Improved Dynamic Latent Variable Modeling for Process Monitoring, Fault Diagnosis and Anomaly Detection

by

Haitian Zhang

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Chemical Engineering

Waterloo, Ontario, Canada, 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Helen Shang Full Professor, Bharti School of Engineering and Computation
Supervisor(s)	Qinqin Zhu Adjunct Assistant Professor, Chemical Engineering Ali Elkamel Full Professor, Chemical Engineering
Internal Member	Alex Penlidis Full Professor, Chemical Engineering Chandra Madhuranthakam Adjunct Associate Professor, Chemical Engineering
Internal-external Member	Zhao Pan Assistant Professor, Mechanical & Mechatronics Engineering

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Statement of Contributions

I was the sole author for Chapters 1-7, which were written under the supervision of Professor Qinqin Zhu and Professor Ali Elkamel.

The thesis is made up of the following manuscripts written for publication.

Citations:

Chapter 3: Zhang, H., & Zhu, Q. 2022. Concurrent Multilayer Fault Monitoring with Nonlinear Latent Variable Regression. Industrial & Engineering Chemistry Research, 61(3), 1423-1442.

Chapters 4 & 5: Zhang, H., & Zhu, Q. 2023. An improved dynamic latent variable regression model for fault diagnosis and causal analysis. The Canadian Journal of Chemical Engineering.

Chapter 6: Zhang, H., Fgaier, H., Zhu, Q., Ahmadian, A., & Elkamel, A. 2023. An Improved Autoencoder with Dynamic Hidden Layer for Anomaly Detection Monitoring, Proceedings of the 6th European Conference on Industrial Engineering and Operations Management (Accepted and Presented)

As the sole author of all the work above, I was responsible for conceptualizing study design, carrying out data process and analysis, and drafting and submitting manuscripts. My coauthors provided guidance during each step of the research and provided feedback on draft manuscripts.

Abstract

Due to the rapid advancement of modern industrial processes, a considerable number of measured variables enhance the complexity of systems, progressively leading to the development of multivariate statistical analysis (MSA) methods to exploit valuable information from the collected data for predictive modeling, fault detection and diagnosis, such as partial least squares (PLS), canonical correlation analysis (CCA) and their extensions. However, these methods suffer from some issues, involving the irrelevant information extracted by PLS, and CCA's inability to exploit quality information. Latent variable regression (LVR) was designed to address these issues, but it has not been fully and systematically studied.

A concurrent kernel LVR (CKLVR) with a regularization term is designed for collinear and nonlinear data to construct a full decomposition of the original nonlinear data space, and to provide comprehensive information of the systems. Further, dynamics are inevitable in practical industrial processes, and thus a dynamic auto-regressive LVR (DALVR) is also proposed based on regularized LVR to capture dynamic variations in both process and quality data. The comprehensive monitoring framework and fault diagnosis and causal analysis scheme based on DALVR are developed. Their superiority can be demonstrated with case studies, involving the Tennessee Eastman process, Dow's refining process and three-phase flow facility process.

In addition to MSA approaches, autoencoder (AE) technology is extensively used in complicated processes to handle the expanding dimensionality caused by the increasing complexity of industrial applications. Apart from modeling and fault diagnosis, anomaly detection draws great attention as well to maintain the performance, avoid economic losses, and ensure safety during the industrial processes. In view of advantages in dimensionality reduction and feature retention, autoencoder (AE) technology is widely applied for anomaly detection monitoring. Considering both high dimensionality and dynamic relations between elements in the hidden layer, an improved autoencoder with dynamic hidden layer (DHL-AE) is proposed and applied for anomaly detection monitoring. Two case studies including Tennessee Eastman process and Wind data are used to show the effectiveness of the proposed algorithm.

Acknowledgements

I would like to express my sincere gratitude to all the individuals and institutions who have supported and contributed to the completion of this doctoral research.

First and foremost, I extend my deepest appreciation to my supervisors, Professors Qinqin Zhu and Ali Elkamel, for their invaluable guidance, unwavering support, and endless patience throughout this journey. Without their support and guidance, this thesis would not have been possible.

Moreover, I am indebted to the members of my thesis committee, Professor Helen Shang, Professor Alex Penlidis, Professor Chandra Madhuranthakam, and Professor Pan Zhao, for their valuable insights, constructive feedback, and critical review of this work. Their diverse perspectives and constructive criticisms have undoubtedly enhanced the quality of this thesis.

I am grateful to my coworker, Dr. Mohamed EL KOUJOK, research scientist at CanmetENERGY in Varennes, for the camaraderie and intellectual discussions, which have enriched my understanding of the subject matter and made the research process more enjoyable.

I am deeply appreciative of my family for their unconditional love, understanding, and encouragement throughout my academic pursuits. Your support has been the bedrock of my perseverance. I want to thank all my friends for providing me with constant encouragement and help during challenging times.

Lastly, I wish to thank the entire department of chemical engineering community for providing an intellectually stimulating environment in which to pursue my degree.

To all those mentioned above and to those whose names I have inadvertently omitted, your contributions, whether big or small, have played a crucial role in making this research possible. Thank you for being a part of this journey.

Dedication

To my dear parents, my beloved husband, and my lovely kid with my everlasting love.

Table of Contents

List of Tables

List of Figures

Nomenclature

$a_{ij,l}$	auto-regressive coefficients ($l = 1, 2, \dots, r$) for two time series $\mathbf{x}_i$ and $\mathbf{x}_j$
$\mathbf{a}_i^{(t)}$	anomaly score for $\mathbf{x}_i^{(t)}$
$[A_{ij}(f)]$	coefficient matrix for Fourier transformation at frequency f (i is the row index, and j is the column index)
AE	autoencoder
AR	accuracy rate
ARIMA	autoregressive integrated moving average
ARX	auto-regressive exogenous
b	regression coefficient of relation between process/input and quality/output scores
$\mathbf{b}_1$	bias of encoder
$\mathbf{b}_2$	bias of decoder
$\mathbf{B}$	diagonal matrix for regression coefficient of relation between process/input and quality/output scores
c	width of Gaussian function
$\mathbf{c}$	loading vector for dynamic part in quality/output matrix
$\mathbf{c}_{st}$	loading vector for static part in quality/output matrix
$\mathbf{C}$	loading matrix for quality/output matrix
CCA	canonical correlation analysis
CDALVR	concurrent dynamic auto-regressive latent variable regression
CKLVR	concurrent kernel latent variable regression

CSGC	conditional spectral Granger causality
$\mathrm{CSGC}_{\mathbf{x}_j \to \mathbf{x}_i \mid \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_v}(f)$	CSGC index from the j^{th} variable to the i^{th} variable $\mathbf{x}_i$ exclude at frequency f
CVS	covariation subspace
d	dynamic order for quality/output variables
d_a	degree of differencing for ARIMA
$\mathbf{D}_c$	rectangular diagonal matrix in SVD for CVS
DALVR	dynamic auto-regressive latent variable regression
DAOS	dynamic auto-regressive output subspace
DAPLS	dynamic auto-regressive partial least squares
DCVS	dynamic covariation subspace
DHL-AE	autoencoder with dynamic hidden layer
DiPCA	dynamic inner principal component analysis
DiPLS	dynamic inner partial least squares
DIPS	dynamic input principal subspace
DKPCA	dynamic kernel principal component analysis
DrLVR	dynamic regularized latent variable regression
$\mathbf{e}$	moving average part in ARIMA
$\boldsymbol{e}_i(k)$	residual/prediction errors of the i^{th} variable
$\boldsymbol{e}_{i(j)}(k)$	prediction error that excludes the j^{th} variable to predict the i^{th} variable
$\mathbf{e}_t$	moving average part existing in the input $\mathbf{z}$ at time t in ARIMA
$\mathbf{e}^{(t)}$	moving average part $\mathbf{e}$ existing in the latent variables at time t in DHL-AE

$E_i(f)$	representation of $A_{ii}(f) \times \mathbf{x}_i(f) + A_{ij}(f) \times \mathbf{x}_j(f)$ in matrix calculation (i is the row index, and j is the column index)
ELU	exponential linear unit
f	frequency
f_i	fault magnitude for a faulty sample
f_i^{φ}	fault magnitude for a faulty sample for φ index
f_x	fault magnitude of input
f_y	fault magnitude of output
f_1	activation function used to construct the hidden layer
f_2	activation function used to construct the output layer
F	Hilbert space
$F_{l,n-l}$	F-distribution with l and $n-l$ degrees of freedom
$F_{\mathbf{x}_j \to \mathbf{x}_i}$	TDGC index from the j^{th} variable to the i^{th} variable
FAR	false alarm rate
FDR	false detection rate
g	representation of $\max(s, d)$
g_{dx}	scaling factor of $\varphi_{dx,\text{cl}}$
g_s	scaling factor
g_{sx}	scaling factor of $Q_{sx,\text{cl}}$
g_{sy}	scaling factor of $Q_{sy,\text{cl}}$
g_x	scaling factor of $Q_{x,\text{cl}}$
g_y	scaling factors of $Q_{y,\text{cl}}$
g^{φ}	scaling factor of control limit of φ statistic

$\mathbf{G}$	spectral transfer function matrix of incomplete system
$\tilde{G}_{ii}$	elements in matrix $\widetilde{\mathrm{G}}_{jj} = \mathrm{G}_{jj} - \mathrm{G}_{ij}^2/\mathrm{G}_{ii}$ (i is the row index, and j is the column index)
GMM	Gaussian mixture model
h	degree of freedom
h_{dx}	degree of freedom of χ^2-distribution used in $\varphi_{dx,\mathrm{cl}}$
h_x	degree of freedom of χ^2-distribution used in $Q_{x,\mathrm{cl}}$
h_{sx}	degree of freedom of χ^2-distribution used in $Q_{sx,\mathrm{cl}}$
h_{sy}	degree of freedom of χ^2-distribution used in $Q_{sy,\mathrm{cl}}$
h_y	degree of freedom of χ^2-distribution used in $Q_{y,\mathrm{cl}}$
h^{φ}	degrees of freedom of control limit of φ statistic
$\mathbf{H}$	spectral transfer function matrix of complete system
$H_{ij}(f)$	element in spectral transfer function matrix at frequency f (i is the row index, and j is the column index)
$\mathbf{I}$	identity matrix
$\mathbf{I}_d$	$d \times d$ identity matrix
$\mathbf{I}_m$	$m \times m$ identity matrix
$\mathbf{I}_p$	$p \times p$ identity matrix
$\mathbf{I}_{s+1}$	$(s+1) \times (s+1)$ identity matrix
ICA	independent component analysis
$\mathrm{Index}(\mathbf{z}_i)$	reconstructed fault detection index for $\mathbf{z}_i$
$\mathrm{Index}(\mathbf{z}_i)^{\varphi}$	reconstructed fault detection index for $\mathbf{z}_i$ based on φ index
IPS	input-principal subspace

IRS	input-residual subspace
k	dimension of compressed representation in autoencoder
$\mathbf{k}_t$	representation of $\mathbf{\Phi}\phi(\mathbf{x}_{\text{new}})$
$k(\mathrm{x}_i, \mathrm{x}_j)$	kernel function
$\mathbf{K}$	kernel matrix
$\mathbf{K}_c$	kernel matrix of $\widetilde{\mathbf{\Phi}}_c$
$\mathbf{K}_t$	kernel matrix for test samples $\mathbf{X}_t$
KPCA	kernel principal component analysis
KLVR	kernel latent variable regression
KPLS	kernel partial least squares
l	latent variable number
l_c	principal component number in CVS
l_{dy}	principal component number in DAOS
l_{sx}	principal component number in SIPS
l_{sy}	principal component number in SOPS
l_x	principal component number in IPS
l_y	principal component number in OPS
$L(\cdot)$	loss function
L_1	predictable quality/output-relevant fault alarm
L_2	unpredictable quality/output-relevant fault alarm
L_3	process/input-relevant and/or quality/output-irrelevant fault alarm
$\mathcal{L}$	Lagrangian function
Leaky ReLU	leaky rectified linear unit

LFP	local-field potential
LVR	latent variable regression
m	process/input variable number
m_f	latent variable number in kernel
m'	latent variable number in hidden layer
$\mathbf{M}$	general matrix determined by corresponding monitoring index
MAE	mean absolute error
MAR	missing alarm rate
MSE	mean squared error
MSA	multivariate statistical analysis
M2D-CCA	multi-objective two-dimensional canonical correlation analysis
n	sample number
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
NN	neural network
NRMSE	normalized root mean squared error
OPS	output-principal subspace
ORS	output-residual subspace
p	quality/output variable number
p_a	number of time lags for ARIMA
$\mathbf{P}$	loading matrix for process/input variables
$\mathbf{P}_{dx}$	loading matrix of dynamic process/input variations
$\mathbf{P}_{sx}$	loading matrix of static process/input variations
$\mathbf{P}_{sy}$	loading matrix of static quality/output principal variations

$\mathbf{P}_x$	input-principal loading matrix in CKLVR
$\mathbf{P}_y$	output-principal loading matrix in CKLVR
PCA	principal component analysis
PCS	principal component subspace
PLS	partial least squares
PR	precision rate
$\mathbf{q}$	quality/output weighting vector
q_a	order of moving average part
$\mathbf{Q}$	quality/output score matrix
Q	Q statistic
$\mathbf{Q}_c$	representation of $\mathbf{V}_c\mathbf{D}_c$
Q_{cl}	control limit of Q statistic
Q_{dx}	Q statistic for DIPS in CDALVR
$Q_{dx,\text{cl}}$	control limit of Q_{dx} statistic
$Q_{ii}(f)$	spectral density of the target variable $\mathbf{x}_i$ in incomplete system at frequency f
Q_{sx}	Q statistic for SIRS in CDALVR
Q_{sy}	Q statistic for SORS in CDALVR
Q_x	Q statistic for IRS in CKLVR
$Q_{x,\text{cl}}$	control limit of Q_x statistic
Q_y	Q statistic for ORS in CKLVR
$Q_{y,\text{cl}}$	control limit of Q_y statistic
r	auto-regressive model order

rLVR	regularized latent variable regression
rRBC	relative reconstruction-based contribution
$\mathrm{rRBC}_i^{\varphi}$	relative reconstruction-based contribution of the i^{th} variable for φ index
$\mathbf{R}$	representation of $\mathbf{W}(\mathbf{P}^{\top}\mathbf{W})^{-1}$
$\mathbf{R}_c$	representation of $\mathbf{R}\mathbf{C}^{\top}\mathbf{V}_c\mathbf{D}_c^{-1}$
$\mathbf{R}_c^{\dagger}$	representation of $(\mathbf{R}_c^{\top}\mathbf{R}_c)^{-1}\mathbf{R}_c^{\top}$
$\mathbf{R}_x$	parameters for $\mathbf{t}_{dx,k}$ determined by DiPCA
$\mathbb{R}^{n\times m}$	$n \times m$ matrix
RBC	reconstruction-based contribution
$\mathrm{RBC}_i^{\mathrm{Index}}$	amount of reconstruction along the fault direction ξ_i
$\mathrm{RBC}_{\mathrm{avg},i}^{\varphi}$	average values of the i^{th} variable calculated with RBC_i^{φ} obtained from normal data
RBC_i^{φ}	amount of reconstruction along the fault direction ξ_i for φ index
ReLU	rectified linear unit
RE	reconstruction error
RMSE	root mean squared error
RS	residual subspace
s	dynamic order for process/input variables
$\mathbf{S}$	spectral density matrix of complete system
$S_{ii}(f)$	spectral density of the target variable $\mathbf{x}_i$ in complete system at frequency f
$\mathbf{S}_x$	covariance matrix of input
SGC	spectral Granger causality

$\mathrm{SGC}_{\mathbf{x}_j \to \mathbf{x}_i}(f)$	SGC index from the j^{th} variable to i^{th} variable at frequency f
SIPS	static input principal subspace
SIRS	static input residual subspace
SOPS	static output principal subspace
SORS	static output residual subspace
Span{·}	span of a set of vectors or matrix
SVD	singular value decomposition
SVM	support vector machine
$\mathbf{t}$	process/input score vector
$\hat{\mathbf{t}}$	predicted process/input score vector
$\mathbf{t}_{dx,k}$	score vector in DIPS for $\mathbf{x}_k$
$\hat{\mathbf{t}}_{dx,k}$	predicted score vector in DIPS for $\mathbf{x}_k$
$\mathbf{t}_i$	process/input score vector corresponding to the i^{th} latent variable $(i = 1, 2, \ldots, l)$
t_k	process/input score of $(\mathbf{x}_k, \mathbf{y}_k)$
$\mathbf{t}_{\text{new}}$	process/input score vector of $\mathbf{x}_{\text{new}}$
$\mathbf{t}_{sx,k}$	score vector in SIPS for $\mathbf{x}_k$
$\mathbf{t}_{sy,k}$	static output score vector in SOPS for $\mathbf{x}_k$
$\mathbf{t}_{x,\text{new}}$	process-relevant/input-principal score vector of $\mathbf{x}_{\text{new}}$
$\mathbf{t}_{y,\text{new}}$	unpredictable quality-relevant/output-principal score vector of $\mathbf{x}_{\text{new}}$
$tr(\cdot)$	trace of a square matrix
$\mathbf{T}$	process/input score matrix
$\hat{\mathbf{T}}_{dx}$	predicted score matrix of dynamic process/input variations

$\mathbf{T}_s$	representation of $\mathbf{Z}_x(\mathbf{I}_{s+1} \otimes \mathbf{w})$
$\mathbf{T}_{sx}$	score matrix of static process/input variations
$\mathbf{T}_{sy}$	score matrix of static quality/output principal variations
$\mathbf{T}_x$	process/input-principal score matrix in CKLVR
$\mathbf{T}_y$	quality/output-principal score matrix in CKLVR
T^2	Hotelling's T^2 statistic
T_c^2	Hotelling's T^2 statistic for CVS in CKLVR / DCVS in CDALVR
$T_{c,\mathrm{cl}}^2$	control limit of T_c^2 statistic
T_{cl}^2	control limit of Hotelling's T^2 statistic
T_{dx}^2	Hotelling's T^2 statistic for DIPS in CDALVR
$T_{dx,\mathrm{cl}}^2$	control limit of T_{dx}^2 statistic
T_{dy}^2	Hotelling's T^2 statistic for DAOS in CDALVR
T_{sx}^2	Hotelling's T^2 statistic for SIPS in CDALVR
T_{sy}^2	Hotelling's T^2 statistic for SOPS in CDALVR
T_x^2	Hotelling's T^2 statistic for IPS in CKLVR
$T_{x,\mathrm{cl}}^2$	control limit of T_x^2 statistic
T_y^2	Hotelling's T^2 statistic for OPS in CKLVR
$T_{y,\mathrm{cl}}^2$	control limit of T_y^2 statistic
TDGC	time-domain Granger causality
TEP	Tennessee Eastman process
TPFF	three-phase flow facility
$\mathbf{u}$	quality/output score vector

$\mathbf{u}_{c,k}$	dynamic output-relevant score vector in DCVS for $\mathbf{x}_k$
$\mathbf{u}_{c,\text{new}}$	predictable quality/output-relevant score vector for $\mathbf{x}_{\text{new}}$ in CKLVR
$\mathbf{u}_g$	quality/output score vector containing dynamic cross- and auto-correlations
$\hat{\mathbf{u}}_g$	predicted quality/output score vector containing dynamic cross- and auto-correlations
$\hat{\mathbf{u}}_{g,i}$	predicted quality/output score vectors containing dynamic cross- and auto-correlations corresponding to the i^{th} latent variable $(i = 1, 2, \ldots, l)$
u_k	quality/output score of $(\mathbf{x}_k, \mathbf{y}_k)$
$\hat{u}_k$	predicted quality/output score of $(\mathbf{x}_k, \mathbf{y}_k)$
$\mathbf{u}_s$	quality/output score vector for $\mathbf{Y}_s$
$\hat{\mathbf{u}}_{\psi,k}$	auto-regressive output score vector in DAOS for $\mathbf{x}_k$
$\mathbf{U}_c$	predictable quality/output score matrix in CKLVR
$\mathbf{U}_d$	representation of $\mathbf{Z}_y(\mathbf{I}_d \otimes \mathbf{q})$
$\mathbf{U}_{in}$	spectral density matrix of system with one variable excluded for the causality test
$\hat{\mathbf{U}}_g$	predicted quality/output score matrix containing dynamic cross- and auto-correlations
$\mathbf{U}_s$	quality/output score matrix for $\mathbf{Y}_s$
$\hat{\mathbf{U}}_s$	predicted quality/output score matrix for $\mathbf{Y}_s$
$\hat{\mathbf{U}}_\psi$	representation of $\mathbf{Z}_y(\boldsymbol{\psi} \otimes \mathbf{Q})$
$\mathcal{U}([0,1])$	uniform distribution in the interval [0,1]
$\mathbf{v}$	input vector for ARIMA

$\mathbf{v}_t$	aggressive part existing in the input $\mathbf{z}$ at time t in ARIMA
$\mathbf{v}^{(t)}$	aggressive part existing in the latent variables at time t in DHL-AE
$\mathrm{var}(\cdot)$	variance of a vector
$\mathbf{V}_c$	complex unitary matrix in SVD for CVS
VAE	variational autoencoder
VAR	vector-autoregressive
$\mathbf{w}$	process/input weighting vector
$\mathbf{W}$	process/input weighting matrix
$\mathbf{W}_x$	input-principal weighting matrix in CKLVR
$\mathbf{W}_1$	weights of encoder
$\mathbf{W}_2$	weights of decoder
$\mathbf{x}$	process/input vector
$\mathbf{x}^*$	fault-free input samples
$\tilde{\mathbf{x}}_c$	process/input vector after extraction of dynamic and auto-regressive variations for $\mathbf{x}_k$
$\mathbf{x}_i$	the i^{th} vector of process/input matrix
$\mathbf{x}_i^{(t)}$	element at time t in the i^{th} vector of process/input matrix
$\tilde{\mathbf{x}}_k$	residual vector in SIRS for $\mathbf{x}_k$
$\mathbf{x}_{\text{new}}$	new input sample for testing
$\hat{\mathbf{x}}_{\text{new}}$	representation of $\mathbf{P}\mathbf{R}^\top \mathbf{x}_{\text{new}}$
$\tilde{\mathbf{x}}_{\text{new}}$	representation of $(\mathbf{I} - \mathbf{P}\mathbf{R}^\top)\,\mathbf{x}_{\text{new}}$
$\mathbf{x}_{\zeta,k}$	representation of $\sum_{i=0}^{s} \zeta_i \mathbf{x}_{k-i}$
$\mathbf{X}$	process/input matrix

$\widetilde{\mathbf{X}}$	residual of process/input matrix in DALVR
$\mathbf{X}_i$	a subset of the collected input samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{s+N}]^\top$
$\mathbf{X}_r$	residual of process/input matrix
$\widetilde{\mathbf{X}}_s$	static process/input residual
$\mathbf{X}_t$	input matrix for testing
$\mathbf{X}_\beta$	representation of $\mathbf{Z}_x(\boldsymbol{\beta} \otimes \mathbf{I}_m)$
$\mathbf{X}_\zeta$	representation of $[\mathbf{Z}_x(\boldsymbol{\zeta} \otimes \mathbf{I})]$
$\mathbf{y}$	quality/output vector
$\mathbf{y}^*$	fault-free output samples
$\tilde{\mathbf{y}}_c$	quality/output vector after extraction of dynamic and auto-regressive variations
$\mathbf{y}_i$	the i^{th} sample of quality/output matrix
$\tilde{\mathbf{y}}_k$	static output residual vector in SORS for $\mathbf{y}_k$
$\hat{\boldsymbol{y}}_{\text{new}}$	predicted output of $\mathbf{x}_{\text{new}}$
$\mathbf{Y}$	quality/output matrix
$\widehat{\mathbf{Y}}$	predicted quality/output matrix
$\widetilde{\mathbf{Y}}$	output residual matrix
$\mathbf{Y}_c$	quality/output matrix after extraction of predictable dynamic and auto-regressive variations
$\widetilde{\mathbf{Y}}_c$	unpredictable output matrix in CKLVR
$\mathbf{Y}_g$	dynamic part in quality/output matrix
$\widehat{\mathbf{Y}}_g$	dynamic predictable quality/output matrix
$\widetilde{\mathbf{Y}}_g$	residual of dynamic part in quality/output matrix

$\mathbf{Y}_r$	residuals of quality/output matrix
$\mathbf{Y}_s$	a subset of the collected output samples $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{s+N}]^\top$
$\widehat{\mathbf{Y}}_{xg}$	*dynamic* cross-correlation quality/output part in quality/output matrix
$\widehat{\mathbf{Y}}_{yg}$	*dynamic* auto-correlation quality/output part in quality/output matrix
$\mathbf{Y}_\delta$	representation of $\mathbf{Z}_y(\boldsymbol{\delta} \otimes \mathbf{I}_p)$
$\mathbf{Y}_{0:g-1}$	static part in quality/output matrix
$\mathbf{z}$	reconstructed vector in hidden layer
$\hat{\mathbf{z}}$	predicted reconstructed vector in hidden layer
$\mathbf{z}_i$	reconstructed vector for a fault-free sample
$\mathbf{z}_t$	component of time series $\mathbf{z}$ at time t
$\mathbf{z}_x$	variables that may influence time serise $\mathbf{x}_i$ and $\mathbf{x}_j$ $(x = 1, 2, \ldots, v)$
$\mathbf{z}^{(t)}$	compressed representation vector for autoencoder at time t $(t = 1, 2, \ldots, n)$
$\hat{\mathbf{z}}^{(t)}$	predicted compressed representation vector for autoencoder at time t $(t = 1, 2, \ldots, n)$
$\mathbf{Z}$	compressed representation matrix for autoencoder
$\hat{\mathbf{Z}}$	predicted compressed representation matrix for autoencoder
$\mathbf{Z}_s$	representation of $[\mathbf{X}_s, \mathbf{X}_{s-1}, \ldots, \mathbf{X}_1]$
$\mathbf{Z}_x$	representation of $[\mathbf{X}_g, \mathbf{X}_{g-1}, \ldots, \mathbf{X}_{g-s}]$
$\mathbf{Z}_y$	representation of $[\mathbf{Y}_{g-1}, \mathbf{Y}_{g-2}, \ldots, \mathbf{Y}_{g-d}]$
Z	space in the hidden layer

α	confidence interval
$\boldsymbol{\alpha}$	nonlinear weighting vector for $\mathbf{K}$
$\boldsymbol{\alpha}_{\boldsymbol{\Phi}}$	nonlinear projection vector for $\boldsymbol{\Phi}$
$\mathbf{A}$	nonlinear weighting matrix
β_x	weighting coefficient elements in $\boldsymbol{\beta}$ $(x = 1, 2, \dots, s)$
$\boldsymbol{\beta}$	weighting coefficient vector for process/input scores
$\beta_{a,i}$	parameters of the autoregressive part for ARIMA $(i = 1, 2, \dots, p_a)$
B	backshift operator
γ	regularized parameter
γ_w	regularized parameter for $\mathbf{w}$
γ_β	regularized parameter for $\boldsymbol{\beta}$
γ_δ	regularized parameter for $\boldsymbol{\delta}$
δ_x	weighting coefficient elements in $\boldsymbol{\delta}$ $(x = 1, 2, \dots, d)$
$\boldsymbol{\delta}$	weighting coefficient vector for quality/output scores
$\boldsymbol{\Gamma}$	error covariance matrix of incomplete system
$\boldsymbol{\Gamma}_c$	representation of $(\mathbf{D}_c^{-\top}\mathbf{V}_c^{\top}\mathbf{C}(\mathbf{T}^{\top}\mathbf{K}\mathbf{A})^{-\top}\mathbf{A}\mathbf{K}\mathbf{A}(\mathbf{T}^{\top}\mathbf{K}\mathbf{A})^{-1}\mathbf{C}^{\top}\mathbf{V}_c\mathbf{D}_c^{-1})^{-1}$ $\mathbf{D}_c^{-\top}\mathbf{V}_c^{\top}\mathbf{C}(\mathbf{T}^{\top}\mathbf{K}\mathbf{A})^{-\top}\mathbf{A}$
$\boldsymbol{\varepsilon}_g$	regression error for dynamic inner structure in DALVR
ε_k	regression error for inner structure
$\boldsymbol{\varepsilon}_{PLS}$	error of relation between process/input and quality/output scores
$\boldsymbol{\varepsilon}_t$	white noise error term at time t
$\boldsymbol{\varepsilon}^{(t)}$	reconstruction error of an input vector $\mathbf{x}^{(t)}$
$\boldsymbol{\varepsilon}_i^{(t)}$	reconstruction error of $\mathbf{x}_i$ at time t

ζ_x	elements in predicted weighting vector of process/input matrix ($x = 0, 1, 2, \dots, s$)
$\boldsymbol{\zeta}$	predicted weighting vector of process/input matrix
θ_i	parameters of moving average part in ARIMA ($i = 1, 2, \dots, q_a$)
$\boldsymbol{\Theta}_x$	dynamic weighting coefficient vector for $\hat{\mathbf{t}}_{dx,k}$
κ	representation of γ/λ_α
κ_q	representation of $\lambda_{q\delta}/\lambda_q$
κ_w	representation of $\gamma_w/\lambda_{w\beta}$
κ_β	representation of $\gamma_\beta/\lambda_{w\beta}$
κ_δ	representation of $\gamma_\delta/\lambda_{q\delta}$
λ_q	Lagrange multiplier for $\mathbf{q}$
$\lambda_{q\delta}$	Lagrange multiplier for $\mathbf{q}$ and $\boldsymbol{\delta}$
$\lambda_{w\beta}$	Lagrange multiplier for $\mathbf{w}$ and $\boldsymbol{\beta}$
λ_α	Lagrange multiplier for $\boldsymbol{\alpha}$
$\boldsymbol{\Lambda}$	variance matrix of latent variables
$\boldsymbol{\Lambda}_c$	variance matrix of $\mathbf{U}_c$
$\boldsymbol{\Lambda}_{dx}$	variance matrix of $\widehat{\mathbf{T}}_{dx}$
$\boldsymbol{\Lambda}_{dy}$	variance matrix of $\widehat{\mathbf{U}}_\psi$
$\boldsymbol{\Lambda}_{sx}$	variance matrix of $\mathbf{T}_{sx}$
$\boldsymbol{\Lambda}_{sy}$	variance matrix of $\mathbf{T}_{sy}$
$\boldsymbol{\Lambda}_x$	variance matrix of process/input variables
$\boldsymbol{\Lambda}_y$	variance matrix of quality/output variables
$\boldsymbol{\mu}$	mean value of a matrix

ξ_i	fault direction for a faulty sample
$\mathbf{\Xi}_x$	process/input matrix with faulty samples
$\mathbf{\Xi}_y$	quality/output matrix with faulty samples
χ	space of input and output data
$\chi^2_{h,\alpha}$	χ^2-distribution with h degrees of freedom
σ	density of Gaussian kernel function
$\mathbf{\Sigma}$	error covariance matrix of complete system
τ	threshold of anomaly scores
$\phi(\cdot)$	nonlinear projection indicator
$\tilde{\phi}_c(\mathbf{x}_{\text{new}})$	nonlinear input vector for $\mathbf{x}_{\text{new}}$ after extraction of predictable variations
$\phi_r(\mathbf{x}_{\text{new}})$	residual of $\phi(\mathbf{x}_{\text{new}})$
φ	single statistic combined T^2 with Q
φ_{cl}	control limit of φ
φ_{dx}	φ statistic for DIPS in CDALVR
$\varphi(\mathbf{x})$	process of encoding
$\varphi: \chi \to Z$	projection from the space of input and output data to the space in the hidden layer
$\mathbf{\Phi}$	input matrix after nonlinear mapping
$\widetilde{\mathbf{\Phi}}$	nonlinear input residual
$\widetilde{\mathbf{\Phi}}_c$	nonlinear input matrix after extraction of predictable variations
$\mathbf{\Phi}_{dx}$	representation of $\frac{\mathbf{P}_{dx}\mathbf{\Lambda}_{dx}^{-1}\mathbf{P}_{dx}^{\top}}{T^2_{dx,\text{cl}}} + \frac{\mathbf{I}-\mathbf{P}_{dx}\mathbf{P}_{dx}^{\top}}{Q_{dx,\text{cl}}}$
$\mathbf{\Phi}_{\varphi}$	representation of $\frac{\mathbf{R}\mathbf{\Lambda}^{-1}\mathbf{R}^{\top}}{T^2_{\text{cl}}} + \frac{\mathbf{I}-\mathbf{P}\mathbf{R}^{\top}}{Q_{\text{cl}}}$

ψ_x	elements in predicted weighting vector of quality/output matrix ($x = 0, 1, 2, \dots, d$)
$\psi[\varphi(\mathbf{x})]$	process of decoding
$\psi: Z \to \chi$	projection from the space in the hidden layer to the space of input and output data
$\boldsymbol{\psi}$	predicted weighting vector of quality/output matrix
$\mathbf{1}_k$	all-one vector of length k
$\mathbf{1}_n$	all-one vector of length n
$\langle \mathbf{a}, \mathbf{b} \rangle$	dot product between vectors **a** and **b**
$\mathbf{M}^*$	corresponding adjoint matrix of a known matrix **M**
$\widetilde{\mathrm{M}}_{jj}$	calculation as $\widetilde{\mathrm{M}}_{jj} = \mathrm{M}_{jj} - \mathrm{M}_{ij}^2/\mathrm{M}_{ii}$ (i is the row index, and j is the column index)
$\otimes$	Kronecker product between vectors $\boldsymbol{\beta}$ and **w**

Chapter 1
Introduction

1.1 Motivation

To assure the safe operation of industrial processes and the quality of their products, multivariate statistical analysis (MSA) has been extensively used for the purpose of predictive modeling [1]-[2], fault detection and diagnosis [3]-[7], and causal analysis [8]-[9], which are critical for the enhancement of safety, reliability, and maintainability of industrial processes. Among them, principal component analysis (PCA) [10], independent component analysis (ICA) [11], partial least squares (PLS) [12], canonical correlation analysis (CCA) [13], and latent variable regression (LVR) [14]-[15] are the most widely used ones.

As an unsupervised algorithm, PCA has been in common usage from its birth. Assuming that the data follows Gaussian distributions, PCA extracts the latent variables, which are also known as principal components, via maximizing the variations of process variables. With PCA, the original process data can be decomposed into two subspaces: principal component subspace, which includes most variances of process variables, and residual subspace [10]. To eliminate the assumption of Gaussian distribution, ICA has been designed to capture non-Gaussian variations via minimizing the dependence between independent components [11]. When the collected data contains both process and quality data, supervised algorithms are preferred to fully exploit the information in the data. Supervised methods are designed to construct the relations between process and quality variables. For instance, PLS maximizes covariances between input and output, but its extracted space may contain extraneous variances of process data, leading to ineffective quality prediction. Alternatively, CCA addresses this issue by maximizing their correlations, but it fails to make full use of the quality information [16]. LVR was proposed to maximize the prediction projection of quality data on the latent space, and it has shown superiority over PLS and CCA in terms of quality-relevant modeling and monitoring [14].

Considering the advantages of LVR, a novel concurrent kernel LVR (CKLVR) is proposed in this thesis to improve the quality-relevant modeling performance and realize a comprehensive nonlinear monitoring framework. In the CKLVR algorithm, the kernel LVR (KLVR) is designed to map the process matrix into a higher-dimensional feature space, where process and quality data are related approximately linearly. It is noted that a regularization term is also designed in KLVR to avoid the negative effects brought by strong collinearity. Further, subsequent decomposition is conducted to

decompose the extracted feature space into different subspaces, including process-relevant variations, potentially quality-relevant variations, and quality variations that are not predictable from process, with each serving a particular monitoring purpose. The monitoring framework is referred to as multi-layer monitoring, since different fault alarming levels are attached to these spaces, where quality and quality-relevant anomalies receive the highest alarming level while process-relevant ones are ranked as the lowest. The superiority of the proposed algorithm is demonstrated through case studies of a numerical example, the Dow's Refining Process and the Tennessee Eastman process (TEP).

In addition, dynamics is also a key issue in the related field since it is one of the typical characteristics of real-world processes, and two kinds of dynamics exist, namely the temporal dependence between adjacent samples, and the dynamic cross-correlations between various variables. It is of key importance to capture temporal relations, and various dynamic models were proposed, such as dynamic PLS [17], dynamic CCA [18], and dynamic rLVR [19]. These methods are effective to capture cross-correlations between the input and output data. To fully exploit both auto and cross temporal relations, dynamic auto-regressive approaches have been proposed inspired by the auto-regressive exogenous (ARX) model, such as dynamic auto-regressive PLS (DAPLS) [20]. DAPLS has better performance in modeling and monitoring, but it still suffers the same issues as the PLS. Considering the superiority of LVR over PLS and CCA, a novel dynamic auto-regressive LVR (DALVR) algorithm is designed to fully exploit system dynamics, which can capture dynamic cross-correlations and auto-correlations simultaneously. DALVR-based concurrent monitoring scheme is also developed for comprehensive monitoring.

After identifying anomalies, the fault diagnosis can be further used to determine the root cause. The existing diagnosis methods include contribution plots [21], subspace extraction methods [22], and reconstruction-based contribution (RBC) [23]. Compared with other approaches, RBC has the great advantage of its non-rigorous diagnosability analysis and the inability to diagnose faults with unknown directions in contribution plots and subspace extraction methods [24]. To further eliminate the smearing effect, causal analysis can be proposed to investigate causal relations between contributing variables selected by RBC. As one of the most commonly used approaches, Granger causality is chosen to construct a fault diagnosis and causal analysis framework in this work.

Apart from MSA methods, it is also valuable for autoencoders (AE) to exploit dynamic relations in the datasets. AE is a current research focus due to its ability to extract features and reduce

dimensionality of datasets [25]. However, due to the assumption of most AE algorithms that data points are considered as independent samples, it is still a challenging issue to deal with complex data, especially dynamic relations. To overcome this issue, an improved AE is proposed to capture dynamics in the hidden layer. The corresponding anomaly detection monitoring framework is developed as well for the safety and security of complex processes in industrial applications [26].

1.2 Research Objectives

The primary objectives of this research are listed below:

1. A nonlinear LVR algorithm with a regularization term is designed to model the nonlinearities and collinearity in complex processes and thus to improve the predictive modeling performance. Based on this nonlinear LVR method, a concurrent modeling and anomaly detection scheme is proposed to extract and monitor nonlinear variations in process and quality spaces comprehensively, where the monitoring index is designed for each extracted subspace with different alarming levels.

2. A novel dynamic auto-regressive regularized LVR (DALVR) algorithm is proposed to exploit dynamic correlations and auto-correlations simultaneously. A DALVR-based concurrent modeling and fault detection framework is developed for multi-layer comprehensive monitoring.

3. An integrated fault diagnosis and root cause identification framework is developed based on DALVR for causal analysis, where a relative RBC (rRBC) is first proposed to narrow down the faulty candidates, and then the integration of time-domain Granger causality (TDGC) and conditional spectral Granger causality (CSGC) is developed to improve the efficiency and effectiveness of root cause analysis.

4. Considering dynamic relations in the hidden layer, a novel autoencoder is designed to address high-dimensional and time-dependent issues existing in the real-world data simultaneously. On the basis of the improved autoencoder, an anomaly detection monitoring scheme is constructed.

5. Simulation and industrial datasets are used for case studies to validate the superiority and effectiveness of proposed methods.

1.3 Thesis Organization

To make a clear scope view to the structure of the thesis, all the chapters are organized as follows:

1. Chapter 1 gives a general introduction to the primal motivation, research objectives and organization of this research.

2. Chapter 2 provides a brief background introduction of MSA methods, process monitoring, fault diagnosis and causal analysis. Some classical MSA approaches are introduced in this chapter, such as principal component analysis (PCA), partial least squares (PLS), canonical correlation analysis (CCA), and latent variable regression (LVR), together with their nonlinear and dynamic extensions. The following is the content of process monitoring, involving fault detection, identification and diagnosis. Two of the most frequently used approaches, contribution plots and RBC, are presented. Additionally, the definition of causal analysis and two causal analysis methods are contained in this chapter, including TDGC and CSGC.

3. Chapter 3 proposed kernel latent variable regression (KLVR) and its concurrent quality and process-related monitoring framework. Case studies on a synthetic case, Tennessee Eastman Process (TEP) and Dow's refining process are demonstrated to show the superiority of CKLVR.

4. Chapter 4 elaborates the details of DALVR with the corresponding concurrent monitoring scheme, and its performance is validated with the additional TEP data.

5. Chapter 5 presents a fault diagnosis and root cause identification framework based on DALVR, and its superiority is shown by case studies on TEP and the three-phase flow facility (TPFF) data.

6. Chapter 6 demonstrates a novel autoencoder with dynamic hidden layer (DHL-AE) and the corresponding anomaly monitoring scheme. The effectiveness can be proven by a case study on TEP data.

7. Finally, Chapter 7 draws the conclusions.

Chapter 2
Background and Literature Review

2.1 Multivariate Statistical Analysis Methods

2.1.1 Multivariate Statistical Analysis

Due to the advanced data sensory techniques, the explosive growth of collected data has brought large values and challenges to the industrial applications. Data-driven statistical process monitoring has been extensively employed in the area of process systems engineering for prediction [1], process monitoring [2]-[3], fault identification and diagnosis [3]-[7], and causal analysis [8]-[9], which plays a vital part in ensuring product quality and operation safety in industrial processes.

Various machine learning algorithms are proposed and adapted to improve monitoring accuracy, including both unsupervised and supervised ones. As an unsupervised modeling algorithm, principal component analysis (PCA) [10] has been widely applied to the extraction of latent variables by maximizing the variations of the projected variables, and it decomposes the original data into principal component subspace (PCS) and residual subspace (RS). PCS contains most variances of the original data, while RS mainly includes the noise [27]. The PCA-based monitoring is constructed to detect faults in PCS and RS via T^2 statistic [28] and Q statistic [29], respectively. Several variants of PCA have also been developed in the past decades. For instance, multiblock PCA was developed by Westerhuis et al. [30] for enhancing the interpretability of multivariate algorithms, which is applicable to data with large number of variables. Multi-scale principal component analysis was designed by Misra et al. [31] for the detection and identification of faults at different scales. Amin et al. [32] combined multivariate exponentially weighted moving average with PCA to handle the monitoring of unobservable faults.

PCA considers statistics (mean and variance) of the data, and it assumes that the data follows a Gaussian distribution. Instead, independent component analysis (ICA) [33] removes the assumption of data distribution, and it decomposes the original data into a linear combination of independent components and captures non-Gaussian variations by minimizing the dependence between independent components [34]. In recent years, modified versions have been proposed by researchers. Kano et al. [35] used ICA to transform the observed multivariate data into statistically independent components, and combined ICA with a novel statistical process control method focusing on data-driven essential variables that could be estimated from measured process variables. PCA-ICA integrated with Bayesian

method was developed by Jiang et al. [36] for non-Gaussian fault diagnosis, which was employed to generate signature evidence of efficient fault. An improved ICA methodology named multi-scale independent component analysis [34] was put forward in order to improve the monitoring performance under noisy environment.

