jiangnan.edu.cn)

Table A1 Summary of Variable Notations

Variable	Description
$\boldsymbol{\theta}_m$	Parameter vector for sub-model m
s	Number of iterations
$\hat{\boldsymbol{\theta}}^s$	Estimated parameter vector at iteration s
$\hat{\imath}_k$, $\hat{\boldsymbol{o}}$	Outlier and outlier vector
y_k	System output at sample k
$y_{m,k}$	Output of sub-model m at sample k
w_k	The scheduling variable
$\eta_{(k,m)}$	The posterior probability of the sub-model m
o_m	The effective width of the sub-model m
ζ_{m,w_k}	The weighting function for sub-model m at sample k
λ	Design factor controlling sparsity of outliers
γ_k	Positive weight for sample k
τ, β	Weight control parameters
α	Step size in gradient descent
$\boldsymbol{\phi}_k$	Information vector at sample k
n_a, n_b	System orders
N	Total number of samples
M	Number of sub-models
$L(\cdot)$, $\mathcal{L}(\cdot)$	Original and relaxed cost functions
$\mathbb{F}(\cdot)$	Weighted cost function
$\|\cdot\|_0$, $\|\cdot\|$	ℓ_0 pseudo-norm and ℓ_2-norm
$\boldsymbol{\chi}_{m,1}^s$	Gradient direction vector at iteration s
$\bar{\boldsymbol{\chi}}_{m,i}^s$	i-th orthonormal direction vector at iteration s
${}^l\bar{\boldsymbol{\chi}}_m^s$	Matrix of l orthonormal directions at iteration s
Q_m	Curvature matrix for sub-model m
υ^s	Step size vector in Arnoldi update at iteration s
$\boldsymbol{P}^s$	Hessenberg matrix from Arnoldi process at iteration s
ξ^s	Norm of gradient direction at iteration s
$\boldsymbol{I}$	Identity matrix
$\rho[\cdot]$	Spectral radius of a matrix
$x(k)$	Displacement in MSD system
$F(k)$	External force in MSD system
$c_d(k)$, $k_s(k)$	Time-varying damping and stiffness coefficients
$C_A(k)$	Outlet reactant concentration in CSTR
$T(k)$	Reactor temperature in CSTR
$q_c(k)$	Coolant flow rate in CSTR

By applying the probability chain rule, $\ln p(C_{mis}, C_{obs}|\boldsymbol{\theta})$ can be further decomposed as

$$\begin{aligned}\ln p(C_{mis}, C_{obs}|\boldsymbol{\theta}) &= \ln p(Y, \mathcal{I}, \mathcal{H}, U|\boldsymbol{\theta}) \\ &= \ln p(Y|\mathcal{I}, \mathcal{H}, U, \boldsymbol{\theta}) + \ln p(\mathcal{I}|\mathcal{H}, U, \boldsymbol{\theta}) + \ln p(\mathcal{H}, U|\boldsymbol{\theta}).\end{aligned} \tag{A2}$$

Since U and $\mathcal{H}$ are measured process signals independent of the model parameters $\boldsymbol{\theta}$, $\ln p(\mathcal{H}, U|\boldsymbol{\theta})$ is constant, denoted as $C = \ln p(\mathcal{H}, U|\boldsymbol{\theta})$.

The term $\ln p(Y|\mathcal{I}, \mathcal{H}, U, \boldsymbol{\theta})$ in (A2) can be further decomposed as

$$\begin{aligned}\ln p(Y|\mathcal{I}, \mathcal{H}, U, \boldsymbol{\theta}) &= \ln p(y_{N:1}|\mathcal{I}_{N:1}, w_{N:1}, u_{N:1}, \boldsymbol{\theta}_m) \\ &= \ln p(y_N|y_{N-1:1}, \mathcal{I}_{N:1}, w_{N:1}, u_{N:1}, \boldsymbol{\theta}_m) + \ln p(y_{N-1}|y_{N-2:1}, \mathcal{I}_{N:1}, w_{N:1}, u_{N:1}, \boldsymbol{\theta}_m) \\ &\quad + \ln p(y_{N-2:1}|\mathcal{I}_{N:1}, w_{N:1}, u_{N:1}, \boldsymbol{\theta}_m) \\ &= \ln p(y_N|y_{N-1:1}, \mathcal{I}_{N:1}, w_{N:1}, u_{N:1}, \boldsymbol{\theta}_m) + \ln p(y_{N-1}|y_{N-2:1}, \mathcal{I}_{N:1}, w_{N:1}, u_{N:1}, \boldsymbol{\theta}_m) \\ &\quad + \ln p(y_{N-2}|y_{N-3:1}, \mathcal{I}_{N:1}, w_{N:1}, u_{N:1}, \boldsymbol{\theta}_m) + \cdots + \ln p(y_1|\mathcal{I}_{N:1}, w_{N:1}, u_{N:1}, \boldsymbol{\theta}_m) \\ &= \sum_{k=1}^{N} \ln p(y_k|y_{k-1:1}, \mathcal{I}_{N:1}, w_{N:1}, u_{N:1}, \boldsymbol{\theta}_m).\end{aligned}$$

The output y_k depends on the historical observations, including past outputs and inputs, which are in the observation set C_{obs}. Hence, the joint likelihood can be expressed as

$$\ln p(Y|\mathcal{I}, \mathcal{H}, U, \boldsymbol{\theta}) = \sum_{k=1}^{N} \ln p(y_k|\mathcal{I}_k, C_{obs}, \boldsymbol{\theta}_m). \tag{A3}$$

Similarly, the term $\ln p(\mathcal{I}|\mathcal{H}, U, \boldsymbol{\theta})$ in (A2) can be further decomposed into

$$\ln p(\mathcal{I}|\mathcal{H}, U, \boldsymbol{\theta}) = \sum_{k=1}^{N} \ln p(\mathcal{I}_k|\mathcal{I}_{k-1:1}, w_{N:1}, u_{N:1}, \boldsymbol{\theta}).$$

The $\mathcal{I}_k$ is only related to the w_k and the $\boldsymbol{\theta}$, so it can be denoted as

$$\ln p(\mathcal{I}|\mathcal{H}, U, \boldsymbol{\theta}) = \sum_{k=1}^{N} \ln p(\mathcal{I}_k|w_k, \boldsymbol{\theta}). \tag{A4}$$

E-step: Construction of the conditional expectation $\mathbb{Q}$-function

$$\begin{aligned}Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^s) &= E_{C_{mis}|C_{obs}, \hat{\boldsymbol{\theta}}^s}\{\ln p(C_{mis}, C_{obs}|\boldsymbol{\theta})\} \\ &= E_{\mathcal{I}|Y,U,\mathcal{H}, \hat{\boldsymbol{\theta}}^s}\Big\{ \sum_{k=1}^{N} \ln p(y_k|\mathcal{I}_k, C_{obs}, \boldsymbol{\theta}_m) + \sum_{k=1}^{N} \ln p(\mathcal{I}_k|w_k, \boldsymbol{\theta}) + C \Big\} \\ &= \sum_{k=1}^{N} \sum_{m=1}^{M} p(\mathcal{I}_k = m|C_{obs}, \hat{\boldsymbol{\theta}}^s) \times \ln p(y_k|\mathcal{I}_k = m, C_{obs}, \hat{\boldsymbol{\theta}}^s_m) \\ &+ \sum_{k=1}^{N} \sum_{m=1}^{M} p(\mathcal{I}_k = m|C_{obs}, \hat{\boldsymbol{\theta}}^s) \times \ln p(\mathcal{I}_k = m|w_k, \hat{\boldsymbol{\theta}}^s) + C,\end{aligned}$$

where,

$$\begin{aligned}p(y_k|\mathcal{I}_k = m, C_{obs}, \hat{\boldsymbol{\theta}}^s_m) &= \frac{1}{\sqrt{2\pi(\hat{\sigma}^2)^s}} \exp\Big\{ - \frac{(y_k - \boldsymbol{\phi}_k^{\mathrm{T}} \hat{\boldsymbol{\theta}}^s_m)^2}{2(\hat{\sigma}^2)^s} \Big\}, \\ p(\mathcal{I}_k = m|w_k, \hat{\boldsymbol{\theta}}^s) &= \frac{\exp(-\frac{(w_k - \mathcal{H}_m)^2}{2(o_m)^2})}{\sum_{m=1}^{M} \exp(-\frac{(w_k - \mathcal{H}_m)^2}{2(o_m)^2})},\end{aligned} \tag{A5}$$

where, o_m is the sub-model effective width. According to Bayes' rule, the posterior probability that the output y_k from the m-th sub-model can be computed as

$$\eta_{(k,m)} = \frac{p(y_k|\mathcal{I}_k = m, C_{\text{obs}}, \hat{\boldsymbol{\theta}}^s_m) p(\mathcal{I}_k = m|w_k, \hat{\boldsymbol{\theta}}^s)}{\sum_{m'=1}^{M} p(y_k|\mathcal{I}_k = m', C_{\text{obs}} \hat{\boldsymbol{\theta}}^s_{m'}) p(\mathcal{I}_k = m'|w_k, \hat{\boldsymbol{\theta}}^s)}, \tag{A6}$$

where $\eta_{(k,m)}$ denotes the posterior probability of the m-th sub-model at time step k.

M − step Taking the partial derivative of the $\mathbb{Q}$-function with respect to the model parameters $\boldsymbol{\theta}_m$ and setting it to zero yields

$$\hat{\boldsymbol{\theta}}_m^{s+1} = \left[\sum_{k=1}^{N} \eta_{(k,m)} \boldsymbol{\phi}_k \boldsymbol{\phi}_k^{\mathrm{T}}\right]^{-1} \sum_{k=1}^{N} \eta_{(k,m)} \boldsymbol{\phi}_k y_k. \tag{A7}$$

Similarly, the estimate of $\hat{\sigma}^2$ is obtained by taking the partial derivative of the Q-function with respect to $\hat{\sigma}^2$ and setting it to zero, giving

$$(\hat{\sigma}^2)^{s+1} = \frac{\sum_{k=1}^{N}\sum_{m=1}^{M} \eta_{(k,m)}(y_k - \boldsymbol{\phi}_k^{\mathrm{T}}\hat{\boldsymbol{\theta}}_m^{s+1})^2}{N}.$$

For the sub-model effective width $o_m, \{m = 1, \cdots, M\}$, the analytical solution of the parameter estimates cannot be derived by maximizing the $\mathbb{Q}$-function. Hence, the parameters are obtained by solving a nonlinear optimization problem formulated as

$$\begin{aligned} \max_{o_m\{m=1,\cdots,M\}} \quad & \sum_{k=1}^{N}\sum_{m=1}^{M} \eta_{(k,m)} \ln p(\mathcal{I}_k = m | w_k, \hat{\boldsymbol{\theta}}^s). \\ s.t \quad & o_{min} \leqslant o_m \leqslant o_{max}, \quad m = 1, \cdots, M \end{aligned}$$

Appendix B

Appendix B.1 Derivation of the outlier estimate

Let $\hat{\imath}_k$ denote the estimate of outlier $\imath_k$, and define $\hat{\boldsymbol{o}} \coloneqq \left[\hat{\imath}_1, \hat{\imath}_2, \ldots, \hat{\imath}_N\right]$. To estimate parameters $\boldsymbol{\theta}_m$, we formulate a cost function that penalizes model residual and outlier sparsity via an ℓ_0 pseudo-norm:

$$L(\hat{\boldsymbol{\theta}}^s, \hat{\imath}_k) = \sum_{k=1}^{N} \left\{ \left(y_k - \sum_{m=1}^{M} \eta_{(k,m)} y_{m,k} - \hat{\imath}_k\right)^2 + \lambda \|\hat{\boldsymbol{o}}\|_0 \right\}, \tag{B1}$$

where $\|\cdot\|_0$ denotes the ℓ_0 pseudo norm, λ is a design factor to control the sparsity of $\hat{\boldsymbol{o}}$. Since (B1) is NP-hard, hence, an approximate solution (B2) is obtained through convex relaxation:

$$\mathcal{L}(\hat{\boldsymbol{\theta}}^s, \hat{\imath}_k) = \sum_{k=1}^{N} \left\{ \left(y_k - \sum_{m=1}^{M} \eta_{(k,m)} y_{m,k} - \hat{\imath}_k\right)^2 + \lambda |\hat{\imath}_k| \right\}. \tag{B2}$$

By setting the derivative of (B2) with respect to $\hat{\imath}_k$ to zero and applying the soft-thresholding principle, the outlier estimate is obtained as:

$$\begin{aligned} \hat{\imath}_k &= y_k - \sum_{m=1}^{M} \eta_{(k,m)} y_{m,k} - \lambda \delta(\hat{\imath}_k)\hat{\imath}_k - \frac{\lambda}{2}\mathrm{sign}(\hat{\imath}_k) \\ &= \begin{cases} y_k - \sum_{m=1}^{M} \eta_{(k,m)} y_{m,k} - \frac{\lambda}{2}, & y_k - \sum_{m=1}^{M} \eta_{(k,m)} y_{m,k} > \frac{\lambda}{2} \\ 0, & |y_k - \sum_{m=1}^{M} \eta_{(k,m)} y_{m,k}| < \frac{\lambda}{2} \\ y_k - \sum_{m=1}^{M} \eta_{(k,m)} y_{m,k} + \frac{\lambda}{2}, & y_k - \sum_{m=1}^{M} \eta_{(k,m)} y_{m,k} < -\frac{\lambda}{2} \end{cases} \end{aligned} \tag{B3}$$

Appendix B.2 Derivation of parameter update via gradient descent

Define the iterative model parameters to be updated as:

$$\boldsymbol{\theta}_m^{s+1} = \boldsymbol{\theta}_m^s + \alpha \nabla_d, \tag{B4}$$

where α is the step size and ∇_d is the direction, which can be calculated from (B4) as follows:

$$-\frac{\nabla \mathbb{F}(\boldsymbol{\theta}_m)}{\nabla \boldsymbol{\theta}_m} = \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} \boldsymbol{\theta}_m - \hat{\imath}_k\right]. \tag{B5}$$

Then, we can get

$$\begin{aligned}\boldsymbol{\theta}_m^{s+1} &= \boldsymbol{\theta}_m^s + \alpha \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} \boldsymbol{\theta}_m^s - \hat{\imath}_k \right], \\ &= \left[\boldsymbol{I} - \alpha \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \boldsymbol{\phi}_k^{\mathrm{T}} \right] \boldsymbol{\theta}_m^s + \alpha \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \hat{\imath}_k \right], \end{aligned} \tag{B6}$$

where $\boldsymbol{I}$ denotes the identity matrix. For the sequence $\boldsymbol{\theta}_m^{s+1}$ in (B6) to converge, it is essential to ensure that:

$$\rho \left[\boldsymbol{I} - \alpha \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \boldsymbol{\phi}_k^{\mathrm{T}} \right] < 1, \tag{B7}$$

$\rho[\boldsymbol{G}]$ in (B7) means the spectral radius of matrix $\boldsymbol{G}$. Hence, the α satisfies

$$\begin{aligned} 0 \;<\; \alpha &< \frac{2}{\alpha_{\max}\left[\boldsymbol{G}\right]}, \\ \boldsymbol{G} \;&=\; \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \boldsymbol{\phi}_k^{\mathrm{T}}, \end{aligned} \tag{B8}$$

where $\alpha_{\max}\left[\boldsymbol{G}\right]$ means the largest eigenvalue of matrix $\boldsymbol{G}$.