Both PCA and ICA pay attention to the input or output only, thus failing to consider the connection between input and output variables, which increases the difficulty to locate the root causes. Alternatively, partial least squares (PLS) [12] serves as a supervised learning algorithm, and it constructs its latent space by maximizing the covariation between input and output data. PLS focuses on both the process and quality data, but some components of the captured principal features are orthogonal or irrelevant to the output data, which may deteriorate the monitoring performance. Further, its residual space may contain large variations, which are not appropriate to employ Q index for monitoring. Improvements of basic PLS have been developed over the past few years. For instance, a novel framework of localized process diagnosis based on multiblock PLS was utilized for the monitoring of a complex chemical process [37]. To capture the characteristics of dynamic processes, a dynamic inner PLS (DiPLS) was designed by Dong and Qin [17], providing an explicit description for dynamic inner and outer models simultaneously. To achieve efficient modeling of processes, Xu and Zhu [20] proposed a novel dynamic auto-regressive PLS (DAPLS) approach to capture both auto-correlations and cross-correlations between the input and output data.

In contrast to PLS, CCA extracts latent variables by maximizing the correlation between input and output data, thus all the captured information is related to the output. Therefore, it eliminates the extraneous variances caused by the input variables, leading to a better prediction power. Various extensions of CCA have also been proposed for the purpose of tackling processes with more complex characteristics. Combined with the statistical local method, an improved CCA-based fault detection approach was put forward for the sake of incipient multiplicative faults in the industrial practice [38]. A generalized CCA-based scheme integrated with the threshold-setting based on the randomized algorithm was developed by Chen et al. [39] to improve the fault detection performance. Jiang et al. [40] designed a multi-objective two-dimensional CCA (M2D-CCA)-based framework for fault detection with respect to successive batch processes. The issue involved with CCA-based monitoring methods is that the variances of the output variables are not exploited, which, however, is important for quality-relevant monitoring [41].

In order to overcome the variance issue involved in PLS and CCA, latent variable regression (LVR) was proposed to consider both the correlation between input and output data and the variances of quality variables. The superiority of LVR over PLS and CCA in terms of quality-relevant modeling and monitoring was demonstrated both theoretically and experimentally in Zhu's work [42].

To elaborate the algorithms proposed in this work, some classical MSA methods, PLS, CCA, and LVR, and their corresponding monitoring methods are reviewed in subsequent subsections.

2.1.1.1 Partial Least Squares

As one of the most extensively used algorithms, PLS [12] lays a solid foundation for further research and investigation on MSA. Assume that the input matrix is defined as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^\top \in \mathbb{R}^{n \times m}$, which contains n samples with each having m process variables, while the output matrix is denoted as $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]^\top \in \mathbb{R}^{n \times p}$ with p quality variables. The objective of PLS is

$$\begin{aligned} \max_{\mathbf{w},\mathbf{q}} \ & J_{\text{outer}} = \mathbf{t}^\top \mathbf{u} = \mathbf{w}^\top \mathbf{X}^\top \mathbf{Y} \mathbf{q} \\ & s.t. \ \|\mathbf{w}\| = 1, \|\mathbf{q}\| = 1 \end{aligned} \tag{2.1}$$

where $\mathbf{t} = \mathbf{X}\mathbf{w}$ and $\mathbf{u} = \mathbf{Y}\mathbf{q}$ are defined as process and quality score vectors, and they represent the projections of input and output matrices, respectively. $\mathbf{w} \in \mathbb{R}^m$ and $\mathbf{q} \in \mathbb{R}^p$ are weighting vectors for $\mathbf{X}$ and $\mathbf{Y}$, respectively. Here, the constraints are used to ensure the stability and solvability of the model. Setting the norm values of $\mathbf{w}$ and $\mathbf{q}$ equal to 1 aims to look for unit vectors that maximize the covariance. Eq. (2.1) defines the objective of PLS outer structure, and its solution can be obtained with the aid of Lagrange multipliers.

For the inner structure, PLS constructs a relation between process and quality scores as

$$\mathbf{t} = b\mathbf{u} + \boldsymbol{\varepsilon}_{PLS} \tag{2.2}$$

where b and $\boldsymbol{\varepsilon}_{PLS}$ represent the regression coefficient and error, respectively. The prediction of process score vector is denoted as $\hat{\mathbf{t}} = b\mathbf{u}$, where $b = \mathbf{u}^\top \mathbf{t} / \mathbf{t}^\top \mathbf{t}$. More details can refer to Literature [12].

With l latent variables extracted by iterative procedure, PLS decomposes $\mathbf{X}$ and $\mathbf{Y}$ as follows:

$$\begin{cases} \mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{X}_r \\ \mathbf{Y} = \mathbf{T}\mathbf{B}\mathbf{Q}^\top + \mathbf{Y}_r \end{cases} \tag{2.3}$$

where $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_l] \in \mathbb{R}^{n\times l}$ is the score matrix, $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_l] \in \mathbb{R}^{m\times l}$ and $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_l] \in \mathbb{R}^{p\times l}$ denote the loading matrix for process variables and quality score matrix, and $\mathbf{B} = \mathrm{diag}[b_1, b_2, \dots, b_l]$. $\mathbf{X}_r$ and $\mathbf{Y}_r$ are residuals of input and output matrices, respectively.

For a new data sample $\mathbf{x}_{\mathrm{new}} \in \mathbb{R}^m$, its quality variables are predicted by PLS as

$$\hat{\mathbf{y}}_{\mathrm{new}} = \mathbf{Q}\mathbf{R}^{\top}\mathbf{x}_{\mathrm{new}} \tag{2.4}$$

where $\mathbf{R} = \mathbf{W}(\mathbf{P}^{\top}\mathbf{W})^{-1} \in \mathbb{R}^{m\times l}$ with $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_l] \in \mathbb{R}^{m\times l}$ as the input weighting matrix. $\mathbf{x}_{\mathrm{new}}$ can be decomposed as $\mathbf{x}_{\mathrm{new}} = \hat{\mathbf{x}}_{\mathrm{new}} + \tilde{\mathbf{x}}_{\mathrm{new}}$, and $\hat{\mathbf{x}}_{\mathrm{new}}$ and $\tilde{\mathbf{x}}_{\mathrm{new}}$ are

$$\hat{\mathbf{x}}_{\mathrm{new}} = \mathbf{P}\mathbf{R}^{\top}\,\mathbf{x}_{\mathrm{new}} \in \mathrm{Span}\{\mathbf{P}\} \tag{2.5}$$

$$\tilde{\mathbf{x}}_{\mathrm{new}} = (\mathbf{I} - \mathbf{P}\mathbf{R}^{\top})\,\mathbf{x}_{\mathrm{new}} \in \mathrm{Span}\{\mathbf{R}\}^{\perp}. \tag{2.6}$$

Assuming that variations in $\mathbf{X}$ and $\mathbf{Y}$ obey normal distribution, monitoring indices, the Hotelling's T^2 [28] and Q indices [29], can then be defined to detect whether anomalies are involved in $\hat{\mathbf{x}}_{\mathrm{new}}$ and $\tilde{\mathbf{x}}_{\mathrm{new}}$. T^2 and Q can be obtained as

$$T^2 = \mathbf{t}_{\mathrm{new}}^{\top}\boldsymbol{\Lambda}^{-1}\mathbf{t}_{\mathrm{new}} \tag{2.7}$$

$$Q = \|\tilde{\mathbf{x}}_{\mathrm{new}}\|^2 = \tilde{\mathbf{x}}_{\mathrm{new}}^{\top}\tilde{\mathbf{x}}_{\mathrm{new}} \tag{2.8}$$

where the latent score vector $\mathbf{t}_{\mathrm{new}} = \mathbf{R}^{\top}\mathbf{x}_{\mathrm{new}}$, and $\boldsymbol{\Lambda} = \frac{1}{n-1}\mathbf{T}^{\top}\mathbf{T}$ contains the variances of the latent variables. T^2 index is used to monitor faults that are relevant to quality variables, while Q index is to capture anomalies existing in the residual space.

Their corresponding control limits are designed as [3]

$$\begin{aligned} T_{\mathrm{cl}}^2 &= \frac{l(n^2-1)}{n(n-l)}F_{l,n-l,\alpha} \\ Q_{\mathrm{cl}} &= g_s\chi_{h,\alpha}^2 \end{aligned} \tag{2.9}$$

where α is the confidence interval, and the confidence level is represented by $(1-\alpha)\times 100\%$. $F_{l,n-l}$ denotes an F-distribution with l and $n-l$ degrees of freedom. $\chi_{h,\alpha}^2$ defines a χ^2-distribution with h degrees of freedom, and g is a scaling factor [58]. The calculation of g_s and h are calculated according to Literature [28]. When the sample number n is large enough, T_{cl}^2 can also be approximated by $T_{\mathrm{cl}}^2 = \chi_{l,\alpha}^2$ [29].

2.1.1.2 Canonical Correlation Analysis

PLS extracts latent variables by maximizing covariances between quality and process data, but it fails to avoid variance information that is not highly correlated to quality data. To overcome this drawback of PLS, CCA focuses on maximizing correlation between quality and process data. In this case, all the captured information is related to quality data. The objective of CCA is formulated as follows [18]:

$$\max_{\mathbf{t},\mathbf{u}} \ J = \frac{\mathbf{t}^{\top}\mathbf{u}}{\|\mathbf{t}\|\|\mathbf{u}\|}. \tag{2.10}$$

where $\mathbf{t}$ and $\mathbf{u}$ are score vectors of process and quality matrices. Considering the definition of correlation, Eq. (2.10) can be reorganized as

$$\begin{aligned} &\max_{\mathbf{w},\mathbf{q}} \ J = \mathbf{t}^{\top}\mathbf{u} = \mathbf{w}^{\top}\mathbf{X}^{\top}\mathbf{Y}\mathbf{q} \\ &s.t. \ \|\mathbf{X}\mathbf{w}\| = 1, \|\mathbf{Y}\mathbf{q}\| = 1. \end{aligned} \tag{2.11}$$

After all latent variables are extracted, CCA decomposes $\mathbf{X}$ and $\mathbf{Y}$ as

$$\begin{cases} \mathbf{X} = \mathbf{T}\mathbf{P}^{\top} + \mathbf{X}_r \\ \mathbf{Y} = \mathbf{T}\mathbf{Q}^{\top} + \mathbf{Y}_r. \end{cases} \tag{2.12}$$

T^2 and Q statistics of CCA for process monitoring can be constructed as shown in Eqs. (2.5) - (2.9). More details can be found in Literature [13] and [18].

2.1.1.3 Latent Variable Regression

LVR shows its superiority due to its better prediction performance than PLS and more sufficient exploitation on the quality information than CCA [14]. LVR considers the correlation between $\mathbf{X}$ and $\mathbf{Y}$ and the variance of $\mathbf{Y}$ simultaneously, which is shown as

$$\begin{aligned} &\max_{\mathbf{w},\mathbf{q}} \ J = \mathbf{w}^{\top}\mathbf{X}^{\top}\mathbf{Y}\mathbf{q} \\ &s.t. \ \|\mathbf{X}\mathbf{w}\| = 1, \|\mathbf{q}\| = 1 \end{aligned} \tag{2.13}$$

where the symbols in Eq. (2.13) are identical to those in Eq. (2.1).

The calculation of matrix inversion $(\mathbf{X}^{\top}\mathbf{X})^{-1}$ is involved when obtaining model parameters in Eq. (2.13), and LVR may get poor modeling performance in the existence of strong collinearity in $\mathbf{X}$ [14]. Thus, a regularization term is introduced into LVR, which constructs a regularized LVR (rLVR) method.

$$\begin{aligned} \max_{\mathbf{w},\mathbf{q}} \ & J = \mathbf{q}^\top \mathbf{Y}^\top \mathbf{X}\mathbf{w} - \frac{\gamma}{2}\|\mathbf{w}\|^2 \\ s.t. \ & \|\mathbf{X}\mathbf{w}\| = 1, \|\mathbf{q}\| = 1 \end{aligned} \tag{2.14}$$

where $\frac{\gamma}{2}\|\mathbf{w}\|^2$ is the regularization term to handle the potential singular value issues in $\mathbf{X}^\top\mathbf{X}$, and γ is the regularized parameter.

With rLVR, **X** and **Y** can be decomposed into

$$\begin{cases} \mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{X}_r \\ \mathbf{Y} = \mathbf{T}\mathbf{C}^\top + \mathbf{Y}_r \end{cases} \tag{2.15}$$

where **T**, **P** , $\mathbf{X}_r$ and $\mathbf{Y}_r$ have the same meaning as those in PLS, and $\mathbf{C} \in \mathbb{R}^{p\times l}$ denotes the loading matrix for quality variables. The detailed algorithm can be found in Literature [14] and [42]. Similar to PLS, T^2 and Q indices and corresponding control limits can be developed for rLVR based on Eqs. (2.5) - (2.9).

2.1.2 Nonlinear Extensions of Multivariate Statistical Analysis

The typical MSA methods including PLS, CCA, and LVR are based on a linearity assumption, and their performance is usually poor in practical industrial processes, since nonlinearity is inevitable in these processes.

The aforementioned algorithms are capable of process monitoring, but they are based on the implicit assumption that there exist linear relations between process and quality variables, leading to poor performance in real-world nonlinear processes. To overcome this issue, their nonlinear extensions were proposed. Wold et al. [43] employed quadratic polynomial to build up the nonlinear relations for the applications such as multivariate calibration and process optimization. Rosipal and Trejo [44] proposed a kernel PLS (KPLS) methodology for nonlinear data by combining the reproducing kernel Hilbert space with PLS to project the original data into a higher-dimensional feature space. After nonlinear mapping, linear relation can be established and extracted in the higher-dimensional feature space, leading to better monitoring performance. The multi-scale KPLS algorithm was designed by Zhang and Hu [45] to exploit the multi-scale nature of nonlinear data by incorporating KPLS with wavelet analysis. Zhou et al. [46] developed a new fault identification method for KPLS-based monitoring to deal with the issue of strong nonlinearity and few fault samples. Nonlinear counterparts were also developed for CCA such as kernel CCA [47] and mixture CCA [48].

Apart from nonlinear extensions of the aforementioned algorithms, other nonlinear machine learning methods are also popularly used to address this issue, such as neural network (NN) and support vector machine (SVM) models. Qin and McAvoy [49] developed a neural net PLS algorithm by embedding multilayer neural networks into the PLS modeling. Shang et al. [50] proposed a NN based soft sensor to model massive nonlinear data. Based on the optimization of neuron number in the hidden layer of NNs, a hybrid artificial neural network model was designed for the fault detection and diagnosis of complex process systems [51]. SVM has also been utilized to handle nonlinear problems by incorporating the kernel trick, which projects the input data into a higher dimensional subspace and obtains solutions via solving quadratic programming optimization [52]. Compared with NNs, SVM can avoid the local minimum problems which exist in neural networks, leading to great generalization capacity. For instance, Li et al. [53] combined SVM and kernel PCA (KPCA) and developed a novel fault recognition algorithm, leading to a higher recognition rate than KPCA. An SVM-based algorithm designed for fault detection in high-speed trains was studied by Liu et al. [54] with cost-sensitive strategy handling imbalanced data and weighted-feature strategy differentiating features. Other nonlinear methods including *k*-nearest neighbor algorithm [55]- [56] are also applied for process monitoring. However, these algorithms are time-consuming due to their computational complexity, and their interpretability is weak compared with kernel MSA methods. Thus, in this work, Gaussian kernel function is chosen to construct a nonlinear variant of MSA methods, and its concept is defined as follows [57]. Suppose $\mathbf{x} \in \mathbb{R}^n$ is a non-empty subset, and ϕ is a nonlinear mapping from $\mathbf{x}$ to a Hilbert space F. $k(\mathrm{x}_i, \mathrm{x}_j)$ is defined as kernel function:

$$k(\mathrm{x}_i, \mathrm{x}_j) = \langle \phi(\mathrm{x}_i), \phi(\mathrm{x}_j) \rangle \ \ \forall \mathrm{x}_i, \mathrm{x}_j \in \mathbf{x} \tag{2.16}$$

where $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the dot product between vectors $\mathbf{a}$ and $\mathbf{b}$. The kernel function changes a non-linear problem into a linear one.

As one of the most popular kernel functions, Gaussian kernel function has better smoothing performance and ability to approximate any nonlinear function using fewer parameters. Thus, it is widely used for nonlinear issues, which is given by

$$k(\mathrm{x}_i, \mathrm{x}_j) = \frac{1}{2\pi} \exp\left(-\frac{\left\| \mathrm{x}_i - \mathrm{x}_j \right\|^2}{2\sigma^2} \right) \tag{2.17}$$

where σ is density of Gaussian kernel function and it is an adjustable parameter.

To simplify the calculation, Eq. (2.17) can be rewritten as

$$k(\mathrm{x}_i, \mathrm{x}_j) = \exp\left(-\frac{\left\|\mathrm{x}_i - \mathrm{x}_j\right\|^2}{c}\right) \tag{2.18}$$

where c represents the width of a Gaussian function, which determines the robustness of kernel mapping.

2.1.3 Dynamic Regularized Latent Variable Regression

Correlation is one of the typical characteristics of real-world processes, and it can be classified into two categories, namely the temporal dependence between adjacent samples, and the dynamic cross-correlations between various variables. Various dynamic extensions were proposed, such as dynamic PLS [17], dynamic CCA [18], and dynamic rLVR [19]. These models consider dynamic cross-correlations, while the dynamic auto-correlations are ignored, which are also very important for the accurate prediction and monitoring performance of quality variables. Due to the superiority of LVR over PLS and CCA which has been discussed in the previous section, the dynamic variant of LVR, namely dynamic rLVR (DrLVR) [19], is reviewed in this section, which serves as a steppingstone to the objectives proposed in this work.

DrLVR maximizes the covariance between current quality score and current and past process scores, and its objective is

$$\begin{aligned} &\max_{\mathbf{w},\mathbf{q},\boldsymbol{\beta}} \; J = \mathbf{q}^\top \mathbf{Y}_s^\top \mathbf{Z}_s(\boldsymbol{\beta} \otimes \mathbf{w}) - \frac{\gamma}{2}\|\boldsymbol{\beta} \otimes \mathbf{w}\|^2 \\ &s.t. \; \|\mathbf{Z}_s(\boldsymbol{\beta} \otimes \mathbf{w})\| = 1, \|\mathbf{q}\| = 1, \|\boldsymbol{\beta}\| = 1 \end{aligned} \tag{2.19}$$

where $\mathbf{Z}_s = [\mathbf{X}_s, \mathbf{X}_{s-1}, \dots, \mathbf{X}_1] \in \mathbb{R}^{N\times(ms)}$, and $\mathbf{X}_i = [\mathbf{x}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_{i+N}]^\top \in \mathbb{R}^{N\times m}$ $(i = 0,1,\dots,s)$ is a subset of the collected input samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{s+N}]^\top \in \mathbb{R}^{(N+s)\times m}$. $\mathbf{Y}_s = [\mathbf{y}_{i+1}, \mathbf{y}_{i+2}, \dots, \mathbf{y}_{i+N}]^\top \in \mathbb{R}^{N\times p}$ is a subset of the collected output samples $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{s+N}]^\top \in R^{(N+s)\times p}$, and s is the time lag which reflects the dynamics degrees in the system. $\boldsymbol{\beta} = [\beta_0, \beta_1, \dots, \beta_s]^\top \in \mathbb{R}^{s+1}$ represents the weighting coefficient vector for $\mathbf{w}$, and $\boldsymbol{\beta} \otimes \mathbf{w}$ denotes the Kronecker product between $\boldsymbol{\beta}$ and $\mathbf{w}$. $\frac{\gamma}{2}\|\boldsymbol{\beta} \otimes \mathbf{w}\|^2$ is the regularization term to deal with the collinearity issue existing in the original data.

To align with the dynamic outer model, the inner structure of DrLVR is constructed by a linear regression between quality score and a set of past process scores as follows:

$$u_k = \beta_0 t_k + \beta_1 t_{k-1} + \cdots + \beta_s t_{k-s} + \varepsilon_k \tag{2.20}$$

where $t_k = \mathbf{x}_k^\top \mathbf{w}$ and $u_k = \mathbf{y}_k^\top \mathbf{q}$ represent process and quality score of the sample $(\mathbf{x}_k, \mathbf{y}_k)$. ε_k is the regression error. To extract other sets of latent variables, deflation should be performed on $\mathbf{X}$ and $\mathbf{Y}_s$ [19].

DrLVR decomposes $\mathbf{X}$ and $\mathbf{Y}_s$ as

$$\begin{cases} \mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{X}_r \\ \mathbf{Y}_s = \widehat{\mathbf{U}}_s \mathbf{C}^\top + \mathbf{Y}_r \end{cases} \tag{2.21}$$

where $\widehat{\mathbf{U}}_s$ is the prediction of quality score matrix $\mathbf{U}_s = \left[\mathbf{u}_{s,1}, \mathbf{u}_{s,2}, \ldots, \mathbf{u}_{s,l}\right] \in \mathbb{R}^{N \times l}$. Here, $\mathbf{u}_s = \mathbf{Y}_s \mathbf{q}$ corresponds to the quality score of the i^{th} latent variable, and $\mathbf{u}_s = [u_{s+1}, u_{s+2}, \ldots, u_{s+N}]^\top \in \mathbb{R}^N$. Other symbols retain the same meanings as those in Eqs. (2.1), (2.13), and (2.14).

Based on Eq. (2.21), the predicted output $\widehat{\mathbf{Y}}_s$ is expressed as

$$\widehat{\mathbf{Y}}_s = \widehat{\mathbf{U}}_s \mathbf{C}^\top. \tag{2.22}$$

Similar to PLS and rLVR, DrLVR-based monitoring is performed on a new sample $(\mathbf{x}_k, \mathbf{y}_k)$ according to Eqs. (2.5) - (2.9). More details on DrLVR and its monitoring scheme can be referred to Literature [19].

2.2 Process Monitoring

Over the past few decades, multivariate analysis has been commonly applied for process monitoring. As one of the most active research domains in process control, statistical process monitoring (SPM) has been developed and used for different industrial applications, such as chemicals, microelectronics, manufacturing, and pharmaceutical processes. The related tasks involve fault detection, fault identification, fault diagnosis, fault estimation, and fault reconstruction [3]. In this work, we focus on fault detection, identification, and diagnosis. Note that for process monitoring, process data is in accordance with the connotation of input data, while quality data is consistent with the definition of output data.

2.2.1 Fault Detection

Fault detection is considered as the first step in process monitoring, and monitoring statistics are used to determine abnormal datapoints. The Hotelling's T^2 and Q statistics are used to detect out-of-control

situations in principal component subspace and residual subspace, respectively. Q is also known as the squared prediction error (SPE) [3]. Combined T^2 with Q, a single index φ was proposed for monitoring as follows [3]:

$$\varphi = \frac{T^2}{T_{\text{cl}}^2} + \frac{Q}{Q_{\text{cl}}} \tag{2.23}$$

where T_{cl}^2 and Q_{cl} represent the control limits of T^2 with Q, respectively.

Assuming that φ follows a χ^2-distribution approximately, its control limit φ_{cl} is obtained by

$$\varphi_{\text{cl}} = g^{\varphi}\chi_{h^{\varphi},\alpha}^2 \tag{2.24}$$

where

$$g^{\varphi} = \frac{tr(\mathbf{S}_x\mathbf{\Phi})^2}{tr(\mathbf{S}_x\mathbf{\Phi})}, h^{\varphi} = \frac{[tr(\mathbf{S}_x\mathbf{\Phi})]^2}{tr(\mathbf{S}_x\mathbf{\Phi})}$$

and $\mathbf{S}_x$ is the covariance matrix of input $\mathbf{X}$.

These statistics are effective for quality-relevant monitoring, but they ignore process-relevant variations and quality variations that are not predictable from process variables, which are also of vital importance for operation safety and product quality. To address this issue, several decomposition methods have been developed to obtain more comprehensive monitoring results. The orthogonal signal correction (OSC) was used to remove information unrelated to the output [59], but it fails to provide enough information on the input data. Later, the total PLS (TPLS) was put forward by Zhou et al. [60] to resolve these issues by decomposing the space of process data into four different subspaces. However, the output-relevant monitoring of TPLS only focuses on quality variances that can be predicted by the process variables. Additionally, it is unnecessary for TPLS to decompose the process space into four subspaces, and it is sufficient to differentiate between output-relevant and input-relevant variances. Alternatively, to monitor process-relevant and quality-relevant faults, a concurrent PLS (CPLS) [61] was developed to partition the original feature space into five subspaces, involving covariation subspace, output-principal subspace, output-residual subspace, input-principal subspace, an input-residual subspace. The CPLS-based monitoring considers both process and quality variables, which constructs comprehensive monitoring of both process-relevant and quality-relevant faults. A similar concurrent framework was also developed for CCA. Similar to CPLS, concurrent CCA was proposed by Zhu et al. [62] for quality-relevant fault detection, decomposes the original data space into five subspaces,

including correlation subspace, process-principal subspace, process-residual subspace, quality-principal subspace, and quality-residual subspace.

2.2.2 Fault Identification and Diagnosis

Once an anomaly is detected with the monitoring statistics, it is necessary to determine its assignable causes. Several approaches are proposed for fault identification and diagnosis, such as contribution plots [21], subspace extraction methods [22], and reconstruction-based contribution (RBC) [23].

As one of the early proposed fault diagnosis methods, contribution plots diagnose the causes by identifying the corresponding contribution of each variable to the monitoring indices, with the assumption that faulty variables contribute more to the monitoring statistics. Hopkins et al. patented the commercial use of contribution plots. A novel contribution plots method with a new indicator was developed by Wang et al. [63] for quality relevant fault diagnosis. Bounoua and Bakdi [64] improved the contribution plots to analyze the cause for abnormal process conditions, indicating more accurate information. Additionally, an enhanced comprehensive contribution plots approach was designed to for Bayesian fusion based distributed multivariate statistical process monitoring, which improved its fault isolation efficiency [65].

Subspace extraction methods are also developed for root cause analysis. For instance, a PCA-based subspace reconstruction approach was proposed by Dunia and Qin [66] for multidimensional fault identification and reconstruction. Subsequently, Gertler et al. [67] put forward an isolation enhanced PCA method. Based on a robust reconstruction error calculation, a novel fault identification method proposed by Choi et al. [68] was formulated to handle nonlinear relations by kernel PCA. Besides, to deal with inaccurate fault subspace extraction and unidentified false alarms, a modified PLS algorithm developed by Hu et al. [69] was designed to gain a precise subspace through orthogonal decomposition and extract purer quality-related and quality-unrelated fault subspaces.

To address the issues involving non-rigorous diagnosability analysis and inability to diagnose faults with unknown directions in the previous methods, Alcala and Qin [23] proposed an RBC approach, which denotes the amount of reconstruction in the direction of each variable to minimize the fault detection statistics. Later, several variants of RBC have also been proposed. Specifically, a generalized RBC approach with a total PLS model was created by Li et al. [70], showing its superiority to contribution plots-based diagnosis methods. To identify the variables that are the most responsible for the fault, He et al. [71] established a generic reconstruction based multivariate contribution analysis

framework for fault diagnosis with a branch and bound algorithm for the efficient solution to the combinatorial optimization problem. Two modified RBC approaches including a generalized RBC method and RBC ratio method were proposed by Mnassri et al. [72] to tackle complex faults and remedy the defective of the traditional RBC method.

As preliminaries of proposed fault diagnosis and causal analysis framework, contribution plots and RBC methods are reviewed in the following subsections.

2.2.2.1 Contribution Plots

T^2 and Q are the most commonly used statistics for fault diagnosis with contribution plots. Contribution plots on T^2 denote the significance of the effect of for each variable, which focus on score vectors of MSA methods. Variables with the largest contributions are selected as contributing variables which cause abnormal situations. Contribution plots on Q indicate the contribution of each variable. If the Q value of a sample vector is abnormal, the corresponding variables with significant contribution need further investigation.

The contribution of Q for each variable is simply dividing the summation of Q into each variable, while the contribution of T^2 is not defined clearly as that of Q due to the definition of T^2 [3]. Nomikos defined a distribution T^2 involving cross talk among variables, potentially resulting in negative contributions [73]. To eliminate the cross talk among variables and locate the root cause of a fault in a decentralized manner, Qin et al. [74] proposed novel definitions of block and variable contributions to T^2 and Q for decentralized monitoring. Westerhuis et al. [75] included all principal components for the generalization of T^2 contributions.

There is no need to provide any prior knowledge required to generate contribution plots, which makes contribution plots method easy to calculate. However, prior knowledge is necessary to interpret the plots and find the root cause. As explained and analyzed by Kourti and MacGregor [76], although contribution plots cannot locate the cause of an abnormal situation explicitly, they narrow down the range of candidates for root cause, since the contribution from each variable is propagated to other variables in obtaining the projection of residual subspace. Such “smearing” effect will reduce the difference between contributing and non-contributing variables, leading to misidentification in some cases [3]. Thus, further studies are necessary to overcome the drawback of contribution plots.

2.2.2.2 Reconstructed-Based Contribution

To eliminate the "smearing" effect, the reconstruction-based contribution (RBC) with known fault directions and fault magnitude has been developed to locate the assignable cause from a set of contributing variables which are considered as candidates of root cause.

Since RBC retains satisfactory fault diagnosis performance, it is used as the basis of our work, and its idea is reviewed briefly in the following. For a faulty sample $\mathbf{x}$, a reconstructed vector along its fault direction ξ_i with a fault magnitude of f_i is expressed as

$$\mathbf{z}_i = \mathbf{x} - \xi_i f_i \tag{2.25}$$

where $\mathbf{z}_i$ represents a fault-free sample. Then, the general form of a reconstructed fault detection index is constructed as

$$\text{Index}(\mathbf{z}_i) = \mathbf{z}_i^\top \mathbf{M} \mathbf{z}_i = \|\mathbf{z}_i\|_{\text{M}}^2 = \|\mathbf{x} - \xi_i f_i\|_{\text{M}}^2 \tag{2.26}$$

where $\mathbf{M}$ is determined by the corresponding monitoring index. For instance, $\mathbf{M}$ for T^2 and Q indices are presented in Table 1 [23].

Table 1: $\mathbf{M}$ for the general form of RBC

Index	T^2	Q
$\mathbf{M}$	$\mathbf{R}\boldsymbol{\Lambda}^{-1}\mathbf{R}^\top$	$\mathbf{I} - \mathbf{P}\mathbf{R}^\top$

RBC identifies the fault detection by minimizing $\text{Index}(\mathbf{z}_i)$ for different variable directions, and it assumes that the direction with the largest fault magnitude is the faulty direction [23]. Taking the first derivative of $\text{Index}(\mathbf{z}_i)$ with regard to f_i and making it equal to zero yield

$$f_i = \left(\xi_i^\top \mathbf{M} \xi_i\right)^{-1} \xi_i^\top \mathbf{M} \mathbf{x}.$$

Then the amount of reconstruction along the fault direction ξ_i, denoted as $\text{RBC}_i^{\text{Index}}$, is

$$\text{RBC}_i^{\text{Index}} = \|\xi_i f_i\|_{\text{M}}^2 = \mathbf{x}^\top \mathbf{M} \xi_i \left(\xi_i^\top \mathbf{M} \xi_i\right)^{-1} \xi_i^\top \mathbf{M} \mathbf{x}. \tag{2.27}$$

2.3 Causal Analysis for Root Cause Identification

To deal with quality-related problems, root cause identification is a key step to identify the main reason for the process faults or failures. To locate the causes of faults, causal analysis, which is also known as root cause analysis, is developed to explore causal relations between components of time series, and it has attracted plenty of attractions in the related fields including public health, policy, finance, physics, medicine, biology, environmental science, and public health [77]. Many efficient methods have been designed for causal analysis, such as dynamic causal model [78], Granger causality [79]-[80], transfer entropy [81], randomization test [82], and phase slope index [83]. Among them, Granger causality and transfer entropy are the most commonly used and reliable approaches, but Granger causality is preferred in complex processes due to its computation efficiency [84].

Originated from econometrics, on the basis of linear vector-autoregressive (VAR) modeling, Granger causality [79] was proposed to handle stochastic time-series datasets, and has been developed as a time series analysis approach [85]. Conceptually, as a measure of the predictive ability from a signal to another, if the predictive ability of the signal is statistically different from that in the opposite direction, there exists Granger causal relationship between these two signals. Due to its simplicity, usability, robustness, and extendibility [86]-[87], Granger causality has been attached great importance to by many researchers [77]. At the early stage, focusing on linear relations, Geweke [88] proposed the definition of causality for multiple time series, which was seen as the supplement and extension of Granger's previous work. Later, to address early clarification of statistical issues, Granger causality was used for local-field potentials (LFPs) [89].

Since Granger causality can determine whether a signal Granger-causes another one, it is feasible to evaluate whether a set of variables includes valuable information that can be used to enhance the predictive performance of another set of variables for root cause analysis, which is meaningful for the causal analysis for dynamic systems [90]. Goebel et al. [89] combined an adaptive multivariate VAR modeling technique with Granger causal analysis to capture fast-changing cortical dynamics. Incorporating sparse regression approaches with Granger causality, a modified Granger causality framework [91] was provided to estimate high-dimensional dynamical models. In consideration of hemodynamics responses between regions in the brain, a measure of Granger causality was validated by David et al. [92] to determine the connectivity from time series. Integrating explicit observation expressions for hemodynamics responses, Ryali et al. [93] analyzed a state-space framework combined

with Granger causality. Integrating the Geweke's spectral approach with the time-varying autoregressive with exogenous input (ARX) modeling method, a novel parametric scheme for conditional Granger causality was established by Li et al. [94] to capture dynamic interaction relations in oscillatory neocortical sensorimotor networks. To reveal dynamic interaction patterns of cortical regions in surgical training on effective brain connectivity, dynamic spectral Granger causality was applied by Kamat et al. [95] via the short-time Fourier transform approach. Considering the existence of nonlinear relations in datasets, Shen et al. [96] proposed a root cause analysis framework which combines a recurrent NN with Granger causality test.

In the following two subsections, two most frequently used Granger causality methods, namely time-domain Granger causality (TDGC) [90] and conditional spectral Granger causality (CSGC) [97], are reviewed in details and will be adapted in the proposed work.

2.3.1 Time-Domain Granger Causality

Given two time series $\mathbf{x}_i(k)$ and $\mathbf{x}_j(k)$ in stationary stochastic processes, a bivariate auto-regressive model at time k is constructed as [90]

$$\begin{aligned}\mathbf{x}_i(k) &= \sum_{l=1}^{r} a_{ii,l}\mathbf{x}_i(k-l) + \sum_{l=1}^{r} a_{ij,l}\mathbf{x}_j(k-l) + \boldsymbol{e}_i(k)\\ \mathbf{x}_j(k) &= \sum_{l=1}^{r} a_{ji,l}\mathbf{x}_i(k-l) + \sum_{l=1}^{r} a_{jj,l}\mathbf{x}_j(k-l) + \boldsymbol{e}_j(k)\end{aligned} \tag{2.28}$$

where r defines the AR model order, $a_{ij,l}$ $(l = 1,2,\dots,r)$ can represent the AR coefficients, and $\boldsymbol{e}_i(k)$ and $\boldsymbol{e}_j(k)$ are the residuals or prediction errors of $\mathbf{x}_i(k)$ and $\mathbf{x}_j(k)$, respectively.

To perform the AR model on each time series, one variable, $\mathbf{x}_i$ or $\mathbf{x}_j$, can be excluded. Therefore, Eq. (2.28) can be simplified into

$$\begin{aligned}\mathbf{x}_i(k) &= \sum_{l=1}^{r} b_{i,l}\mathbf{x}_i(k-l) + \boldsymbol{e}_{i(j)}(k)\\ \mathbf{x}_j(k) &= \sum_{l=1}^{r} b_{j,l}\mathbf{x}_j(k-l) + \boldsymbol{e}_{j(i)}(k)\end{aligned} \tag{2.29}$$

where $\boldsymbol{e}_{i(j)}(k)$ expresses the prediction error that excludes the j^{th} variable to predict the i^{th} variable, which can be used to determine whether a causal-and-effect relationship exists.

It is defined that $\mathbf{x}_j$ Granger-causes $\mathbf{x}_i$ if the exclusion of $\mathbf{x}_j$ reduces the ability to predict $\mathbf{x}_i$ when all the other variables are involved in the regression model. The TDGC from Variable j to Variable i is measured by the following index

$$F_{\mathbf{x}_j \to \mathbf{x}_i} = \ln \frac{\mathrm{var}\left(\boldsymbol{e}_{i(j)}\right)}{\mathrm{var}\left(\boldsymbol{e}_i\right)} \tag{2.30}$$

where $\mathrm{var}(\cdot)$ denotes the variance. Thereafter, the statistical significance is assessed by an F test. More details can be found in Literature [90].

2.3.2 Conditional Spectral Granger Causality

As the theoretical foundation of CSGC, it is necessary to review spectral Granger causality (SGC) first [98]. Due to oscillating variables in real-world industrial processes, it is feasible and appropriate to use SGC to investigate cause-and-effect relationship among them. SGC is the spectral decomposition of TDGC for multiple time series. Compared with TDGC, SGC provides more information on the granger causality at a specific frequency range, so it can be used to combine with TDGC for Granger causality analysis. Theoretically, SGC is defined as

$$\mathrm{SGC}_{\mathbf{x}_j \to \mathbf{x}_i}(f) = \ln \frac{S_{ii}(f)}{S_{ii}(f) - \left(\boldsymbol{\Gamma}_{jj} - \dfrac{\boldsymbol{\Gamma}_{ij}^2}{\boldsymbol{\Gamma}_{ii}}\right)\left|\mathbf{H}_{ij}(f)\right|^2} \tag{2.31}$$

where $S_{ii}(f)$ defines the spectral density of the target variable $\mathbf{x}_i$ at frequency f, which represents components in spectral density matrix $\mathbf{S}$. The detailed calculation of $S_{ii}(f)$ is shown in Appendix A.

In addition to the mutual relations between $\mathbf{x}_i$ and $\mathbf{x}_j$, other variables $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_v$ may also affect them, which should be included to analyze the causality. However, SGC fails to consider this effect. Alternatively, conditional SGC (CSGC) was developed [97]. With the spectral transformation and Wilson factorization [99], a multivariate system with $v + 2$ variables $\left(\mathbf{x}_j, \mathbf{x}_i, \mathbf{z}_1, \ldots, \mathbf{z}_v\right)$ is formulated as

$$\begin{aligned} \mathbf{S}\left(\mathbf{x}_j, \mathbf{x}_i, \mathbf{z}_1, \ldots, \mathbf{z}_v, f\right) &= \mathbf{H}(f)\boldsymbol{\Sigma}\mathbf{H}^*(f) \\ \mathbf{U}_{in}\left(\mathbf{x}_j, \mathbf{z}_1, \ldots, \mathbf{z}_v, f\right) &= \mathbf{G}(f)\boldsymbol{\Gamma}\mathbf{G}^*(f) \end{aligned} \tag{2.32}$$

where $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_v$ are variables that may influence $\mathbf{x}_i$ and $\mathbf{x}_j$. $\mathbf{S}$ and $\mathbf{U}_{in}$ define the spectral matrices of complete system and system with Variable $\mathbf{x}_i$ excluded for the causality test. $\mathbf{H}$ and $\mathbf{G}$ represent the

spectral transfer function matrices, while $\mathbf{\Sigma}$ and $\mathbf{\Gamma}$ are the error covariance matrix of complete and incomplete systems, respectively. Thus, the corresponding CSGC index can be computed as

$$\mathrm{CSGC}_{\mathbf{x}_j \to \mathbf{x}_i | \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_v}(f) = \ln \frac{\mathbf{\Gamma}_{ii}}{\left| \mathbf{Q}_{ii}(f) \mathbf{\Gamma}_{jj} \mathbf{Q}_{ii}^*(f) \right|} \tag{2.33}$$

where

$$\mathbf{Q} = \begin{bmatrix} \tilde{G}_{ii} & 0 & \tilde{G}_{i\mathbf{z}_1} & \cdots & \tilde{G}_{i\mathbf{z}_v} \\ 0 & 1 & 0 & \cdots & 0 \\ \tilde{G}_{\mathbf{z}_1 i} & 0 & \tilde{G}_{\mathbf{z}_1 \mathbf{z}_1} & \cdots & \tilde{G}_{\mathbf{z}_1 \mathbf{z}_v} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{G}_{\mathbf{z}_v i} & 0 & \tilde{G}_{\mathbf{z}_v \mathbf{z}_1} & \cdots & \tilde{G}_{\mathbf{z}_v \mathbf{z}_v} \end{bmatrix}^{-1} \times \begin{bmatrix} \widetilde{H}_{ii} & \widetilde{H}_{ij} & \cdots & \cdots & \widetilde{H}_{i\mathbf{z}_v} \\ \widetilde{H}_{ji} & \cdots & \cdots & \cdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \widetilde{H}_{\mathbf{z}_v i} & 0 & \widetilde{H}_{\mathbf{z}_v j} & \cdots & \widetilde{H}_{\mathbf{z}_v \mathbf{z}_v} \end{bmatrix}.$$

The calculation of $\tilde{G}_{ii}$ can refer to Eqs. (A.1) - (A.6).

If $\mathbf{x}_j$ does not Granger-cause $\mathbf{x}_i$, but other variables among $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_v$ affect both $\mathbf{x}_j$ and $\mathbf{x}_i$, $\mathbf{\Gamma}_{ii}$ will be equal to $\left| \mathbf{Q}_{ii}(f) \mathbf{\Gamma}_{jj} \mathbf{Q}_{ii}^*(f) \right|$, thus leading to $\mathrm{CSGC}_{\mathbf{x}_j \to \mathbf{x}_i | \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_v}(f) = 0$; if there is Granger causality between $\mathbf{x}_j$ and $\mathbf{x}_i$, $\mathrm{CSGC}_{\mathbf{x}_j \to \mathbf{x}_i | \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_v}(f) > 0$. Moreover, to determine whether $\mathbf{x}_j$ Granger-causes $\mathbf{x}_i$, the robustness of CSGC is supposed to be tested against the null hypothesis.

2.4 Autoencoder

Considerable technological progress such as artificial intelligence and real-time signal processing methods has equipped existing industrial systems with highly sophisticated technologies in diverse fields, leading to significant enhancement on the complexity of real-world industrial applications, especially chemical industry. The collected data from complicated practical industry processes primarily consists of multivariate time series data with high dimensionality. To ensure the operational safety and security of complex industrial systems, anomaly detection has been a critical issue for identifying abnormal observations or events that deviate from normal patterns, which may cause potential problems in industrial practice [100].

Several classical machine learning and deep learning methods have been widely adopted in industry and academia. More concretely, multivariate statistical approaches such as clustering [101], principal component analysis (PCA) [102], and Gaussian mixture model (GMM) [103] have been used to detect abnormal points that exhibit substantial deviations from the normal data. In terms of machine learning-based algorithms, support vector machines (SVM) [104], random forests [105], and neural networks

[106] are also suitable for anomaly detection by model training. Deep learning methods have also been employed for identifying abnormal events or observations that deviate significantly from the expected or typical behavior of the data [107]. Due to the ability to capture complex patterns, feature extraction and dimensionality reduction, autoencoder (AE) technology has been considered as a popular choice for anomaly detection in chemical practice [108]. However, they pay less attention to dynamic behaviors existing in industrial datasets. Thus, it still remains a challenging issue carrying out the reliable results of anomaly detection from high-dimensional data that changes over time.

For reasons of identifying anomalies in real-world data with high dimensional and time-dependent properties, some classical algorithms such PCA and AE, and their corresponding variants can be employed for dimensionality reduction and feature extraction. Specifically, to handle time-dependent samples, dynamic PCA has been created to extract dynamics in the original data and detect irregular observations or samples [109]. To overcome the assumption of most AE algorithms that data points are considered as independent samples, considering the dynamic relations existing in the original collected data from industrial applications, some pre-processing approaches for time series data have been adopted for the construction of frameworks combined with AE methods, such as mutual information [110] and dynamic thresholding mechanism [111]. However, these proposed schemes focus on the pre-processing methods, without any consideration for the improvement of AE structure. Pre-processing increases the complexity of algorithms, rendering disadvantages involving high computational costs, prolonged running time, difficulty of implementation, and higher demand on data quality [112].

Compared with PCA, AE methods possess the strengths including handling missing data, learning hierarchical representations, new data generation, and data processing capabilities [113]. Thus, AE is adopted to build up the anomaly detection scheme proposed in this work.

2.5 Auto-Regressive Integrated Moving Average Model

Autoregressive Integrated Moving Average (ARIMA) models have been commonly utilized in time series analysis. Due to its flexibility and applicability, ARIMA models have been developed and applied for processing data with trends, irregular fluctuations, or even seasonality. Typically, an ARIMA model consists of auto-regressive part and moving average part. The autoregressive part captures the temporal dependence of current value based on its past values, while the differencing process depends on the integrated part. Additionally, the error term is processed as a linear combination of past error terms by

the moving average part [114]. Given an input vector $\mathbf{v} \in \mathbb{R}^n$, which can also be expressed as time series, the autoregressive part at time t $(t = 1, 2, \dots, n)$ is expressed by

$$\mathbf{v}_t = \sum_{i=1}^{p_a} \beta_{a,i} \mathbf{v}_{t-i} + \boldsymbol{\varepsilon}_t \tag{2.34}$$

where $\mathbf{v}_t$ denotes the aggressive part $\mathbf{v}$ existing in the input $\mathbf{z}$ at time t. p_a is the number of time lags, and $\beta_{a,i}$ $(i = 1, 2, \dots, p_a)$ are parameters of the autoregressive part. $\boldsymbol{\varepsilon}_t$ denotes the white noise error term at time t, which is typically assumed to be independent and conform to a normal distribution with zero mean.

The moving average part is given by

$$\mathbf{e}_t = \boldsymbol{\mu} + \sum_{i=1}^{q_a} \theta_i \boldsymbol{\varepsilon}_{t-i} + \boldsymbol{\varepsilon}_t \tag{2.35}$$

where $\mathbf{e}_t$ denotes the moving average part $\mathbf{e}$ existing in the input $\mathbf{z}$ at time *t*. $\boldsymbol{\mu}$ represents the mean of $\mathbf{e}$. q_a is the order of the moving average part. θ_i $(i = 1, 2, \dots, q_a)$ are parameters of the moving average part. $\boldsymbol{\varepsilon}_t$ is the white noise error term at time t for the moving average part as well.

Combining Eq. (2.34) with Eq. (2.35), the ARIMA model is defined by

$$\mathbf{z}_t = \left(1 - \sum_{i=1}^{p_a} \beta_{a,i} B^i\right)^{-1} (1 - B)^{-d_a} \left(1 - \sum_{i=1}^{q_a} \theta_i B^i\right) \boldsymbol{\varepsilon}_t \tag{2.36}$$

where $\mathbf{z}_t$ represents the component of time series $\mathbf{z}$ at time t. B denotes the backshift operator, which operates on an element of the compressed representation to construct the previous element and can be expressed as $B\mathbf{z}_t = \mathbf{z}_{t-1}, \forall t > 1$. d_a denotes the degree of differencing, which is the number of times the time series $\mathbf{z}$ has that had past values subtracted. Here, p_a, d_a, and q_a are non-negative integers.

2.6 Anomaly Detection

Anomaly detection refers to the identification of anomalies, which are defined as abnormal events, observations, or samples that significantly deviate from the normal or expected behavior of given data to indicate the potential problems in the dataset [115]. The early detection of anomalies plays an essential role in industrial practice to raise the system safety, prevent failures, reduce downtime, and improve

production in manufacturing, since anomalies are considered as indicative of potential issues involving safety risks, equipment failures, and maintenance problems [116]. To detect anomalies in a given dataset, the model that captures the normal observations or behaviors is constructed by the training period, and the test data is applied for the identification of abnormal points which are not fitted with the training model.

Anomaly detection using dimensionality reduction has been developed based on the assumption that variables in the input data are correlated with each other and their features can be captured by projecting them into a lower dimensional subspace where the samples are significantly different from the original ones [25]. For AE, in the training period, the input matrix $\mathbf{X} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\right] \in \mathbb{R}^{n \times m}$ with normal samples is considered as the training set, where *n* and *m* denote the number of samples and variables in the original data. Then, the corresponding compressed representation matrix $\mathbf{Z} = \left[\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(n)}\right] \in \mathbb{R}^{n \times k}$ is reconstructed as $\hat{\mathbf{Z}} = \left[\hat{\mathbf{z}}^{(1)}, \hat{\mathbf{z}}^{(2)}, \dots, \hat{\mathbf{z}}^{(n)}\right] \in \mathbb{R}^{n \times k}$ to capture the features of $\mathbf{X}$, where *k* is the dimension of the compressed representation. The output of the model is reproduced by decoder as $\widehat{\mathbf{X}} = \left[\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \dots, \hat{\mathbf{x}}^{(n)}\right] \in \mathbb{R}^{n \times m'}$ $(m' \leq m)$, and mean squared error (MSE) is used as the reconstruction error (RE) between input and output is calculated as [113]

$$\boldsymbol{\varepsilon}^{(t)} = \left\|\mathbf{x}^{(t)} - \hat{\mathbf{x}}^{(t)}\right\|^2 \tag{2.37}$$

where $\boldsymbol{\varepsilon}^{(t)}$ denotes the RE of an input vector $\mathbf{x}^{(t)}$ at time t $(t = 1, 2, \dots, n)$. RE is obligated to update the parameters of encoder and decoder and calculate the anomaly score for anomaly detection.

Chapter 3
Nonlinear Latent Variable Regression for Process Monitoring

With the advancement of modern science and technologies, the industrial processes tend to be increasingly complex, leading to collinearity and nonlinearity. Collinearity in collected adjacent samples can be solved by introducing a regularization term in matrix inversions, while nonlinearity can be handled via nonlinear variants of MSA approaches. However, the existing nonlinear MSA methods such as KPLS and KCCA suffer from the same issues as described in Chapter 1. Considering the advantage of LVR over PLS and CCA [14], a nonlinear extension of LVR is proposed in this chapter, and a concurrent decomposition with subsequent PCA operation is conducted to obtain a comprehensive modeling and monitoring performance.

3.1 Kernel Latent Variable Regression and Its Monitoring Scheme

Define ϕ as a nonlinear projection indicator to map process variables from the original space into the feature space F, which is a higher-dimensional feature space. The original process matrix $\mathbf{X}$ is transformed into $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_n)]^\top \in \mathbb{R}^{n \times m_f}$ in the feature space. It is assumed that in the feature space, variables are linearly related with each other. Based on Eq. (2.16), the kernel matrix is constructed as $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top \in \mathbb{R}^{n \times n}$. For simplicity, With the kernel function in Eq. (2.18), the optimization objective of KLVR is designed as

$$\begin{aligned} \max_{\boldsymbol{\alpha}_{\mathbf{\Phi}}, \mathbf{q}} \; & J = \mathbf{q}^\top \mathbf{Y}^\top \mathbf{\Phi} \boldsymbol{\alpha}_{\mathbf{\Phi}} - \frac{\gamma}{2} \|\boldsymbol{\alpha}_{\mathbf{\Phi}}\|^2 \\ s.t. \; & \|\mathbf{\Phi}\boldsymbol{\alpha}_{\mathbf{\Phi}}\| = 1, \|\mathbf{q}\| = 1 \end{aligned} \tag{3.1}$$

where $\boldsymbol{\alpha}_{\mathbf{\Phi}} = \mathbf{\Phi}^\top \boldsymbol{\alpha}$ and $\mathbf{q}$ are projection vectors for $\mathbf{\Phi}$ and $\mathbf{Y}$, respectively. Replacing the calculation of $\mathbf{\Phi}$ with kernel matrix $\mathbf{K}$ simplifies Eq. (3.1), leading to

$$\begin{aligned} \max_{\boldsymbol{\alpha}, \mathbf{q}} \; & J = \mathbf{q}^\top \mathbf{Y}^\top \mathbf{K} \boldsymbol{\alpha} - \frac{\gamma}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \\ s.t. \; & \|\mathbf{K}\boldsymbol{\alpha}\| = 1, \|\mathbf{q}\| = 1 \end{aligned} \tag{3.2}$$

where $\boldsymbol{\alpha}$ and $\mathbf{q}$ are weighting vectors for $\mathbf{K}$ and $\mathbf{Y}$, respectively. In KLVR, the process score vector can be rewritten as $\mathbf{t} = \mathbf{\Phi}\boldsymbol{\alpha}_{\mathbf{\Phi}} = \mathbf{K}\boldsymbol{\alpha}$.

Lagrange multipliers are used to solve Eq. (3.2):

$$\mathcal{L} = \mathbf{q}^\top \mathbf{Y}^\top \mathbf{K}\boldsymbol{\alpha} - \frac{\gamma}{2}\boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha} + \frac{\lambda_\alpha}{2}(1 - \boldsymbol{\alpha}^\top \mathbf{K}^2\boldsymbol{\alpha}) + \frac{\lambda_q}{2}(1 - \mathbf{q}^\top \mathbf{q}). \tag{3.3}$$

After taking derivatives with respect to $\boldsymbol{\alpha}$ and $\mathbf{q}$ and setting them equal to zero, the following expressions are retained from Eq. (3.3).

$$\begin{aligned} \mathbf{K}\mathbf{Y}\mathbf{q} &= \lambda_\alpha(\mathbf{K} + \kappa\mathbf{I})\mathbf{K}\boldsymbol{\alpha} \\ \mathbf{Y}^\top \mathbf{K}\boldsymbol{\alpha} &= \lambda_q \mathbf{q} \end{aligned} \tag{3.4}$$

where $\kappa = \gamma/\lambda_\alpha$.

Arrange terms in Eq. (3.4), obtaining

$$\begin{aligned} (\mathbf{K} + \kappa\mathbf{I})^{-1}\mathbf{Y}\mathbf{Y}^\top \mathbf{K}\boldsymbol{\alpha} &= \lambda_\alpha \lambda_q \boldsymbol{\alpha} \\ \mathbf{Y}^\top (\mathbf{K} + \kappa\mathbf{I})^{-1}\mathbf{K}\mathbf{Y}\mathbf{q} &= \lambda_q \lambda_\alpha \mathbf{q}. \end{aligned} \tag{3.5}$$

Eq. (3.5) implies that λ_α and λ_q are eigenvalues of $(\mathbf{K} + \kappa\mathbf{I})^{-1}\mathbf{Y}\mathbf{Y}^\top \mathbf{K}$ and $\mathbf{Y}^\top (\mathbf{K} + \kappa\mathbf{I})^{-1}\mathbf{K}\mathbf{Y}$, respectively, while $\boldsymbol{\alpha}$ and $\mathbf{q}$ are their corresponding eigenvectors.

Accordingly, the outer structure of KLVR can be obtained by iterating the following relations until convergence.

(1) $\boldsymbol{\alpha} = (\mathbf{K} + \kappa\mathbf{I})^{-1}\mathbf{u}$, $\boldsymbol{\alpha} = \boldsymbol{\alpha}/\|\mathbf{K}\boldsymbol{\alpha}\|$;

(2) $\mathbf{t} = \mathbf{K}\boldsymbol{\alpha}$;

(3) $\mathbf{q} = \mathbf{Y}^\top \mathbf{t}$, $\mathbf{q} = \mathbf{q}/\|\mathbf{q}\|$;

(4) $\mathbf{u} = \mathbf{Y}\mathbf{q}$.

Afterwards, perform deflation on $\boldsymbol{\Phi}$ and $\mathbf{Y}$ to remove the effects of the extracted latent variables as follows:

$$\begin{aligned} \boldsymbol{\Phi} &\coloneqq (\mathbf{I} - \mathbf{t}\mathbf{t}^\top)\boldsymbol{\Phi} \\ \mathbf{Y} &\coloneqq (\mathbf{I} - \mathbf{t}\mathbf{t}^\top)\mathbf{Y}. \end{aligned}$$

Given $\mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top$, $\mathbf{K}$ is deflated as

$$\mathbf{K} \coloneqq (\mathbf{I} - \mathbf{t}\mathbf{t}^\top)\mathbf{K}(\mathbf{I} - \mathbf{t}\mathbf{t}^\top).$$

Due to the consistent inner and outer objectives, similar to LVR, inner model is not needed in KLVR. The detailed KLVR algorithm is summarized as follows:

1. Pre-process process and quality matrices to obtain centered **K** and scaled **Y**.

2. The iteration number is set as $i = 1$, and initialize **K** and **Y** as $\mathbf{K}_1 = \mathbf{K}$, $\mathbf{Y}_1 = \mathbf{Y}$, and $\mathbf{u}_i$ as the first column of **Y**.

3. Perform the following (1)-(4) iteratively until convergence.

(1) $\boldsymbol{\alpha}_i = (\mathbf{K}_i + \kappa\mathbf{I})^{-1}\mathbf{u}_i$, $\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_i / \|\mathbf{K}_i\boldsymbol{\alpha}_i\|$;

(2) $\mathbf{t}_i = \mathbf{K}_i\boldsymbol{\alpha}_i$;

(3) $\mathbf{q}_i = \mathbf{Y}_i^{\top}\mathbf{t}_i$, $\mathbf{q}_i = \mathbf{q}_i / \|\mathbf{q}_i\|$;

(4) $\mathbf{u}_i = \mathbf{Y}_i\mathbf{q}_i$.

4. Perform deflation on **K** and **Y** as

$$\mathbf{K}_{i+1} = (\mathbf{I} - \mathbf{t}_i\mathbf{t}_i^{\top})\mathbf{K}_i(\mathbf{I} - \mathbf{t}_i\mathbf{t}_i^{\top})$$
$$\mathbf{Y}_{i+1} = \left(\mathbf{I} - \mathbf{t}_i\mathbf{t}_i^{\top}\right)\mathbf{Y}_i.$$

5. Set $i \coloneqq i + 1$, and return to Step 3 until *l* latent variables are extracted.

With KLVR, $\boldsymbol{\Phi}$ and **Y** are decomposed by KLVR as

$$\boldsymbol{\Phi} = \mathbf{T}\mathbf{P}^{\top} + \boldsymbol{\Phi}_r$$
$$\mathbf{Y} = \mathbf{T}\mathbf{C}^{\top} + \mathbf{Y}_r$$

where $\mathbf{P} = \boldsymbol{\Phi}^{\top}\mathbf{T}$ and $\mathbf{C} = \mathbf{Y}^{\top}\mathbf{T}$ represent the loading matrices for $\boldsymbol{\Phi}$ and **Y**, while $\boldsymbol{\Phi}_r$ and $\mathbf{Y}_r$ are residuals of $\boldsymbol{\Phi}$ and **Y**, respectively.

According to the score matrix $\mathbf{T} = \boldsymbol{\Phi}\mathbf{R}$,

$$\mathbf{R} = \mathbf{W}(\mathbf{P}^{\top}\mathbf{W})^{-1} = \boldsymbol{\Phi}^{\top}\mathbf{A}(\mathbf{T}^{\top}\mathbf{K}\mathbf{A})^{-1} \tag{3.6}$$

where $\mathbf{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_n]^{\top} \in R^{n\times l}$, and $\mathbf{W} = \boldsymbol{\Phi}^{\top}\mathbf{A}$.

For test samples $\mathbf{X}_t$, its kernel matrix $\mathbf{K}_t$ should be first centered as

$$\mathbf{K}_t \coloneqq (\mathbf{K}_t - \frac{1}{n}\mathbf{1}_k\mathbf{1}_n{}^{\top}\mathbf{K})(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n{}^{\top})$$

where **K** is the kernel matrix of the training data **X**.

For a single test sample $\mathbf{x}_{\text{new}}$, its kernel is designed as $k(\mathbf{x}_i, \mathbf{x}_{\text{new}}) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_{\text{new}}) \rangle$, where $\mathbf{x}_i$ represents the i^{th} sample of the training data. The score vector $\mathbf{t}_{\text{new}}$ is calculated as

$$\mathbf{t}_{\text{new}} = \mathbf{R}^{\top} \phi(\mathbf{x}_{\text{new}}) = \mathbf{\Phi}^{\top} \mathbf{A} (\mathbf{T}^{\top} \mathbf{K} \mathbf{A})^{-1} \mathbf{k}_t \tag{3.7}$$

where $\mathbf{k}_t = \mathbf{\Phi} \phi(\mathbf{x}_{\text{new}})$.

Two statistics, T^2 and Q, are employed to detect the variations in the principal component subspace and residual subspace, which are decomposed with KLVR. T^2 and Q indices are defined as

$$T^2 = \mathbf{t}_{\text{new}}^{\top} \mathbf{\Lambda}^{-1} \mathbf{t}_{\text{new}} \tag{3.8}$$

$$Q = \| \phi_r(\mathbf{x}_{\text{new}}) \|^2 \tag{3.9}$$

where $\mathbf{\Lambda} = \frac{1}{n-1} \mathbf{T}^{\top} \mathbf{T}$ includes the variances of principal components, and $\phi_r(\mathbf{x}_{\text{new}}) = \phi(\mathbf{x}_{\text{new}}) - \mathbf{P} \mathbf{t}_{\text{new}}$ represents the residual of $\phi(\mathbf{x}_{\text{new}})$. With the model parameters obtained from KLVR model, the Q index is denoted as

$$Q = \phi(\mathbf{x}_{\text{new}})^{\top} \phi(\mathbf{x}_{\text{new}}) - 2 \mathbf{k}_t^{\top} \mathbf{T} \mathbf{t}_{\text{new}} + \mathbf{t}_{\text{new}}^{\top} \mathbf{T}^{\top} \mathbf{K} \mathbf{T} \mathbf{t}_{\text{new}} \tag{3.10}$$

where $\phi(\mathbf{x}_{\text{new}})^{\top} \phi(\mathbf{x}_{\text{new}}) = 1 - \frac{2}{n} \sum_{i=1}^{n} k(\mathbf{x}_i, \mathbf{x}_{\text{new}}) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(\mathbf{x}_i, \mathbf{x}_j)$ $(i, j = 1, 2, \ldots, n)$.

The control limits of T^2 and Q indices are designed in the same way as in Eq. (2.9) [3].

3.2 Concurrent KLVR-based Monitoring

KLVR-based monitoring only focuses on the monitoring of quality anomalies, failing to provide a comprehensive monitoring for variations in both process and quality variables. Thus, a concurrent modeling and fault detection framework is developed based on KLVR, which constructs a multilayer monitoring structure to capture and monitor variations in both quality and process spaces. The proposed method is named concurrent KLVR (CKLVR).

CKLVR decomposes the original data space into five subspaces, including covariation subspace (CVS), input-principal subspace (IPS), input-residual subspace (IRS), output-principal subspace (OPS) and output-residual subspace (ORS). The details of CKLVR algorithm are outlined as follows.

1. Scale **X** and **Y** to zero mean and unit variance, and **X** is processed to obtain the mean-centered **K**.

2. Perform KLVR on the mean-centered **K** and **Y** with l latent variables to obtain **T**, **C**, and **R**.

3. Construct the predictable output matrix in CVS as $\widehat{\mathbf{Y}} = \mathbf{T}\mathbf{C}^{\top}$, and then perform singular value decomposition (SVD) on $\widehat{\mathbf{Y}}$ to capture the predictable variations with l_c principal components as

$$\widehat{\mathbf{Y}} = \mathbf{U}_c\mathbf{D}_c\mathbf{V}_c^{\top} \equiv \mathbf{U}_c\mathbf{Q}_c^{\top} \tag{3.11}$$

where $\mathbf{U}_c$ reflects the covariations in $\boldsymbol{\Phi}$ which are predictable from $\widehat{\mathbf{Y}}$. $\mathbf{Q}_c = \mathbf{V}_c\mathbf{D}_c$ includes all l_c nonzero singular values in descending order and their corresponding right singular vectors. Since $\mathbf{V}_c$ is orthonormal,

$$\mathbf{U}_c = \widehat{\mathbf{Y}}\mathbf{V}_c\mathbf{D}_c^{-1} = \mathbf{T}\mathbf{C}^{\top}\mathbf{V}_c\mathbf{D}_c^{-1} = \boldsymbol{\Phi}\mathbf{R}\mathbf{C}^{\top}\mathbf{V}_c\mathbf{D}_c^{-1} \equiv \boldsymbol{\Phi}\mathbf{R}_c \tag{3.12}$$

where $\mathbf{R}_c = \mathbf{R}\mathbf{C}^{\top}\mathbf{V}_c\mathbf{D}_c^{-1} = \boldsymbol{\Phi}^{\top}\mathbf{A}(\mathbf{T}^{\top}\mathbf{K}\mathbf{A})^{-1}\mathbf{C}^{\top}\mathbf{V}_c\mathbf{D}_c^{-1}$.

4. The unpredictable output matrix is denoted as $\widetilde{\mathbf{Y}}_c = \mathbf{Y} - \mathbf{U}_c\mathbf{Q}_c^{\top}$, and perform PCA on $\widetilde{\mathbf{Y}}_c$ with l_y principal components

$$\widetilde{\mathbf{Y}}_c = \mathbf{T}_y\mathbf{P}_y^{\top} + \widetilde{\mathbf{Y}} \tag{3.13}$$

where $\mathbf{T}_y = \widetilde{\mathbf{Y}}_c\mathbf{P}_y$ is the output-principal score matrix, $\mathbf{P}_y$ denotes the output-principal loading matrix, and $\widetilde{\mathbf{Y}}$ represents the output residual matrix.

5. Variations that are quality-irrelevant but process-relevant can be extracted by $\widetilde{\boldsymbol{\Phi}}_c = \boldsymbol{\Phi} - \mathbf{U}_c\mathbf{R}_c^{\dagger}$, where $\mathbf{R}_c^{\dagger} = (\mathbf{R}_c^{\top}\mathbf{R}_c)^{-1}\mathbf{R}_c^{\top}$. Theoretically, the next step is to perform PCA on $\widetilde{\boldsymbol{\Phi}}_c$ with l_x principal components as

$$\widetilde{\boldsymbol{\Phi}}_c = \mathbf{T}_x\mathbf{P}_x^{\top} + \widetilde{\boldsymbol{\Phi}} \tag{3.14}$$

where $\mathbf{T}_x = \widetilde{\boldsymbol{\Phi}}_c\mathbf{P}_x$ denotes the input-principal score matrix, $\mathbf{P}_x$ is the input-principal loading matrix, and $\widetilde{\boldsymbol{\Phi}}$ represents the input residuals.

The explicit nonlinear calculation can be circumvented by performing PCA on $\mathbf{K}_c = \widetilde{\boldsymbol{\Phi}}_c\widetilde{\boldsymbol{\Phi}}_c^{\top}$, and the input-principal score matrix $\mathbf{T}_x$ can be expressed as

$$\mathbf{T}_x = \widetilde{\boldsymbol{\Phi}}_c\mathbf{P}_x = \mathbf{K}_c\mathbf{W}_x \tag{3.15}$$

where $\mathbf{W}_x$ contains scaled eigenvectors of $\frac{1}{n}\widetilde{\boldsymbol{\Phi}}_c\widetilde{\boldsymbol{\Phi}}_c^{\top}$ corresponding to the l_x largest eigenvalues. Besides, the input-principal loading matrix $\mathbf{P}_x$ can be represented as $\mathbf{P}_x = \widetilde{\boldsymbol{\Phi}}_c^{\top}\mathbf{W}_x$.

Consequently, CKLVR decomposes the process and quality data as

$$\begin{cases} \boldsymbol{\Phi} = \mathbf{U}_c\mathbf{R}_c^{\dagger} + \mathbf{T}_x\mathbf{P}_x^{\top} + \widetilde{\boldsymbol{\Phi}} \\ \mathbf{Y} = \mathbf{U}_c\mathbf{Q}_c^{\top} + \mathbf{T}_y\mathbf{P}_y^{\top} + \widetilde{\mathbf{Y}} \end{cases} \tag{3.16}$$

where $\mathbf{U}_c\mathbf{R}_c^{\dagger}$ represents the predictable quality-relevant variations in process variables, showing the relations between process and quality variables, while $\mathbf{U}_c\mathbf{Q}_c^{\top}$ includes predictable quality variations, and they build up CVS together. $\mathbf{T}_x\mathbf{P}_x^{\top}$ in IPS captures major process-relevant but quality-irrelevant variations, while $\widetilde{\boldsymbol{\Phi}}$ in IRS represents the process residuals. $\mathbf{T}_y\mathbf{P}_y^{\top}$ in OPS contains major quality-relevant variations that are unobservable for the process data, and ORS is constructed by the residuals $\widetilde{\mathbf{Y}}$.

For a new data sample pair $(\mathbf{x}_{\text{new}}, \mathbf{y}_{\text{new}})$ can be decomposed by CKLVR into

$$\begin{cases} \phi(\mathbf{x}_{\text{new}}) = \mathbf{R}_c^{\dagger\top}\mathbf{u}_{c,\text{new}} + \mathbf{P}_x\mathbf{t}_{x,\text{new}} + \tilde{\phi}(\mathbf{x}_{\text{new}}) \\ \mathbf{y}_{\text{new}} = \mathbf{Q}_c\mathbf{u}_{c,\text{new}} + \mathbf{P}_y\mathbf{t}_{y,\text{new}} + \tilde{\mathbf{y}}_{\text{new}} \end{cases} \tag{3.17}$$

where score vectors $\mathbf{u}_{c,\text{new}} = \mathbf{R}_c^{\top}\tilde{\phi}(\mathbf{x}_{\text{new}})$, $\mathbf{t}_{x,\text{new}} = \mathbf{P}_x^{\top}\tilde{\phi}_c(\mathbf{x}_{\text{new}})$, and $\mathbf{t}_{y,\text{new}} = \mathbf{P}_y^{\top}\tilde{\mathbf{y}}_{c,\text{new}}$ are calculated directly through the kernel form $\mathbf{k}_t = \boldsymbol{\Phi}\phi(\mathbf{x}_{\text{new}})$, which are specifically presented as follows:

$$\mathbf{u}_{c,\text{new}} = \mathbf{D}_c^{-\top}\mathbf{V}_c^{\top}\mathbf{C}(\mathbf{T}^{\top}\mathbf{K}\mathbf{A})^{-\top}\mathbf{A}^{\top}\mathbf{k}_t \tag{3.18}$$

$$\mathbf{t}_{x,\text{new}} = \mathbf{W}_x^{\top}\left(\mathbf{k}_t - \mathbf{U}_c\boldsymbol{\Gamma}_c\mathbf{k}_t - \mathbf{K}\boldsymbol{\Gamma}_c^{\top}\mathbf{u}_{c,\text{new}} + \mathbf{U}_c\boldsymbol{\Gamma}_c\mathbf{K}\boldsymbol{\Gamma}_c^{\top}\mathbf{u}_{c,\text{new}}\right) \tag{3.19}$$

$$\mathbf{t}_{y,\text{new}} = \mathbf{P}_y^{\top}\mathbf{Y} - \mathbf{P}_y^{\top}\mathbf{Q}_c\mathbf{D}_c^{-\top}\mathbf{V}_c^{\top}\mathbf{C}(\mathbf{T}^{\top}\mathbf{K}\mathbf{A})^{-\top}\mathbf{A}^{\top}\mathbf{k}_t \tag{3.20}$$

where $\boldsymbol{\Gamma}_c = (\mathbf{D}_c^{-\top}\mathbf{V}_c^{\top}\mathbf{C}(\mathbf{T}^{\top}\mathbf{K}\mathbf{A})^{-\top}\mathbf{A}\mathbf{K}\mathbf{A}(\mathbf{T}^{\top}\mathbf{K}\mathbf{A})^{-1}\mathbf{C}^{\top}\mathbf{V}_c\mathbf{D}_c^{-1})^{-1}\mathbf{D}_c^{-\top}\mathbf{V}_c^{\top}\mathbf{C}(\mathbf{T}^{\top}\mathbf{K}\mathbf{A})^{-\top}\mathbf{A}$ is defined for ease of presentation. In addition, $\tilde{\phi}(\mathbf{x}_{\text{new}})$ and $\tilde{\mathbf{y}}_{\text{new}}$ can be attained by

$$\begin{cases} \tilde{\phi}(\mathbf{x}_{\text{new}}) = \tilde{\phi}_c(\mathbf{x}_{\text{new}}) - \mathbf{P}_x\mathbf{t}_{x,\text{new}} \\ \tilde{\mathbf{y}}_{\text{new}} = \left(\mathbf{I} - \mathbf{P}_y\mathbf{P}_y^{\top}\right)\tilde{\mathbf{y}}_{c,\text{new}}. \end{cases} \tag{3.21}$$

Then, the monitoring statistic of each subspace is designed to indicate process and quality anomalies with different alarming levels.

Since the predictable quality-relevant score vector $\mathbf{u}_{c,\text{new}}$, process-relevant score vector $\mathbf{t}_{x,\text{new}}$, and unpredictable quality-relevant score vector $\mathbf{t}_{y,\text{new}}$ involves variations of the whole system, for a new testing sample pair $(\mathbf{x}_{\text{new}}, \mathbf{y}_{\text{new}})$, the Hotelling's T^2, T_c^2, T_x^2, and T_y^2 are applied for monitoring:

$$T_c^2 = (n-1)\mathbf{u}_{c,\text{new}}^\top \mathbf{u}_{c,\text{new}} \tag{3.22}$$

$$T_x^2 = \mathbf{t}_{x,\text{new}}^\top \boldsymbol{\Lambda}_x^{-1} \mathbf{t}_{x,\text{new}} \tag{3.23}$$

$$T_y^2 = \mathbf{t}_{y,\text{new}}^\top \boldsymbol{\Lambda}_y^{-1} \mathbf{t}_{y,\text{new}} \tag{3.24}$$

where $\boldsymbol{\Lambda}_x = \frac{1}{n-1}\mathbf{T}_x^\top \mathbf{T}_x$ and $\boldsymbol{\Lambda}_y = \frac{1}{n-1}\mathbf{T}_y^\top \mathbf{T}_y$ represent the variances of process and quality variables with respect to principal components l_x and l_y, respectively.

The Q_x and Q_y indices are designed for residuals of process variations, where Q_x index is

$$\begin{aligned} Q_x &= \left\|\tilde{\phi}(\mathbf{x}_{\text{new}})\right\|^2 = \left\|\tilde{\phi}_c(\mathbf{x}_{\text{new}}) - \mathbf{P}_x \mathbf{t}_{x,\text{new}}\right\|^2 \\ &= \tilde{\phi}_c^\top(\mathbf{x}_{\text{new}})\tilde{\phi}_c(\mathbf{x}_{\text{new}}) - \mathbf{2}\tilde{\phi}_x^\top(\mathbf{x}_{\text{new}})\mathbf{P}_x \mathbf{t}_{x,\text{new}} + \mathbf{t}_{x,\text{new}}^\top \mathbf{P}_x^\top \mathbf{P}_x \mathbf{t}_{x,\text{new}} \end{aligned} \tag{3.25}$$

where $\tilde{\phi}_c(\mathbf{x}_{\text{new}}) = \tilde{\phi}(\mathbf{x}_{\text{new}}) - \mathbf{R}_c^{\dagger\top}\mathbf{u}_{c,\text{new}} = \phi(\mathbf{x}_{\text{new}}) - \boldsymbol{\Phi}^\top \boldsymbol{\Gamma}_c^\top \mathbf{u}_{c,\text{new}}$, and every term in Eq. (3.25) can be computed as

$$\tilde{\phi}_c^\top(\mathbf{x}_{\text{new}})\tilde{\phi}_c(\mathbf{x}_{\text{new}}) = 1 - \mathbf{2}\mathbf{u}_{c,\text{new}}^\top \boldsymbol{\Gamma}_c \mathbf{k}_t + \mathbf{u}_{c,\text{new}}^\top \boldsymbol{\Gamma}\mathbf{K}\boldsymbol{\Gamma}_c^\top \mathbf{u}_{c,\text{new}}$$

$$\tilde{\phi}_x^\top(\mathbf{x}_{\text{new}})\mathbf{P}_x \mathbf{t}_{x,\text{new}} = \left(\mathbf{k}_t^\top - \mathbf{u}_{c,\text{new}}^\top \boldsymbol{\Gamma}_c \mathbf{K}\right)(\mathbf{I} - \mathbf{U}_c\boldsymbol{\Gamma}_c)^\top \mathbf{W}_x \mathbf{t}_{x,\text{new}}$$

$$\mathbf{t}_{x,\text{new}}^\top \mathbf{P}_x^\top \mathbf{P}_x \mathbf{t}_{x,\text{new}} = \mathbf{t}_{x,\text{new}}^\top \mathbf{W}_x^\top (\mathbf{I} - \mathbf{U}_c\boldsymbol{\Gamma}_c)\mathbf{K}(\mathbf{I} - \boldsymbol{\Gamma}_c^\top \mathbf{U}_c^\top)\mathbf{W}_x \mathbf{t}_{x,\text{new}}$$

where $\tilde{\boldsymbol{\Phi}}_c = \boldsymbol{\Phi} - \mathbf{U}_c\mathbf{R}_c^\dagger = \boldsymbol{\Phi} - \mathbf{U}_c\,\boldsymbol{\Gamma}_c\,\boldsymbol{\Phi}$, $\mathbf{P}_x = \tilde{\boldsymbol{\Phi}}_c^\top \mathbf{W}_x$, and $\tilde{\boldsymbol{\Phi}}_c\tilde{\boldsymbol{\Phi}}_c^\top = (\mathbf{I} - \mathbf{U}_c\boldsymbol{\Gamma}_c)\mathbf{K}(\mathbf{I} - \boldsymbol{\Gamma}_c^\top \mathbf{U}_c^\top)$.

In addition, the Q_y statistic can be calculated as

$$\begin{aligned} Q_y &= \|\tilde{\mathbf{y}}_{\text{new}}\|^2 = \left\|\tilde{\mathbf{y}}_{c,\text{new}} - \mathbf{P}_y \mathbf{t}_{y,\text{new}}\right\|^2 = \tilde{\mathbf{y}}_{c,\text{new}}^\top \left(\mathbf{I} - \mathbf{P}_y\mathbf{P}_y^\top\right)\tilde{\mathbf{y}}_{c,\text{new}} \\ &= \left(\mathbf{y}_{\text{new}} - \mathbf{Q}_c\mathbf{u}_{c,\text{new}}\right)^\top \left(\mathbf{I} - \mathbf{P}_y\mathbf{P}_y^\top\right)\left(\mathbf{y}_{\text{new}} - \mathbf{Q}_c\mathbf{u}_{c,\text{new}}\right). \end{aligned} \tag{3.26}$$

Table 2: Concurrent monitoring statistics and corresponding control limits

Statistics	Control limits	Alarming level
T_c^2	$T_{c,\text{cl}}^2 = \chi_{\alpha,l_c}^2$	L_1
T_x^2	$T_{x,\text{cl}}^2 = \chi_{\alpha,l_x}^2$	L_3
Q_x	$Q_{x,\text{cl}} = g_x\chi_{\alpha,h_x}^2$	L_2
T_y^2	$T_{y,\text{cl}}^2 = \chi_{\alpha,l_y}^2$	L_2
Q_y	$Q_{y,\text{cl}} = g_y\chi_{\alpha,h_y}^2$	L_3

The corresponding control limits of T_c^2, T_x^2, T_y^2, Q_x, and Q_y indices are summarized in Table 2, where g_x and g_y represent scaling factors of control limits, while h_x and h_y denote the degrees of freedom of χ^2-distribution used in the computation of $Q_{x,\mathrm{cl}}$ and $Q_{y,\mathrm{cl}}$, respectively [117].

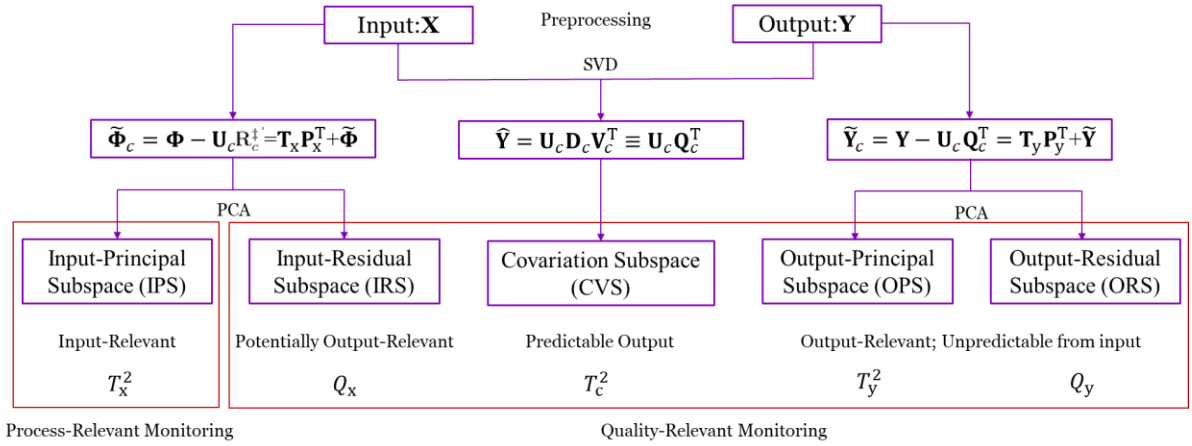

Figure 1: Multilayer monitoring framework based on CKLVR

The CKLVR-based multilayer monitoring framework is illustrated in Figure 1, which decomposes the original process and quality data into five subspaces, namely CVS, OPS, ORS, IPS, and IRS. Different fault alarms align with faults detected in different subspaces, which is founded on whether anomalies observed by statistics are quality-relevant or not. More concretely, L_1, L_2, and L_3 are attached for faults with descending importance. The monitoring procedure for a new sample is summarized.

1. If $T_c^2 > T_{c,\mathrm{cl}}^2$, the L_1 fault alarm is fired in CVS with $(1-\alpha)\times 100\%$ confidence, indicating that a quality-relevant fault is detected.

2. If $T_y^2 > T_{y,\mathrm{cl}}^2$, a quality-relevant fault is detected with $(1-\alpha)\times 100\%$ confidence, and it is attached with a L_2 alarming level. This fault is in OPS, and it is unpredictable from process variables.

3. If $Q_y > Q_{y,\mathrm{cl}}$, a quality-irrelevant fault with a L_3 alarming level is detected with $(1-\alpha)\times 100\%$ confidence.

4. If $T_x^2 > T_{x,\mathrm{cl}}^2$, a quality-relevant but process-irrelevant fault is detectable for IPS with $(1-\alpha)\times 100\%$ confidence, showing that the fault is attached to the L_3 alarming level.

5. If $Q_x > Q_{x,\mathrm{cl}}$, a potentially quality-relevant fault can be observed in IRS with $(1-\alpha)\times 100\%$ confidence, which implies the fault pertains to the L_2 alarm.

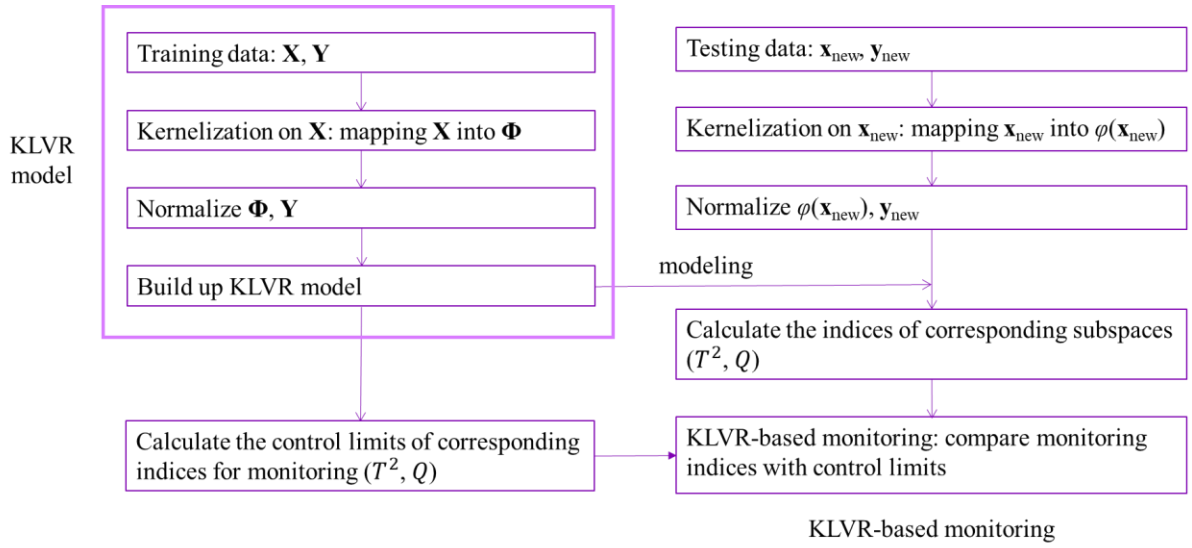

Figure 2: CKLVR-based monitoring diagram

Additionally, the whole CKLVR-based monitoring scheme is demonstrated in Figure 2 for better understanding.

3.3 Synthetic Case Study

In this section, a synthetic case study is employed to illustrate the effectiveness of CKLVR-based multilayer monitoring. The fault-free process and quality variables are generated as follows:

$$\begin{aligned}
\mathbf{x}_1 &= \mathbf{z}_1 + 3\sin(\mathbf{z}_2) + \mathbf{z}_3 + \mathbf{e}_1 \\
\mathbf{x}_2 &= 3\mathbf{z}_1 + \cos(\mathbf{z}_3) + \mathbf{e}_2 \\
\mathbf{x}_3 &= 4\mathbf{z}_1^2 + \mathbf{z}_2 + 3\mathbf{z}_3 + \mathbf{e}_3 \\
\mathbf{x}_4 &= 4\mathbf{z}_1 + 4\sin(\mathbf{z}_2) + \mathbf{e}_4 \\
\mathbf{x}_5 &= \sin(\mathbf{z}_2) + \mathbf{e}_5
\end{aligned}$$

$$\begin{aligned}
\mathbf{y}_1 &= \sin(\mathbf{x}_1) + 2\mathbf{x}_2 + \cos(\mathbf{x}_3) + \mathbf{x}_4 + \mathbf{v}_1 \\
\mathbf{y}_2 &= 3\mathbf{x}_1 + \cos(\mathbf{x}_2) + 4\mathbf{x}_4 + \mathbf{v}_2
\end{aligned}$$

where $z_k \sim \mathcal{U}([0,1])$, $e_k \sim \mathcal{N}(0, 0.2^2)$, and $v_k \sim \mathcal{N}(0, 0.1^2)$. Specifically, $\mathcal{U}([0,1])$ represents a uniform distribution in the interval [0,1], and $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian distribution with mean μ and variance σ^2.

Here, 200 samples are generated under normal conditions to form a CKLVR model, and 400 samples are generated for testing. The model parameters are determined by a 10-fold cross validation jointly:

For CKLVR, $l = 1$, $\kappa = 0.004$, $c = 110$, $l_c = 1$, $l_x = 4$, and $l_y = 2$, while for KLVR, $l = 1$, $\kappa = 0.004$, and $c = 110$. Note that the Q_y statistic is null since l_y is equal to the number of quality data p. The confidence level of control limit is chosen as 95%.