Appendix C

Appendix C.1 Detailed steps for generating the orthonormal basis

Arnoldi method reduces the large-scale problem to an eigenvalue problem for a smaller Hessenberg matrix via Krylov subspace projection.

Consider the Krylov subspace

$$\bar{\boldsymbol{\chi}}_{m,l}^s = \mathrm{span}\{\bar{\boldsymbol{\chi}}_{m,1}^s, Q_m \bar{\boldsymbol{\chi}}_{m,1}^s, \cdots, (Q_m)^{l-1} \bar{\boldsymbol{\chi}}_{m,1}^s\}, \tag{C1}$$

where, $Q_m \in \mathbb{R}^{(n_a+n_b)\times(n_a+n_b)}$ is nonsingular, $\bar{\boldsymbol{\chi}}_{m,1}^s \in \mathbb{R}^{(n_a+n_b)}$ with $\|\bar{\boldsymbol{\chi}}_{m,1}^s\| = 1$. An orthonormal basis $\{\bar{\boldsymbol{\chi}}_{m,1}^s, \bar{\boldsymbol{\chi}}_{m,2}^s, \cdots, \bar{\boldsymbol{\chi}}_{m,l}^s\}$ is generated by the following procedure.

1. Initialize with a vector $\bar{\boldsymbol{\chi}}_{m,1}^s$ such that $\|\bar{\boldsymbol{\chi}}_{m,1}^s\| = 1$, and set $i = 1$.
2. For $j = 1, 2, \ldots, i$, compute the inner products $\left(Q_m \bar{\boldsymbol{\chi}}_{m,i}^s, \bar{\boldsymbol{\chi}}_{m,j}^s\right)$.
3. Update the direction:

$$\boldsymbol{\chi}_{m,i+1}^s = Q_m \bar{\boldsymbol{\chi}}_{m,i}^s - \sum_{j=1}^{i} \left(Q_m \bar{\boldsymbol{\chi}}_{m,i}^s, \bar{\boldsymbol{\chi}}_{m,j}^s\right) \bar{\boldsymbol{\chi}}_{m,j}^s.$$

4. Normalize the new vector:

$$\bar{\boldsymbol{\chi}}_{m,i+1}^s \leftarrow \frac{\boldsymbol{\chi}_{m,i+1}^s}{\|\boldsymbol{\chi}_{m,i+1}^s\|}.$$

5. Increment $i \leftarrow i + 1$, and repeat from step 2.

Here, $(\cdot, \cdot)$ denotes the inner product.

Appendix C.2 Parameter update in Krylov subspace

An orthogonal basis is constructed using the gradient direction vector. Based on (B5), it is then reformulated as follows:

$$\boldsymbol{\chi}_{m,1}^s = \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} \boldsymbol{\theta}_m - \hat{\imath}_k \right], \tag{C2}$$

$$\bar{\boldsymbol{\chi}}_{m,1}^s = \frac{\boldsymbol{\chi}_{m,1}^s}{\|\boldsymbol{\chi}_{m,1}^s\|} = \frac{\boldsymbol{\chi}_{m,1}^s}{\xi^s},$$

let

$$Q_m^s = \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \boldsymbol{\phi}_k^{\mathrm{T}}. \tag{C3}$$

Construct the following l-orthonormal directions as:

$$ {}^{l}\bar{\boldsymbol{\chi}}_m^s = \left[\bar{\boldsymbol{\chi}}_{m,1}^s, \bar{\boldsymbol{\chi}}_{m,2}^s, \cdots, \bar{\boldsymbol{\chi}}_{m,l}^s\right], \tag{C4} $$

where, $l<(n_a+n_b)$ and $[\bar{\boldsymbol{\chi}}_{m,i}^s]^{\mathrm{T}}\bar{\boldsymbol{\chi}}_{m,j}^s = 0,\ i \neq j$.

Therefore, the model parameter estimates at iteration $s+1$ are given by:

$$ \hat{\boldsymbol{\theta}}_m^{s+1} = \hat{\boldsymbol{\theta}}_m^s + {}^{l}\bar{\boldsymbol{\chi}}_m^{s+1}\upsilon^{s+1}, \tag{C5} $$

where, ${}^{l}\bar{\boldsymbol{\chi}}_m^{s+1}$ contains the information of l-orthonormal directions, the key step is to select an appropriate υ^{s+1} that satisfies:

$$ \sum_{k=1}^{N}\gamma_k\eta_{(k,m)}\boldsymbol{\phi}_k\left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}}\hat{\boldsymbol{\theta}}_m^{s+1} - \hat{\imath}_k\right] \leqslant \sum_{k=1}^{N}\gamma_k\eta_{(k,m)}\boldsymbol{\phi}_k\left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}}\hat{\boldsymbol{\theta}}_m^{s} - \hat{\imath}_k\right]. \tag{C6} $$

Substituting (C5) into the left side of (C6) yields

$$ \sum_{k=1}^{N}\gamma_k\eta_{(k,m)}\boldsymbol{\phi}_k\left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\theta}_m^{s+1} - \hat{\imath}_k\right] = \bar{\boldsymbol{\chi}}_{m,1}^{s+1}\xi^{s+1} - \left[Q_m^{s+1}\bar{\boldsymbol{\chi}}_{m,1}^s, Q_m^{s+1}\bar{\boldsymbol{\chi}}_{m,2}^s, \cdots, Q_m^{s+1}\bar{\boldsymbol{\chi}}_{m,l}^s\right]\upsilon^s. \tag{C7} $$

Based on Arnoldi's method, we get

$$ \begin{aligned} Q_m^{s+1\,l}\bar{\boldsymbol{\chi}}_m^{s+1} &= {}^{l+1}\bar{\boldsymbol{\chi}}_m^{s+1}P^s, \\ \boldsymbol{P}^s &= \begin{bmatrix} \bar{\boldsymbol{P}}^s \\ p_{l+1}^s r_{l,l}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{(l+1)\times l}, \\ \bar{\boldsymbol{P}}^s &= \begin{bmatrix} p_{1,1}^s & p_{1,2}^s & \cdots & p_{1,l-1}^s & p_{1,l}^s \\ p_{2,1}^s & p_{2,2}^s & \cdots & p_{2,l-1}^s & p_{2,l}^s \\ 0 & p_{3,2}^s & \cdots & p_{3,l-1}^s & p_{3,l}^s \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & p_{l,l-1}^s & p_{l,l}^s \end{bmatrix} \in \mathbb{R}^{l\times l}, \end{aligned} \tag{C8} $$

$$ \boldsymbol{r}_{l,l} = [0,0,\cdots,0,1]^{\mathrm{T}} \in \mathbb{R}^l, \tag{C9} $$

where $p_{j,i}^s = (Q_m^{s+1}\bar{\boldsymbol{\chi}}_{m,i}^{s+1}, \bar{\boldsymbol{\chi}}_{m,j}^s)$, $j \leqslant i$ and $p_{i+1,i}^s = \|\boldsymbol{\chi}_{m,i+1}^{s+1}\|$ in (C8). The first subscript l in $\boldsymbol{r}_{l,l}$ in (C9) indicates the order of the vector, while the second subscript denotes that the l-th element of the vector is 1.