Thus, to verify the monitoring performance, four faults are added from the 201st samples to the end in subspaces including CVS, OPS, IPS, and IRS as follows:

$$\mathbf{x} = \mathbf{x}^* + \Xi_x f_x$$
$$\mathbf{y} = \mathbf{y}^* + \Xi_y f_y$$

where $\mathbf{x}^*$ and $\mathbf{y}^*$ are fault-free samples, Ξ_x and Ξ_y represent matrices of different faulty cases, and f_x and f_y denote fault magnitudes, respectively.

To show the ground truth for each case, KPCA-based monitoring is performed in each faulty case on process variables and quality variables, respectively. T_y^2 is used to describe quality variables, while T_x^2 and Q_x are statistics for process monitoring.

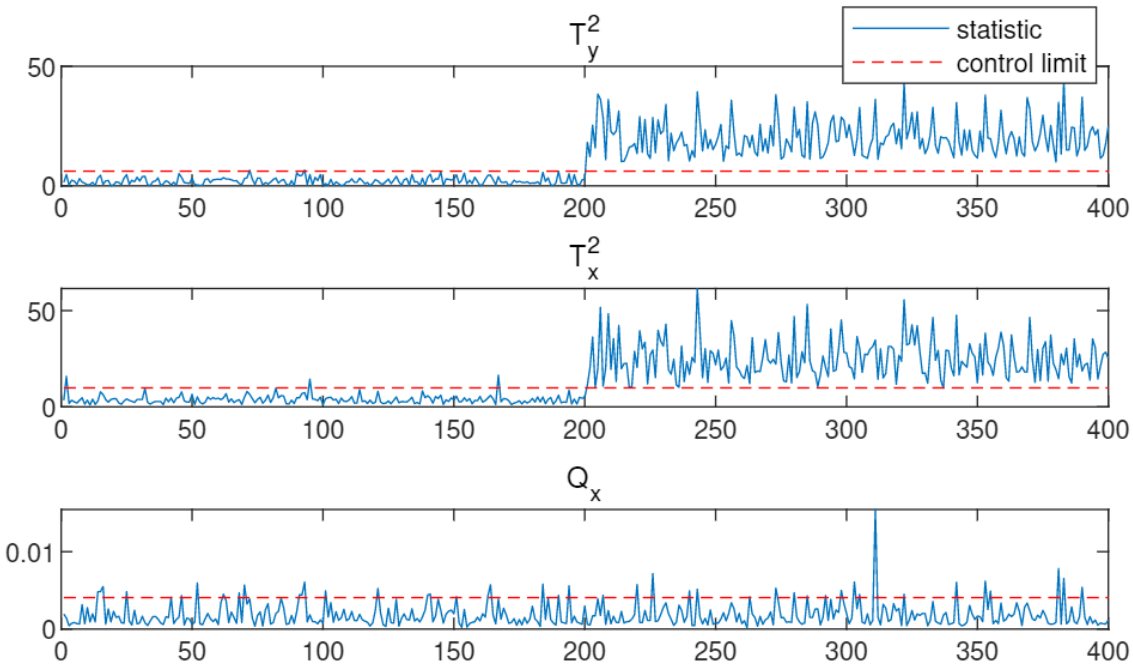

Figure 3: KPCA-based monitoring results with fault in CVS ($f_x = 1$)

Concretely, the monitoring results of KPCA, CKLVR, and KLVR for each faulty scenario are demonstrated as follows: Figures 3, 4, and 5 describe the corresponding results of the fault in CVS;

Figures 6, 7, and 8 show the monitoring results of the fault in OPS; Figures 9, 10, and 11 display the monitoring results of the fault in IPS; and Figures 12, 13, and 14 demonstrate the results of the fault in IRS.

3.3.1 Fault in Covariation Subspace

To create a fault in CVS, $\boldsymbol{\Xi}_x$ is chosen as the first column of $\mathbf{R}_c$ that is normalized to unit norm. As is shown in Figure 3, although the fault has influence on both process and quality variables, only the process part relevant to quality variable is affected, and Q_x is within the control limit.

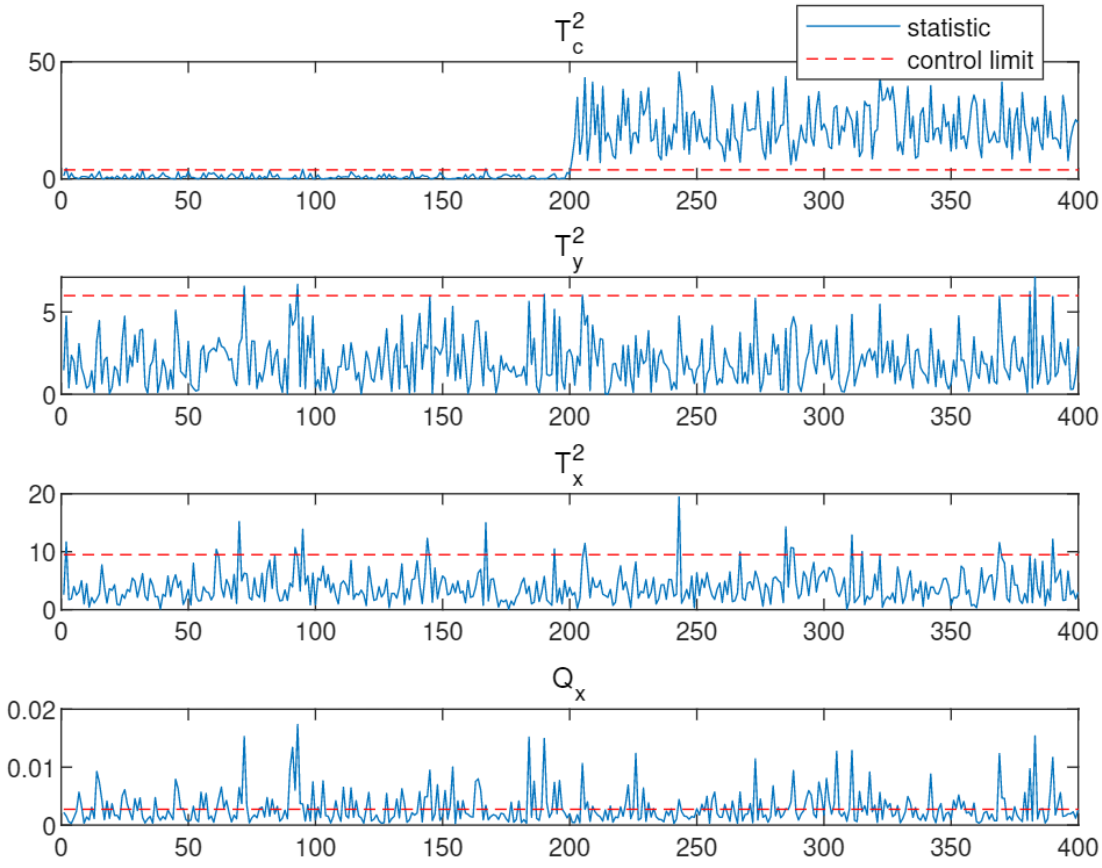

Figure 4: CKLVR-based monitoring results with fault in CVS ($f_x = 1$)

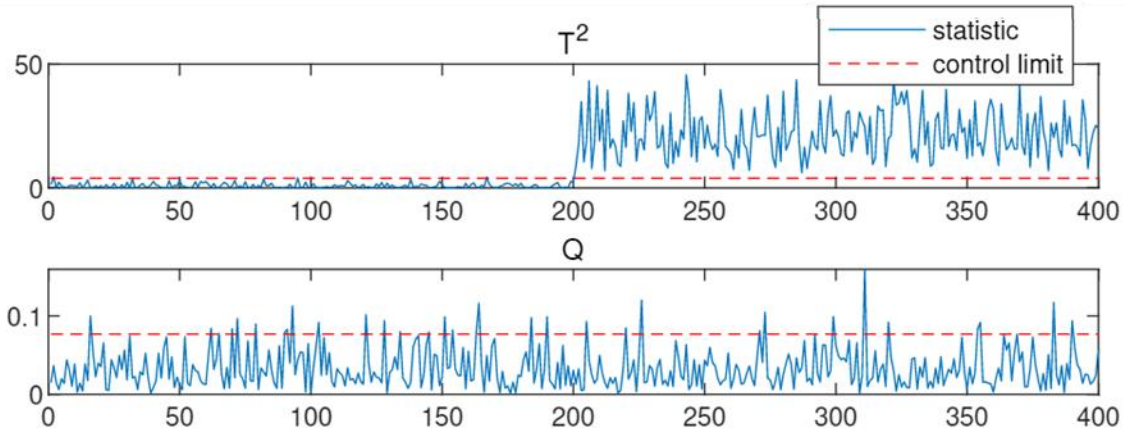

Figure 5: KLVR-based monitoring results with fault in CVS ($f_x = 1$)

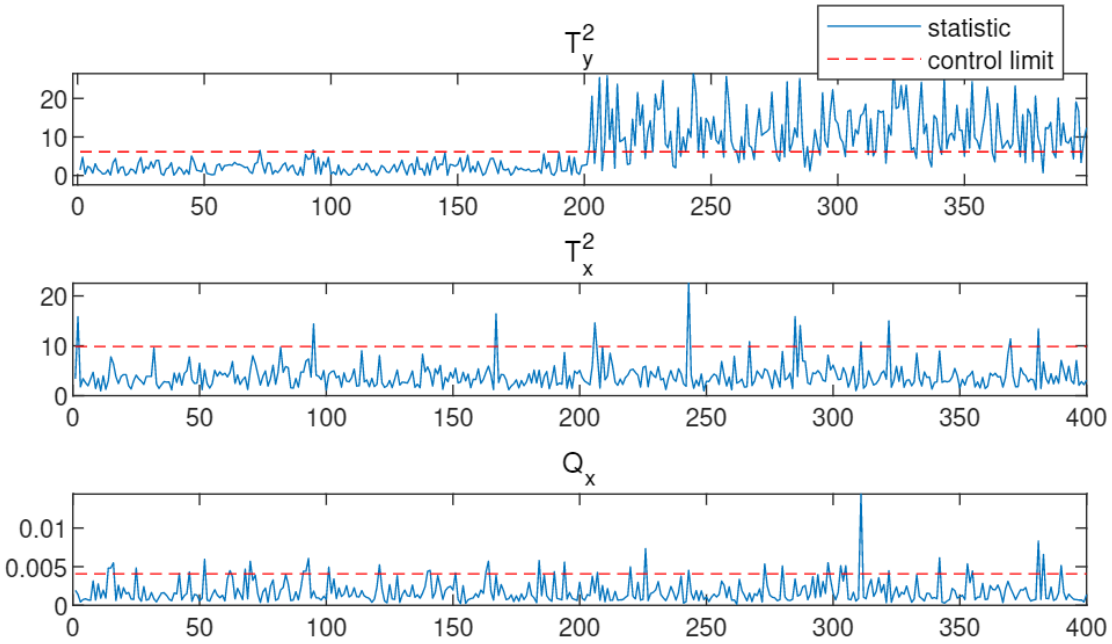

Figure 6: KPCA-based monitoring results with fault in OPS ($f_y = 4$)

The fault detection results of CKLVR and KVLR are displayed in Figures 4 and 5, respectively. As is shown in Figure 4, CKLVR-based monitoring provides a clear and accurate monitoring result for each subspace: only the T_c^2 statistic raises the L_1 fault alarm, while other monitoring indices are not affected, implying that it is a predictable output-relevant fault, which is in line with the result of KPCA-based monitoring. By contrast, although KLVR in Figure 5 observes the fault with the T^2 statistic, no

further information is provided. Meanwhile, process-relevant variations and unpredictable quality-variations remain unexploited.

3.3.2 Fault in Output-Principal Subspace

To introduce a fault to OPS, the fault direction matrix $\boldsymbol{\Xi}_y$ is selected as the first column of $\mathbf{P}_y$ which is normalized to unit norm. Figure 6 shows that only the T_y^2 index exceeds its control limit, which indicates that the fault is input-irrelevant but output-relevant.

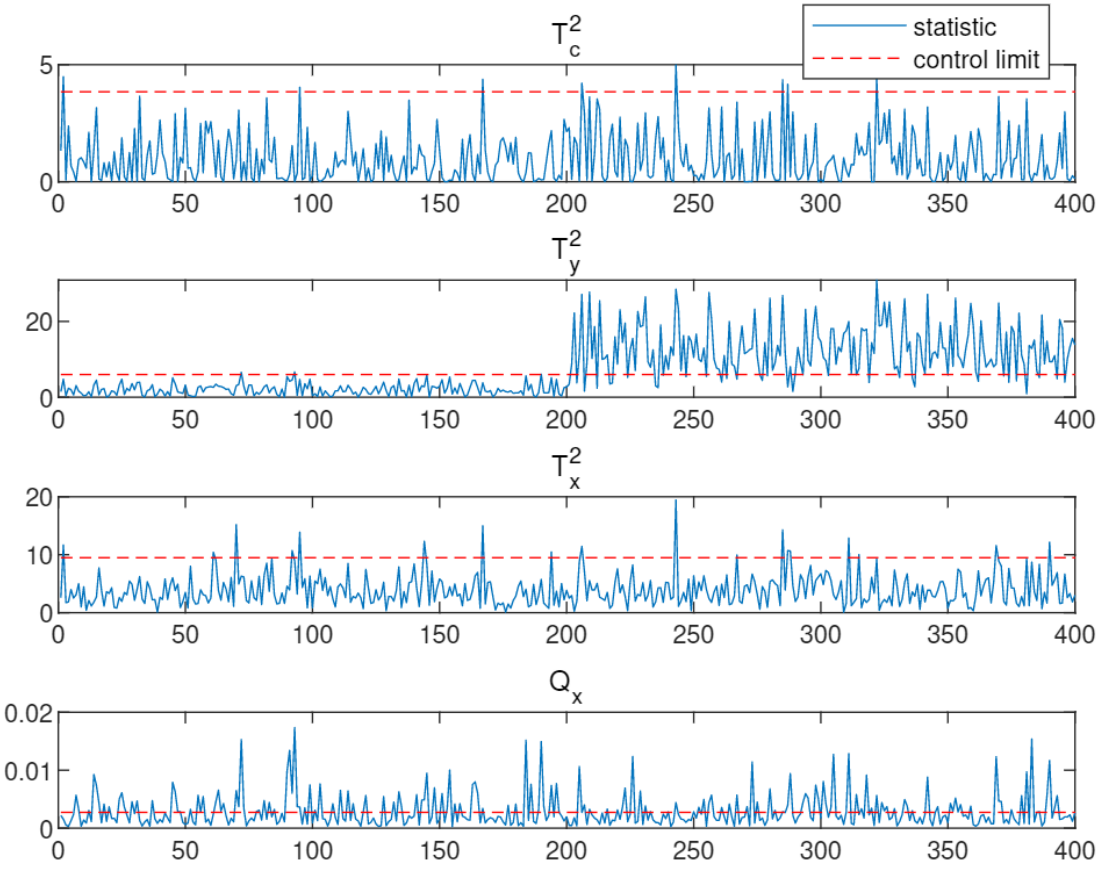

Figure 7: CKLVR-based monitoring results with fault in OPS ($f_y = 4$)

Figure 7 demonstrates CKLVR-based monitoring. Since this quality-relevant fault is not related to process data, T_c^2 remains unchanged. Only T_y^2 successfully identifies the L_1 fault alarm, which indicates it is quality-relevant but process-irrelevant. T_x^2 and Q_x that are related to process variables are within their corresponding limits. Accordingly, the monitoring results of CKLVR are consistent with

KPCA-based monitoring. However, KLVR-based monitoring in Figure 8 illustrates that the fault is unobservable for both T^2 and Q statistics, and thus it is concluded that the monitoring performance of CKLVR is more comprehensive than that of KLVR.

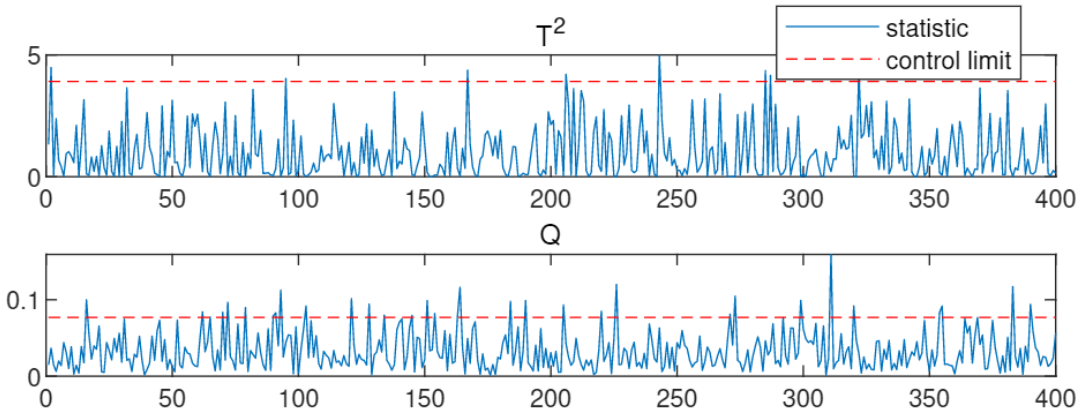

Figure 8: KLVR-based monitoring results with fault in OPS ($f_y = 4$)

3.3.3 Fault in Input-Principal Subspace

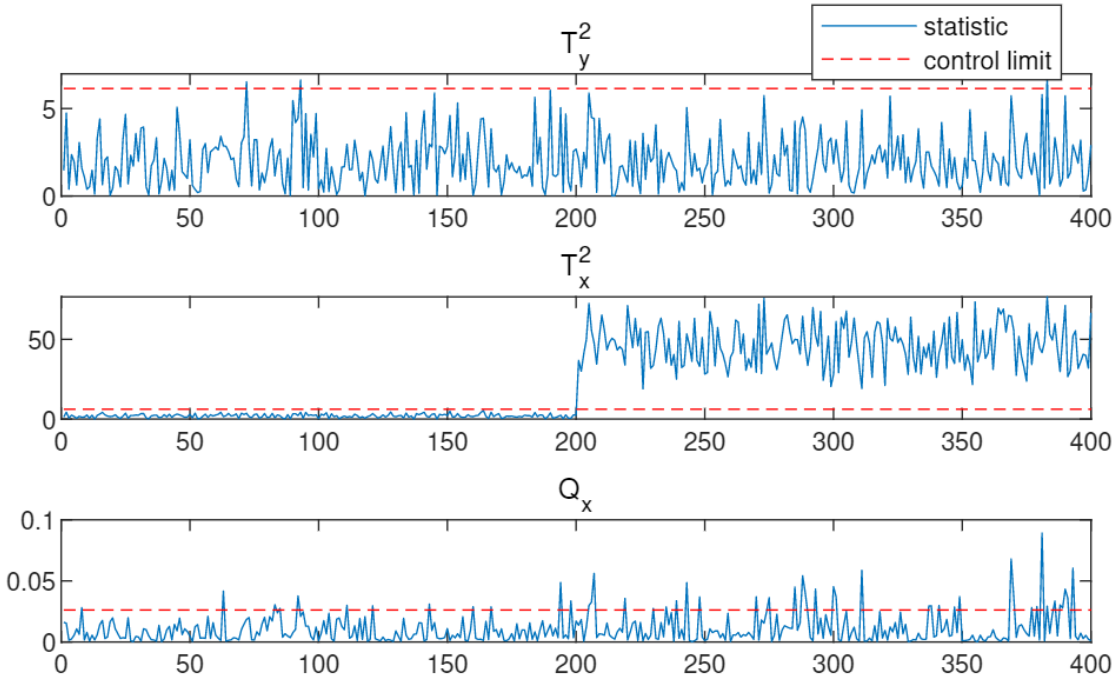

Figure 9: KPCA-based monitoring results with fault in IPS ($f_x = 1$)

The fault direction matrix $\boldsymbol{\Xi}_x$ is selected as the first column of $\mathbf{P}_x$ to introduce a fault to IPS which is also normalized to unit norm. According to Figure 9, KPCA-based monitoring shows that the added fault is quality-irrelevant but process-relevant.

Figure 10 presents CKLVR-based monitoring, which illustrates that the L_3 fault alarm is only fired by T_x^2 while other statistics remain normal. This indicates that the fault only affects process variations, which aligns with the fault design. In contrast, KLVR only observes the fault by the Q index, as is shown in Figure 11. Since the Q statistic reflects variations that are not excited in the training phase but are potentially quality-relevant, no valuable conclusions can be conducted from Figure 11.

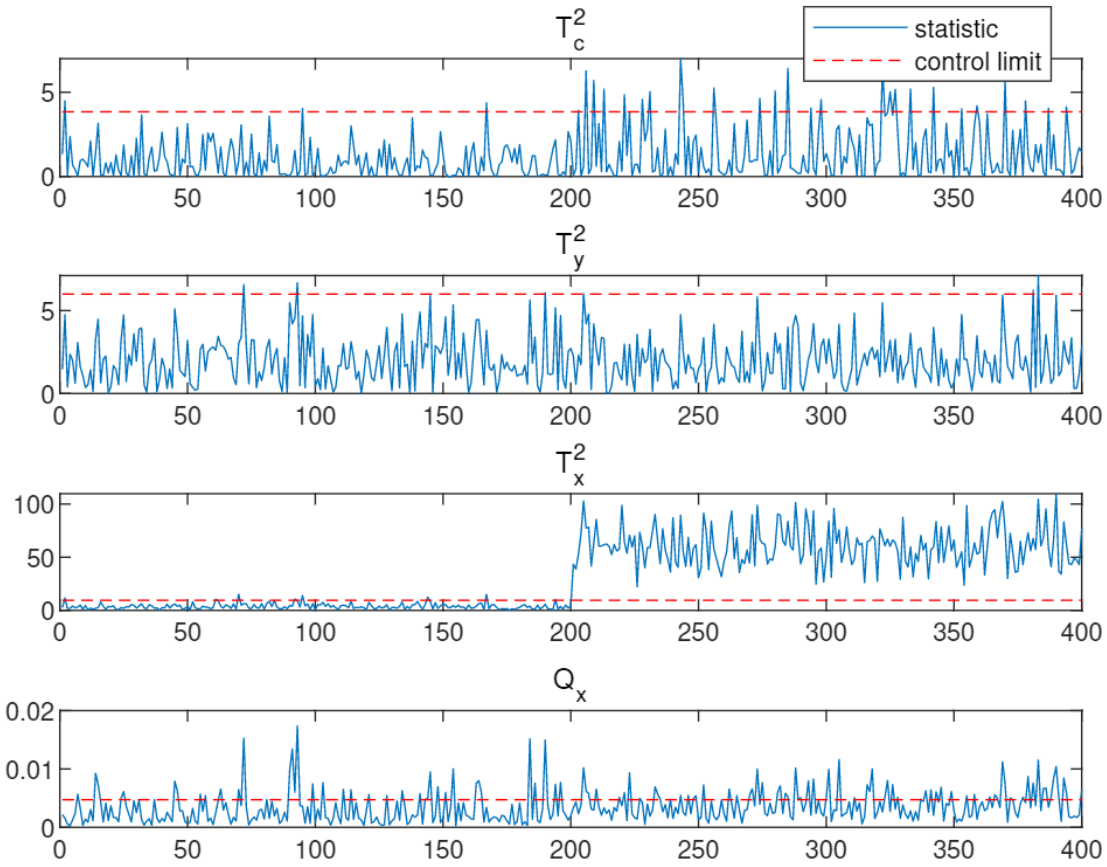

Figure 10: CKLVR-based monitoring results with fault in IPS ($f_x = 1$)

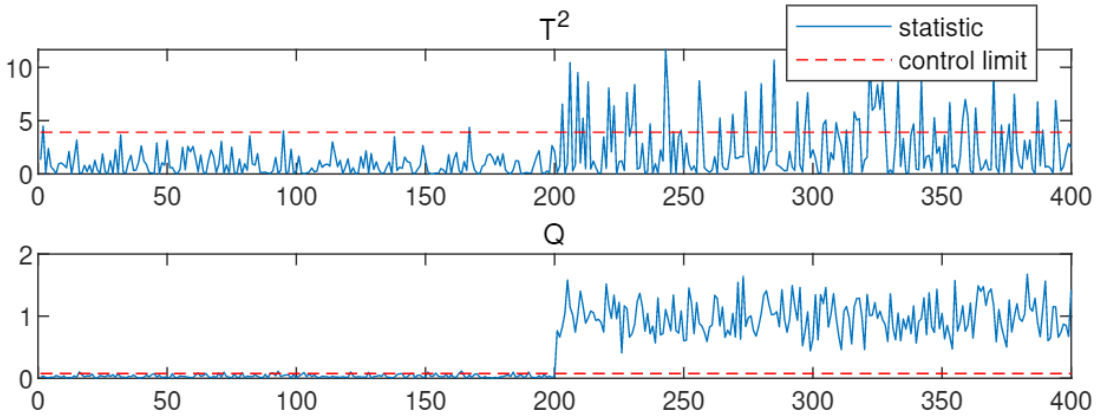

Figure 11: KLVR-based monitoring results with fault in IPS ($f_x = 1$)

3.3.4 Fault in Input-Residual Subspace

The input residual can be calculated as

$$\tilde{\phi}(\mathbf{x}) = (\mathbf{I} - \mathbf{P}_x\mathbf{P}_x^{\top})\left(\mathbf{I} - \mathbf{R}_c\mathbf{R}_c^{\dagger}\right)\phi(\mathbf{x}). \tag{3.27}$$

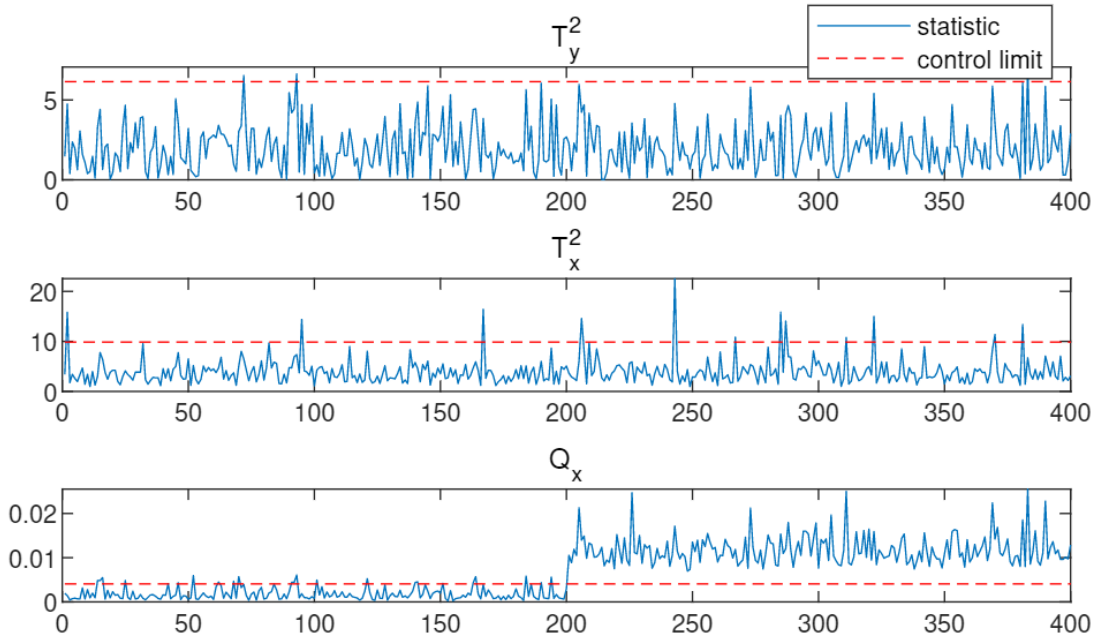

Figure 12: KPCA-based monitoring results with fault in IRS ($f_x = 0.1$)

Thus, to make sure that the fault only occurs in IRS, the basis vector $\boldsymbol{\Xi}_x$ of IRS is chosen as the left singular vectors of $(\mathbf{I} - \mathbf{P}_x\mathbf{P}_x^{\top})\left(\mathbf{I} - \mathbf{R}_c\mathbf{R}_c^{\dagger}\right)$ that is related to non-zero singular values, and $\boldsymbol{\Xi}_x$ needs to

be normalized to unit norm. From Figure 12, KPCA-based monitoring demonstrates that the potential L_2 fault alarm is only observable for the Q_x index.

The monitoring performance of CKLVR is demonstrated in Figure 13. It is noted that only Q_x successfully detects the fault, while other statistics remain normal, which aligns with Figure 12. KLVR shown in Figure 14 does not observe the fault with T^2 and Q. This only implies that the predictable quality variations remain normal, and without further decomposition, the status of other subspaces cannot be obtained. Thus, as shown in the discussions above, CKLVR realizes the comprehensive monitoring of all subspaces, which is also beneficial for further analysis.

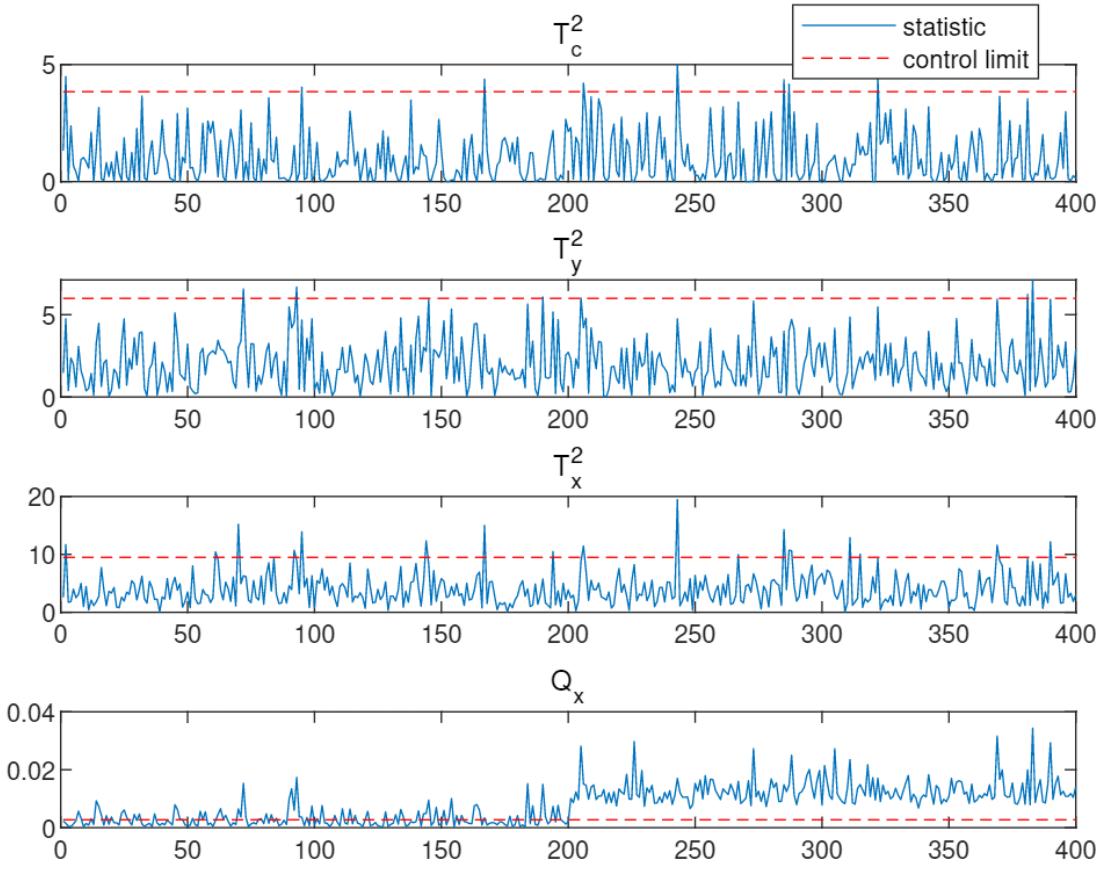

Figure 13: CKLVR-based monitoring results with fault in IRS ($f_x = 0.1$)

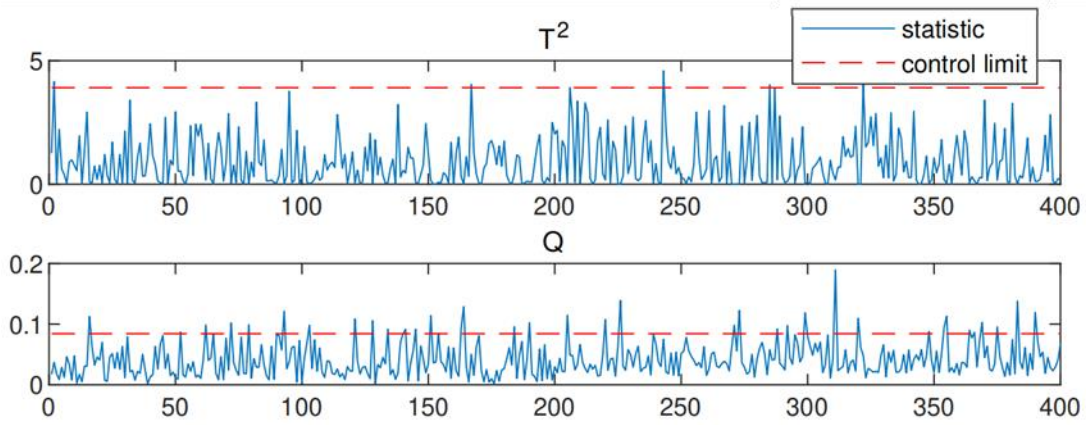

Figure 14: KLVR-based monitoring results with fault in IRS ($f_x = 0.1$)

3.4 Tennessee Eastman Process Case Study

In this section, the Tennessee Eastman Process (TEP) proposed by Downs and Vogel [118] is used for case study to further investigate the effectiveness of the CKLVR-based multilayer monitoring framework. TEP is a benchmark industrial process to evaluate different methods of process monitoring. The whole process contains five major unit operation plants, involving a chemical reactor, product condenser, recycle compressor, vapor-liquid separator and product stripper. Eight components (A-H) are included in TEP: four gaseous reactants (A, C, D, E), the inert material (B), and two liquid products (G,H) along with a by-product (F), which can be shown as follows:

$$A(g) + C(g) + D(g) \rightarrow G(l) \quad \text{(Product)}$$
$$A(g) + C(g) + E(g) \rightarrow H(l) \quad \text{(Product)}$$
$$A(g) + E(g) \rightarrow F(l) \quad \text{(By-product)}$$
$$3D(g) \rightarrow 2F(l) \quad \text{(By-product)}$$

Reactions involved in TEP are exothermic, irreversible and approximately 1-order with respect to concentrations of reactants. In addition, TEP is under closed-loop control. Readers can refer to Literature [60] and [62] for more details.

TEP is composed of two blocks of variables with a total number of 53, including 12 manipulated variables (XMV (1-12)) and 41 measured variables (XMEAS (1-41)). Here, XMEAS (1-22) and XMV (1-11) are chosen as process variables to construct **X**, while XMEAS (35-36) are selected as quality variables to form **Y**.

500 samples are simulated in TEP for training and 960 samples for testing. Process variables XMEAS (1-22) and XMV (1-11) are sampled at an interval of 3 minutes, while quality variables XMEAS (35-36) are sampled every 6 minutes, which gives rise to the irregular sampling intervals for process and quality variables. To balance the sampling discrepancy, data sampling is necessary, and only 1/2 samples without duplicate samples are employed [60]. Accordingly, only half of the data (250 normal samples and 480 faulty samples) are used for modeling and testing.

Table 3: FDRs and FARs for quality-relevant disturbances in TEP

Disturbance	FDR					FAR				
	KLVR	CKLVR				KLVR	CKLVR			
	T^2	T^2	T_c^2	T_y^2	T_x^2	T^2	T^2	T_c^2	T_y^2	T_x^2
IDV (1)	0.8750	**1**	0.7639	0.9028	**1**	0.1397	0.0956	0.0931	**0.0025**	0.0122
IDV (2)	0.9536	0.9969	0.8824	0.9907	**1**	0.1847	0.1083	0.1019	**0.0064**	0.0227
IDV (5)	0.7778	**1**	0.7556	0.9778	0.9429	0.0920	0.0598	0.0552	**0.0046**	0.0187
IDV (6)	0.0731	**1**	0.0705	**1**	0.8333	0.0825	0.0722	0.0722	**0**	**0**
IDV (7)	0.8769	**1**	0.8462	0.9231	**1**	0.1687	0.1446	0.1398	**0.0048**	0.0125
IDV (8)	0.8811	**1**	0.8238	0.9736	0.9974	0.4466	0.4229	0.4190	**0.0079**	0.0217
IDV (10)	0.7308	**1**	0.6538	0.9615	0.8721	0.1740	0.1410	0.1410	**0**	0.0309
IDV (12)	0.8371	**1**	0.7841	0.9773	0.9949	0.4213	0.3889	0.3843	0.0139	**0**
IDV (13)	0.8697	**1**	0.8275	0.9718	0.9894	0.3469	0.3163	0.3163	0.0051	**0**

Table 4: FARs for quality-irrelevant disturbances in TEP

Disturbance	KLVR	CKLVR			
	T^2	T^2	T_c^2	T_y^2	T_x^2
IDV (3)	0.0972	0.0346	0.0302	**0.0043**	0.0128
IDV (4)	0.0395	0.0148	0.0148	**0**	0.2578
IDV (9)	0.1017	0.0297	0.0275	**0.0021**	0.0411
IDV (11)	0.1199	0.0407	0.0321	**0.0086**	0.1446
IDV (14)	0.0169	0.0021	**0**	0.0021	0.0361
IDV (15)	0.1064	0.0489	0.0468	**0.0021**	0.0176

By reference to the criteria established by Zhou et al. [60], 15 disturbances (IDV (1-15)) can be classified into two categories, quality-relevant and quality-irrelevant disturbances. IDV (1-2, 5-8, 10, 12-13) are identified as quality-relevant disturbances, while IDV (3-4, 9, 11, 14-15) are quality-

irrelevant disturbances. For the purpose of better illustration, IDV (5, 11) are chosen as examples of two kinds of disturbances to show the monitoring performance of CKLVR in detail.

By 10-fold cross validation, the numbers of principal components for CKLVR are determined as $l = 1$, $l_c = 2$, $l_x = 14$, and $l_y = 2$, while for KLVR, $l = 1$. The width of Gaussian kernel function is $c = 5000$. For CKLVR-based monitoring, Q_y is null, since $l_y = 2$ while $p = 2$. The confidence level of control limit is selected to be 99 %.

False detection rates (FDRs) and false alarm rates (FARs) of quality-relevant and quality-irrelevant disturbances [60] are summarized in Tables 3 and 4, respectively. The FDR and FAR are defined on the premise of KPCA-based monitoring which is considered as ground truth and taken as the criterion to n classify normal and faulty samples, where FDR is the ratio of number of actual faults that are detectable for the proposed algorithm and number of total faulty samples, while FAR is the ratio of number of normal samples that are incorrectly categorized as faults and number of total normal samples.

To investigate quality-relevant monitoring performance, T^2 of CKLVR is defined as a combination of T_c^2 and T_y^2, which illustrates the monitoring results of predictable and unpredictable quality-relevant disturbances.

As shown in Table 3, for quality-relevant disturbances in TEP, CKLVR achieves higher quality-relevant monitoring FDRs than KLVR, since it monitors quality-relevant anomalies that are both predictable and unpredictable from process variable, while KLVR only has the process information that are related with quality variables. In terms of FARs of quality-relevant disturbances, except IDV (13), the Hotelling's T^2 statistics of CKLVR and subspaces are lower than T^2 of KLVR, which indicates improved monitoring performance of CKLVR. For IDV (13), the difference of T^2 between KLVR and CKLVR is in a reasonable range, which will not affect the advantage of CKLVR over KLVR.

For quality-irrelevant disturbances, FARs are of the top priority to focus on, since the total number of faulty samples in FDR is close to zero and it is meaningless to calculate FDRs. As shown in Table 4, CKLVR gains better monitoring results than KLVR due to its smaller FARs for all cases. Furthermore, process-relevant FDRs and FARs are also available in CKLVR with T_x^2, which yet attracts lower attention than T_c^2 and T_y^2. Consequently, it is concluded that CKLVR is superior to KLVR because of its comprehensive monitoring ability and fault detection performance.

To further validate the effectiveness of CKLVR over KLVR, two disturbances, IDV (5) and IDV (11), are applied for visual comparison.

3.4.1 Monitoring of IDV (5)

IDV (5) is caused by a step change of condenser cooling water inlet temperature [118]. As shown in Figure 15, KPCA-based monitoring on the process and quality variables illustrates that IDV (5) has influence on both process and quality variables transiently. More concretely, quality variations are abnormal in the interval [83rd, 130th], while in terms of process variables, faulty samples are detectable for T_x^2 from 81st to 168th and the anomalies detected by Q_x are from 81st to 182nd.

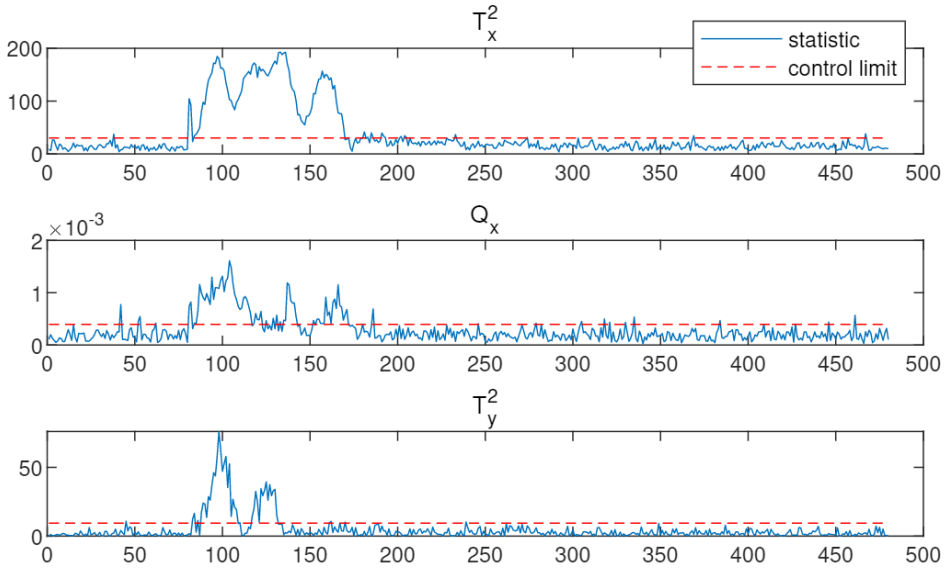

Figure 15: KPCA-based process and quality monitoring results for IDV (5)

The monitoring results of CKLVR and KLVR are demonstrated in Figures 16 and 17, respectively. All the monitoring indices of CKLVR in Figure 16 successfully observe the disturbance, and later return to normal, which is consistent with the process and quality monitoring results shown in Figure 15. For L_1-level alarming with respect to T_c^2 and T_y^2, it is noted that the disturbance of IDV (5) on quality variables is predictable from process variables, and thus most quality deviations are captured in CVS subspace with T_c^2 around the interval of faulty samples from 88th to 132nd. At the same time, T_y^2 observes additional quality-relevant but process-irrelevant anomalies in the interval of faulty samples [83rd, 130th], which is in accordance with T_y^2 of KPCA-monitoring. Furthermore, CKLVR also detects

anomalies by T_x^2 around the sample range from 81st to 171st, while Q_x around the sample range from 81st to 172nd, and the disturbance has longer influence on process-relevant indices T_x^2 and Q_x than quality-relevant indices T_c^2 and T_y^2.

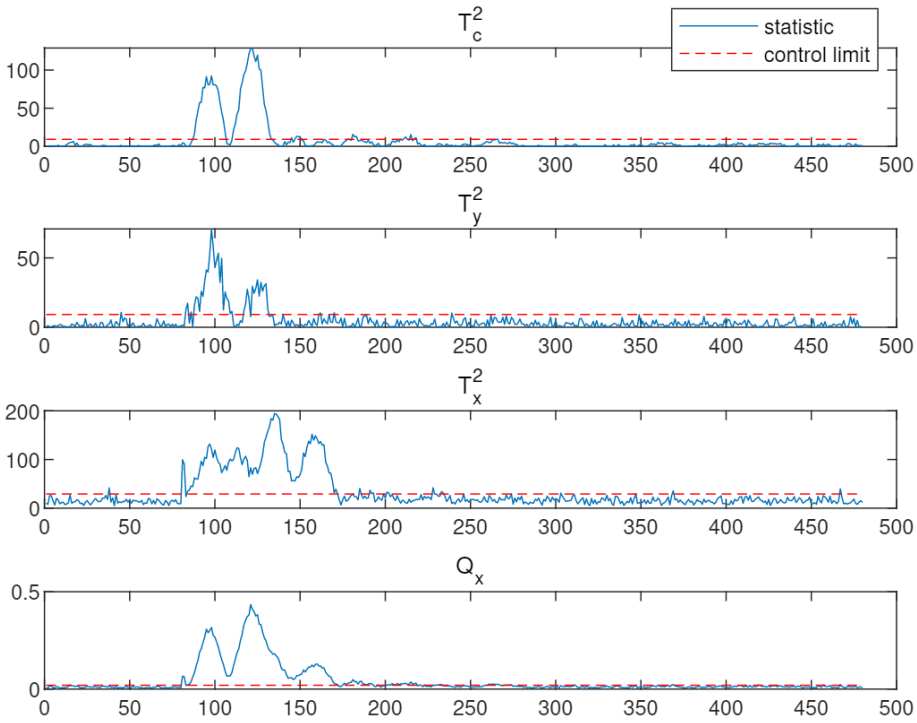

Figure 16: CKLVR-based monitoring results for IDV (5)

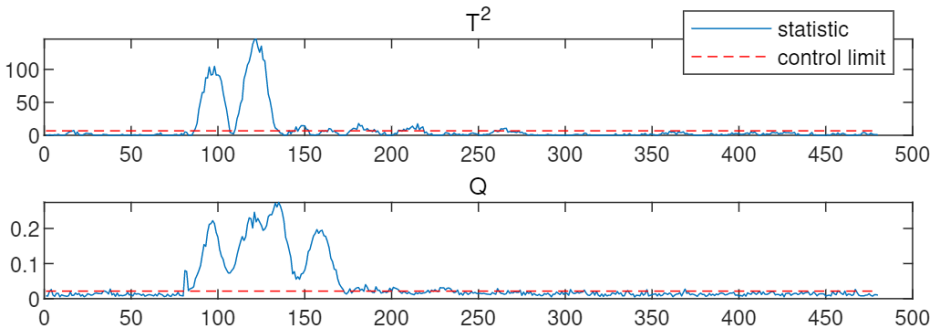

Figure 17: KLVR-based monitoring results for IDV (5)

By contrast, the monitoring result with KLVR in Figure 17 presents that the disturbance can be detected by T_c^2 from 87th to 134th and Q from 81st to 172nd, which indicates that KLVR-based monitoring only reflects the transient effect of the disturbance on the principal and residual subspaces,

and it has no knowledge about the process and remaining quality variations, leading to less informative presentation.

3.4.2 Monitoring of IDV (11)

IDV (11) results from random variations of reactor cooling water inlet temperature [118]. From KPCA-based monitoring in Figure 18, IDV (11) only affects the T_x^2 and Q_x statistics for process variables from 83rd, while T_y^2 remains unaffected, demonstrating that the disturbance is process-relevant but quality-irrelevant.

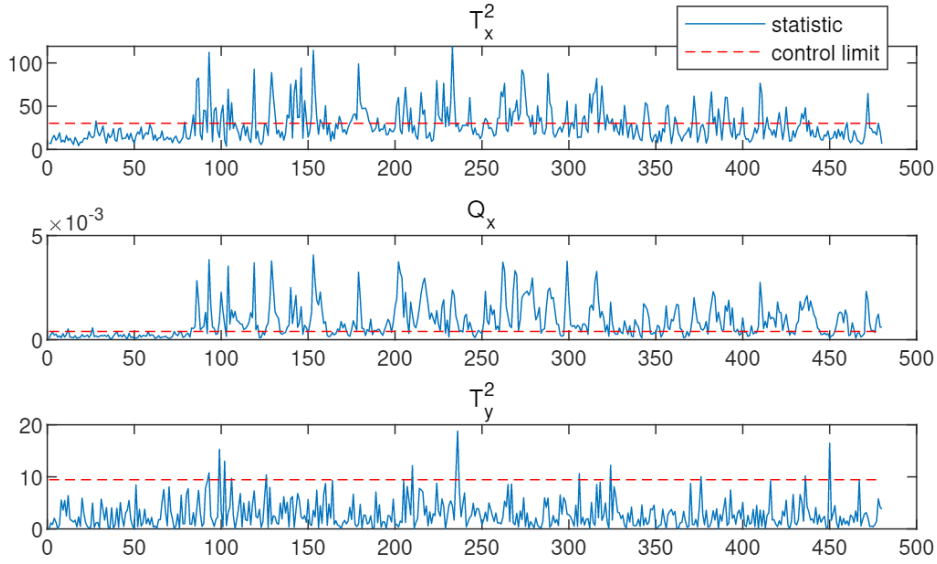

Figure 18: KPCA-based process and quality monitoring results for IDV (11)

Figures 19 and 20 illustrate CKLVR-based and KLVR-based monitoring results, respectively. The monitoring results of CKLVR is in line with Figure 18: only process-relevant indices T_x^2 and Q_x observe anomalies with the L_3-level alarming and L_2-level alarming from the 84th and 83rd faulty sample, respectively, and quality-relevant statistics T_c^2 and T_y^2 are within their corresponding control limits, indicating that the disturbance is process-relevant but it is unrelated to quality variables.

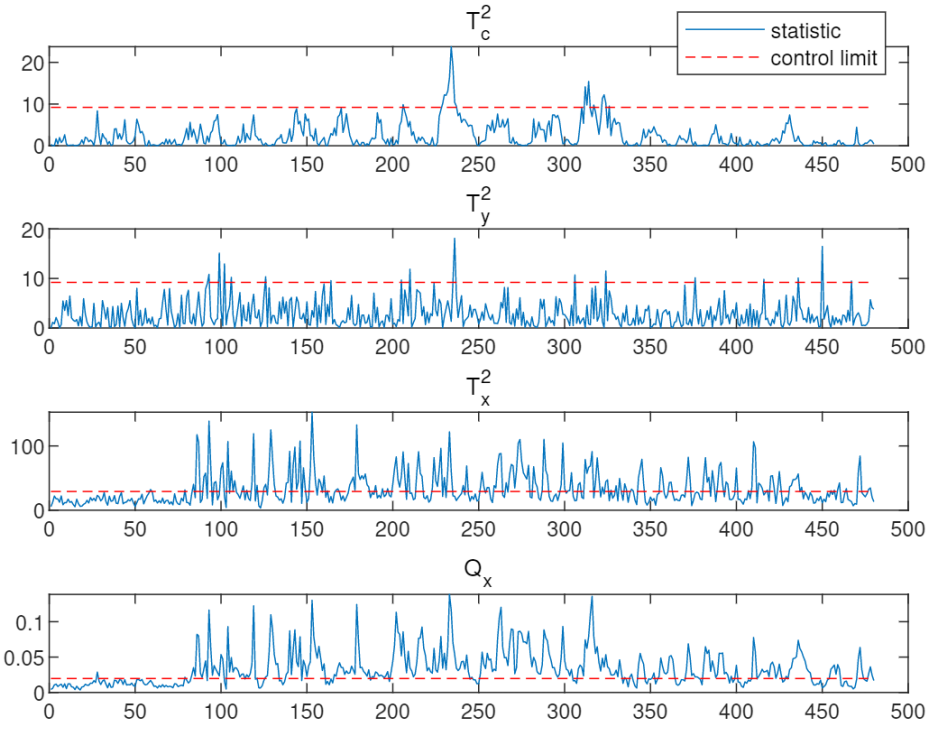

Figure 19: CKLVR-based monitoring results for IDV (11)

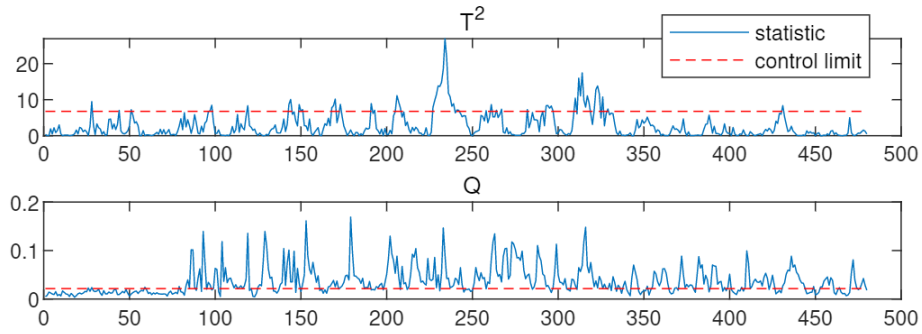

Figure 20: KLVR-based monitoring results for IDV (11)

KLVR-based monitoring is illustrated in Figure 20. The disturbance which affects the faulty samples is undetectable for T^2 statistics, since T^2 values of most samples are within control limits. Q observes faulty samples from the 83rd in the residual subspace caused by the disturbance, but any further information on the root causes is difficult to obtain.

3.5 Dow's Refining Process Case Study

The Dow's refining process was presented by Braun et al. [119] as a data science challenge problem, and the variables are anonymized due to privacy issues. The whole process consists of three distillation columns, including Primary Column, Feed Column and Secondary Column, and the Primary Column unit is controlled based on the reflux to feed ratio. 45 variables are collected from the process, including 44 process variables ($x1$-$x44$) and a quality variable (y). Among the process variables, Variables $x1$-$x21$ and $x41$-$x44$ are from the Primary Column, Variables $x22$-$x35$ are from the Secondary Column, and $x36$-$x40$ are from the Feed Column. The quality variable y is the impurity level sampled from the Primary Column. The sampling interval of the dataset is one per hour. More details of the process can be found in Literature [119].

In this case study, the dataset is first preprocessed to remove outliers and missing values. Then 3000 samples are selected as training data, and 1753 samples are chosen as test data, among which the first 1153 samples are normal data, while the latter 600 are faulty samples. The confidence level of control limit is selected as 99 %. KLVR, KPLS, rLVR and PLS models are developed with the training data, and their parameters are determined by cross-validation, which are listed in Table 5. Mean squared errors (MSEs) [119] of KLVR, KPLS, rLVR and PLS for normal test samples are also summarized in Table 5. For KLVR and rLVR, $\kappa = 0.005$. As shown in the table, KLVR and KPLS obtain lower MSEs than rLVR and PLS, implying the importance to employ kernel techniques for nonlinear processes. Further, KLVR has the lowest MSE, which indicates the superiority of KLVR in terms of prediction performance over KPLS.

Table 5: Parameters and MSEs of KLVR, KPLS, rLVR and PLS in Dow's refining process

Algorithm	KLVR	KPLS	rLVR	PLS
Principal component number l	4	5	2	1
Kernel width c	900	2000	-	-
MSEs	0.5647	0.6496	1.1427	4.2561

Table 6 summarizes the FDRs and FARs of CKLVR, CKPLS, KLVR, KPLS, rLVR and PLS for faulty test samples. For both CKLVR and CKPLS, $l_c = 1$, $l_x = 5$, and $l_y = 1$. As shown in Table 6, the concurrent decomposition divides the input and output data into different subspaces including CVS, OPS, IPS and IRS, and of T_c^2, T_y^2, T_x^2, and Q_x indices present their monitoring performance, respectively. It is noted that the ORS subspace does not exist since the output only has one variable in

this case study. Due to the subsequent decomposition, concurrent kernel algorithms retain more comprehensive monitoring results than others. In addition, as shown in Table 6, CKLVR performs better than CKPLS with lower FARs, while its FDRs are comparable to those of CKPLS.

Table 6: FDRs and FARs of algorithms for faulty case in Dow's refining process

Algorithm	FDR					FAR				
	T^2	T_c^2	T_y^2	T_x^2	Q_x	T^2	T_c^2	T_y^2	T_x^2	Q_x
CKLVR	**0.9921**	0.9868	0.9553	0.9868	0	0.1545	**0**	0.1545	**0**	**0**
CKPLS	0.9921	0.9842	0.9263	**0.9973**	0.0297	0.2773	**0**	0.2773	**0**	0.0076
KLVR	0.9921	-	-	-	-	0	-	-	-	-
KPLS	0.9816	-	-	-	-	0.0182	-	-	-	-
rLVR	0.9829	-	-	-	-	0	-	-	-	-
PLS	0.9816	-	-	-	-	0	-	-	-	-

For better visualization, Figure 21 illustrates the KPCA-based monitoring result which shows the real situation of this case. Figures 22-27 present the monitoring results of CKLVR, CKPLS, KLVR, KPLS, rLVR and PLS, respectively. As shown in the monitoring results of KLVR, KPLS, rLVR and PLS, faults are detected with both T^2 and Q indices; however, no further detailed information can be drawn from Figures 22-27. Instead, CKLVR in Figure 22 detects the L_1 alarm fault with T_c^2 index indicating that Samples [113rd, 122nd] and [190th, 554th] are quality-relevant faulty samples that can be predicted from process variables, which aligns with the quality monitoring results on quality data as shown in Figure 21. Further, monitoring results with T_y^2 of CKLVR indicate that faulty samples in the range of [183rd, 546th] also affect the quality variables in the OPS subspace in the way of L_1 alarm fault. Process variations that are not related to quality in IPS subspace also deviate from their normal conditions, while the process residual subspace is unaffected. It is noted that the monitoring results of CKPLS are similar to those of CKLVR; however, due to the incomplete exploitation of KPLS, anomalies are still detected in the process residual subspace. Therefore, the superiority of CKLVR-based multilayer monitoring scheme over other algorithms is observed.

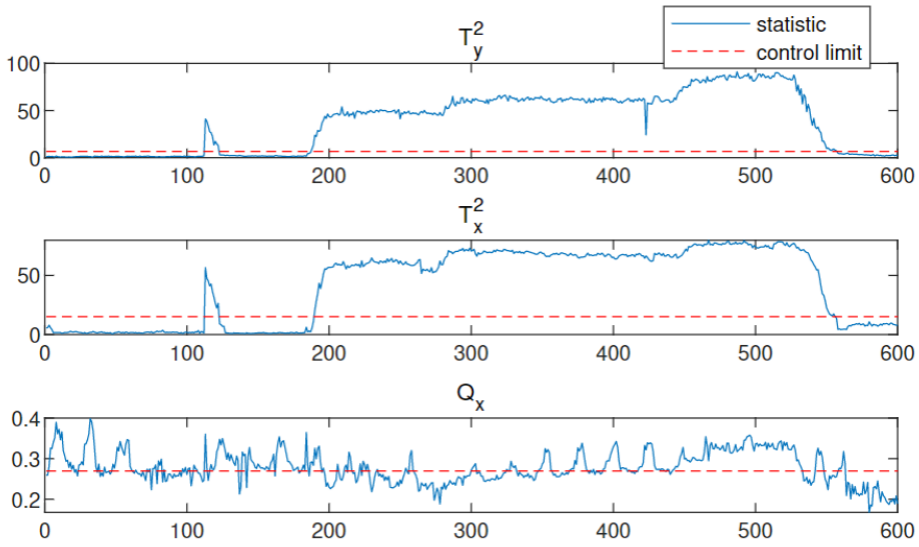

Figure 21: KPCA-based process and quality monitoring results in Dow's refining process

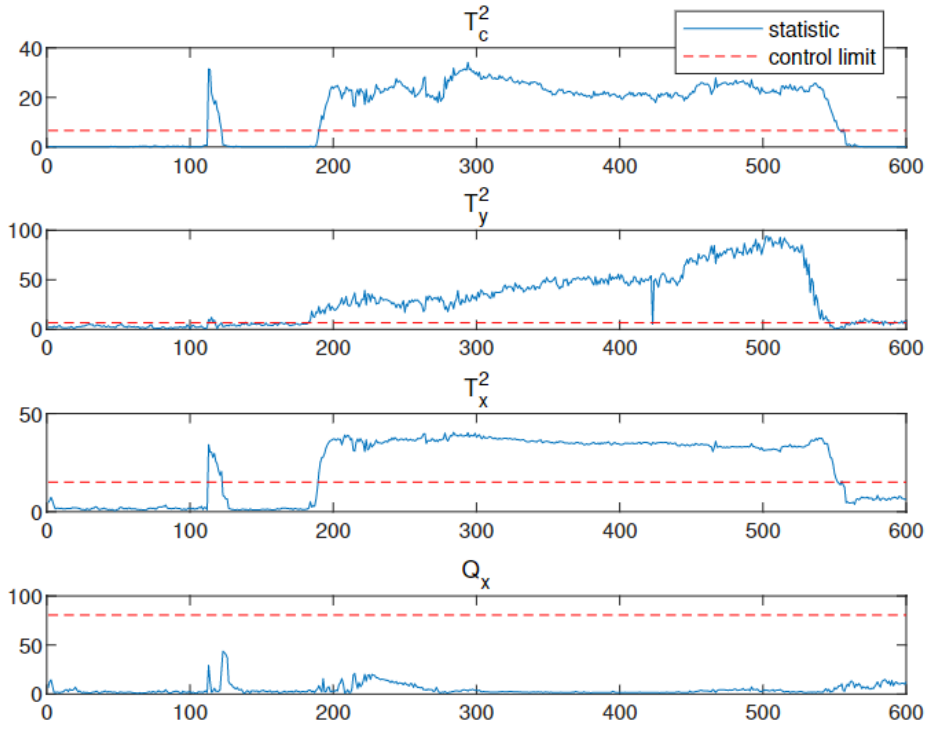

Figure 22: CKLVR-based monitoring results in Dow's refining process

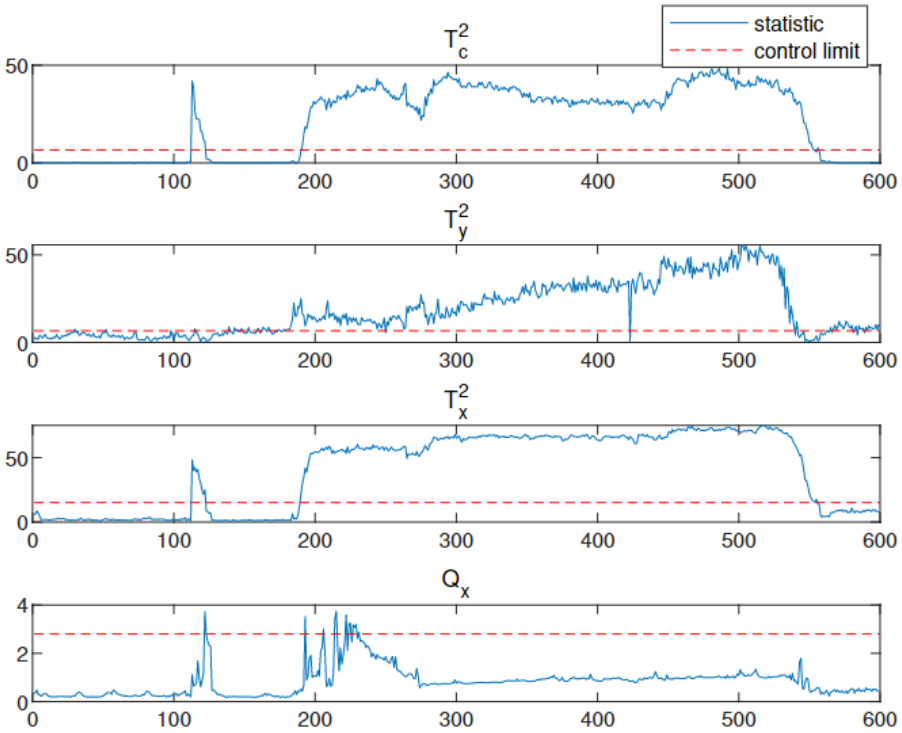

Figure 23: CKPLS-based monitoring results in Dow's refining process

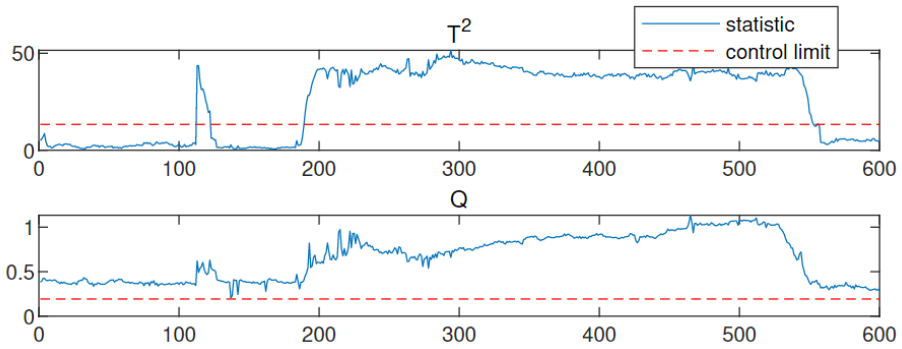

Figure 24: KLVR-based monitoring results in Dow's refining process

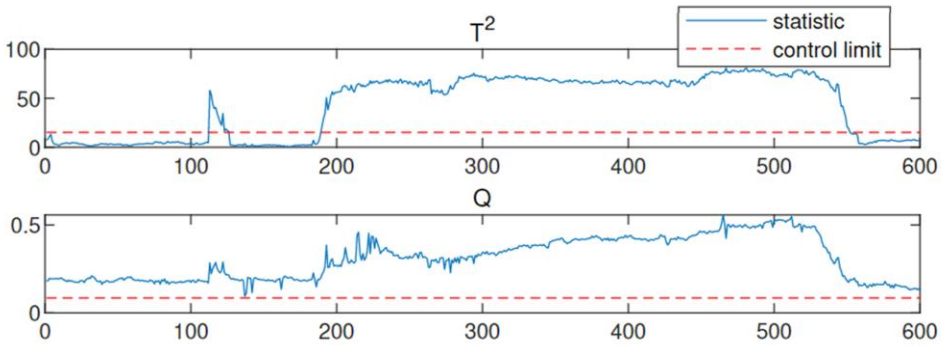

Figure 25: KPLS-based monitoring results in Dow's refining process

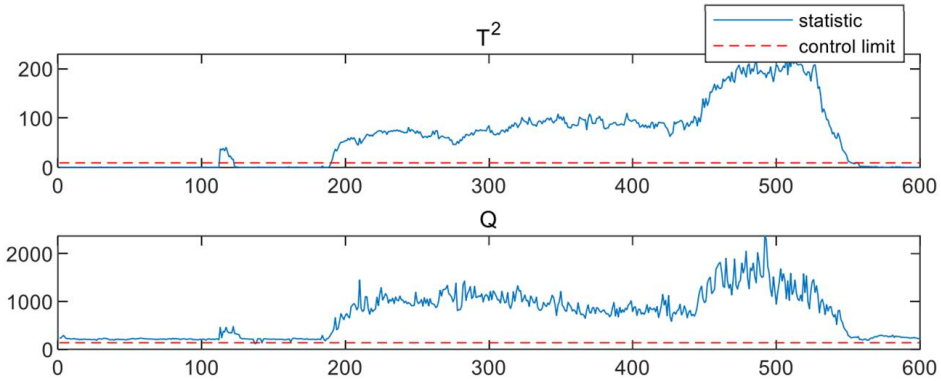

Figure 26: LVR-based monitoring results in Dow's refining process

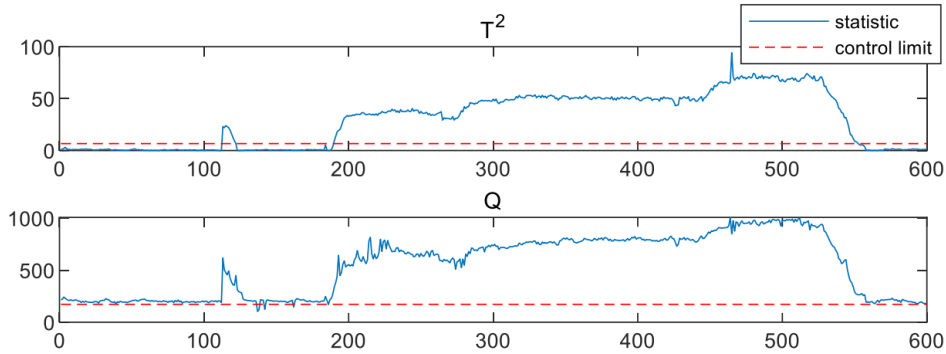

Figure 27: PLS-based monitoring results in Dow's refining process

3.6 Summary

In this chapter, CKLVR is proposed by incorporating the concurrent modeling of process and quality variables with KLVR algorithm, which improves the nonlinear modeling performance and enhances

the comprehensiveness of data exploitation. After performing CKLVR, the process and quality data are concurrently projected into five subspaces, and their corresponding indices designed for comprehensive process monitoring.

Different fault scenarios in different subspaces are simulated in the numerical simulation case study to evaluate the effectiveness of the proposed CKLVR method. The case studies on Tennessee Eastman Process and Dow's refining process demonstrate that CKLVR achieves a more comprehensive and accurate modeling and monitoring performance than other algorithms such as CKPLS and KPLS.

Chapter 4
Dynamic Latent Variable Modeling for Temporal Modeling and Monitoring

Due to the advancement of industrial technologies, another typical characteristic involved in the process and quality data is the complicated dynamics. The dynamic MSA methods reviewed in Chapter 1 focus only on capturing the dynamic cross-correlations between process and quality variables, leaving the dynamics in remaining spaces unexploited. However, both dynamic cross-correlations and auto-correlations are important for accurate dynamic quality modeling. Thus, inspired by the work of Xu and Zhu [20], a dynamic auto-regressive model is developed for LVR, namely dynamic auto-regressive LVR (DALVR), which realizes a comprehensive exploitation of dynamics in quality data.

4.1 Dynamic Auto-Regressive Latent Variable Regression

Different from the dynamic algorithms reviewed in Chapter 2, DALVR considers both the dynamic cross-correlations between process variations and quality variations, and the dynamic relations among quality variables. With the same denotations as in DrLVR, the score vectors t_k and u_k for $(\mathbf{x}_k, \mathbf{y}_k)$ are

$$\begin{aligned} t_k &= \mathbf{x}_k^\top \mathbf{w} \\ u_k &= \mathbf{y}_k^\top \mathbf{q}. \end{aligned} \tag{4.1}$$

DALVR assumes that the current output scores u_k are dependent on the past output scores and input scores, which is shown as follows:

$$u_k = \beta_0 t_k + \beta_1 t_{k-1} + \cdots + \beta_s t_{k-s} + \delta_1 u_{k-1} + \cdots + \delta_d u_{k-d} + \varepsilon_k \tag{4.2}$$

where s and d represent the dynamic orders for input and output variables, respectively. β_i $(i = 0,1,\ldots,s)$ and δ_i $(i = 0,1,\ldots,d)$ are weighting parameters for input and output scores. ε_k is the regression error. The prediction of u_k can be presented as

$$\hat{u}_k = \sum_{j=1}^{s} \beta_j \, t_{k-j} + \sum_{j=1}^{d} \delta_j \, u_{k-j} = \left[\mathbf{x}_k^\top, \mathbf{x}_{k-1}^\top, \ldots, \mathbf{x}_{k-s}^\top\right](\boldsymbol{\beta} \otimes \mathbf{w}) + \left[\mathbf{y}_{k-1}^\top, \ldots, \mathbf{y}_{k-d}^\top\right](\boldsymbol{\delta} \otimes \mathbf{q}) \tag{4.3}$$

where $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_s]^\top$ is the weighting coefficient vector for $\mathbf{w}$ and $\boldsymbol{\delta} = [\delta_1, \delta_2, \ldots, \delta_d]^\top$ is the weighting coefficient vector for $\mathbf{q}$. Eq. (4.3) can be transformed into a matrix form for the whole training data $\mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{g+N+1}\right]^\top$ and $\mathbf{Y} = \left[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{g+N+1}\right]^\top$, which is

$$\hat{\mathbf{u}}_g = \mathbf{Z}_x(\boldsymbol{\beta} \otimes \mathbf{w}) + \mathbf{Z}_y(\boldsymbol{\delta} \otimes \mathbf{q}) \tag{4.4}$$

where $g = \max(s, d)$, and

$$\begin{aligned} \mathbf{Z}_x &= \left[\mathbf{X}_g, \mathbf{X}_{g-1}, \dots, \mathbf{X}_{g-s}\right] \in \mathbb{R}^{N\times(s+1)m} \\ \mathbf{Z}_y &= \left[\mathbf{Y}_{g-1}, \mathbf{Y}_{g-2}, \dots, \mathbf{Y}_{g-d}\right] \in \mathbb{R}^{N\times dp} \\ \mathbf{X}_i &= [\mathbf{x}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_{i+N}]^\top \in \mathbb{R}^{N\times m} \\ \mathbf{Y}_i &= [\mathbf{y}_i, \mathbf{y}_{i+1}, \dots, \mathbf{y}_{i+N}]^\top \in \mathbb{R}^{N\times p} \end{aligned} \tag{4.5}$$

where $i = 0,1,2, \dots, g$.

It is also noted that the score vectors $\mathbf{t}_i$ and $\mathbf{u}_i$ can be calculated for each $\mathbf{X}_i$ and $\mathbf{Y}_i$ as follows:

$$\begin{aligned} \mathbf{t}_i &= \mathbf{X}_i\mathbf{w} \\ \mathbf{u}_i &= \mathbf{Y}_i\mathbf{q}. \end{aligned} \tag{4.6}$$

DALVR is designed to maximize the covariance between $\mathbf{u}_g$ and $\hat{\mathbf{u}}_g$, and its objective is

$$\begin{gathered} \max_{\mathbf{w},\mathbf{q},\boldsymbol{\beta},\boldsymbol{\delta}} \ \mathbf{q}^\top\mathbf{Y}_g^\top\left[\mathbf{Z}_x(\boldsymbol{\beta} \otimes \mathbf{w}) + \mathbf{Z}_y(\boldsymbol{\delta} \otimes \mathbf{q})\right] - \frac{\gamma_w}{2}\|\mathbf{w}\|^2 - \frac{\gamma_\beta}{2}\|\boldsymbol{\beta}\|^2 - \frac{\gamma_\delta}{2}\|\boldsymbol{\delta}\|^2 \\ s.t. \ \|\boldsymbol{\beta} \otimes \mathbf{w}\| = 1, \|\boldsymbol{\delta} \otimes \mathbf{q}\| = 1, \|\mathbf{q}\| = 1 \end{gathered} \tag{4.7}$$

where γ_w, γ_β and γ_δ are regularized parameters to handle collinearity issues. For ease of presentation, define $\mathbf{X}_\beta$, $\mathbf{T}_s$, $\mathbf{Y}_\delta$, and $\mathbf{U}_d$ as

$$\begin{aligned} \mathbf{X}_\beta &= \mathbf{Z}_x(\boldsymbol{\beta} \otimes \mathbf{I}_m) = \sum_{i=0}^{s} \beta_i \mathbf{X}_{g-i} \\ \mathbf{T}_s &= \mathbf{Z}_x(\mathbf{I}_{s+1} \otimes \mathbf{w}) = \left[\mathbf{t}_g, \mathbf{t}_{g-1}, \dots, \mathbf{t}_{g-s}\right] \\ \mathbf{Y}_\delta &= \mathbf{Z}_y\left(\boldsymbol{\delta} \otimes \mathbf{I}_p\right) = \sum_{i=1}^{d} \delta_i \mathbf{Y}_{g-i} \\ \mathbf{U}_d &= \mathbf{Z}_y(\mathbf{I}_d \otimes \mathbf{q}) = \left[\mathbf{u}_{g-1}, \mathbf{u}_{g-2}, \dots, \mathbf{u}_{g-d}\right]. \end{aligned} \tag{4.8}$$

Then the objective in Eq. (4.7) can be rewritten as

$$\begin{gathered} \max_{\mathbf{w},\mathbf{q},\boldsymbol{\beta},\boldsymbol{\delta}} \ \mathbf{q}^\top\mathbf{Y}_g^\top(\mathbf{T}_s\boldsymbol{\beta} + \mathbf{U}_d\boldsymbol{\delta}) - \frac{\gamma_w}{2}\|\mathbf{w}\|^2 - \frac{\gamma_\beta}{2}\|\boldsymbol{\beta}\|^2 - \frac{\gamma_\delta}{2}\|\boldsymbol{\delta}\|^2 \\ s.t. \ \|\mathbf{T}_s\boldsymbol{\beta}\| = 1, \|\mathbf{U}_d\boldsymbol{\delta}\| = 1, \|\mathbf{q}\| = 1 \end{gathered} \tag{4.9}$$

which defines the outer structure of DALVR.

The Lagrange multipliers $\lambda_{q\delta}$, $\lambda_{w\beta}$, and λ_q are employed to solve the objective in Eq. (4.7):

$$\mathcal{L} = \mathbf{q}^{\top}\mathbf{Y}_g^{\top}(\mathbf{T}_s\boldsymbol{\beta} + \mathbf{U}_d\boldsymbol{\delta}) - \frac{\gamma_w}{2}\|\mathbf{w}\|^2 - \frac{\gamma_\beta}{2}\|\boldsymbol{\beta}\|^2 - \frac{\gamma_\delta}{2}\|\boldsymbol{\delta}\|^2 + \frac{\lambda_{w\beta}}{2}(1 - \boldsymbol{\beta}^{\top}\mathbf{T}_s^{\top}\mathbf{T}_s\boldsymbol{\beta}) + \frac{\lambda_{q\delta}}{2}\left(1 - \boldsymbol{\delta}^{\top}\mathbf{U}_d^{\top}\mathbf{U}_d\boldsymbol{\delta}\right) + \frac{\lambda_q}{2}(1 - \mathbf{q}^{\top}\mathbf{q}). \quad (4.10)$$

Take derivatives with regard to $\mathbf{w}$, $\mathbf{q}$, $\boldsymbol{\beta}$, and $\boldsymbol{\delta}$ and set these equations to zero, leading to

$$\left(\mathbf{X}_\beta^{\top}\mathbf{X}_\beta + \kappa_w\mathbf{I}\right)^{-1}\mathbf{X}_\beta^{\top}\mathbf{u}_g = \lambda_{w\beta}\mathbf{w} \quad (4.11)$$

$$\left(\mathbf{Y}_\delta^{\top}\mathbf{Y}_\delta + \kappa_q\mathbf{I}\right)^{-1}\left(\mathbf{Y}_g^{\top}\mathbf{T}_s\boldsymbol{\beta} + \mathbf{Y}_g^{\top}\mathbf{U}_d\boldsymbol{\delta} + \mathbf{Y}_\delta^{\top}\mathbf{u}_g\right) = \lambda_{q\delta}\mathbf{q} \quad (4.12)$$

$$\left(\mathbf{T}_s^{\top}\mathbf{T}_s + \kappa_\beta\mathbf{I}\right)^{-1}\mathbf{T}_s^{\top}\mathbf{u}_g = \lambda_{w\beta}\boldsymbol{\beta} \quad (4.13)$$

$$\left(\mathbf{U}_d^{\top}\mathbf{U}_d + \kappa_\delta\mathbf{I}\right)^{-1}\mathbf{U}_d^{\top}\mathbf{u}_g = \lambda_{q\delta}\boldsymbol{\delta} \quad (4.14)$$

where $\kappa_w = \gamma_w/\lambda_{w\beta}$, $\kappa_\beta = \gamma_\beta/\lambda_{w\beta}$, $\kappa_\delta = \gamma_\delta/\lambda_{q\delta}$, and $\kappa_q = \lambda_{q\delta}/\lambda_q$. κ_w, κ_β, and κ_δ can be processed by the iteration calculation of Eqs. (4.11) - (4.14).

As shown in Eqs. (4.11) - (4.14), the compact form of $\mathbf{w}$, $\mathbf{q}$, $\boldsymbol{\beta}$, and $\boldsymbol{\delta}$ cannot be obtained. Thus, the model parameters are calculated iteratively as follows.

1. Scale $\mathbf{X}$ and $\mathbf{Y}$ to zero mean and unit variance.

2. Initialize $\mathbf{u}$ as some column of $\mathbf{Y}$, $\mathbf{u}_g$ as some column of $\mathbf{Y}_g$, $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ as unit random vectors.

3. Repeat the following relations iteratively until convergence.

(1) Construct $\mathbf{X}_\beta = \sum_{i=0}^{s}\beta_i\mathbf{X}_{g-i}$, $\mathbf{Y}_\delta = \sum_{i=1}^{d}\delta_i\mathbf{Y}_{g-i}$, and $\mathbf{U}_d = \left[\mathbf{u}_{g-1}, \mathbf{u}_{g-2}, \ldots, \mathbf{u}_{g-d}\right]$;

(2) $\mathbf{w} = \left(\mathbf{X}_\beta^{\top}\mathbf{X}_\beta + \kappa_w\mathbf{I}\right)^{-1}\mathbf{X}_\beta^{\top}\mathbf{u}_g$;

(3) $\mathbf{t} = \mathbf{X}\mathbf{w}$, and form $\mathbf{T}_s = \left[\mathbf{t}_g, \mathbf{t}_{g-1}, \ldots, \mathbf{t}_{g-s}\right]$;

(4) $\mathbf{q} = \left(\mathbf{Y}_\delta^{\top}\mathbf{Y}_\delta + \kappa_q\mathbf{I}\right)^{-1}\left(\mathbf{Y}_g^{\top}\mathbf{T}_s\boldsymbol{\beta} + \mathbf{Y}_g^{\top}\mathbf{U}_d\boldsymbol{\delta} + \mathbf{Y}_\delta^{\top}\mathbf{u}_g\right)$, and $\mathbf{q} = \mathbf{q}/\|\mathbf{q}\|$;

(5) $\mathbf{u}_g = \mathbf{Y}_g\mathbf{q}$;

(6) $\boldsymbol{\beta} = \left(\mathbf{T}_s^{\top}\mathbf{T}_s + \kappa_\beta\mathbf{I}\right)^{-1}\mathbf{T}_s^{\top}\mathbf{u}_g$, and $\boldsymbol{\beta} = \boldsymbol{\beta}/\|\mathbf{T}_s\boldsymbol{\beta}\|$;

(7) $\boldsymbol{\delta} = \left(\mathbf{U}_d^{\top}\mathbf{U}_d + \kappa_\delta\mathbf{I}\right)^{-1}\mathbf{U}_d^{\top}\mathbf{u}_g$, and $\boldsymbol{\delta} = \boldsymbol{\delta}/\|\mathbf{U}_d\boldsymbol{\delta}\|$.

To align with the modeling of dynamic outer structure, a dynamic inner structure is built based on the latent variable modeling structure to describe the relations among $\mathbf{u}_g$, $\mathbf{T}_s$ and $\mathbf{U}_d$:

$$\mathbf{u}_g = \mathbf{T}_s\boldsymbol{\zeta} + \mathbf{U}_d\boldsymbol{\psi} + \boldsymbol{\varepsilon}_g \tag{4.15}$$

where $\boldsymbol{\zeta} = [\zeta_0, \zeta_1, \ldots, \zeta_s]^\top$ and $\boldsymbol{\psi} = [\psi_1, \psi_2, \ldots, \psi_d]^\top$ are weighting vectors of input and output matrices respectively, and $\boldsymbol{\varepsilon}_g$ is the regression error. With the obtained $\boldsymbol{\zeta}$ and $\boldsymbol{\psi}$, $\mathbf{u}_g$ can be predicted by

$$\hat{\mathbf{u}}_g = \mathbf{T}_s\boldsymbol{\zeta} + \mathbf{U}_d\boldsymbol{\psi}. \tag{4.16}$$

After the outer and inner models are obtained, deflation should be performed on $\mathbf{X}$ and $\mathbf{Y}$ to eliminate the effects of the extracted latent variables. The input matrix $\mathbf{X}$ can be deflated by

$$\mathbf{X} \coloneqq \mathbf{X} - \mathbf{t}\mathbf{p}^\top \tag{4.17}$$

where $\mathbf{p} = \mathbf{X}^\top\mathbf{t}/\mathbf{t}^\top\mathbf{t}$.

The deflation of $\mathbf{Y}$ needs to be partitioned into two parts: the static part $\mathbf{Y}_{0:g-1} \equiv \{\mathbf{y}_i\}_{i=0}^{g-1}$ and the dynamic part $\mathbf{Y}_g \equiv \{\mathbf{y}_i\}_{i=g}^{n}$. $\mathbf{Y}_{0:g-1}$ is deflated by

$$\mathbf{Y}_{0:g-1} \coloneqq \mathbf{Y}_{0:g-1} - \mathbf{t}\mathbf{c}_{st}^\top \tag{4.18}$$

where $\mathbf{c}_{st} = \mathbf{Y}_{0:g-1}^\top\mathbf{t}/\mathbf{t}^\top\mathbf{t}$; while $\mathbf{Y}_g$ is deflated by

$$\mathbf{Y}_g \coloneqq \mathbf{Y}_g - \hat{\mathbf{u}}_g\mathbf{c}^\top \tag{4.19}$$

where $\mathbf{c} = \mathbf{Y}_g^\top\hat{\mathbf{u}}_g/\hat{\mathbf{u}}_g^\top\hat{\mathbf{u}}_g$.

After modeling with DALVR, $\mathbf{X}$ and $\mathbf{Y}_g$ are decomposed as

$$\begin{cases} \mathbf{X} = \sum_{i=1}^{l}\mathbf{t}_i\mathbf{p}_i^\top + \tilde{\mathbf{X}} = \mathbf{T}\mathbf{P}^\top + \tilde{\mathbf{X}} \\ \mathbf{Y}_g = \sum_{i=1}^{l}\hat{\mathbf{u}}_{g,i}\mathbf{c}_i^\top + \tilde{\mathbf{Y}}_g = \hat{\mathbf{U}}_g\mathbf{C}^\top + \tilde{\mathbf{Y}}_g \end{cases} \tag{4.20}$$

where $\hat{\mathbf{U}}_g = \left[\hat{\mathbf{u}}_{g,1}, \hat{\mathbf{u}}_{g,2}, \ldots, \hat{\mathbf{u}}_{g,l}\right] \in \mathbb{R}^{N\times l}$, and $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}_g$ are residuals of $\mathbf{X}$ and $\mathbf{Y}_g$. $\mathbf{t}_i$, $\mathbf{p}_i$, $\hat{\mathbf{u}}_{g,i}$, and $\mathbf{c}_i$ correspond to the i^{th} latent variable ($i = 1, 2, \ldots, l$), respectively.

4.2 Concurrent Dynamic Decomposition

To realize a comprehensive modeling and monitoring of the process and quality spaces as for KLVR in Chapter 2, the modified concurrent dynamic modeling framework is also developed for DALVR, which is summarized as follows.