Hence, (C7) can be transform into:

$$ \begin{aligned} \sum_{k=1}^{N}\gamma_k\eta_{(k,m)}\boldsymbol{\phi}_k\left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\theta}_m^{s+1} - \hat{\imath}_k\right] &= \bar{\boldsymbol{\chi}}_{m,1}^{s+1}\xi^{s+1} - {}^{l+1}\bar{\boldsymbol{\chi}}_m^{s+1}\boldsymbol{P}^s\upsilon^s \\ &= {}^{l+1}\bar{\boldsymbol{\chi}}_m^{s+1}\left[\xi^{s+1}\boldsymbol{r}_{l+1,1} - \boldsymbol{P}^s\upsilon^s\right]. \end{aligned} \tag{C10} $$

Since ${}^{l+1}\bar{\boldsymbol{\chi}}_m^{s+1} \in \mathbb{R}^{(n_a+n_b)\times(l+1)}$ and $(n_a+n_b) > (l+1)$, the following equality holds:

$$ \left\|\sum_{k=1}^{N}\gamma_k\eta_{(k,m)}\boldsymbol{\phi}_k\left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\theta}_m^{s+1} - \hat{\imath}_k\right]\right\| = \left\|\xi^{s+1}\boldsymbol{r}_{l+1,1} - \boldsymbol{P}^s\upsilon^s\right\|. \tag{C11} $$

By matrix theory, there exists an orthogonal matrix $\boldsymbol{J} \in \mathbb{R}^{(l+1)\times(l+1)}$, such that (C11) is satisfied:

$$ \begin{aligned} &\left\|\sum_{k=1}^{N}\gamma_k\eta_{(k,m)}\boldsymbol{\phi}_k\left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\theta}_m^{s+1} - \hat{\imath}_k\right]\right\| \\ &= \left\|\boldsymbol{J}^{\mathrm{T}}\right\|\left\|\boldsymbol{J}\xi^{s+1}\boldsymbol{r}_{l+1,1} - \boldsymbol{J}\boldsymbol{P}^s\upsilon^s\right\| \\ &= \left\|\left(\begin{bmatrix} \boldsymbol{\mu}^{\mathrm{T}} \\ \mu^{s+1} \end{bmatrix} - \begin{bmatrix} \tilde{\boldsymbol{P}}^s \\ \boldsymbol{0} \end{bmatrix}\upsilon^s\right)\right\|, \end{aligned} $$

where

$$ \boldsymbol{J}\xi^{s+1}\boldsymbol{r}_{l+1,1} = \begin{bmatrix} \boldsymbol{\mu}^{\mathrm{T}} \\ \mu^{s+1} \end{bmatrix}, $$

$$ \boldsymbol{J}\boldsymbol{P}^s = \begin{bmatrix} \tilde{\boldsymbol{P}}^s \\ \boldsymbol{0} \end{bmatrix}, $$

$$ \boldsymbol{\mu}^{\mathrm{T}} \in \mathbb{R}^l, \tilde{\boldsymbol{P}}^s \in \mathbb{R}^{l\times l}, \boldsymbol{0} \in \mathbb{R}^{1\times l}. $$

Appendix C.3 Theorem 1 and its proof

Theorem 1: Let the model parameter estimate $\boldsymbol{\theta}_m$ be obtained using the SRE algorithm in (C5), and define the l-direction vectors ${}^{l}\bar{\boldsymbol{\chi}}_m^s$ as in (C4). When $\upsilon^s = \frac{\boldsymbol{\mu}^{\mathrm{T}}}{\tilde{\boldsymbol{P}}^s}$, the following inequality holds:

$$\left\| \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} \boldsymbol{\theta}_m^{s+1} - \hat{\imath}_k \right] \right\| \leqslant \left\| \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} \boldsymbol{\theta}_m^{s} - \hat{\imath}_k \right] \right\|. \tag{C12}$$

Proof: Let $l = 1$, then

$$\begin{aligned}
&{}^{l}\bar{\boldsymbol{\chi}}_m^s = \left[\bar{\boldsymbol{\chi}}_{m,1}^s \right], \\
&\sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} \boldsymbol{\theta}_m - \hat{\imath}_k \right] = \boldsymbol{\chi}_{m,1}^s, \\
&Q_m^{s+1} {}^{1}\bar{\boldsymbol{\chi}}_m^{s+1} = [\bar{\boldsymbol{\chi}}_{m,1}^s, \bar{\boldsymbol{\chi}}_{m,2}^s] \begin{bmatrix} p_{1,1}^s \\ p_{2,1}^s \end{bmatrix}.
\end{aligned}$$

The orthogonal matrix $\boldsymbol{J} \in \mathbb{R}^{2\times 2}$ is expressed as

$$\boldsymbol{J} = \begin{bmatrix} \frac{p_{1,1}^s}{\sqrt{(p_{1,1}^s)^2 + (p_{2,1}^s)^2}} & \frac{p_{2,1}^s}{\sqrt{(p_{1,1}^s)^2 + (p_{2,1}^s)^2}} \\ \frac{-p_{2,1}^s}{\sqrt{(p_{1,1}^s)^2 + (p_{2,1}^s)^2}} & \frac{p_{1,1}^s}{\sqrt{(p_{1,1}^s)^2 + (p_{2,1}^s)^2}} \end{bmatrix}. \tag{C13}$$

Then, we get

$$\left\| \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} (\boldsymbol{\theta}_m^1)^{s+1} - \hat{\imath}_k \right] \right\| = \left\| \left(\boldsymbol{J} \begin{bmatrix} \xi^{s+1} \\ 0 \end{bmatrix} - \boldsymbol{J} \begin{bmatrix} p_{1,1}^s \\ p_{2,1}^s \end{bmatrix} \upsilon^s \right) \right\|, \tag{C14}$$

where

$$(\boldsymbol{\theta}_m^1)^{s+1} = \boldsymbol{\theta}_m^s + {}^{l}\bar{\boldsymbol{\chi}}_m^{s+1} \upsilon^{s+1} = \boldsymbol{\theta}_m^s + \bar{\boldsymbol{\chi}}_{m,1}^{s+1} \upsilon^{s+1}, \tag{C15}$$

the superscript 1 in the parameter estimate $(\boldsymbol{\theta}_m^1)^{s+1}$ denotes that a single search direction is used. By substituting (C13) into (C14), we obtain

$$\left\| \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} (\boldsymbol{\theta}_m^1)^{s+1} - \hat{\imath}_k \right] \right\| = \left\| \left(\begin{bmatrix} \frac{p_{1,1}^s}{\sqrt{(p_{1,1}^s)^2 + (p_{2,1}^s)^2}} \xi^{s+1} \\ \frac{-p_{2,1}^s}{\sqrt{(p_{1,1}^s)^2 + (p_{2,1}^s)^2}} \xi^{s+1} \end{bmatrix} - \begin{bmatrix} \sqrt{(p_{1,1}^s)^2 + (p_{2,1}^s)^2} \upsilon^s \\ 0 \end{bmatrix} \right) \right\|,$$

hence, the step size can be expressed as

$$\upsilon^s = \frac{p_{1,1}^s}{(p_{1,1}^s)^2 + (p_{2,1}^s)^2} \xi^{s+1},$$

which lead to

$$\left\| \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} (\boldsymbol{\theta}_m^1)^{s+1} - \hat{\imath}_k \right] \right\| = \left\| \frac{-p_{2,1}^s}{\sqrt{(p_{1,1}^s)^2 + (p_{2,1}^s)^2}} \xi^{s+1} \right\|.$$