1. DALVR is performed on scaled $\mathbf{X}$ and $\mathbf{Y}$ for the modeling parameters $\mathbf{R}$, $\mathbf{Q}$, $\mathbf{C}$, $\boldsymbol{\zeta}$ and $\boldsymbol{\psi}$.

2. Divide the dynamic predictable output $\widehat{\mathbf{Y}}_g$ into the *dynamic* cross-correlation output part $\widehat{\mathbf{Y}}_{xg}$ and *dynamic* auto-correlation output part $\widehat{\mathbf{Y}}_{yg}$:

$$\widehat{\mathbf{Y}}_g = \widehat{\mathbf{Y}}_{xg} + \widehat{\mathbf{Y}}_{yg} = [\mathbf{Z}_x(\boldsymbol{\zeta} \otimes \mathbf{R})]\mathbf{C}^\top + \left[\mathbf{Z}_y(\boldsymbol{\psi} \otimes \mathbf{Q})\right]\mathbf{C}^\top \tag{4.21}$$

where $\widehat{\mathbf{Y}}_{xg}$ are the quality variations that can be predicted from input variables, and $\widehat{\mathbf{Y}}_{yg}$ are auto-regressive quality variations.

(1) SVD is performed on $\widehat{\mathbf{Y}}_{xg}$ to capture the *dynamic* cross-correlations that are observable from input variables:

$$\widehat{\mathbf{Y}}_{xg} = \mathbf{U}_c\mathbf{D}_c\mathbf{V}_c \equiv \mathbf{U}_c\mathbf{Q}_c^\top \tag{4.22}$$

where $\mathbf{U}_c$ represents the covariations in $\mathbf{X}$ that are relevant to $\widehat{\mathbf{Y}}_{xg}$; $\mathbf{Q}_c = \mathbf{V}_c\mathbf{D}_c$ contains all non-zero singular values in a descending order and the corresponding right singular vectors. Due to the orthogonality of $\mathbf{V}_c$, $\mathbf{U}_c$ can be rewritten as

$$\mathbf{U}_c = \widehat{\mathbf{Y}}_{xg}\mathbf{V}_c\mathbf{D}_c^{-1} \equiv \mathbf{X}_\zeta\mathbf{R}_c \tag{4.23}$$

where $\mathbf{X}_\zeta = \mathbf{Z}_x(\boldsymbol{\zeta} \otimes \mathbf{I}) = \sum_{i=0}^{s} \zeta_i\mathbf{X}_{s-i}$, and $\mathbf{R}_c = \mathbf{R}\mathbf{C}^\top\mathbf{V}_c\mathbf{D}_c^{-1}$.

(2) Define $\widehat{\mathbf{U}}_\psi = \mathbf{Z}_y(\boldsymbol{\psi} \otimes \mathbf{Q})$, and the auto-regressive part $\widehat{\mathbf{Y}}_{yg}$ can be written as

$$\widehat{\mathbf{Y}}_{yg} = \left[\mathbf{Z}_y(\boldsymbol{\psi} \otimes \mathbf{Q})\right]\mathbf{C}^\top \equiv \widehat{\mathbf{U}}_\psi\mathbf{C}^\top. \tag{4.24}$$

3. The quality residual $\mathbf{Y}_c = \mathbf{Y} - \widehat{\mathbf{Y}}_{xg} - \widehat{\mathbf{Y}}_{yg}$ only contain *static* variations, and thus perform static PCA on $\mathbf{Y}_c$ to extract *static* principal variations:

$$\mathbf{Y}_c = \mathbf{T}_{sy}\mathbf{P}_{sy}^\top + \widetilde{\mathbf{Y}} \tag{4.25}$$

where $\mathbf{T}_{sy}$ and $\mathbf{P}_{sy}$ are the score and loading matrices of static output principal variations, respectively. $\widetilde{\mathbf{Y}}$ is the static output residual.

4. Dynamic inner PCA (DiPCA) [120] is performed on $\mathbf{X}$ to capture both dynamic and static input variations simultaneously:

$$\mathbf{X} = \hat{\mathbf{T}}_{dx}\mathbf{P}_{dx}^{\top} + \mathbf{T}_{sx}\mathbf{P}_{sx}^{\top} + \tilde{\mathbf{X}}_{s} \tag{4.26}$$

where $\hat{\mathbf{T}}_{dx}$ is the score matrix of dynamic input variations and $\mathbf{P}_{dx}$ is the loading matrix of dynamic input variations, which are predictable for the past input scores. $\mathbf{T}_{sx}$ is the score matrix of static input variations and $\mathbf{P}_{sx}$ is the loading matrix of static input variations, and $\tilde{\mathbf{X}}_{s}$ represents the static input residuals.

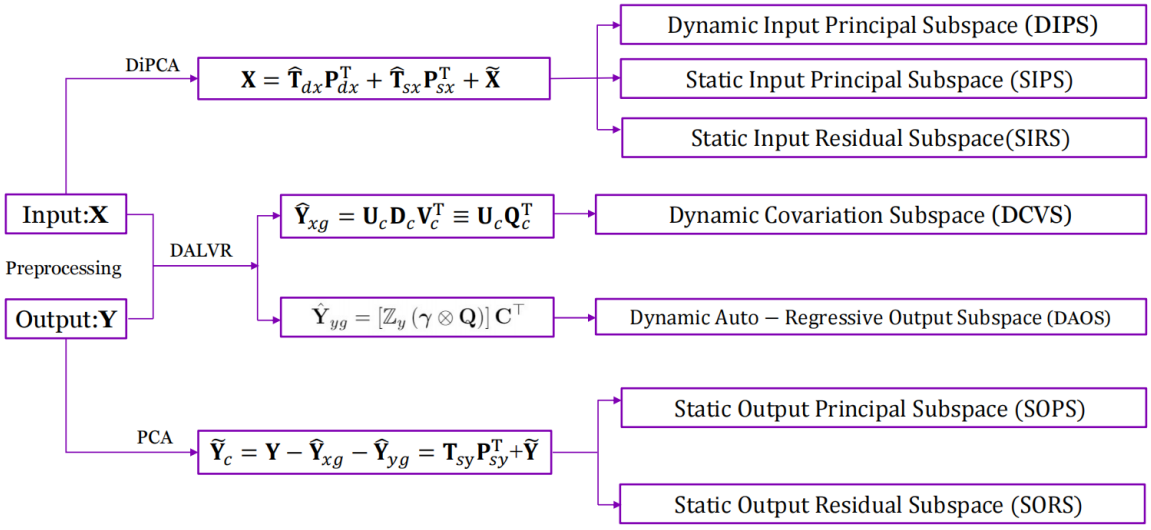

Figure 28: CDALVR-based monitoring framework

For better understanding, the concurrent DALVR (CDALVR) based modeling framework is summarized in Figure 28, where the subspaces names are also defined. With CDALVR, $\mathbf{X}$ and $\mathbf{Y}$ are decomposed as

$$\begin{cases} \mathbf{X} = \hat{\mathbf{T}}_{dx}\mathbf{P}_{dx}^{\top} + \mathbf{T}_{sx}\mathbf{P}_{sx}^{\top} + \tilde{\mathbf{X}}_{s} \\ \mathbf{Y}_{g} = \mathbf{U}_c\mathbf{Q}_c^{\top} + \hat{\mathbf{U}}_{\psi}\mathbf{C}^{\top} + \mathbf{T}_{sy}\mathbf{P}_{sy}^{\top} + \tilde{\mathbf{Y}}. \end{cases} \tag{4.27}$$

where $\mathbf{U}_c\mathbf{Q}_c^{\top}$ denotes predictable dynamic variations in process variables related to quality data, which is projected into the dynamic covariation subspace (DCVS). $\hat{\mathbf{U}}_{\psi}\mathbf{C}^{\top}$ represents auto-regressive relations in quality data, corresponding to the dynamic auto-regressive output subspace (DAOS). $\mathbf{T}_{sy}\mathbf{P}_{sy}^{\top}$ in the static output principal subspace (SOPS) contains major static quality-relevant variations, and $\tilde{\mathbf{Y}}$ shows the static output residual subspace (SORS). $\hat{\mathbf{T}}_{dx}\mathbf{P}_{dx}^{\top}$ in the dynamic input principal subspace (DIPS)

captures dynamic process-relevant but quality-irrelevant variations. $\mathbf{T}_{sx}\mathbf{P}_{sx}^{\top}$ includes static process-relevant but quality-irrelevant variations to form static input principal subspace (SIPS). The static input residual subspace (SIRS) is constructed by $\widetilde{\mathbf{X}}_s$.

After the DALVR model is obtained with training data, it can be employed for online modeling, and it decomposes a new test sample $(\mathbf{x}_k, \mathbf{y}_k)$ into

$$\begin{cases} \mathbf{x}_k = \mathbf{P}_{dx}\hat{\mathbf{t}}_{dx,k} + \mathbf{P}_{sx}\mathbf{t}_{sx,k} + \tilde{\mathbf{x}}_k \\ \mathbf{y}_k = \mathbf{Q}_c\mathbf{u}_{c,k} + \mathbf{C}\hat{\mathbf{u}}_{\psi,k} + \mathbf{P}_{sy}\mathbf{t}_{sy,k} + \tilde{\mathbf{y}}_k \end{cases} \tag{4.28}$$

where for $\mathbf{x}_k$, $\hat{\mathbf{t}}_{dx,k}$ is the score vector in DIPS, $\mathbf{t}_{sx,k}$ is the score vector in SIPS, and $\tilde{\mathbf{x}}_k$ is the residual vector in SIRS, which are obtained by

$$\hat{\mathbf{t}}_{dx,k} = \sum_{i=1}^{s} \mathbf{\Theta}_{x,i}\mathbf{t}_{dx,k-i} \tag{4.29}$$

$$\mathbf{t}_{sx,k} = \mathbf{P}_{sx}\tilde{\mathbf{x}}_c \tag{4.30}$$

$$\tilde{\mathbf{x}}_k = (\mathbf{I} - \mathbf{P}_{sx}\mathbf{P}_{sx}^{\top})\tilde{\mathbf{x}}_c \tag{4.31}$$

where $\mathbf{t}_{dx,k} = \mathbf{R}_x\mathbf{x}_k$, and $\tilde{\mathbf{x}}_c = \mathbf{x}_k - \mathbf{P}_{dx}\hat{\mathbf{t}}_{dx,k}$. $\mathbf{P}_{dx}$, $\mathbf{P}_{sx}$, $\mathbf{\Theta}_x$, and $\mathbf{R}_x$ are parameters determined by DiPCA [120].

For the decomposition of $\mathbf{y}_k$ in Eq. (4.28), $\mathbf{u}_{c,k}$ is the score vector in DCVS, $\hat{\mathbf{u}}_{\psi,k}$ is the score vector in DAOS, $\mathbf{t}_{sy,k}$ is the score vector in SOPS, and $\tilde{\mathbf{y}}_k$ is residual vector in SORS, which are

$$\mathbf{u}_{c,k} = \mathbf{R}_c^{\top}\mathbf{x}_{\zeta,k} \tag{4.32}$$

$$\hat{\mathbf{u}}_{\psi,k} = \sum_{i=1}^{s} \psi_i\mathbf{y}_{k-i}\mathbf{Q} \tag{4.33}$$

$$\mathbf{t}_{sy,k} = \mathbf{P}_{sy}^{\top}\tilde{\mathbf{y}}_c \tag{4.34}$$

$$\tilde{\mathbf{y}}_k = \left(\mathbf{I} - \mathbf{P}_{sy}\mathbf{P}_{sy}^{\top}\right)\tilde{\mathbf{y}}_c \tag{4.35}$$

where $\mathbf{x}_{\zeta,k} = \sum_{i=0}^{s} \zeta_i\mathbf{x}_{k-i}$, and $\tilde{\mathbf{y}}_c = \mathbf{y}_k - \mathbf{Q}_c\mathbf{u}_{c,k} - \mathbf{C}\hat{\mathbf{u}}_{\psi,k}$.

4.3 Comprehensive Dynamic Monitoring Scheme

To develop a comprehensive dynamic monitoring scheme for the extracted subspaces, their monitoring statistics are designed, which is called concurrent DALVR (CDALVR).

For a test sample pair $(\mathbf{x}_k, \mathbf{y}_k)$, the dynamic output-relevant score vector $\mathbf{u}_{c,k}$ is monitored with the T_c^2 index in DCVS.

$$T_c^2 = \mathbf{u}_{c,k}^\top \mathbf{\Lambda}_c^{-1} \mathbf{u}_{c,k} = \mathbf{x}_{\zeta,k}^\top \mathbf{R}_c \mathbf{\Lambda}_c^{-1} \mathbf{R}_c^\top \mathbf{x}_{\zeta,k} \tag{4.36}$$

where $\mathbf{\Lambda}_c$ is the variance matrix of $\mathbf{U}_c$. Note that the variations in this subspace are predictable from input variables.

For the auto-regressive output score vector $\hat{\mathbf{u}}_{\psi,k}$, which is not predictable from input, its corresponding statistic T_{dy}^2 is

$$T_{dy}^2 = \hat{\mathbf{u}}_{\psi,k}^\top \mathbf{\Lambda}_{dy}^{-1} \hat{\mathbf{u}}_{\psi,k} \tag{4.37}$$

where $\mathbf{\Lambda}_{dy}$ is the variance matrix of $\hat{\mathbf{U}}_\psi$.

The static output score vector $\mathbf{t}_{sy,k}$ and the static output residual vector $\tilde{\mathbf{y}}_k$ are monitored by

$$T_{sy}^2 = \mathbf{t}_{sy,k}^\top \mathbf{\Lambda}_{sy}^{-1} \mathbf{t}_{sy,k} \tag{4.38}$$

$$Q_{sy} = \|\tilde{\mathbf{y}}_k\|^2 = \tilde{\mathbf{y}}_c^\top \left(\mathbf{I} - \mathbf{P}_{sy} \mathbf{P}_{sy}^\top\right) \tilde{\mathbf{y}}_c \tag{4.39}$$

where $\mathbf{\Lambda}_{sy}$ is the variance matrix of $\mathbf{T}_{sy}$.

For the decomposed input subspaces, variations in DIPS, SIPS and SIRS are captured by φ_{dx}, T_{sx}^2 and Q_{sx} respectively, which are defined as

$$\varphi_{dx} = \hat{\mathbf{t}}_{dx,k}^\top \mathbf{\Phi}_{dx} \hat{\mathbf{t}}_{dx,k} = \frac{T_{dx}^2}{T_{dx,\mathrm{cl}}^2} + \frac{Q_{dx}}{Q_{dx,\mathrm{cl}}} \tag{4.40}$$

$$T_{sx}^2 = \mathbf{t}_{sx,k}^\top \mathbf{\Lambda}_{sx}^{-1} \mathbf{t}_{sx,k} \tag{4.41}$$

$$Q_{sx} = \|\tilde{\mathbf{x}}_k\|^2 = \tilde{\mathbf{x}}_c^\top (\mathbf{I} - \mathbf{P}_{sx} \mathbf{P}_{sx}^\top) \tilde{\mathbf{x}}_c \tag{4.42}$$

where φ_{dx} is a combined index integrating Hotelling's T^2 with Q statistics [3], and $\mathbf{\Phi}_{dx}$ is

$$\mathbf{\Phi}_{dx} = \frac{\mathbf{P}_{dx} \mathbf{\Lambda}_{dx}^{-1} \mathbf{P}_{dx}^\top}{T_{dx,\mathrm{cl}}^2} + \frac{\mathbf{I} - \mathbf{P}_{dx} \mathbf{P}_{dx}^\top}{Q_{dx,\mathrm{cl}}} \tag{4.43}$$

where $\mathbf{\Lambda}_{dx}$ is the variance matrix of $\widehat{\mathbf{T}}_{dx}$, $\mathbf{P}_{dx}$ is the loading score matrix of $\widehat{\mathbf{T}}_{dx}$, and $T^2_{dx,\mathrm{cl}}$ and $Q_{dx,\mathrm{cl}}$ are the corresponding control limits [60]. $\mathbf{\Lambda}_{sx}$ in Eq. (4.41) represents the variance matrix of $\mathbf{T}_{sx}$.

The details of the control limits of the aforementioned statistics are summarized in Table 7, where l_c, l_{dy}, l_{sy}, and l_{sx} denote the number of latent variables for $\mathbf{u}_{c,k}$, $\widehat{\mathbf{u}}_{\psi,k}$, $\tilde{\mathbf{y}}_c$ and $\tilde{\mathbf{x}}_c$, respectively. The calculations of control limits corresponding to each subspace are obtained according to Literature [3] and [60].

To differentiate the importance of detected anomalies, as shown in Table 7, different alarming levels L_1, L_2, and L_3 are attached to faults or disturbances observed in subspaces decomposed by CDALVR, where L_1 and L_3 represent the highest and lowest alarming level, respectively. More concretely, the faults with L_1 alarming level will affect output variables, the L_2 alarming faults have potential influence on the output, and the L_3 alarms are raised for the input-relevant and output-irrelevant faults.

Table 7: Monitoring statistics and control limits for CDALVR

Subspace		Characteristics	Statistics	Control limit	Alarming level
Output	DCVS	Dynamic; predictable from input	T_c^2	$T^2_{c,\mathrm{cl}} = \chi^2_{\alpha,l_c}$	L_1
	DAOS	Auto-regressive; unpredictable from input	T_{dy}^2	$T^2_{dy,\mathrm{cl}} = \chi^2_{\alpha,l_{dy}}$	L_1
	SOPS	Static; unpredictable from input	T_{sy}^2	$T^2_{sy,\mathrm{cl}} = \chi^2_{\alpha,l_{sy}}$	L_1
	SORS	Static; unpredictable from input	Q_{sy}	$Q_{sy,\mathrm{cl}} = g_{sy}\chi^2_{\alpha,h_{sy}}$	L_3
Input	DIPS	Dynamic	φ_{dx}	$\varphi_{dx,\mathrm{cl}} = g_{dx}\chi^2_{\alpha,h_{dx}}$	L_3
	SIPS	Static	T_{sx}^2	$T^2_{sx,\mathrm{cl}} = \chi^2_{\alpha,l_{sx}}$	L_3
	SIRS	Static, potentially relevant to output	Q_{sx}	$Q_{sx,\mathrm{cl}} = g_{sx}\chi^2_{\alpha,h_{sx}}$	L_2

The concurrent dynamic monitoring procedure is outlined as follows:

1. If $T_c^2 > T^2_{c,\mathrm{cl}}$, the L_1 fault alarm is raised with $(1-\alpha)\times 100\%$ confidence in DCVS, denoting that the fault is *dynamic* output-relevant and predictable from input data.

2. If $T_{dy}^2 > T^2_{dy,\mathrm{cl}}$, with $(1-\alpha)\times 100\%$ confidence, a dynamic auto-regressive output-relevant fault is fired in DAOS with L_1 alarming level, which is unpredictable from input variables.

3. If $T_{sy}^2 > T^2_{sy,\mathrm{cl}}$, a static output-relevant and input-irrelevant fault is detected in SOPS with $(1-\alpha)\times 100\%$ confidence, which indicates that a L_1 fault alarm occurs.

4. If $Q_{sx} > Q_{sx,\mathrm{cl}}$, a static output-irrelevant fault is observed in SIRS with $(1-\alpha)\times 100\%$ confidence, which means that a L_2 fault alarm is raised, and this fault is input-relevant and potentially relevant to output data.

5. If $Q_{sy} > Q_{sy,\mathrm{cl}}$, a static output-relevant fault with a L_3 alarming level is detectable for SORS with $(1-\alpha)\times 100\%$ confidence.

6.If $\varphi_{dx} > \varphi_{dx,\mathrm{cl}}$, a L_3 fault alarm is activated in DIPS with $(1-\alpha)\times 100\%$ confidence, showing that a dynamic input-relevant fault is observable.

7. If $T_x^2 > T_{x,\mathrm{cl}}^2$, a static input-relevant fault is detectable for SIPS with $(1-\alpha)\times 100\%$ confidence, indicating that the fault is attached to the L_3 alarming level.

4.4 Tennessee Eastman Process Case Study

In this section, the additional simulation data of TEP [121] is utilized to investigate the modeling and monitoring performance of DALVR and CDALVR. The dataset contains 50,000 training samples and 96,000 testing samples for each disturbance. 500 simulation runs are included in the dataset, and each run contains 41 measured variables (XMEAS (1-41)) and 11 manipulated variables (XMV (1-11)).

To illustrate the dynamic modeling performance of DALVR, XMEAS (1-22) and XMV (1-11) are selected as input variables **X**, while XMEAS (38) is chosen as the output variable **Y**. Note that the variables in **X** are sampled with an interval of 3 minutes, while the output variable **Y** is sampled every 15 minutes. Thus, to address the irregular sampling frequency, the filtering approach described by Zhu et al. [62] is employed to balance the sampling discrepancy.

4.4.1 Modeling Performance

Determined by the cross-validation method, the modeling parameters for DALVR are $l = 2$, $s = 2$, and $d = 2$; for DrLVR, $l = 5$ and $s = 1$. The regularized parameters for DALVR are $\gamma_w = 0.005$, $\gamma_\beta = 0.005$, and $\gamma_\delta = 0.005$, while for DrLVR, $\kappa = 0.005$. The confidence level is selected to be 99 %. The prediction performance is measured by MSEs, which are summarized in Table 8, and the output variations predicted by DALVR and DrLVR are presented in Figure 29. Compared with DrLVR, DALVR achieves a more comprehensive exploitation on the quality space, and thus as shown in Table 8, its MSE is lower than that of DrLVR, which is in line with the predictive performance shown in Figure 29.

Table 8: MSEs for DrLVR and DALVR in additional simulation of TEP

Variable	DrLVR	DALVR
XMEAS (38)	0.3570	0.3308

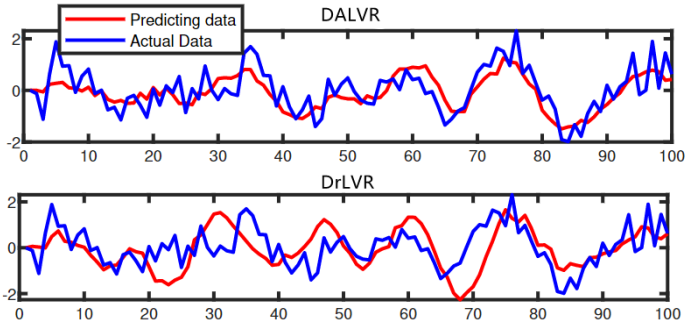

Figure 29: Predicted output variations for normal case in additional simulation of TEP

4.4.2 Concurrent Monitoring Performance

The TEP simulates 20 disturbances, and the first 15 known disturbances (IDV (1-15)) are adopted for analysis in this case study. With the control limit selected as 99%, the FDRs and FARs [62] results of output-relevant and output-irrelevant disturbances are summarized in Tables 9 and 10, respectively. Note that the T^2 statistic of CDALVR in these two tables is the combination of the T_c^2, T_{dy}^2, and T_{sy}^2 statistics, which is for output-relevant anomaly detection, including both predictable and unpredictable, dynamic and static faults. The T^2 statistic realizes a comprehensive monitoring for output relevant data.

As shown in Table 9, CDALVR achieves comparable FDRs with DrLVR, with better performance for IDV (7) and IDV (10), and its false alarms are raised much less than DrLVR. Furthermore, CDALVR-based monitoring provides detailed information for each subspace, monitoring dynamic and static variations in both input and output data. For output-irrelevant disturbances, only FARs are listed in Table 10, since there are few faulty samples and FDRs are noisy. As shown in the table, in most cases, CDALVR achieves better monitoring results than DrLVR with smaller FARs. The input-relevant FARs are also available in CDALVR with T_{sx}^2, which yet should receive lower attention than output-related monitoring indices.

Table 9: FDRs and FARs for output-relevant disturbances in additional simulation of TEP

Disturbance	FDR							FAR						
	DrLVR	CDALVR						DrLVR	CDALVR					
	T^2	T^2	T_c^2	T_{dy}^2	T_{sy}^2	φ_{dx}	T_{sx}^2	T^2	T^2	T_c^2	T_{dy}^2	T_{sy}^2	φ_{dx}	T_{sx}^2
IDV (1)	**1**	0.96	0.8429	0.4411	0.3711	0.8229	0.9998	0.9282	0.3723	0.1956	**0**	0.2501	**0**	0.1672
IDV (2)	**1**	0.8696	0.7446	0.0365	0.1394	0.7912	0.9998	0.9034	0.3556	0.1267	**0**	0.2899	**0**	0.1707
IDV (5)	**1**	0.9986	0.5994	0.247	0.9951	0.6667	0.6783	0.9455	0.9428	0.9126	**0**	0.938	0.2143	0.0261
IDV (6)	**1**	**1**	0.9689	0.8585	0.9468	0.9583	0.9934	0.687	0.4457	0.3783	0.5556	0.3065	**0**	0.2888
IDV (7)	0.9497	0.9749	0.9421	0.5014	0.2703	0.8958	**1**	0.4395	0.1703	0.1241	**0**	0.0793	0.3846	0.0202
IDV (8)	**0.994**	0.9741	0.9374	0.5892	0.2602	0.8077	0.9831	0.7742	0.2325	0.1755	0.3	0.1267	0.3636	0.1383
IDV (10)	0.9001	**0.9565**	0.8195	0.1435	0.758	0.3585	0.2824	0.7418	0.7081	0.5625	0.1667	0.6691	0.1489	**0.0311**
IDV (12)	**0.995**	0.9876	0.867	0.6576	0.6759	0.7952	0.9746	0.8092	0.6175	0.4327	0.3333	0.5613	0.2353	**0.1812**
IDV (13)	**0.9993**	0.9946	0.804	0.6721	0.7802	0.9451	0.9732	0.6194	0.4907	0.3565	0.3846	0.4546	**0**	0.1134

Table 10: FARs for output-irrelevant disturbances in additional simulation of TEP

Disturbance	DrLVR	CDALVR					
	T^2	T^2	T_c^2	T_{dy}^2	T_{sy}^2	φ_{dx}	T_{sx}^2
IDV (3)	0.0123	0.0161	0.0100	**0**	0.0072	**0**	0.0067
IDV (4)	0.1907	0.0178	0.0106	**0**	0.0077	**0**	0.2253
IDV (9)	0.0124	0.0196	0.0118	**0**	0.0077	**0**	0.2253
IDV (11)	0.3765	0.0208	0.0133	**0**	0.0080	**0**	0.0768
IDV (14)	0.9595	0.0814	0.0260	**0**	0.0590	**0**	0.0752
IDV (15)	0.0145	0.0211	0.0144	**0**	0.0070	**0**	0.0096

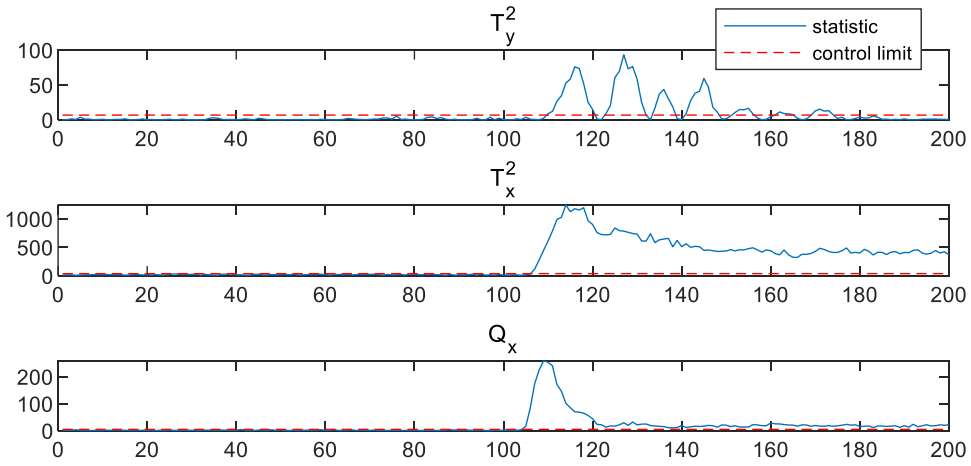

Figure 30: PCA-based process and quality monitoring results for IDV (1)

As observed, CDALVR shows superiority to DrLVR due to its comprehensive monitoring ability and improved anomaly detection performance.

For better illustration, IDV (1) (an output-relevant disturbance) and IDV (14) (an output-irrelevant disturbance) are chosen to visualize the monitoring performance of CDALVR and DrLVR. The first 100 samples are training data, while the 101st-200th samples are the first round of testing data.

IDV (1) introduces a step change of A/C feed ratio in Stream 4 [118], and as shown in the PCA-based monitoring results in Figure 30, it affects the output variable transiently. Specifically, output variable is abnormal in the interval [110th, 174th], while for input variables, T_x^2 and Q_x detect the faults from 106th and 105th sample respectively.

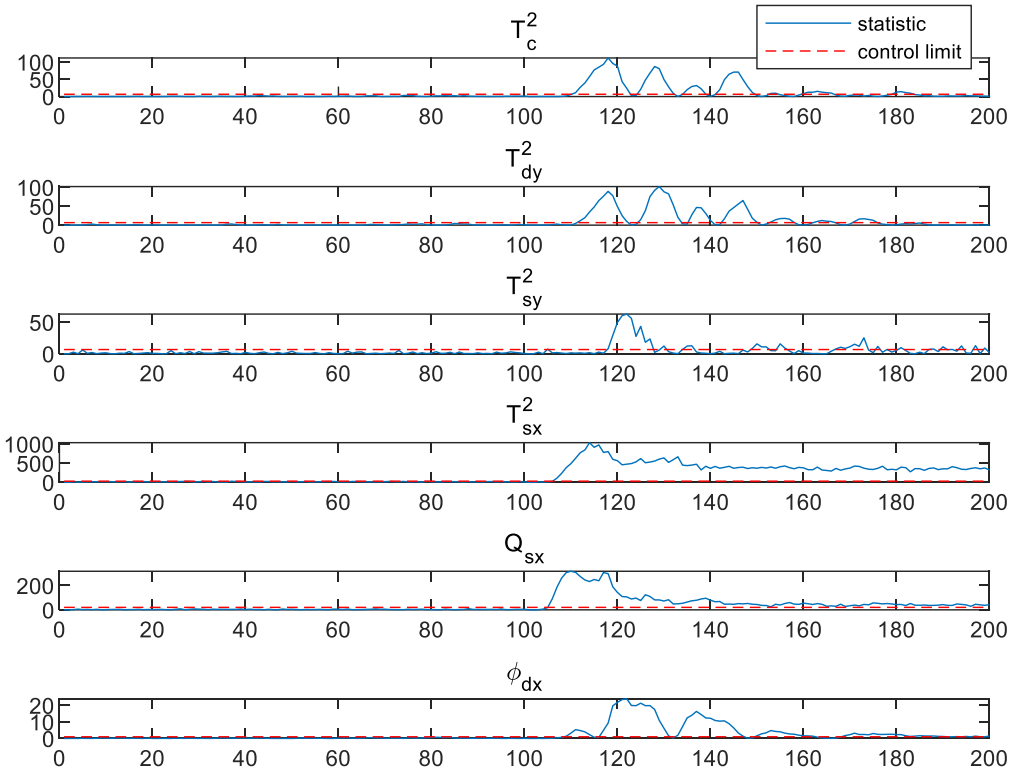

Figure 31: CDALVR-based monitoring results for IDV (1)

The monitoring results of CDALVR and DrLVR are presented in Figures 31 and 32, respectively. Figure 31 shows that all quality relevant monitoring indices of CDALVR detect the anomalies, and they return to normal conditions later, which is consistent with the PCA-based monitoring results in Figure 30. T_c^2 is constructed by the predictable dynamic output-relevant variations, and in it captures the anomalies in DCVS for the 111st-150th faulty samples. T_{dy}^2 detects dynamic output-relevant faulty samples that are unpredicted from the input in the range of [112nd, 175th]. T_{sy}^2 also successfully detects additional static output-relevant anomalies in the interval of [118th, 173rd]. The monitoring results of output-relevant statistics in CDALVR are in line with the T_y^2 of PCA-monitoring in Figure 30. In addition to quality-relevant anomalies, CDALVR also detects anomalies in the input data with φ_{dx} from the 109th sample, T_{sx}^2 from the 107th sample and Q_x from the 106th sample. It is observed that the IDV (1) disturbance affects input data longer than quality variables. In contrast, both monitoring indices

of DrLVR keep alarming from the 105th sample, and it cannot provide a detailed analysis for the sources of the detected faults.

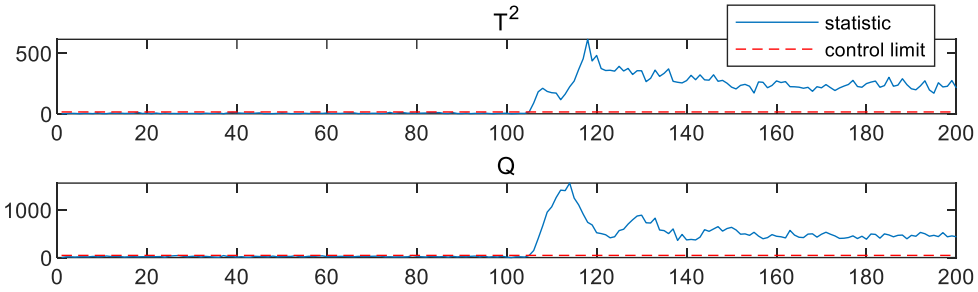

Figure 32: DrLVR-based monitoring results for IDV (1)

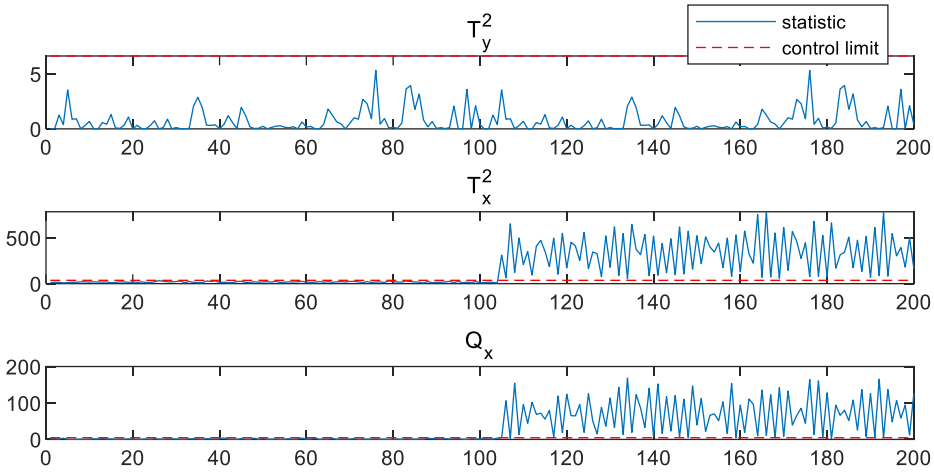

Figure 33: PCA-based process and quality monitoring results for IDV (14)

The IDV (14) disturbance is caused by the sticking issue of the reactor cooling water valve [118]. The PCA-based monitoring in Figure 33 indicates that it affects T_x^2 and Q_x statistics from the 105th sample, while its T_y^2 index remains unaffected, showing that this disturbance is input-relevant and output-irrelevant. Figures 34 and 35 summarize the CDALVR-based and DrLVR-based monitoring results, respectively. CDALVR 's monitoring results generally align with Figure 33: only the input-relevant statistics T_{sx}^2 and Q_x raise the L_3-level alarms from the 105th and 106th faulty samples, respectively, and quality-relevant indices T_c^2, T_{dy}^2, T_{sy}^2 and dynamic input-relevant index φ_{dx} are within their corresponding control limits. DrLVR has both monitoring indices exceeding their control

limit, and it is hard to collect any further information from Figure 35. Thus, DrLVR based monitoring method is less informative than CDALVR.

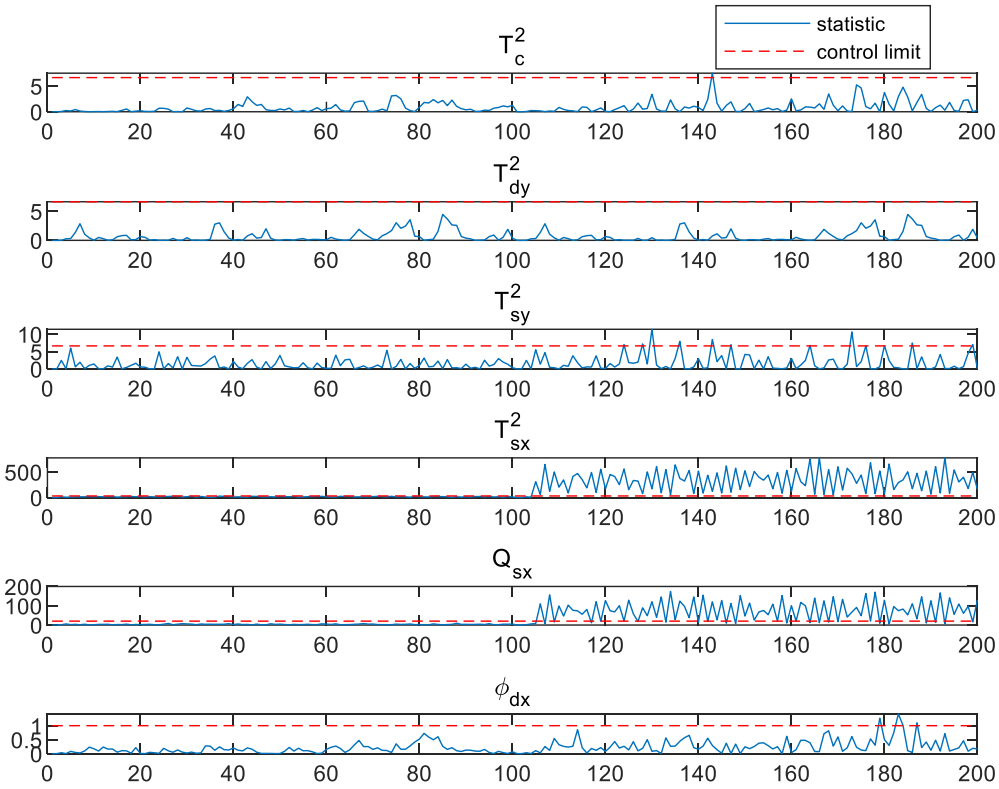

Figure 34: CDALVR-based monitoring results for IDV (14)

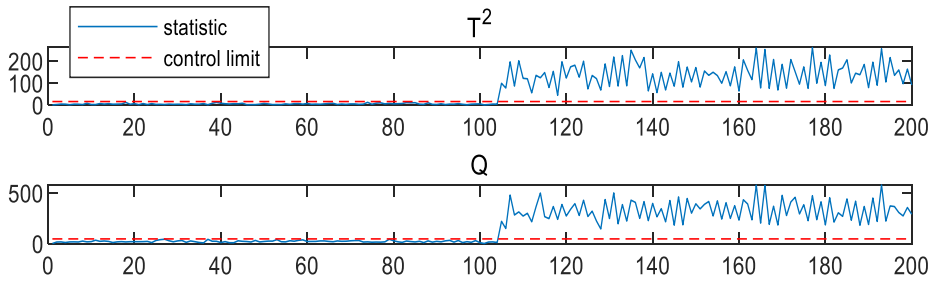

Figure 35: DrLVR-based monitoring results for IDV (14)

4.5 Summary

This chapter proposes a DALVR method by modeling both dynamic cross-correlations and dynamic auto-correlations simultaneously, leading to the enhancement of ability to handle dynamic relations. A dynamic comprehensive modeling and monitoring framework is also developed with subsequent decompositions. Case study on the TEP data illustrates that DALVR establishes a more predictive relationship between input and output data compared with DrLVR. Meanwhile, CDALVR accomplishes a comprehensive multilayer monitoring on both input- and output-relevant faults, which can distinguish detected anomalies in different subspaces given dynamic or static characteristics.

Chapter 5
DALVR-based Fault Diagnosis and Causal Analysis Framework

As shown in previous chapters, KLVR was designed to deal with nonlinearity in practical processes, and its concurrent modeling and monitoring scheme realizes a comprehensive anomaly detection in both quality and process spaces. A dynamic auto-regressive LVR was further proposed to capture system dynamics in practical processes. DALVR fully extracts the information in both input and output data by considering cross-correlations between input and output and auto-correlations from current and past output information. A comprehensive monitoring scheme was also developed to monitor dynamic anomalies. However, fault detection is only the first step, and it is necessary to analyze the root causes of the detected faults. In this chapter, a fault diagnosis and causal analysis scheme is designed for further analysis.

5.1 DALVR-based Fault Diagnosis and Causal Analysis Framework

Root cause identification is a key issue for fault diagnosis. After an anomaly is detected, it is necessary to locate their root causes to ensure the operation safety and efficiency, and the increasingly complex processes in industrial practice have increased the difficulty of fault diagnosis. Traditional fault diagnosis approaches can only determine variables contributing to the fault initially, failing to locate the root cause accurately due to ignorance of a cause-and-effect relationship between selected variables. It is necessary to integrate traditional fault diagnosis approaches with causality analysis to improve the ability of root cause identification, leading to the proposed DALVR-based fault diagnosis and causal analysis framework.

The monitoring scheme based on DALVR includes T^2 and Q statistics, which can be obtained in a similar way. To ensure the monitoring and diagnosis efficiency, based on Eq. (2.23), a combined monitoring index φ is first designed by integrating T^2 and Q indices as

$$\varphi = \frac{T^2}{T_{\text{cl}}^2} + \frac{Q}{Q_{\text{cl}}} = \mathbf{x}^\top \boldsymbol{\Phi}_\varphi \mathbf{x} \tag{5.1}$$

where

$$\boldsymbol{\Phi}_\varphi = \frac{\mathbf{R}\boldsymbol{\Lambda}^{-1}\mathbf{R}^\top}{T_{\text{cl}}^2} + \frac{\mathbf{I} - \mathbf{P}\mathbf{R}^\top}{Q_{\text{cl}}}. \tag{5.2}$$

The control limit of φ is obtained by Eq. (2.24).