Therefore, we can know that

$$\left\| \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} (\boldsymbol{\theta}_m^1)^{s+1} - \hat{\imath}_k \right] \right\| \leqslant \left\| \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} \boldsymbol{\theta}_m^{s} - \hat{\imath}_k \right] \right\|.$$

The following inequality can be held in the same manner.

$$\left\| \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} \boldsymbol{\theta}_m^{s+1} - \hat{\imath}_k \right] \right\| \leqslant \left\| \sum_{k=1}^{N} \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k \left[y_k - \boldsymbol{\phi}_k^{\mathrm{T}} \boldsymbol{\theta}_m^{s} - \hat{\imath}_k \right] \right\|.$$

□

Algorithm C1 Synergistic Reweighted Expectation (SRE) Algorithm

Require: Input-output data $\{u_k, y_k, w_k\}_{k=1}^N$, model orders n_a, n_b, number of sub-models M, operating points $\mathcal{H}$, relative error ϵ, subspace dimension l.

Ensure: Parameter estimates $\hat{\boldsymbol{\theta}}_m$ for $m = 1, \ldots, M$, outlier estimates $\hat{\lambda}_k$.

Initialize: $\hat{\boldsymbol{\theta}}_m^0$, $(\hat{\sigma}^2)^0$, $\{\hat{o}_m^0\}$, λ via 3σ rule.
for $s = 1$ **do**
  E-step:
  for each $k = 1$ **to** N **do**
    Compute weighting ζ_{m,w_k} and posterior probability $\eta_{(k,m)}$.
  end for
  Outlier Estimation & Reweighting:
  for each $k = 1$ **to** N **do**
    Estimate outlier $\hat{\lambda}_k$ via soft-thresholding.
    Compute adaptive weight $\gamma_k = \frac{1}{(\tau+\beta|\hat{\lambda}_k|)}$.
  end for
  M-step (Krylov Subspace Optimization):
  for each sub-model $m = 1$ **to** M **do**
    Compute gradient direction: $\boldsymbol{\chi}_{m,1}^s = \sum_{k=1}^N \gamma_k \eta_{(k,m)} \boldsymbol{\phi}_k [y_k - \boldsymbol{\phi}_k^{\mathrm{T}} \hat{\boldsymbol{\theta}}_m^s - \hat{\lambda}_k]$.
    Normalize: $\bar{\boldsymbol{\chi}}_{m,1}^s = \boldsymbol{\chi}_{m,1}^s / \|\boldsymbol{\chi}_{m,1}^s\|$.
    Construct Krylov subspace basis $\{\bar{\boldsymbol{\chi}}_{m,1}^s, \ldots, \bar{\boldsymbol{\chi}}_{m,l}^s\}$ via Arnoldi process.
    Form basis matrix ${}^l\bar{\boldsymbol{\chi}}_m^s = [\bar{\boldsymbol{\chi}}_{m,1}^s, \ldots, \bar{\boldsymbol{\chi}}_{m,l}^s]$.
    Solve for v^s via the projected linear system (Theorem 1).
    Update parameters: $\hat{\boldsymbol{\theta}}_m^{s+1} = \hat{\boldsymbol{\theta}}_m^s + {}^l\bar{\boldsymbol{\chi}}_m^s v^s$.
  end for
  Update variance $(\hat{\sigma}^2)^{s+1}$ and widths $\{\hat{o}_m^{s+1}\}$ via nonlinear optimization.
  if $\|\hat{\boldsymbol{\theta}}^{s+1} - \hat{\boldsymbol{\theta}}^s\| < \epsilon$ **then**
    break
  end if
  $s = s + 1$
end for

Appendix C.4 A comprehensive pseudo code of the proposed algorithm

Appendix C.5 Complexity theory analysis

Table C1 Computational complexity of the shared steps in R-EM and SRE (per-iteration, per-submodel m).

Computation task	**Multiplications**	**Additions**
Precompute predictions $y_{m,k} = \boldsymbol{\phi}_k^\top \boldsymbol{\theta}_m$ (for all k)	$N(n_a + n_b)$	$N((n_a + n_b) - 1)$
Compute posterior weights $\eta(k, m)$	N	N
Compute residuals and weighted residual squares for σ^2 update	N	$N - 1$
Update noise variance σ_m^2	N	$N - 1$
Update prior-width objective evaluations	N	$N - 1$
Subtotal cost	$4N + N(n_a + n_b)$	$4N + N(n_a + n_b - 1) - 4$

Table C2 The main computational complexity of the SRE algorithm at each step.

Step	**Details**
Computation task	Update the parameter $\boldsymbol{\theta}_m$
Multiplications	$(n_a + n_b)l + l^3$
Additions	$(n_a + n_b)(l - 1) + l^3 - l^2$
Computation task	Construct the matrix Q_m^s
Multiplications	$N(n_a + n_b)^2$
Additions	$N(n_a + n_b)^2$
Computation task	Construct the vector $\boldsymbol{\chi}_{m,1}^s$
Multiplications	$N[2(n_a + n_b) + 1]$
Additions	$N[2(n_a + n_b) + 1]$
Total cost	$C_{\mathrm{SRE}} = 2l^3 - l^2 + (2l - 1)(n_a + n_b) + 2N(n_a + n_b)^2 + 4N(n_a + n_b) + 2N$

Table C3 The main computational complexity of the R-EM algorithm at each step.

Step	**Details**
Computation task	Update the parameter $\boldsymbol{\theta}_m$
Multiplications	$2N(n_a + n_b)^2 + 2N(n_a + n_b) + (n_a + n_b)^3$
Additions	$N[(n_a + n_b)^2 - 1] + N[(n_a + n_b) - 1] + (n_a + n_b)^3$
Total cost	$C_{\mathrm{R-EM}} = 2(n_a + n_b)^3 + 3N(n_a + n_b)^2 + 3N(n_a + n_b) - 2N$

The count of multiplication and addition of the SRE algorithm and the R-EM algorithm at each step is shown in Tables C2 and C3. The difference in computational load between the two algorithms is

$$\begin{aligned}
&C_{R-EM} - C_{SRE} \\
&= 2(n_a + n_b)^3 + 3N(n_a + n_b)^2 + 3N(n_a + n_b) - 2N \\
&\quad - [2l^3 - l^2 + (2l - 1)(n_a + n_b) + 2N(n_a + n_b)^2 + 4N(n_a + n_b) + 2N] \\
&= 2[(n_a + n_b)^3 - l^3] + l^2 - (2l - 1)(n_a + n_b) + N(n_a + n_b)^2 - N(n_a + n_b) - 4N \\
&= 2[(n_a + n_b)^3 - l^3] + l^2 - (2l - 1)(n_a + n_b) + N[(n_a + n_b)^2 - (n_a + n_b) - 4]. \qquad \text{(C16)}
\end{aligned}$$

Sine $n_a + n_b \geqslant l$, N is much larger than l and $n_a + n_b$, the difference (C16) is positive. For example, let $n_a + n_b = 15$, $l = 5$, $N = 1000$, the difference at each step is

$$C_{EM} - C_{SRE} = 2 \times [15^3 - 5^3] + 5^2 - (2 \times 5 - 1) \times 15 + 1000 \times [15^2 - 15 - 4] = 212390.$$

Table D1 The evaluation of the SRE under different percentages outliers.