Then, RBC is adopted and modified as the first step of root cause analysis. RBC obtains the amount of reconstruction along each variable by minimizing its effect over the detection statistics [66]. Its key assumption is that reconstruction magnitude in the fault direction should be larger than other non-faulty variables. Assuming that $\mathbf{x}i$ is affected by a fault, according to Eqs. (2.25)-(2.28), the reconstructed combined monitoring index based on φ index is formulated as

$$\text{Index}(\mathbf{z}_i)^{\varphi} = \mathbf{z}_i^{\top}\boldsymbol{\Phi}_{\varphi}\mathbf{z}_i = \|\mathbf{z}_i\|_{\Phi}^2 = \|\mathbf{x} - \xi_i f_i\|_{\Phi}^2. \tag{5.3}$$

To make $\text{Index}(\mathbf{z}_i)$ within the control limit, the fault magnitude f_i is determined by minimizing $\text{Index}(\mathbf{z}_i)$. Thus, take the first derivative of $\text{Index}(\mathbf{z}_i)$ with regard to f_i and make it equal to zero yields:

$$f_i^{\varphi} = \xi_i\left(\xi_i^{\top}\boldsymbol{\Phi}_{\varphi}\xi_i\right)^{-1}\xi_i^{\top}\boldsymbol{\Phi}_{\varphi}\mathbf{x}. \tag{5.4}$$

Then the reconstruction amount along fault direction ξ_i, RBC_i^{φ}, is

$$\text{RBC}_i^{\varphi} = \|\xi_i f_i\|_{\Phi}^2 = \mathbf{x}^{\top}\boldsymbol{\Phi}_{\varphi}\xi_i\left(\xi_i^{\top}\boldsymbol{\Phi}\xi_i\right)^{-1}\xi_i^{\top}\boldsymbol{\Phi}_{\varphi}\mathbf{x}. \tag{5.5}$$

RBC is an effective approach for fault diagnosis, but sometimes it is not reliable enough, since for each variable, the RBC value may be affected by other variables. To alleviate this issue, a relative RBC index, namely rRBC_i^{φ}, is employed instead. rRBC_i^{φ} is defined as the ratio of RBC_i^{φ} to the average values of the i^{th} variable calculated with $\text{RBC}_{\text{avg},i}^{\varphi}$ obtained from the normal data. rRBC_i^{φ} is formulated as

$$\text{rRBC}_i^{\varphi} = \frac{\text{RBC}_i^{\varphi}}{\text{RBC}_{\text{avg},i}^{\varphi}} = \frac{\mathbf{x}^{\top}\boldsymbol{\Phi}_{\varphi}\xi_i\left(\xi_i^{\top}\boldsymbol{\Phi}_{\varphi}\xi_i\right)^{-1}\xi_i^{\top}\boldsymbol{\Phi}_{\varphi}\mathbf{x}}{\text{RBC}_{\text{avg},i}^{\varphi}}. \tag{5.6}$$

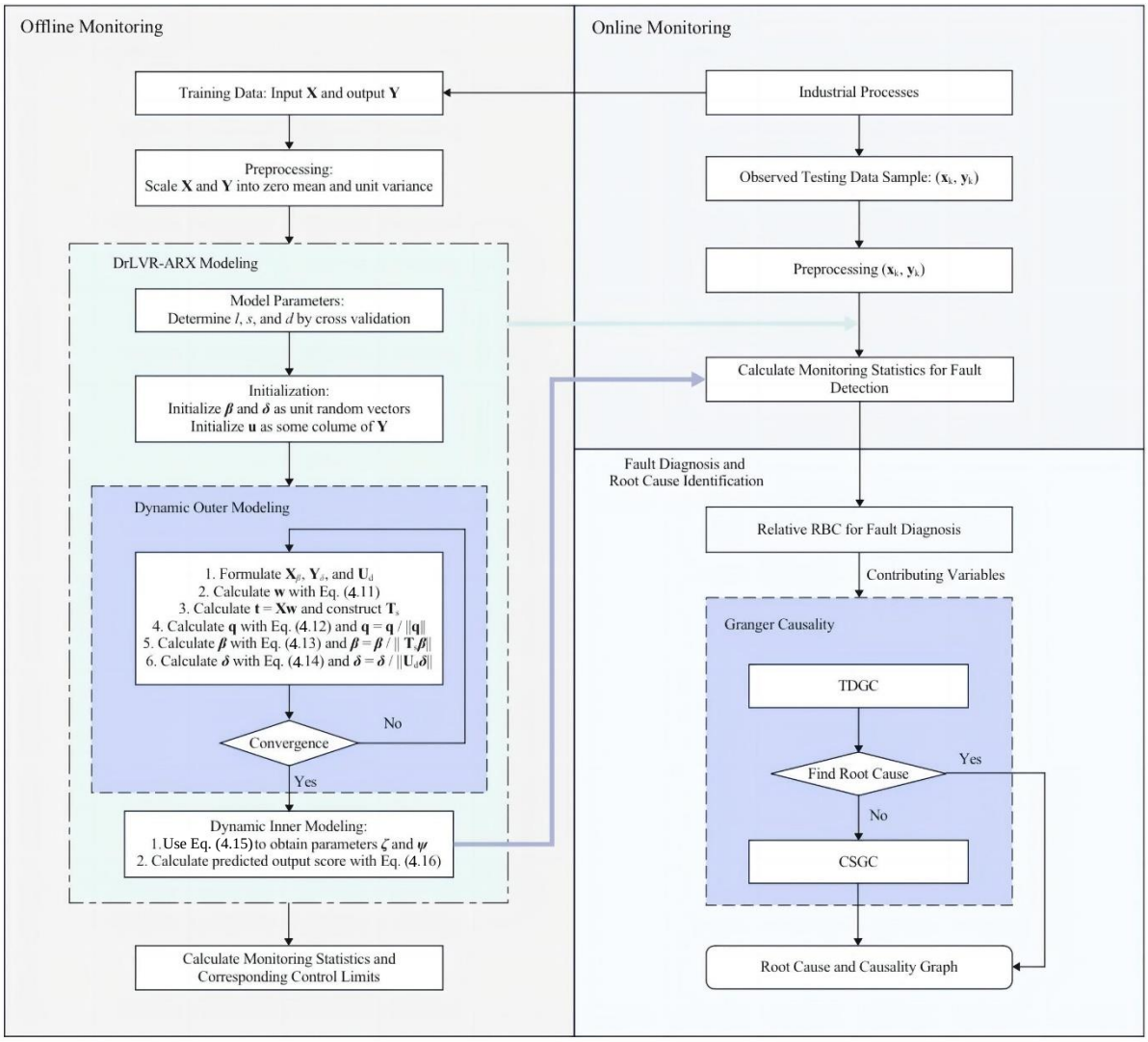

Figure 36: Fault diagnosis and root cause identification framework based on DALVR

$\mathrm{rRBC}_i^{\varphi}$ can remove the interference brought by the correlations among various variables. The variables that have larger $\mathrm{rRBC}_i^{\varphi}$ values are regarded as the main contributing variables, which are used for further root cause identification. To further attribute the root causes of the detected fault, TDGC [90] and CSGC [97] described in Chapter 2 are employed to identify the causal relations of the contributing variables identified by RBC. TDGC is easy to implement and understand, so it is extensively used in root cause analysis. However, sometimes it fails to provide enough information to locate the root causes due to the existence of other potential causes. Thus, it is necessary to enhance reliability. In this work, we integrate another causality approach with TDGC to address this issue.

CGSC has good causal inference performance since it can identify the causal relationship in the frequency domain, providing strong support for further root cause analysis. However, its computational cost is relatively large. If the range of variables is narrowed down, it will be more efficient. To combine their advantages and improve the performance of causal analysis, the integration of TDGC and CSGC will lead to the improvement of reliability and effectiveness.

The integrated framework is depicted in Figure 36. In the framework, DALVR is first performed on historical normal data to obtain model parameters, and fault detection and diagnosis statistics are developed with DALVR. For online monitoring, new samples are processed and monitored with the established statistics. Once a faulty sample is identified, relative RBC is conducted to obtain contributing variables, which will then be employed for further root cause identification with TDGC and CSGC.

5.2 Tennessee Eastman Process Case Study

The Tennessee Eastman Process (TEP) [118] is analyzed to show the efficacy of the proposed framework. In this section, XMEAS (1-22) and XMV (1-11) are selected as input variables (Variables 1-33), while XMEAS (38) serves as the output variable.

Table 11: MSEs for DrLVR, DAPLS, and DALVR in TEP

Variable	DrLVR	DAPLS	DALVR
XMEAS (38)	0.3872	0.1726	**0.1464**

Table 12: MSEs for different parts of DALVR in TEP

Variable	DALVR	Cross-correlation part	Auto-correlation part
XMEAS (38)	**0.1464**	0.6799	0.1920

The modelling performance of DrLVR and DALVR are first compared, which are trained with 500 normal samples. With the cross-validation method, the parameters are determined as $l = 1$, $s = 1$, $d = 1$, $\gamma_\omega = 0.005$, $\gamma_\beta = 0.005$, and $\gamma_\delta = 0.005$ for DALVR; and $l = 1$, $s = 3$, and $\kappa = 0.005$ for DrLVR. Apart from DrLVR, DAPLS is also used to reflect the effectiveness of DALVR on prediction. For DAPLS, $l = 3$, $s = 3$, and $d = 1$. The confidence level is chosen as 99 %. The prediction performance is evaluated with MSEs, which are summarized in Table 11.

Table 13: Monitoring metrics of T^2 for quality-relevant disturbances in TEP

Fault Type	FDR			FAR			AR			PR		
	DrLVR	DAPLS	DALVR	DrLVR	DAPLS	DALVR	DrLVR	DAPLS	DALVR	DrLVR	DAPLS	DALVR
IDV (1)	0.7826	**1**	0.8261	0.1575	0.7808	**0.1370**	0.8281	0.4063	**0.8542**	0.6102	0.2875	**0.6552**
IDV (2)	0.9167	**1**	**1**	0.7667	0.7889	**0.7389**	0.2760	0.2604	0.2969	0.0738	**0.0779**	0.0764
IDV (5)	**1**	0.8414	**1**	0.8025	**0.0896**	0.7963	0.3187	**0.5000**	0.3240	0.1815	**0.3256**	0.1826
IDV (6)	0.9905	**1**	**1**	0.2818	**0.2364**	0.2773	0.9281	0.9458	**0.9533**	0.9220	0.9343	**0.9436**
IDV (7)	0.8182	0.9273	**0.8364**	0.3431	0.2336	**0.1387**	0.7031	0.8125	**0.8542**	0.4891	0.6145	**0.6970**
IDV (8)	0.8860	**0.9485**	0.9051	0.3263	0.5158	**0.1789**	0.7708	0.7188	**0.8125**	0.7304	0.6525	**0.8298**
IDV (10)	0.5818	0.7091	**0.7818**	0.3577	**0.2774**	0.2818	0.6250	**0.7188**	0.6938	0.3951	**0.5065**	0.4538
IDV (12)	0.7964	**0.9273**	0.8291	0.5244	0.4878	**0.4268**	0.7010	**0.7500**	0.7281	0.7145	**0.7183**	0.6796
IDV (13)	0.8145	0.9274	**0.8306**	0.2353	0.3676	**0.1018**	0.7969	0.8229	**0.8333**	0.8632	0.8214	**0.9035**

Table 14: FARs of T^2 for quality-irrelevant disturbances in TEP

Fault Type	DrLVR	DAPLS	DALVR
IDV (3)	0.0957	0.0213	0.0319
IDV (4)	0.2116	0.0212	0.0847
IDV (9)	0.0851	0.0319	0.0266
IDV (11)	0.1341	0.0391	0.0391
IDV (14)	0.4063	0.0885	0.0521
IDV (15)	0.0737	0.0105	0.0211

Compared with DrLVR, DALVR achieves a much lower MSE than DrLVR, since it exploits more information with the aid of historical quality data. The MSEs of DALVR, its cross-correlation part, and auto-correlation part are shown in Table 12. As shown in Table 12, both cross-correlation part and auto-correlation part have higher MSEs than the overall MSE value of DALVR, which implies that the modelling performance with past process samples only or past quality samples only is not satisfactory. With the control limit selected as 99%, the false detection rates (FDRs) and false alarm rates (FARs) of these two categories are summarized in Tables 13 and 14, respectively. FDR is the ratio of the number of actual faults that are detected with the selected algorithm and the number of total faulty samples, while FAR is the ratio of the number of normal samples that are incorrectly categorized as faults and the number of total normal samples. FDR and FAR are used to measure the monitoring performance of algorithms. The larger the FDR is, the better the monitoring performance is, while the smaller the FAR is, the better the monitoring performance is. Compared with DrLVR, DALVR can capture more variance of the data, leading to better modelling performance. In this case, theoretically,

the performance of DALVR on fault detection and alarm will be better than DrLVR. Apart from FDR and FAR, accuracy rate (AR) and precision rate (PR) can be also used as measures of observational error to compare the monitoring performance. AR is the percentage of correct predictions for a given dataset, which evaluates how close or far off the samples are to their true values. PR measures that fraction of examples classified as faulty samples that are truly faulty, which measures how close or dispersed the samples are to each other [122].

As shown in Table 13, for output-relevant disturbances, for IDV (1, 5, 8, 12, and 13), DALVR achieves the smallest FARs and the largest ARs and PRs, while its FDRs are comparable to those of DrLVR and DAPLS. For IDV (2, 6, 7, and 10), FDRs of DALVR are largest among these algorithms, while their ARs and PRs are either the largest or relatively large, and FARs are either the smallest or relatively small. Thus, in general, DALVR has better fault detection ability, accuracy, and precision performance in quality-relevant monitoring, since compared with other static and dynamic algorithms, DALVR employs both past process and quality information for predictive modelling, and it can capture the data variations more accurately. For output-irrelevant disturbances, only FARs are listed in Table 14, since there are few faulty samples and FDRs, ARs, and PRs are noisy. As shown in Table 14, DALVR obtains comparable or smaller FARs compared with other algorithms. FARs of DALVR are comparable or smaller compared to other algorithms, which means that there are only a few normal samples classified as faulty samples erroneously. Thus, as observed, DALVR shows superiority due to its comprehensive monitoring ability and improved anomaly detection performance.

One of the quality-relevant faults, IDV (1), is selected as an instance to illustrate the proposed fault diagnosis and root cause identification framework. IDV (1) is a step change in the A/C feed ratio in Stream 4. Specifically, it causes a decrease of 3% in the A feed and an increase of 3% in the C feed [118]. The DALVR-based monitoring result with the combined index φ is shown in Figure 37, and the fault is detected since the 165th sample. The rRBC$^{\varphi}$ of the 165th sample is then illustrated in Figure 38. As shown in Figure 38, RBC identifies many faulty variables due to the interrelations among variables, including Variables 7, 15, 17, 18, 20, 29–31, and 33. To further accurately locate the root causes, these variables are selected as the potential contributing candidates in the subsequent Granger causal analysis.

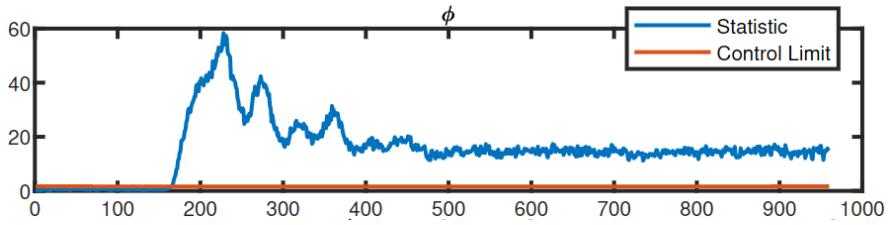

Figure 37: DALVR-based monitoring result for IDV (1)

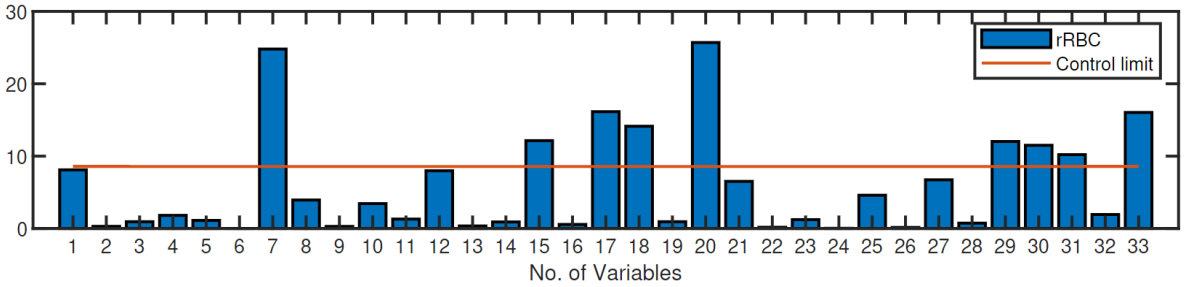

Figure 38: Relative reconstruction-based contribution results for IDV (1)

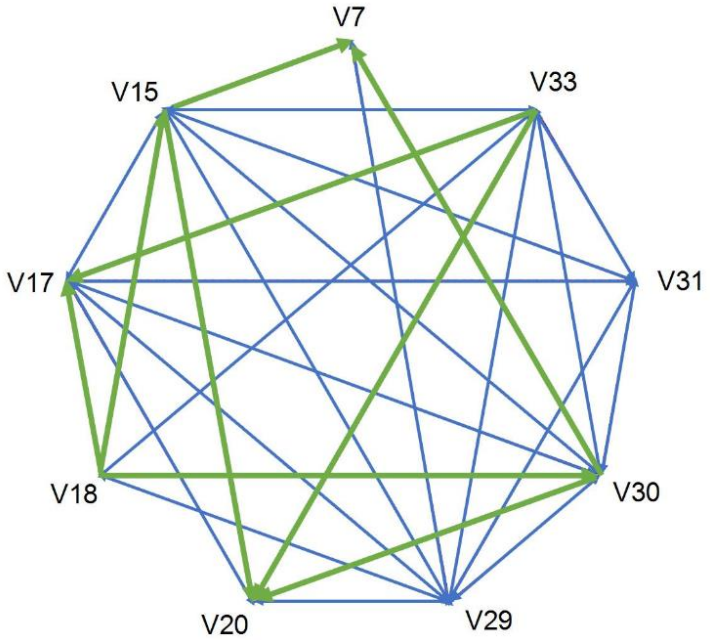

Figure 39: Time-domain Granger causality map for IDV (1)

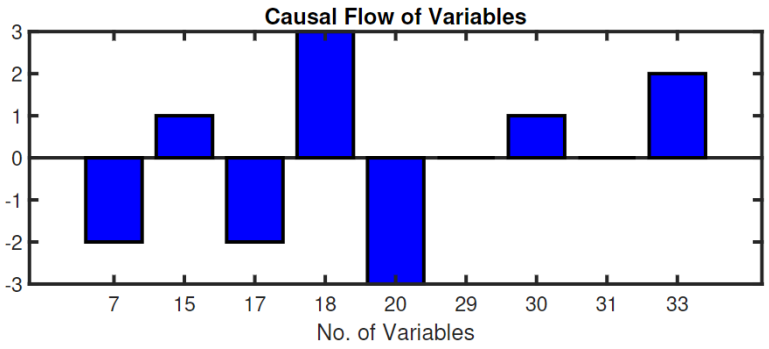

Figure 40: Causal flow results for IDV (1)

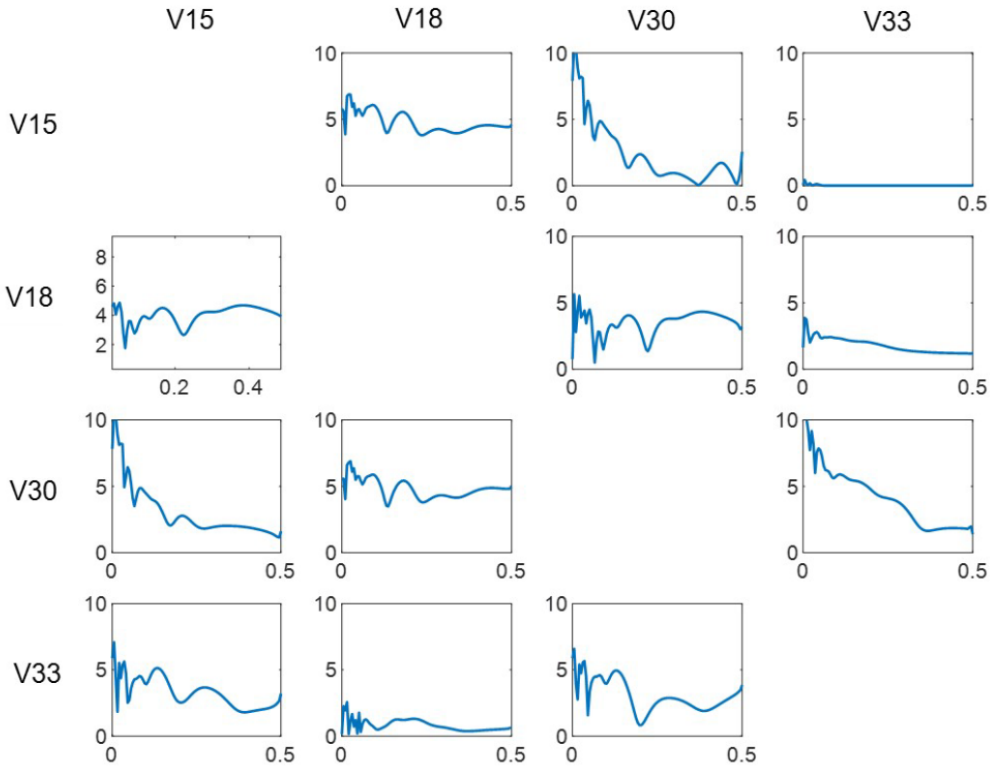

Figure 41: Conditional spectral Granger causality results for IDV (1)

Figure 39 shows the TDGC map of the 165th sample in IDV (1), where V7, V15, V17, V18, V20, V29, V30, V31, and V33 represent Variables 7, 15, 17, 18, 20, 29–31, and 33, respectively. In Figure 39, the green arrows connecting two nodes represent the causality between these two variables, while the blue lines denote the mutual effects between those two variables, failing to provide evidence of their causal relations. Moreover, by reference to the definition of causal flow in a node as the number of

outgoing flows minus that of incoming flows [90], the causal flows of each variable are shown in Figure 40. It can be seen that Variables 15, 18, 30, and 33 have a positive causal flow, so they are potentially supposed to be the root causes, while Variables 7 and 17 have a negative causal flow, indicating that they are less likely to be the faulty variables.

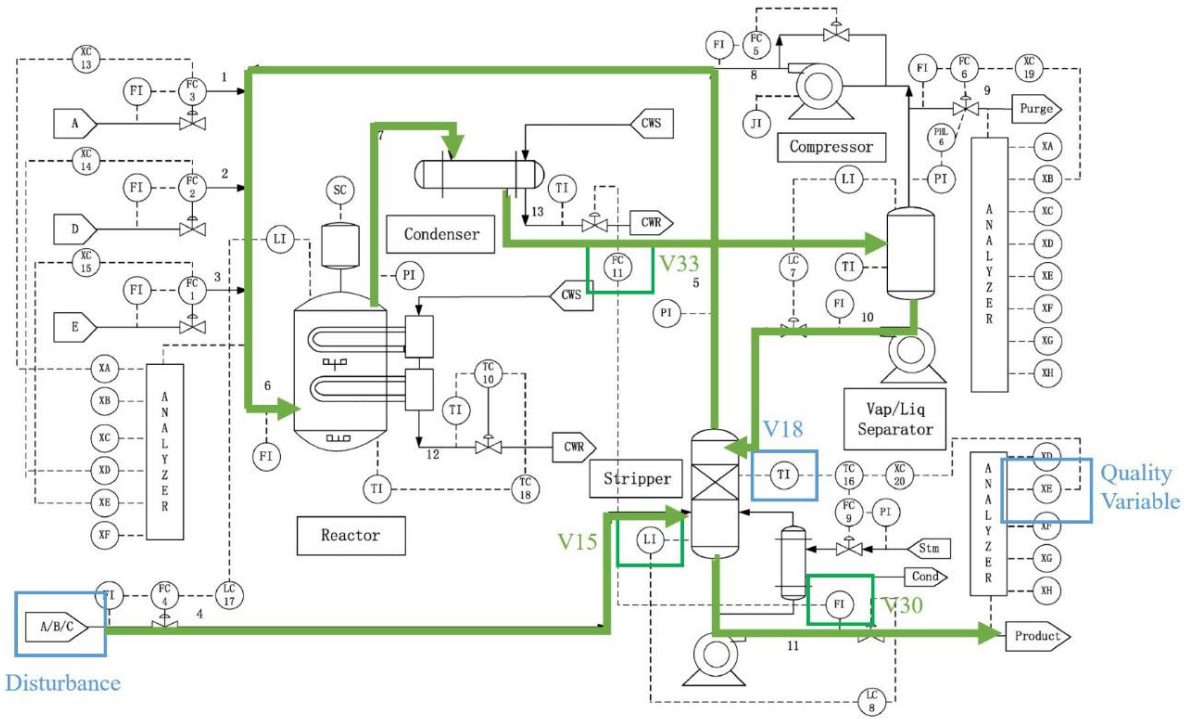

Figure 42: Direct causal analysis for IDV (1)

From the TDGC map, it is observed that Variable 18 Granger-causes Variables 15 and 30, but it fails to provide further information on their causal relations. Thus, CSGC is further applied on these contributing variables (Variables 15, 18, 30, and 33). Figure 41 depicts the CSGC graph for these variables of the 165th sample. As illustrated in Figure 41, the subplot under Column V18 on Row V33 has strong variations, while the fluctuation of the subplot under Column V33 on Row V18 is smaller, showing that Variable 18 has larger causal effect on Variable 33. Thus, it is concluded that Variable 18 is the root cause of IDV (1), which is validated with the actual variations in the process shown in Figure 42. At the beginning, IDV (1) causes the composition change of A and C feeds in Stream 4. Then Stream 4 flows to the stripper, leading to direct influence on the level (Variable 15) and temperature (Variable 18) of the stripper (Variable 18). Afterwards, the stripper steam flows to the reactor, which affects the subsequent components, including the compressor, reactor, condenser, and separator. Finally,

the flow reaches the stripper again and affects the output variable in the product, which is the composition of Product E. Consequently, it is reasonable to conclude Variable 18 as the root cause, which aligns with the available information of IDV (1), and the affected variables align with the extracted contributing variables, showing the effectiveness of the proposed fault diagnosis and root cause identification framework.

5.3 Three-phase Flow Facility Case Study

The three-phase flow facility (TPFF) is provided by Cranfield University, and it is a pressurized system with controlled and measured flowrates of water, oil, and air [123]. TPFF includes three normal datasets (T1, T2, and T3) and six faulty cases with 24 variables. In this case study, Variables 1–7, 10, 11, 12, 19–21, and 23 are chosen as process variables and Variable 13 is the quality variable, which is the flow density in the top riser.

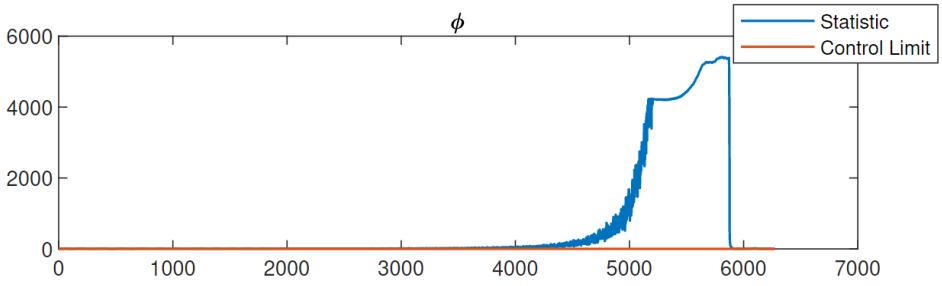

Figure 43: DALVR-based monitoring result for faulty case 3

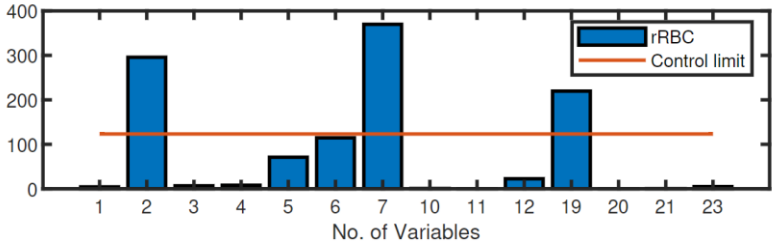

Figure 44: Relative reconstruction-based contribution results for faulty case 3

The DrLVR, DAPLS, and DALVR models are trained with the samples in T2, and their parameters are determined as follows: l = 1, s = 3, and κ = 0.005 for DrLVR; l = 5, s = 1, d = 1, γ_w = 0.005, γ_β = 0.005, and γ_δ = 0.005 for DAPLS; l = 1, s = 1, d = 1, γ_w = 0.005, γ_β = 0.005, and γ_δ = 0.005 for DALVR. T3 is employed as the test data, and the monitoring metrics of T^2 are summarized in Table 15. The confidence level of control limit is chosen as 99%. The faults in the three-phase flow facility (TPFF)

can be divided into quality-relevant ones (faulty cases 1, 2 and 4–6) and quality-irrelevant ones (faulty case 3) [20], and the monitoring performance of several algorithms are summarized in Table 5. For quality-relevant faults, DALVR obtains the largest ARs, and PRs for faulty cases 1, 2, and 5, while its FDRs are either the largest or relatively high, and it maintains the smallest FARs for faulty case 2 and comparable FARs for faulty cases 1 and 5. For faulty case 2, although the FDR of DrLVR is the largest, its FAR is also very high, which shows poor monitoring performance. For faulty case 4, the performance of DALVR is mediocre, and for faulty case 6, DALVR has the smallest FAR, and its FDR, AR, and PR are comparable to the largest values. For quality-irrelevant cases, FAR is the main index that needs to be considered. The FAR of DALVR is the smallest one. Generally, the monitoring performance of DALVR is the best among these algorithms.

Table 15: Monitoring metrics of T^2 for the three-phase flow facility

Fault Type	FDR			FAR			AR			PR		
	DrLVR	DAPLS	DALVR	DrLVR	DAPLS	DALVR	DrLVR	DAPLS	DALVR	DrLVR	DAPLS	DALVR
1	0.9976	**1**	**1**	**0.1548**	0.2541	0.1550	0.8442	0.7692	**0.8592**	0.3707	0.2845	**0.3946**
2	0	**1**	0.1429	0.1078	1	**0.0957**	0.8904	0.0020	**0.9027**	0	0.0020	**0.0030**
4	0.8386	**0.9609**	0.8723	**0.2131**	0.9973	0.2403	**0.7753**	0.0687	0.7398	**0.1771**	0.0666	0.1271
5	0.9162	**0.9746**	0.9626	**0.1296**	0.5787	0.1534	0.8784	0.5546	**0.8805**	0.6916	0.3483	**0.7109**
6	0.0054	**0.0325**	0.0049	0.0140	0.0278	**0**	**0.9238**	0.9004	0.9236	0.0667	**0.0422**	0.0778
3	**0**	**0**	**0**	0.4978	1	**0.4225**	**0**	**0**	**0**	**0**	**0**	**0**

Faulty case 3 is selected for further investigation, where the top separator input blockage is introduced into the system [123]. Figure 43 presents the monitoring result based on DALVR with the combined index, which shows that the fault is detected from the 3554th sample. The relative RBC of the 3554th sample is shown in Figure 44, and Variables 2, 6, and 7 (V2, V6, and V7) are selected as the contributing variables for further Granger causality. As shown in Figure 45, Variable 2 Granger-causes Variable 7, but the relationships between Variables 2, 6 and 6, 7 cannot be identified with TDGC in Figure 45. To further determine the cause-and-effect relations of Variables 2 and 6, and 6 and 7, the root cause analysis with CSGC is further conducted. The result of CSGC in Figure 46 demonstrates that the subplot under Column V2 on Row V6 has stronger peak than the subplot under Column V6 on Row V2, while the subplot under Column V6 on Row V7 has smaller variations than the subplot under Column V7 on Row V6, which indicates that in frequency domain, Variable 2 Granger-causes Variable 6, and Variable 7 also affects Variable 6. Thus, it is concluded that Variable 2 is supposed to be the root cause.

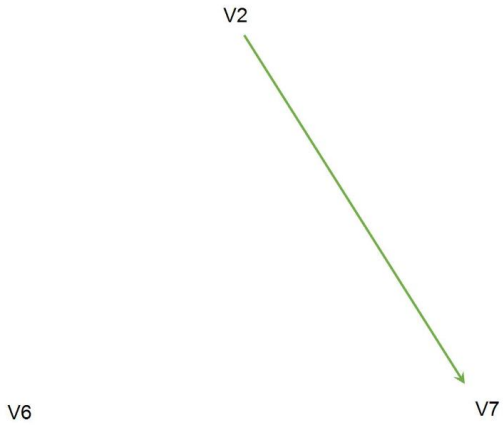

Figure 45: Time-domain Granger causality map for faulty case 3

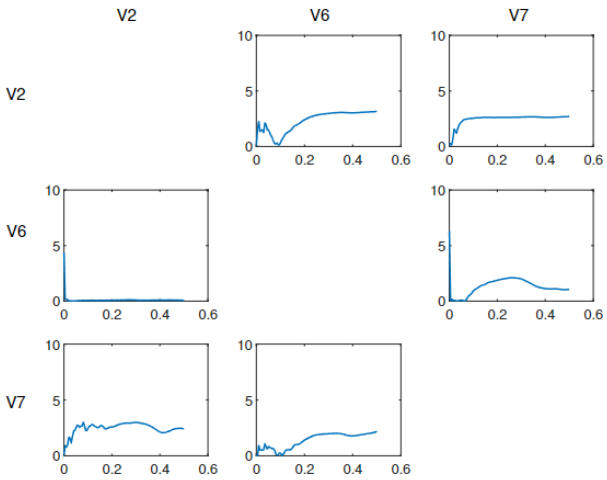

Figure 46: Conditional spectral Graner causality results for faulty case 3

Based on the ground truth of faulty case 3, the fault is caused by the blockage of top separator input. Variable 2 measures the pressure in the bottom of the riser, Variable 6 is the difference pressure between the bottom of the riser and the top separator input, and Variable 7 denotes the differential pressure over VC404. As shown in Figure 47, it is noted that the top separator input blockage reduces the pressure in the top separator input, imposing influence on the pressure in the 4-inch riser. Thus, the value of Variable 2 is affected, which can reasonably be considered as the root cause. This causal inference is denoted by green lines. The difference pressure between the bottom of the riser and the top separator input will be affected slightly and return to normal rapidly, which makes no sense to the influence of other variables. In terms of the causal relationship between Variables 2 and 7, which is illustrated by

orange lines in Figure 47, Variable 2 affects the level in 3-phase separator directly, and the level in a series of separator facilities will also change subsequently. Through the 2-inch riser, Variable 7 is affected correspondingly. Therefore, Variable 2 acts as the root cause in faulty case 3, and the effectiveness of the framework is also proven.

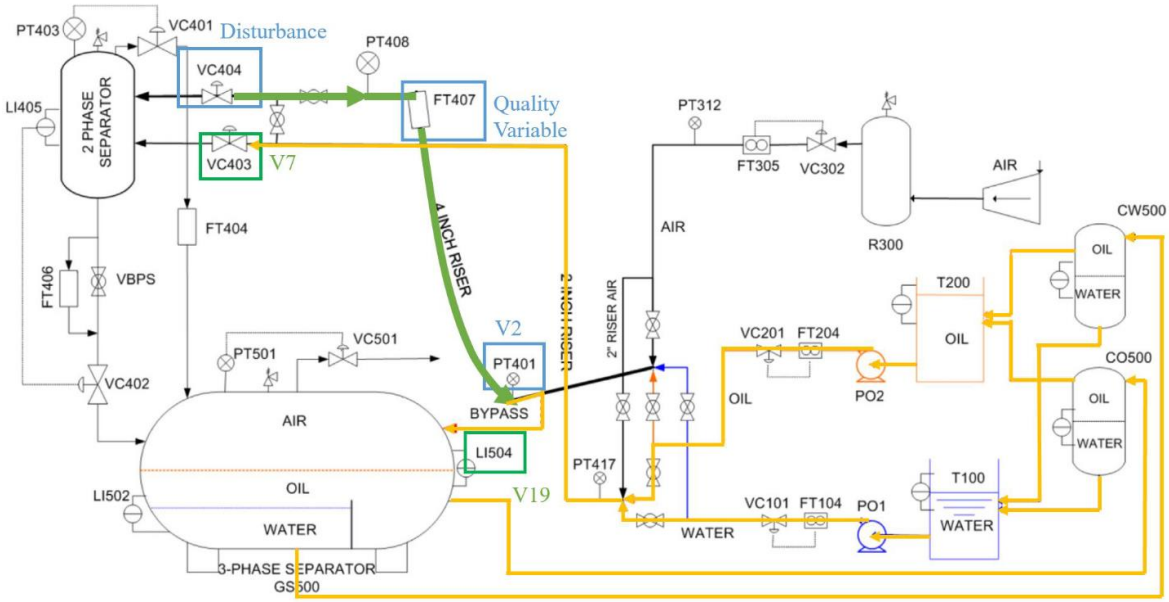

Figure 47: Direct causal analysis for faulty case 3

5.4 Summary

In this chapter, an improved fault diagnosis and root cause identification framework is also developed based on DALVR to identify, investigate, and categorize the root causes of faulty samples. The framework effectively integrates the relative RBC for fault diagnosis, and TDGC and CSGC for root cause analysis. The case studies with the Tennessee Eastman process and three-phase flow facility have demonstrated the effectiveness of the proposed DALVR method and the corresponding fault diagnosis and root cause identification framework.

Chapter 6
Improved Autoencoder with Dynamic Hidden Layer for Anomaly Detection

Inspired by autoregressive integrated moving average (ARIMA) model [114], the hidden layer of AE is improved to handle the issue of dynamics in this section. Then, a modified autoencoder method with dynamic hidden layer (DHL-AE) is proposed to address both high dimensionality and dynamic relations simultaneously and the corresponding anomaly detection monitoring scheme is designed to improve the performance of identifying abnormal points. The validity and superiority of DHL-AE and its corresponding anomaly detection monitoring are proved by case studies.

6.1 Autoencoder with Dynamic Hidden Layer

As a specific type of neural networks, AE has been created by encoding the input into a compressed representation and decoding it back to reconstruct the input that is nearly identical to the original input to greatest extent possible [124]. The purpose of AE is to learn a lower-dimensional and meaningful representation of the input in an unsupervised way via minimizing the reconstruction error based on a loss function in training [113]. Typically, the mean squared error (MSE) between the original input and the corresponding reconstructed output is adopted to be the loss function.

Assuming $\mathbf{x} \in \mathbb{R}^n$ as the input, the loss function is expressed as [113]

$$L(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \tag{6.1}$$

where $L(\cdot)$ expresses the loss function and $\hat{\mathbf{x}}$ is the corresponding reconstructed output vector of $\mathbf{x}$. The encoder projects the input into the hidden layer to obtain a compressed representation $\mathbf{z} \in \mathbb{R}^k$, and the decoder reconstructs the output of AE based on $\mathbf{z}$ as $\hat{\mathbf{x}}$. To construct the encoder by the projection from the space of input and output data to the space in the hidden layer $\varphi: \chi \to Z$ and decoder by the projection from the space in the hidden layer to the space of input and output data $\psi: Z \to \chi$, the objective of AE is defined as [125]

$$\underset{\varphi,\psi}{\operatorname{argmin}} \|\mathbf{x} - \psi[\varphi(\mathbf{x})]\|^2 \tag{6.2}$$

where $\varphi(\mathbf{x})$ denotes the process of encoding, and $\mathbf{z} = \varphi(\mathbf{x})$. $\psi[\varphi(\mathbf{x})]$ represents the process of decoding, and $\hat{\mathbf{x}} = \psi[\varphi(\mathbf{x})]$.

More concretely, encoding and decoding can be expressed as

$$\begin{aligned}\mathbf{z} &= f_1(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)\\ \hat{\mathbf{x}} &= f_2(\mathbf{W}_2\mathbf{x} + \mathbf{b}_2)\end{aligned} \tag{6.3}$$

where f_1 and f_2 represent the activation functions used to construct the hidden and output layers, respectively. $\mathbf{W}_1$ and $\mathbf{b}_1$ are the weights and bias of encoder, while $\mathbf{W}_2$ and $\mathbf{b}_2$ are the weights and bias of decoder.

Based on the structure of AE, DHL-AE algorithm is constructed by processing dynamics in the hidden layer. Premised on the assumption of normalized input, the encoder maps the original input into the latent space in hidden layer, and the dynamics in the original data are captured by the compressed representation. After decoding, dynamic relations are also retained in the output, which is considered as the reconstructed input. Thus, to deal with dynamic datasets, it is reasonable to handle dynamic relations in the hidden layer.

For the dynamic input data $\mathbf{X} \in \mathbb{R}^{n\times m}$, after normalization, the encoder projects $\mathbf{X}$ into the latent space in the hidden layer, obtaining $\mathbf{Z} = f_1(\mathbf{W}_1\mathbf{X} + \mathbf{b}_1) \in \mathbb{R}^{n\times k}$. Considering dynamic relations existing in the input $\mathbf{X}$, motivated by ARIMA, the components of compressed representation in the hidden layer can be constructed as follows [126]:

$$\begin{aligned}\hat{\mathbf{z}}^{(t)} &= \mathbf{v}^{(t)} + \mathbf{e}^{(t)}\\ \mathbf{v}^{(t)} &= \sum_{i=1}^{p} \beta_i \mathbf{z}^{(t-i)}\\ \mathbf{e}^{(t)} &= \sum_{i=1}^{q} \theta_i \boldsymbol{\varepsilon}^{(t-i)} + \boldsymbol{\varepsilon}^{(t)}\end{aligned} \tag{6.4}$$

where $\hat{\mathbf{z}}^{(t)} \in \mathbb{R}^k$ represents the compressed representation reconstructed considering dynamic relations in $\mathbf{z}^{(t)}$ obtained by encoding at time t, which consists of the autoregressive part $\mathbf{v}^{(t)}$ and the moving part $\mathbf{e}^{(t)}$.

Similar to Eq. (2.36), Eq. (6.4) can also be reorganized as

$$\hat{\mathbf{z}}^{(t)} = \left(1 - \sum_{i=1}^{p_a} \beta_{a,i} B^i\right)^{-1} (1-B)^{-d_a} \left(1 - \sum_{i=1}^{q_a} \theta_i B^i\right) \boldsymbol{\varepsilon}^{(t)} \tag{6.5}$$

where $\boldsymbol{\varepsilon}^{(t)}$ is the white noise error at time t $(t = 1, 2, \ldots, n)$ for the moving average part. After the process in the hidden layer, the decoder reconstructs the output from $\hat{\mathbf{z}}$ via $\hat{\mathbf{x}} = f_2(\mathbf{W}_2\hat{\mathbf{z}} + \mathbf{b}_2) \in \mathbb{R}^{m\prime}$.

The structure of DHL-AE algorithm is visualized as Figure 48, and the detailed DHL-AE algorithm is summarized as follows:

1. Scale the input data $\mathbf{X} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\right]$ into zero mean and unit variance.

2. The normalized input data is fed to DHL-AE.

3. After encoding via $\mathbf{Z} = f_1(\mathbf{W}_1\mathbf{X} + \mathbf{b}_1)$, the input $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}$ can be projected to the latent space in the hidden layer as the compressed representation $\mathbf{Z} = \left[\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(n)}\right]$.

4. Decompose $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(n)}$ into two parts, involving the autoregressive part $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(n)}$ and the moving average part $\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \ldots, \mathbf{e}^{(n)}$, leading to the reconstructed compressed representation $\hat{\mathbf{z}}^{(t)} = \mathbf{v}^{(t)} + \mathbf{e}^{(t)}$ (t = 1, 2, ..., n).

5. The decoder reconstructs the input as the output of DHL-AE by $\hat{\mathbf{X}} = f_2\left(\mathbf{W}_2\hat{\mathbf{Z}} + \mathbf{b}_2\right)$, where $\hat{\mathbf{Z}} = \left[\hat{\mathbf{z}}^{(1)}, \hat{\mathbf{z}}^{(2)}, \ldots, \hat{\mathbf{z}}^{(n)}\right]$.