Percentage	5%	10%	15%	20%	25%	30%
RMSE	0.1839	0.2068	0.2204	0.2289	0.2472	0.2781

Appendix D

Simulation Environment: All experiments were conducted using MATLAB R2020a on a computer equipped with an Intel Core i5-11500 processor @ 2.70 GHz and 16 GB of RAM. The computational time reported for each algorithm was measured using the built-in *tic* and *toc* functions, ensuring statistical reliability.

The performance of the SRE algorithm is assessed using two examples and compared against other existing methods. To quantitatively evaluate prediction accuracy, the root mean square error (RMSE) and relative error δ are used. These two statistical metrics are defined as follows:

$$\begin{aligned} \mathrm{RMSE}(\hat{y}) &= \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k-\hat{y}_k)^2}, \\ \delta &= \frac{||\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}||}{||\boldsymbol{\theta}||}\times 100\%, \end{aligned}$$

where $\hat{y}_k$ represents the predicted value of y_k, and $\hat{\boldsymbol{\theta}}$ denotes the estimated parameters $\boldsymbol{\theta}$. The reweighted expectation maximization (R-EM), reweighted gradient descent (R-GD) and robust principal component analysis expectation maximization (RPCA-EM) [1] are chosen as comparative algorithms for comparison with the proposed SRE.

Appendix D.1 Mass-spring-damper

To demonstrate the validity and robustness of the proposed SRE algorithm, a mass-spring-damper (MSD) system is employed, as illustrated in Fig. D1. Following the modeling framework outlined in [2], the dynamics of the MSD system can be described by the following differential equation:

$$\frac{d}{dk}\left(m\frac{d}{dk}x(k)\right)+c_d(k)\frac{d}{dk}x(k)+k_s(k)x(k)=F(k), \tag{D1}$$

where $x(k)$ denotes the displacement of the mass at discrete time k, $F(k)$ is the externally applied force, and $c_d(k)$, $k_s(k)$ represent the time-varying damping coefficient and spring stiffness, respectively.

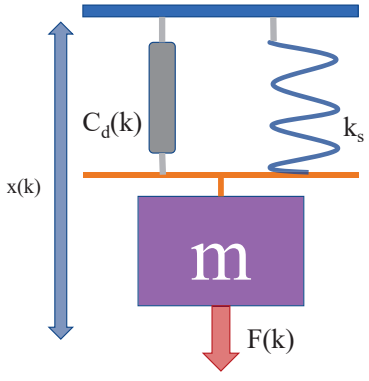

Figure D1 The MSD system.

The time-varying damping behavior influences the system. Therefore, $c_d(k)$ is designated as the scheduling variable. $F(k)$ serves as the system input, while $x(k)$ is treated as the measured output.

To excite the system across varying regimes, a random binary signal is used as the input. Three operating points are defined to reflect distinct dynamic behaviors: $\mathcal{H}_1 = 0.3$, $\mathcal{H}_2 = 0.7$, and $\mathcal{H}_3 = 1.2$. As shown in Fig. D2, the scheduling variable, input, and output signals are depicted, with 10% of the output samples intentionally corrupted by outliers to assess robustness.

The predict outputs of the SRE under the percentage of 10% outliers is provided in Fig. D3. It illustrates that the influence of outliers is minimized through reweighting with a small weight, and the SRE demonstrates a strong ability to capture the dynamics of the true outputs.

To evaluate the robustness of the proposed algorithm, different outlier percentages ranging from 5% to 30% are considered, as shown in Table D1. The data in Table D1 indicate that, although the proportion of outliers increases, the RMSE of the

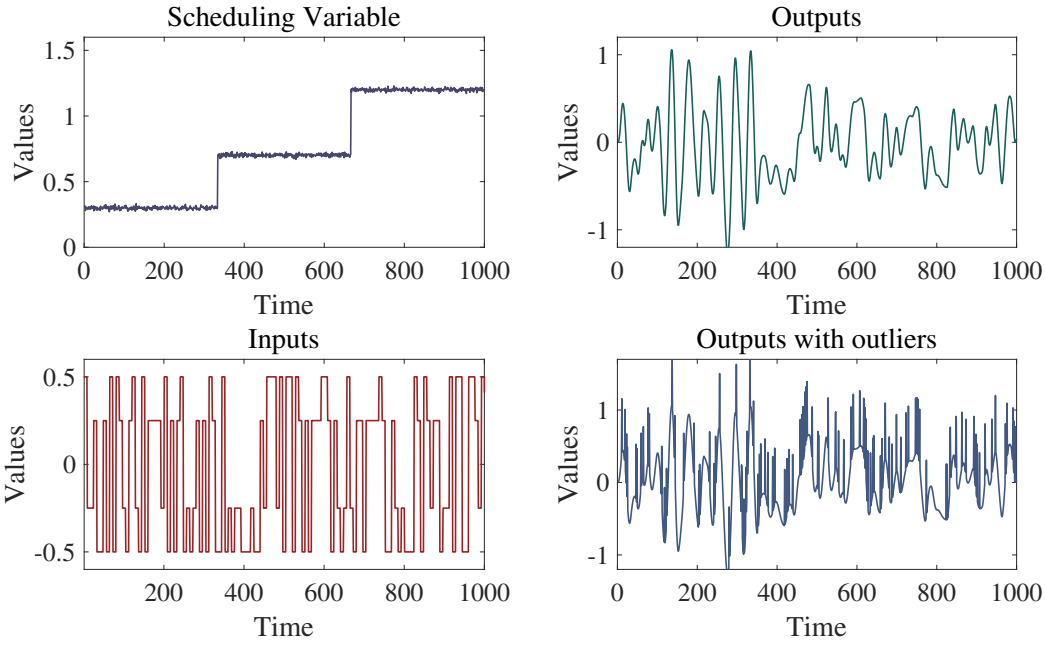

Figure D2 Variables in the MSD system.

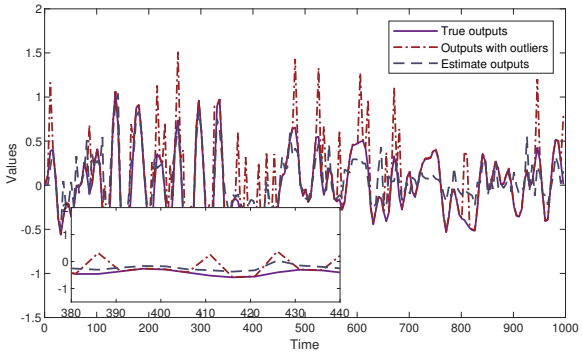

Figure D3 Predict outputs of the SRE under the percentage of 10% outliers.

prediction results exhibits only slight changes. This demonstrates that the algorithm can maintain stable performance even in the different percentages of outliers.

The computational time of the proposed SRE algorithm is evaluated against three established methods using the MSD system benchmark. Table D2 summarizes the key performance metrics. The proposed SRE algorithm completes the identification process in 71.42 seconds. This represents a reduction of 31.27 seconds compared to the RPCA-EM method and 11.81 seconds compared to the R-EM method.

Beyond computation time, the SRE algorithm offers a substantial advantage in memory utilization. The RPCA-EM method, which relies on robust principal component analysis, requires the construction and decomposition of an $N \times N$ matrix. For N=1000 data points, this corresponds to a matrix of 1,000,000 elements. In contrast, the SRE algorithm avoids such large-scale matrices.

The performance of the SRE algorithm with different orthonormal directions for estimating the MSD system with outliers is shown in Fig. D4. It is evident that all orthogonal directions yield good convergence results, with convergence achieved after approximately 350 iterations. This demonstrates that, despite comparing different orthogonal directions, SRE maintains stability.

Table D2 Computation time performance for MSD system

Method	RPCA-EM	R-EM	R-GD	SRE
Elapsed Time (seconds)	102.6943	83.2343	81.5143	71.4236

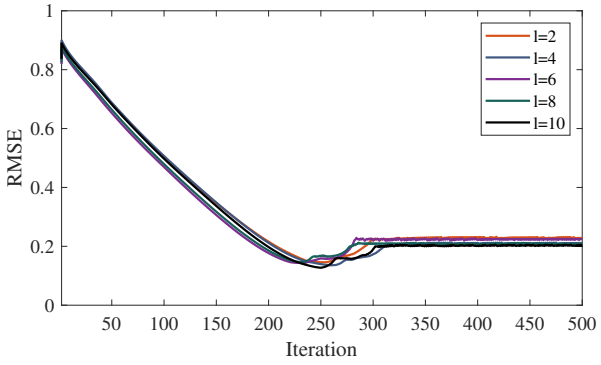

Figure D4 RMSE in different orthogonal directions.

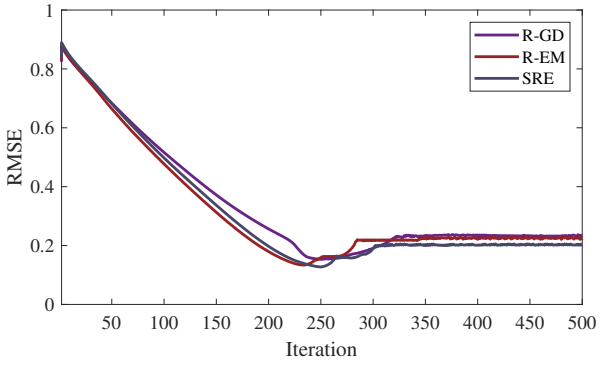

Figure D5 Comparison of RMSE for the MSD under different algorithms.

Fig. D5 illustrates the RMSE of the prediction output, respectively. Under the influence of outliers, all three algorithms converge asymptotically to the estimated model. The convergence rate of the R-GD algorithm is the slowest, while the SRE and R-EM algorithms converge at a faster rate. The prediction performance of the R-GD algorithm is slightly weaker compared to the other two. When the orthogonal direction is $l = 10$, the proposed SRE algorithm requires fewer computational resources than the R-EM algorithm while maintaining similar convergence rates and prediction accuracy.

To further validate the estimated model, another set of operating points, $\mathcal{H} = [0.5, 1.0, 1.5]$, is selected. As shown in Fig. D6, the prediction results yield an RMSE of 0.0698, confirming the model's effectiveness. In comparison, the RPCA-EM method produces a higher RMSE of 0.0783, indicating that the proposed approach achieves more accurate and reliable predictions.

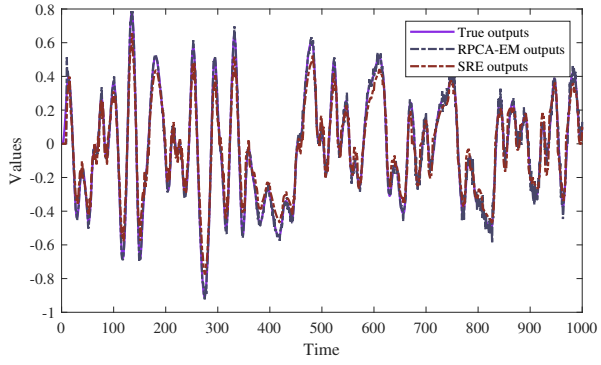

Figure D6 Verification outputs of MSD.

Table D3 Computation time performance for CSTR system

Method	RPCA-EM	R-EM	R-GD	SRE
Elapsed Time (seconds)	783.9417	501.8515	475.3567	452.8139

Appendix D.2 Continuous stirred tank reactor

The continuous stirred tank reactor (CSTR) is a core component in chemical engineering as shown in D7. Its model, derived from heat and mass balance principles, is given by the following two differential equations [3]:

$$\frac{dC_A(k)}{dk} = \frac{q(k)}{V}[C_{A0}(k) - C_A(k)] - k_0 C_A(k)\exp(\frac{-E}{RT(k)}),$$
$$\frac{dT(k)}{dk} = \frac{q(k)}{V}[T_0(k) - T(k)] - \frac{(\Delta\mathcal{H})k_0 C_A(k)}{\rho C_p}\exp(\frac{-E}{RT(k)}) + \frac{\rho_c C_{pc}}{\rho C_p V}q_c(k)\{1 - \exp(\frac{-hA}{q_c(k)\rho C_p})\}[T_{c0}(k) - T(k)],$$

where, $C_A(k)$ represents the outlet reagent concentration, $q_c(k)$ denotes the coolant flow rate, $T(k)$ is the reactor temperature and k denotes the time step. The variable definitions and their steady-state values are referenced from [1].

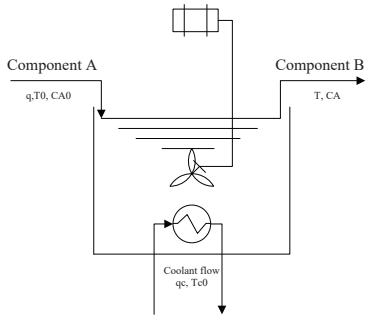

Figure D7 The CSTR system.

We designate the temperature $T(k)$, as the output variable, capturing the thermal dynamics of the system. The control input $q_c(k)$, serves dual roles as both the input and a scheduling variable. The scheduling variable $q_c(k)$ delineates four operational states, given by the set $\mathcal{H} = [93,\ 96,\ 93,\ 96]$. The input and output with outliers are displayed in Fig. D8.

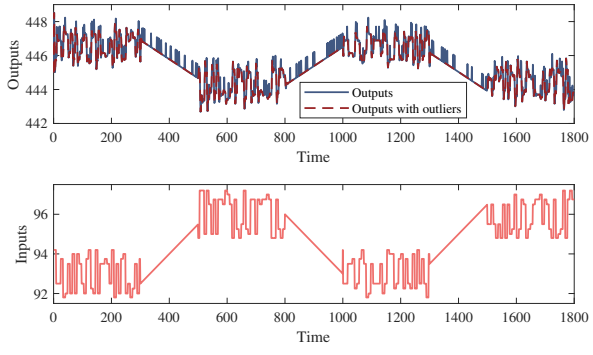

Figure D8 Inputs and outputs with outliers of CSTR.

The proposed SRE algorithm achieves a computation time of 452.81 seconds, representing a reduction of 331.13 seconds compared to the RPCA-EM method as shown in Table D3.

The memory efficiency advantage becomes more pronounced with increased data volume. For N=1800, the RPCA-EM method requires handling an $N \times N$ matrix containing 3,240,000 elements. In contrast, the SRE algorithm maintains its efficient memory utilization pattern of $N \times (n_a + n_b)$, requiring only approximately 10800 elements for typical parameter dimensions, which is approximately 300 times fewer than RPCA-EM.

The predicted outputs of the CSTR with 10% outliers, using the SRE with $l = 5$ orthonormal directions, are presented in Fig. D9. The results demonstrate that the combined sparsity-inducing penalty and reweighting strategy effectively mitigate the impact of outliers on parameter estimation accuracy, thereby enhancing predictive performance.

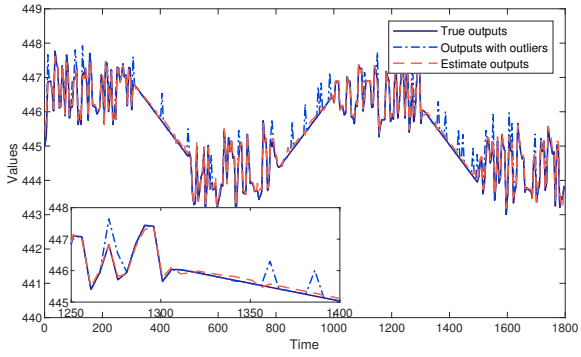

Figure D9 Predicted outputs of the CSTR using SRE with 10% outliers.

Table D4 The evaluation of the CSTR by SRE under different percentages outliers.

Percentage	5%	10%	15%	20%	25%	30%
RMSE	0.1419	0.1221	0.1168	0.1815	0.1793	0.2015

The prediction results for different outlier ratios are presented in Table D4. It indicates that, despite the increasing percentage of outliers, the RMSE of the prediction output fluctuates steadily, demonstrating the algorithm's robustness to outliers.

Fig. D10 compares the prediction results under different orthogonal directions. Furthermore, after 600 iterations, the convergence behavior stabilizes, and the predicted RMSE reaches a steady value, regardless of the number of orthogonal directions. Fig. D11 presents the prediction outputs of different algorithms. It can be observed that the convergence rate

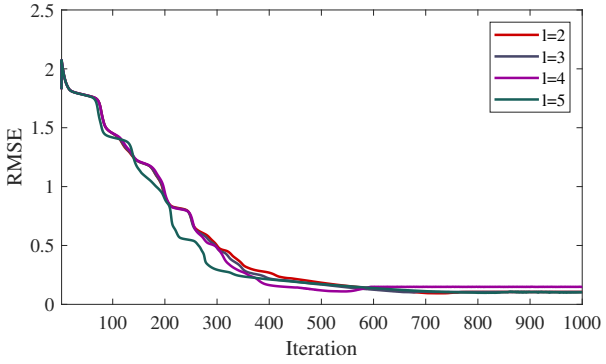

Figure D10 RMSE of CSTR in different orthogonal directions.

of the R-GD algorithm is the slowest, while the SRE and R-EM algorithms exhibit similar convergence rates. To validate the effectiveness of the estimated model parameters, we define a new set of operating points as $\mathcal{H} = [94,\ 97,\ 94,\ 97]$ for verification. As shown in Fig. D12, the proposed algorithm attains an RMSE of 0.0772, whereas the RPCA-EM method yields 0.2871. The lower error confirms the robustness and parameter identification accuracy of the proposed framework in modeling the nonlinear dynamics.

Appendix D.3 Limitations and Future Work

While the proposed algorithm achieves robust identification performance for multi-model systems, several limitations merit further study. The computational gain from the Krylov subspace projection is most significant when the parameter dimension is much larger than the subspace dimension, yet the cost of subspace construction may still be notable in large-scale problems. Moreover, the current framework assumes Gaussian process noise; extending it to explicitly address heavy-tailed or non-Gaussian disturbances would improve generality. In addition, hyperparameters such as the subspace dimension and sparsity factor are determined empirically. Future research will focus on developing data-driven tuning strategies, extending the SRE framework to nonlinear multi-model structures, and formulating an online or recursive version for real-time monitoring and adaptive control.

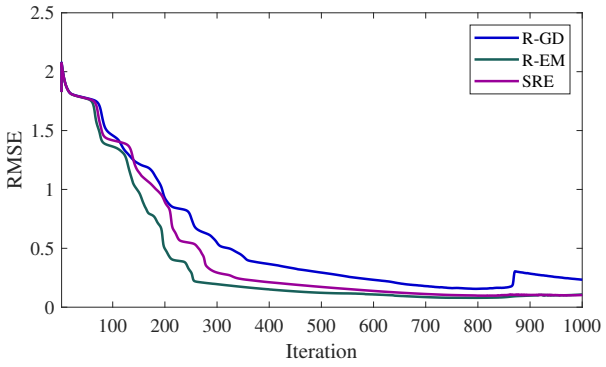

Figure D11 Comparison of RMSE for the CSTR under different algorithms.

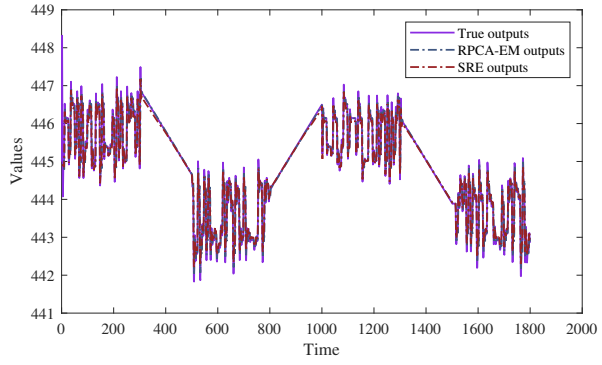

Figure D12 Verification outputs of CSTR.

Appendix D.4 Conclusion

This paper proposed the SRE algorithm to address computational inefficiency and outlier sensitivity in multi-model system identification. The method combines a Krylov subspace-based parameter update, which avoids large matrix inversions through low-dimensional orthogonal projections, with an adaptive reweighting mechanism that dynamically mitigates outlier effects without prior statistical assumptions.

Studies confirmed the algorithm's robustness and efficiency. In the MSD example, the SRE algorithm achieved a computation time of 71.42 seconds, representing a 30.5% reduction compared to the RPCA-EM method (102.69 seconds) while maintaining accuracy under 10% outlier conditions. For the more complex CSTR system with 1800 data points, the computational advantage became more pronounced, with SRE completing identification in 452.81 seconds versus 783.94 seconds for RPCA-EM - a 42.2% improvement in computational efficiency. Beyond time efficiency, the SRE algorithm demonstrated superior memory utilization, requiring storage for approximately 10,000 elements for the MSD system compared to the 1,000,000 elements needed by RPCA-EM - a two-order-of-magnitude reduction that becomes increasingly critical for large-scale industrial applications.

Overall, the SRE algorithm provides a computationally efficient and robust identification tool for noisy and uncertain environments, offering potential for real-time industrial process modeling and control.

References

1 Ma J, Li R, Ma Y, et al. A sparse low-rank matrix recovery strategy to deal with robust identification for multi-model systems with time-varying delays. Signal Process., 2025, 229: 109783
2 Toth R, Laurain V, Gilson M, et al. Instrumental variable scheme for closed-loop LPV model identification. Automatica, 2012, 48: 2314–2320
3 Morningred J D, Paden B E, Seborg D E, et al. An adaptive nonlinear predictive controller. Chem. Eng. Sci., 1992, 47: 755–762