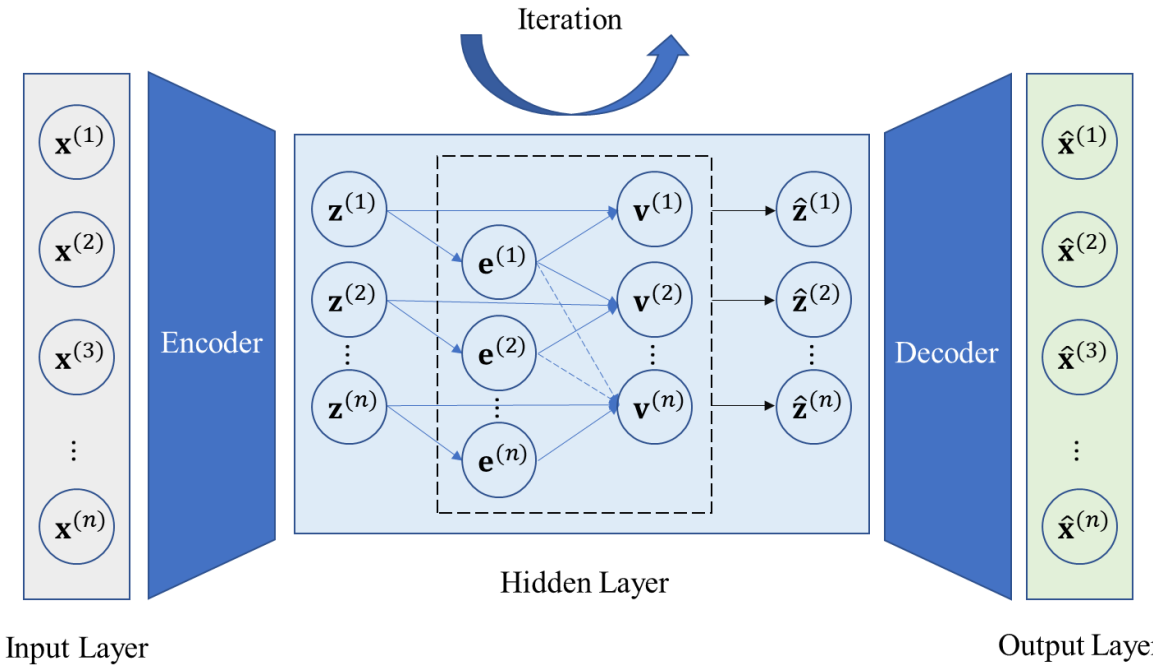

Figure 48: Structure of autoencoder with dynamic hidden layer

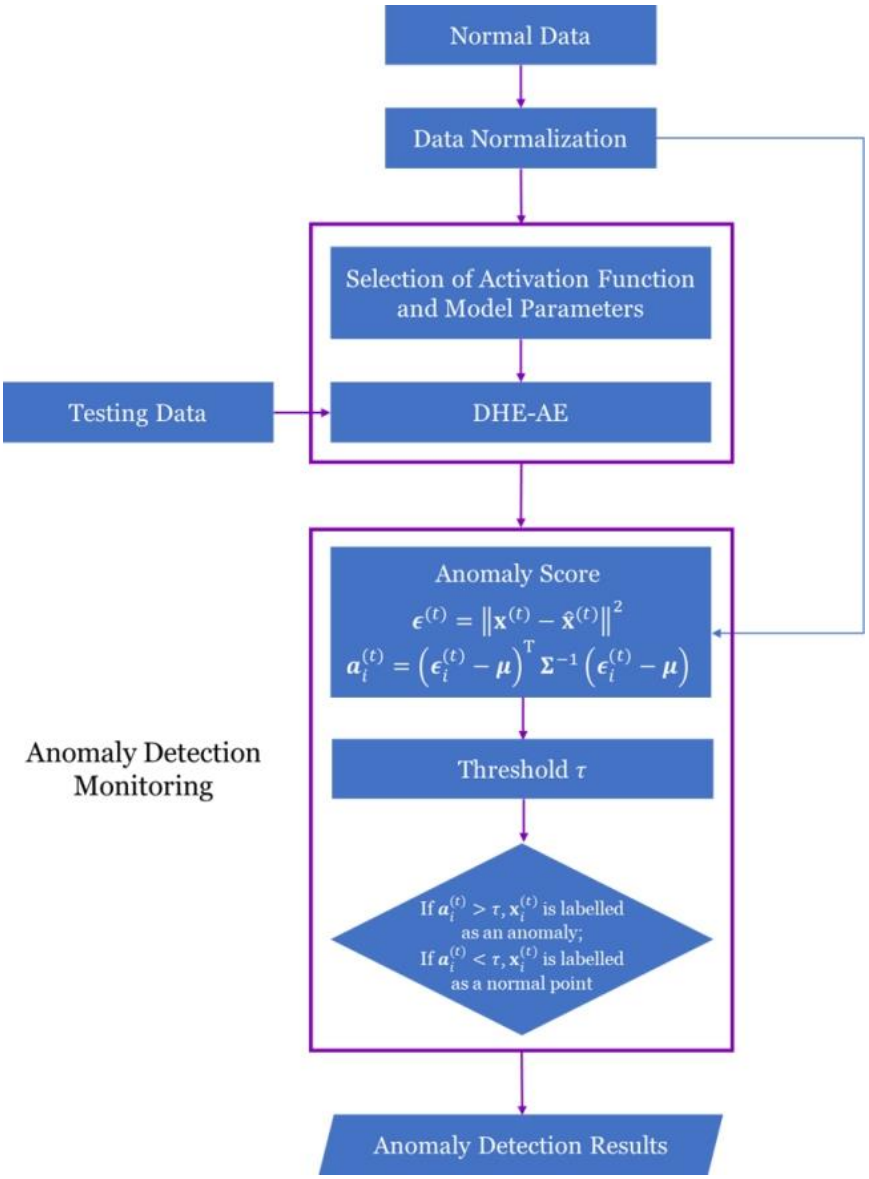

Figure 49: Anomaly detection monitoring scheme based on DHL-AE

6.2 Anomaly Detection Scheme Based on DHL-AE

To enhance the performance, improve the safety, and decrease the potential losses of the process, in this chapter, DHL-AE is integrated with anomaly detection to identify anomalies in the process monitoring. Anomaly detection is based on the conversion from reconstruction errors to anomaly scores. The anomaly score for a datapoint $\mathbf{x}_i^{(t)}$ is computed as [127]

$$\mathbf{a}_i^{(t)} = \left(\boldsymbol{\varepsilon}_i^{(t)} - \boldsymbol{\mu}\right)^{\top} \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{\varepsilon}_i^{(t)} - \boldsymbol{\mu}\right) \tag{6.6}$$

where $\mathbf{a}_i^{(t)}$ is the anomaly score for $\mathbf{x}_i^{(t)}$. $\boldsymbol{\varepsilon}_i^{(t)}$ represents the corresponding RE. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are obtained from a normal distribution using maximum likelihood estimation based on $\boldsymbol{\varepsilon}_i^{(t)}$ [128].

To determine whether a sample is an anomaly or not, the threshold of anomaly scores, which is denoted by τ, is learned using the normal data [127]. If $\mathbf{a}_i^{(t)} > \tau$, the corresponding sample $\mathbf{x}_i^{(t)}$ is labelled as an anomaly, while if $\mathbf{a}_i^{(t)} < \tau$, $\mathbf{x}_i^{(t)}$ is considered as a normal point. For better illustration, the integrated scheme is depicted in Figure 49. In the scheme, DHL-AE is first performed on the normal data after normalization to obtain the proper activation function and model parameters. Then samples of testing data are processed by anomaly detection monitoring to identify irregular datapoints.

6.3 Tennessee Eastman Process Case Study

In this section, the Tennessee Eastman Process (TEP) is applied for case study to demonstrate the effectiveness of the proposed anomaly detection monitoring framework, with 15 disturbances (IDV (1–15)) simulated in the dataset [118]. XMEAS (1-9) are chosen as input variables, and 500 normal datapoints in d00 are selected as training data. Faulty data d02_te is utilized for testing, which represents the disturbance of IDV (2) and has 960 samples in total.

The activation function and model parameters are determined by the training data. The first three-quarters of the training data are designated as the training set, while the complement of the training set is assigned as the test set. Mean squared error (MSE), root mean squared error (RMSE), normalized root mean squared error (NRMSE), and mean absolute error (MAE) [129] are considered as the candidates of model selection criteria to determine the best activation function and model parameters involving the dimension of output and compressed representation of DHL-AE. The activation function is selected from rectified linear unit (ReLU) [130], leaky rectified linear unit (Leaky ReLU) [131], and exponential linear unit (ELU) [132]. As mentioned in Section 6.1, m' denotes the dimension of output $\widehat{\mathbf{X}}$ after decoding, and k refers to the dimension of the compressed representation of input after encoding.

The results of activation function and parameter selection for modeling are shown in Table 16. In terms of activation function, Leaky ReLU and ELU perform better than ReLU, since they have lower MSE, RMSE, and NRMSE for all the metrics which are used for activation function and parameter

selection. Considering the selection criteria and process time simultaneously, the combination of ELU as the activation function and RMSE as the selection criteria is the best choice due to the smallest values of metrics including MSE, RMSE, NRMSE, and MAE, as well as the shortest process time. In this case, the parameters are determined as follows: the dimension of the output $m' = 8$, and the dimension of the compressed representation $k = 5$. Thus, it is concluded that for TEP, ELU is selected as the activation function, and the model parameters are selected as $m' = 8$ and $k = 5$.

In this case study, the proposed anomaly detection monitoring framework based on DHL-AE is applied for the detection of the anomalies in d02_te. Taken as the ground truth, PCA-based anomaly detection monitoring is assigned as the criterion to classify a sample into normal or abnormal datapoints. To manifest the superiority and effectiveness of the DHL-AE based anomaly detection scheme, other methods involving dynamic kernel PCA (DKPCA) [133] and variational autoencoder (VAE) [134] are selected to compare with DHL-AE, since DKPCA and VAE are both suitable for processing complex data with nonlinearity and dynamic relations. Determined by cross validation, the model parameters of DKPCA are selected as follows: the number of principal components $l = 1$, the dynamic order for input variables $s = 20$, and the width of Gaussian kernel function $c = 50000$. For VAE, the model parameters are chosen as follows: the batch size is 100; the dimensionality of latent variable is 3; Leaky ReLU is adopted in both encoder and decoder, while binary cross-entropy loss [135] and Kullback-Leibler divergence [136] are used for loss function.

Figures 50-52 illustrate the anomaly detection monitoring results of DHL-AE, DKPCA, and VAE for comparison. As is shown in these figures, the anomaly scores of the first approximately 210 data points are below the corresponding threshold, while the anomaly scores of the remaining data points exceed the threshold, indicating that they are all detected as anomalies.

For better understanding, the following metrics are introduced to demonstrate the comparison results statistically: accuracy, precision, recall, F1-score [137], false detection rate (FDR), false alarm rate (FAR) [138], and missing alarm rate (MAR) [139], and the corresponding results are presented in Table 17. Among the performance of the algorithms, the proposed algorithm performs best on all the metrics except FDR and MAR. Despite not being the best in FDR and MAR, DHL-AE is still comparable to the FDR and MAR values of VAE, which is the best-performing algorithm on these two metrics. Thus, it is concluded that DHL-AE takes advantages over other DKPCA and VAE in this case study.

Table 16: Activation function and parameter selection for IDV (2) in TEP

Activation Function	Selection Criteria	*m*'	*K*	MSE	RMSE	NRMSE	MAE	Time/s
ReLU	MSE	6	13	0.2537	0.5037	1.3305	0.3145	3288.84
	RMSE	6	13	0.2537	0.5037	1.3305	0.3145	3535.73
	NRMSE	6	13	0.2537	0.5037	1.3305	0.3145	4398.42
	MAE	7	1	0.2178	0.4667	1.2327	0.3283	3254.39
Leaky ReLU	MSE	8	5	**0.2034**	**0.4510**	**1.1914**	**0.3141**	3716.17
	RMSE	8	5	**0.2034**	**0.4510**	**1.1914**	**0.3141**	3309.23
	NRMSE	8	5	**0.2034**	**0.4510**	**1.1914**	**0.3141**	3925.47
	MAE	8	5	**0.2034**	**0.4510**	**1.1914**	**0.3141**	3108.95
ELU	MSE	8	5	**0.2034**	**0.4510**	**1.1914**	**0.3141**	3298.11
	RMSE	8	5	**0.2034**	**0.4510**	**1.1914**	**0.3141**	**3006.76**
	NRMSE	8	5	**0.2034**	**0.4510**	**1.1914**	**0.3141**	3122.58
	MAE	7	1	0.2178	0.4667	1.2327	0.3283	3603.27

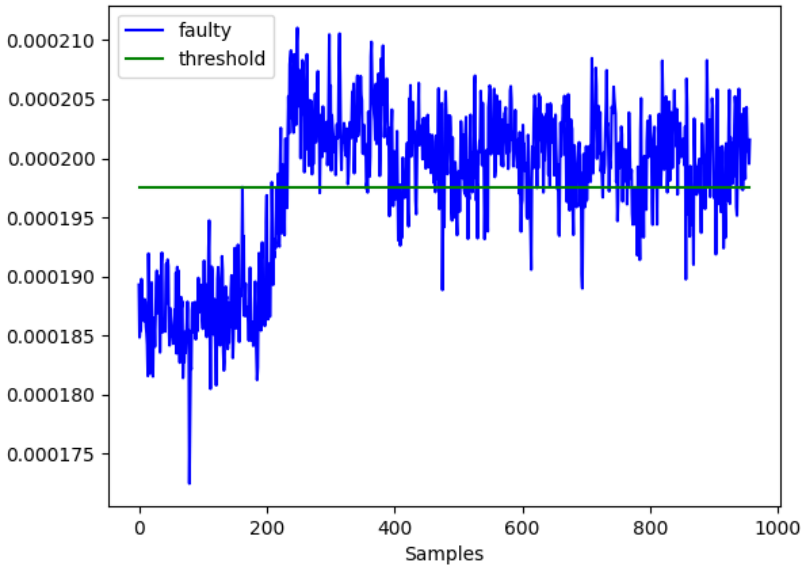

Figure 50: Anomaly detection monitoring results of DHL-AE for IDV (2) in TEP

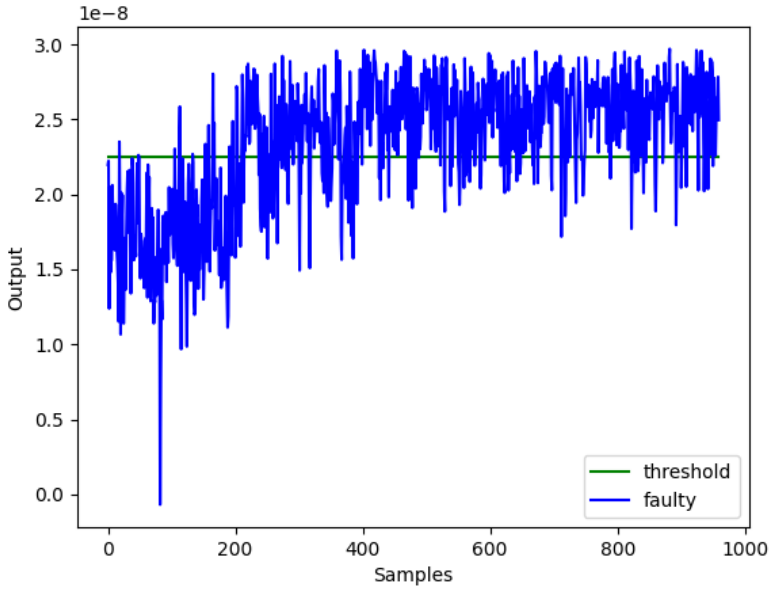

Figure 51: Anomaly detection monitoring results of DKPCA for IDV (2) in TEP

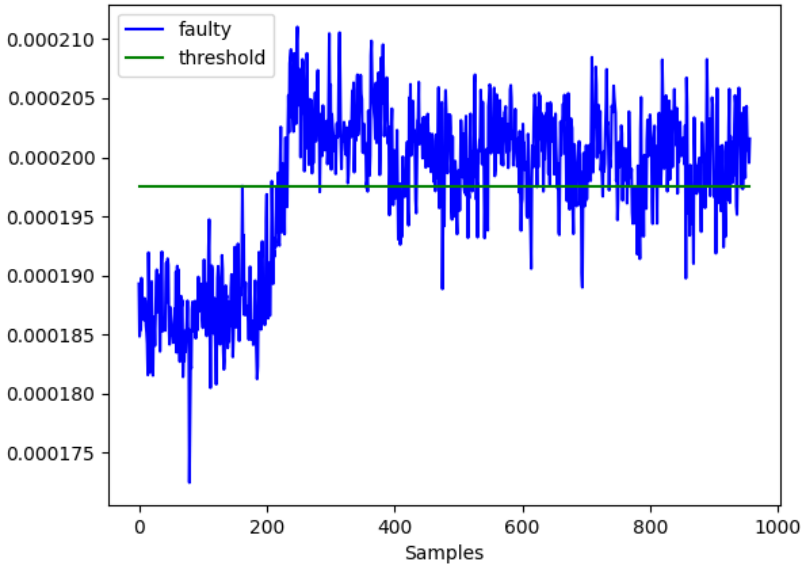

Figure 52: Anomaly detection monitoring results of VAE for IDV (2) in TEP

Table 17: Statistical metrics results of algorithms for IDV (2) in TEP

Algorithms	Accuracy	Precision	Recall	F1-Score	FDR	FAR	MAR
DHL-AE	**0.7615**	**0.7721**	**0.8270**	**0.7986**	0.6740	**0.1730**	0.3260
DKPCA	0.6437	0.6661	0.7559	0.7082	0.4939	0.2441	0.3260
VAE	0.6374	0.7137	0.6055	0.6522	**0.6800**	0.3945	**0.3200**

6.4 Summary

To deal with issues of high-dimensionality and time-dependence existing in the real-world datasets, DHL-AE is proposed in this chapter, and a DHL-AE based anomaly detection monitoring framework is also developed to identify abnormal samples, which is conducive to ensuring the safety of industrial processes and solving the potential problem in the complex industrial applications. The Tennessee Eastman process is used for the case study to show the effectiveness and superiority of the proposed DHL-AE method and the corresponding scheme for anomaly detection monitoring.

Chapter 7
Conclusions

In this thesis, novel methodologies on dynamic latent variable modeling and monitoring are explored and developed to address the key challenges in dealing with complex data containing nonlinearities, dynamic relations, and high dimensionality.

To handle collinear and nonlinear data, adopting Gaussian kernel function, a concurrent kernel latent variable regression (CKLVR) with a regularization term is constructed by performing a full decomposition of original process data space. Based on CKLVR, a concurrent monitoring scheme is designed for fault detection. A synthetic simulation and Tennessee Eastman process (TEP) are used for case studies to demonstrate the effectiveness of CKLVR-based monitoring scheme.

Apart from nonlinearities, dynamic relations are also necessary to be considered for modeling and monitoring. In this case, a dynamic auto-regressive LVR (DALVR) algorithm is created based on regularized LVR to extract dynamic cross-correlations and auto-correlations simultaneously from both process and quality data. To obtain comprehensive monitoring results, the corresponding concurrent DALVR-based monitoring framework is constructed. The additional simulation of TEP is chosen for case study to show the effectiveness and validity.

Combining reconstruction-based contribution (RBC) with time-domain Granger causality (TDGC) and conditional spectral Granger causality (CSGC), DALVR-based fault diagnosis and causal analysis framework is also constructed to locate the root cause of a fault and obtain more accurate fault diagnosis results. The effectiveness of the proposed framework is validated by case studies on the TEP and three-phase flow facility process.

Not only multivariate statistical analysis (MSA) methods, but also autoencoder (AE) techniques are applied for handling complex processes with nonlinearities and dynamics. Moreover, AE approaches are also effective to cope with high dimensionality. To address the issues of both high dimensionality and dynamic relations between elements in the hidden layer, a novel autoencoder with dynamic hidden layer (DHL-AE) is proposed and employed for anomaly detection, which plays an important role in stability and safety of real-world industrial applications. A case study on TEP is performed to show the effectiveness and advantages of DHL-AE over existing methods including dynamic kernel PCA (DKPCA) and variational autoencoder (VAE).

References

[1] Gillo, M. W., & Shelly, M. W. 1974. Predictive modeling of multivariable and multivariate data. Journal of the American Statistical Association, 69(347), 646-653.

[2] Veljanovski, T. 2000. Understanding Roman settlement patterns through multivariate statistics and predictive modelling. Beyond the map: Archaeology and spatial technologies, 321, 147.

[3] Qin S. J. 2003. Statistical process monitoring: basics and beyond. Journal of Chemometrics: A Journal of the Chemometrics Society, 17(8-9): 480-502.

[4] Jardine, A. K., Lin, D., & Banjevic, D. 2006. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical systems and signal processing, 20(7), 1483-1510.

[5] Yin, S., Ding, S. X., Xie, X., & Luo, H. 2014. A review on basic data-driven approaches for industrial process monitoring. IEEE Transactions on Industrial Electronics, 61(11), 6418-6428.

[6] Wang, Y., Si, Y., Huang, B., & Lou, Z. 2018. Survey on the theoretical research and engineering applications of multivariate statistics process monitoring algorithms: 2008–2017. The Canadian Journal of Chemical Engineering, 96(10), 2073-2085.

[7] Pilario, K. E., Shafiee, M., Cao, Y., Lao, L., & Yang, S. H. 2020. A review of kernel methods for feature extraction in nonlinear process monitoring. Processes, 8(1), 24.

[8] Rooney, J. J., & Heuvel, L. N. V. 2004. Root cause analysis for beginners. Quality progress, 37(7), 45-56.

[9] Peerally, M. F., Carr, S., Waring, J., & Dixon-Woods, M. 2017. The problem with root cause analysis. BMJ quality & safety, 26(5), 417-422.

[10] Wold, S., Esbensen, K., & Geladi, P. 1987. Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52.

[11] Comon, P. 1994. Independent component analysis, a new concept?. Signal processing, 36(3), 287-314.

[12] Geladi, P., & Kowalski, B. R. 1986. Partial least-squares regression: a tutorial. Analytica Chimica acta, 185, 1-17.

[13] Hardoon, D. R., Szedmak, S., & Shawe-Taylor, J. 2004. Canonical correlation analysis: An overview with application to learning methods. Neural computation, 16(12), 2639-2664.

[14] Zhu, Q. 2020. Latent variable regression for supervised modeling and monitoring. IEEE/CAA Journal of Automatica Sinica, 7(3), 800-811.

[15] Qin, S. J., Dong, Y., Zhu, Q., Wang, J., & Liu, Q. 2020. Bridging systems theory and data science: A unifying review of dynamic latent variable analytics and process monitoring. Annual Reviews in Control.

[16] Liu, Q., Zhu, Q., Qin, S. J., & Chai, T. 2018. Dynamic concurrent kernel CCA for strip-thickness relevant fault diagnosis of continuous annealing processes. Journal of Process Control, 67, 12-22.

[17] Dong, Y., & Qin, S. J. 2015. Dynamic-inner partial least squares for dynamic data modeling. IFAC-PapersOnLine, 48(8), 117-122.

[18] Dong, Y., & Qin, S. J. 2018. Dynamic-inner canonical correlation and causality analysis for high dimensional time series data. IFAC-PapersOnLine, 51(18), 476-481.

[19] Zhu, Q., Qin, S. J., & Dong, Y. 2020. Dynamic latent variable regression for inferential sensor modeling and monitoring. Computers & Chemical Engineering, 137, 106809.

[20] Xu, B., & Zhu, Q. 2021. Concurrent auto-regressive latent variable model for dynamic anomaly detection. Journal of Process Control, 108, 1-11.

[21] Miller, P., Swanson, R. E., & Heckler, C.E. 1998. Contribution plots: a missing link in multivariate quality control.

[22] Dunia, R., & Qin, S. J. 1998. Subspace approach to multidimensional fault identification and reconstruction. AICHE Journal, 44(8), 1813-1831.

[23] Alcala, C. F., & Qin, S. J. 2009. Reconstruction-based contribution for process monitoring. Automatica, 45(7), 1593-1600.

[24] Zhang, H., & Zhu, Q. 2023. An improved dynamic latent variable regression model for fault diagnosis and causal analysis. The Canadian Journal of Chemical Engineering, 101(6), 3009-3648.

[25] Sakurada, M., & Yairi, T. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, 4-11.

[26] Zhou, X., Hu, Y., Liang, W., Ma, J. & Jin, Q. 2020. Variational LSTM enhanced anomaly detection for industrial big data. IEEE Transactions on Industrial Informatics, 17(5), 3469-3477.

[27] Jackson, J. E. 2005. A user's guide to principal components. John Wiley & Sons.

[28] MacGregor, J. F., Jaeckle, C., Kiparissides, C., & Koutoudi, M. 1994. Process monitoring and diagnosis by multiblock PLS methods. AIChE Journal, 40(5), 826-838.

[29] Qin, S. J. 2012. Survey on data-driven industrial process monitoring and diagnosis. Annual Reviews in Control, 36(2), 220-234.

[30] Westerhuis, J. A., Kourti, T. & MacGregor, J. F. 1998. Analysis of multiblock and hierarchical PCA and PLS models. Journal of Chemometrics: A Journal of the Chemometrics Society, 12(5), 301-321.

[31] Misra, M., Yue, H. H., Qin, S. J., & Ling, C. 2002. Multivariate process monitoring and fault diagnosis by multi-scale PCA. Computers & Chemical Engineering, 26(9), 1281-1293.

[32] Amin, M. T., Khan, F., Imtiaz, S., and Ahmed, S. 2019. Robust process monitoring methodology for detection and diagnosis of unobservable faults. Industrial & Engineering Chemistry Research, 58(41), 19149-19165.

[33] Comon, P. 1994. Independent component analysis, a new concept?. Signal Processing, 36(3), 287-314.

[34] Kini, K. R. & Madakyaru, M. 2022. Improved process monitoring scheme using multi-scale independent component analysis. Arabian Journal for Science and Engineering, 47(5), 5985-6000.

[35] Kano, M., Tanaka, S., Hasebe, S., Hashimoto, I., & Ohno, H. 2003. Monitoring independent components for fault detection. AIChE Journal, 49(4), 969-976.

[36] Jiang, Q., Yan, X. & Li, J. 2016. PCA-ICA integrated with Bayesian method for non-Gaussian fault diagnosis. Industrial & Engineering Chemistry Research, 55(17), 4979-4986.

[37] Choi, S. W. & Lee, I. B. 2005. Multiblock PLS-based localized process diagnosis. Journal of Process Control, 15(3), 295-306.

[38] Chen, Z., Zhang, K., Ding, S. X., Shardt, Y.A., & Hu, Z. 2016. Improved canonical correlation analysis-based fault detection methods for industrial processes. Journal of Process Control, 41, 26-34.

[39] Chen, Z., Ding, S. X., Zhang, K., Li, Z., & Hu, Z. 2016. Canonical correlation analysis-based fault detection methods with application to alumina evaporation process. Control Engineering Practice, 46, 51-58.

[40] Jiang, Q., Gao, F., Yan, X., & Yi, H. 2018. Multiobjective two-dimensional CCA-based monitoring for successive batch processes with industrial injection molding application. IEEE Transactions on Industrial Electronics, 66(5), 3825-3834.

[41] Zhu, Q. & Qin, S. J. 2019. Supervised diagnosis of quality and process faults with canonical correlation analysis. Industrial & Engineering Chemistry Research, 58(26), 11213-11223.

[42] Zhu, Q. 2020. Latent variable regression for supervised modeling and monitoring. IEEE/CAA Journal of Automatica Sinica, 7(3), 800-811.

[43] Wold, S., Kettaneh-Wold, N. & Skagerberg, B. 1989. Nonlinear PLS modeling. Chemometrics and Intelligent Laboratory Systems, 7(1-2), 53-65.

[44] Rosipal, R. & Trejo, L. J. 2001. Kernel partial least squares regression in reproducing kernel hilbert space. Journal of Machine Learning Research, 2(Dec), 97-123.

[45] Zhang, Y. & Hu, Z. 2011. Multivariate process monitoring and analysis based on multi-scale KPLS. Chemical Engineering Research and Design, 89(12), 2667-2678.

[46] Zhou, P., Zhang, R., Liang, M., Fu, J., Wang, H., & Chai, T. 2020. Fault identification for quality monitoring of molten iron in blast furnace ironmaking based on KPLS with improved contribution rate. Control Engineering Practice, 97, 104354.

[47] Lai, P. L., & Fyfe, C. 2000. Kernel and nonlinear canonical correlation analysis. International Journal of Neural Systems, 10(5), 365-377.

[48] Liu, Y., Liu, B., Zhao, X., & Xie, M. 2017. A mixture of variational canonical correlation analysis for nonlinear and quality-relevant process monitoring. IEEE Transactions on Industrial Electronics, 65(8), 6478-6486.

[49] Qin, S. J., & McAvoy, T. J. 1992. Nonlinear PLS modeling using neural networks. Computers & Chemical Engineering, 16(4), 379-391.

[50] Shang, C., Yang, F., Huang, D., & Lyu, W. 2014. Data-driven soft sensor development based on deep learning technique. Journal of Process Control, 24(3), 223-233.

[51] Alauddin, M., Khan, F., Imtiaz, S., & Ahmed, S. 2020. A variable mosquito flying optimization-based hybrid artificial neural network model for the alarm tuning of process fault detection systems. Process Safety Progress, 39, e12122.

[52] Yin, Z., & Hou, J. 2016. Recent advances on SVM based fault diagnosis and process monitoring in complicated industrial processes. Neurocomputing, 174, 643-650.

[53] Li, X., Ming, S., Ding, L., Gang, X. and Li., J. 2009. Novel HVAC fan machinery fault diagnosis method based on KPCA and SVM. In 2009 International Conference on Industrial Mechatronics and Automation, 492-496.

[54] Liu, J., Hu, Y., & Yang, S. 2021. A SVM-based framework for fault detection in high-speed trains. Measurement, 172, 108779.

[55] Patrick, E. A., & Fischer III, F. P. 1970. A generalized k-nearest neighbor rule. Information and control, 16(2), 128-152.

[56] Anava, O., & Levy, K. 2016. k-nearest neighbors: From global to local. Advances in Neural Information Processing Systems, 29.

[57] Liu, M., Zou, L., Yu, X., Zhou, Y., Wang, X., & Tang, B. 2020. Knowledge aided covariance matrix estimation via Gaussian kernel function for airborne SR-STAP. IEEE Access, 8, 5970-5978.

[58] Zhang, Y., Zhou, H., Qin, S. J., & Chai, T. 2009. Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares. IEEE Transactions on Industrial Informatics, 6(1), 3-10.

[59] Wold, S., Antti, H., Lindgren, F., & Öhman, J. 1998. Orthogonal signal correction of near-infrared spectra. Chemometrics and Intelligent Laboratory Systems, 44(1-2), 175-185.

[60] Zhou, D., Li, G., & Qin, S. J., 2010. Total projection to latent structures for process monitoring. AIChE Journal, 56(1), 168-178.

[61] Qin, S. J. & Zheng, Y. 2013. Quality‐relevant and process‐relevant fault monitoring with concurrent projection to latent structures. AIChE Journal, 59(2), 496-504.

[62] Zhu, Q., Liu, Q. & Qin, S. J. 2017. Concurrent quality and process monitoring with canonical correlation analysis. Journal of Process Control, 60, 95-103.

[63] Wang, G., Li, J., Sun, C., & Jiao, J. 2018. Least squares and contribution plot based approach for quality-related process monitoring. IEEE Access, 6, 54158-54166.

[64] Bounoua, W., & Bakdi, A. 2021. Fault detection and diagnosis of nonlinear dynamical processes through correlation dimension and fractal analysis based dynamic kernel PCA. Chemical Engineering Science, 229, 116099.

[65] Mou, T., Li, S., & Zou, Y. 2021, July. Enhancing Comprehensive Contribution Plot for Fault Isolation of Distributed Systems. In 2021 40th Chinese Control Conference (CCC) (pp. 4479-4484). IEEE.

[66] Dunia, R., & Joe Qin, S. 1998. Subspace approach to multidimensional fault identification and reconstruction. AICHE journal, 44(8), 1813-1831.

[67] Gertler, J., Li, W., Huang, Y., & McAvoy, T. 1999. Isolation enhanced principal component analysis. AIChE Journal, 45(2), 323-334.

[68] Choi, S. W., Lee, C., Lee, J. M., Park, J. H., & Lee, I. B. 2005. Fault detection and identification of nonlinear processes based on kernel PCA. Chemometrics and intelligent laboratory systems, 75(1), 55-67.

[69] Hu, C., Luo, J., Kong, X., & Feng, X. 2021. Novel fault subspace extraction methods for the reconstruction-based fault diagnosis. Journal of Process Control, 105, 129-140.

[70] Li, G., Joe Qin, S., & Zhou, D. 2010. Output relevant fault reconstruction and fault subspace extraction in total projection to latent structures models. Industrial & Engineering Chemistry Research, 49(19), 9175-9183.

[71] He, B., Yang, X., Chen, T., & Zhang, J. 2012. Reconstruction-based multivariate contribution analysis for fault isolation: A branch and bound approach. Journal of Process Control, 22(7), 1228-1236.

[72] Mnassri, B., & Ouladsine, M. 2015. Reconstruction-based contribution approaches for improved fault diagnosis using principal component analysis. Journal of Process Control, 33, 60-76.

[73] Nomikos, P. 1997. Statistical monitoring of batch processes. Prepr. Joint Statistical Meet., Anaheim, CA.

[74] Qin, S. J., Valle, S., & Piovoso, M. J., 2001. On unifying multiblock analysis with application to decentralized process monitoring. Journal of Chemometrics: A Journal of the Chemometrics Society, 15(9), 715-742.

[75] Westerhuis, J. A., Gurden, S. P., & Smilde, A.K., 2000. Generalized contribution plots in multivariate statistical process monitoring. Chemometrics and Intelligent Laboratory Systems, 51 (1), 95-114.

[76] Kourti, T., & MacGregor, J. F., 1994. Multivariate SPC methods for monitoring and diagnosing of process performance. Proceedings of PSE, 739–746.

[77] Liu, Y., Taniguchi, M., & Ombao, H. 2021. Statistical Inference for Local Granger Causality. arXiv preprint arXiv:2103.00209.

[78] Friston, K. J., Harrison, L., & Penny, W. 2003. Dynamic causal modelling. Neuroimage, 19(4), 1273-1302.

[79] Granger, C. W. J. 1963. Economic processes involving feedback. Information and Control, 6(1), 28-48.

[80] Granger, C. W. 1969. Investigating causal relations by econometric models and cross-spectral methods. Econometrica: Journal of the Econometric Society, 424-438.

[81] Vicente, R., Wibral, M., Lindner, M., & Pipa, G. 2011. Transfer entropy – a model-free measure of effective connectivity for the neurosciences. Journal of computational neuroscience, 30(1), 45-67.

[82] La Fond, T., & Neville, J. 2010, April. Randomization tests for distinguishing social influence and homophily effects. In Proceedings of the 19th International Conference on World Wide Web, 601-610.

[83] Nolte, G., Ziehe, A., Nikulin, V. V., Schlögl, A., Krämer, N., Brismar, T., & Müller, K. R. 2008. Robustly estimating the flow direction of information in complex physical systems. Physical review letters, 100(23), 234101.

[84] Lindner, B., Auret, L., Bauer, M., & Groenewald, J. W. 2019. Comparative analysis of Granger causality and transfer entropy to present a decision flow for the application of oscillation diagnosis. Journal of Process Control, 79, 72-84.

[85] Lütkepohl, H. 2013. Vector autoregressive models. In Handbook of research methods and applications in empirical macroeconomics. Edward Elgar Publishing.

[86] Diks, C., & Panchenko, V. 2004. Modified hiemstra-jones test for Granger non-causality (No. 192). Society for Computational Economics.

[87] Brovelli, A., Ding, M., Ledberg, A., Chen, Y., Nakamura, R., & Bressler, S. L. 2004. Beta oscillations in a large-scale sensorimotor cortical network: directional influences revealed by Granger causality. Proceedings of the National Academy of Sciences, 101(26), 9849-9854.

[88] Geweke, J. 1982. Measurement of linear dependence and feedback between multiple time series. Journal of the American statistical association, 77(378), 304-313.

[89] Goebel, R., Roebroeck, A., Kim, D. S., & Formisano, E. 2003. Investigating directed cortical interactions in time-resolved fMRI data using vector autoregressive modeling and Granger causality mapping. Magnetic resonance imaging, 21(10), 1251-1261.

[90] Yuan, T., & Qin, S. J. 2014. Root cause diagnosis of plant-wide oscillations using Granger causality. Journal of Process Control, 24(2), 450-459.

[91] Valdés-Sosa, P. A., Sánchez-Bornot, J. M., Lage-Castellanos, A., Vega-Hernández, M., Bosch-Bayard, J., Melie-García, L., & Canales-Rodríguez, E. 2005. Estimating brain functional connectivity with sparse multivariate autoregression. Philosophical Transactions of the Royal Society B: Biological Sciences, 360(1457), 969-981.

[92] David, O., Guillemain, I., Saillet, S., Reyt, S., Deransart, C., Segebarth, C., & Depaulis, A. 2008. Identifying neural drivers with functional MRI: an electrophysiological validation. PLoS Biology, 6(12), e315.

[93] Ryali, S., Supekar, K., Chen, T., & Menon, V. 2011. Multivariate dynamical systems models for estimating causal interactions in fMRI. Neuroimage, 54(2), 807-823.

[94] Li, Y., Lei, M., Cui, W., Guo, Y., & Wei, H. L. 2019. A parametric time-frequency conditional Granger causality method using ultra-regularized orthogonal least squares and multiwavelets for dynamic connectivity analysis in EEGs. IEEE Transactions on Biomedical Engineering, 66(12), 3509-3525.

[95] Kamat, A., Makled, B., Norfleet, J., Intes, X., Dutta, A., & De, S. 2021. November. Brain network effects related to physical and virtual surgical training revealed by Granger causality. In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 1014-1017.

[96] Shen, G., Wang, P., Hu, K., & Ye, Q. 2021. December. Fault Root Cause Diagnosis Method Based on Recurrent Neural Network and Granger Causality. In 2021 CAA Symposium on Fault Detection, Supervision, and Safety for Technical Processes (SAFEPROCESS). IEEE, 1-6.

[97] Claessen, J., Molini, A., Martens, B., Detto, M., Demuzere, M., & Miralles, D. G. 2019. Global biosphere–climate interaction: a causal appraisal of observations and models over multiple temporal scales. Biogeosciences, 16(24), 4851-4874.

[98] Dhamala, M., Rangarajan, G., & Ding, M. 2008. Estimating Granger causality from Fourier and wavelet transforms of time series data. Physical Review Letters, 100(1), 018701.

[99] Wilson, G. T. 1972. The factorization of matricial spectral densities. SIAM Journal on Applied Mathematics, 23(4), 420-426.

[100] Zhou, X., Hu, Y., Liang, W., Ma, J., & Jin, Q. 2020. Variational LSTM enhanced anomaly detection for industrial big data. IEEE Transactions on Industrial Informatics, 17(5), 3469-3477.

[101] Xu, R., & Wunsch, D. 2005. Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3), 645-678.

[102] Wold, S., Esbensen, K., & Geladi, P. 1987. Principal component analysis. Chemometrics and Intelligent Laboratory Systems, 2, 37-52.

[103] Reynolds, D. A. 2009. Gaussian mixture models. Encyclopedia of Biometrics, 741, 659-663.

[104] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. 1998. Support vector machines. IEEE Intelligent Systems and their Applications, 13(4), 18-28.

[105] Breiman, L., & Cutler, R. A. 2001. Random forests machine learning. Journal of Clinical Microbiology, 2, 199-228.

[106] Bishop, C. M. 1994. Neural networks and their applications. Review of Scientific Instruments, 65(6), 1803-1832.

[107] Pang, G., Shen, C., Cao, L., Hengel, A. V. D. 2021. Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR), 54(2), 1-38.

[108] Zugasti, E., Iturbe, M., Garitano, I., & Zurutuza, U. 2018. Null is not always empty: Monitoring the null space for field-level anomaly detection in industrial IoT environments. In 2018 Global Internet of Things Summit (GIoTS). IEEE.

[109] Tsung, F. G. 2000. Statistical monitoring and diagnosis of automatic controlled processes using dynamic PCA. International Journal of Production Research, 38(3), 625-637.

[110] Yin, J. and Yan, X., 2019. Mutual information – dynamic stacked sparse autoencoders for fault detection. Industrial & Engineering Chemistry Research, 58(47), pp.21614-21624.

[111] Tayeh, T., Aburakhia, S., Myers, R., & Shami, A., 2022. An attention-based ConvLSTM autoencoder with dynamic thresholding for unsupervised anomaly detection in multivariate time series. Machine Learning and Knowledge Extraction, 4(2), 350-370.

[112] Bishop, C. M., & Nasrabadi, N. M. 2006. Pattern Recognition and Machine. New York: springer, 4(4), 738.

[113] Hinton, G. E., & Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507.

[114] Schaffer, A. L., Dobbins, T. A., & Pearson, S. A. 2021. Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions. BMC Medical Research Methodology, 21(1), 1-12.

[115] Aggarwal, C. C., & Aggarwal, C. C. 2017. An Introduction to Outlier Analysis. Springer International Publishing.

[116] Chalapathy, R., & Chawla, S. 2019. Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407.

[117] Qin, S. J., & Zheng, Y. 2013. Quality-relevant and process-relevant fault monitoring with concurrent projection to latent structures. AIChE Journal, 59(2), 496-504.

[118] Downs, J. J., & Vogel, E. F. 1993. A plant-wide industrial process control problem. Computers & Chemical Engineering, 17(3), 245-255.

[119] Braun, B., Castillo, I., Joswiak, M., Peng, Y., Rendall, R., Schmidt, A., Wang, Z., Chiang, L., & Colegrove, B. 2020. Data science challenges in chemical manufacturing. IFAC Preprints.

[120] Dong, Y., & Qin, S. J. 2018. A novel dynamic PCA algorithm for dynamic data modeling and process monitoring. Journal of Process Control, 67, 1-11.

[121] Rieth, C. A., Amsel, B. D., Tran, R., & Cook, M. B. 2017. Additional Tennessee Eastman Process Simulation Data for Anomaly Detection Evaluation. https://doi.org/10.7910/DVN/6C3JR1, Harvard Dataverse, V1.

[122] Davis, J., & Goadrich, M. 2006. The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd International Conference on Machine Learning, 233-240.

[123] Ruiz-Cárcel, C., Cao, Y., Mba, D., Lao, L., & Samuel, R. T. 2015. Statistical process monitoring of a multiphase flow facility. Control Engineering Practice, 42, 74-88.

[124] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1985. Learning internal representations by error propagation.

[125] Goodfellow, I., Bengio, Y., & Courville, A. 2016. Deep Learning. MIT Press.

[126] Hyndman, R. J., & Athanasopoulos, G. 2018. Forecasting: Principles and Practice. OTexts.

[127] Ahmad, S., Styp-Rekowski, K., Nedelkoski, S., & Kao, O. 2020. Autoencoder-based condition monitoring and anomaly detection method for rotating machines. In 2020 IEEE International Conference on Big Data, 4093-4102.

[128] Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148.

[129] Müller, A. C., & Guido, S. 2016. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, Inc.

[130] Arora, R., Basu, A., Mianjy, P., & Mukherjee, A. 2016. Understanding deep neural networks with rectified linear units. arXiv preprint arXiv:1611.01491.

[131] Maas, A. L., Hannun, A. Y., & Ng, A. Y. 2013. Rectifier nonlinearities improve neural network acoustic models. Proceedings of the 30th International Conference on Machine Learning, Atlanta, USA.

[132] Clevert, D. A., Unterthiner, T., & Hochreiter, S. 2015. Fast and accurate deep network learning by exponential linear units (elus), arXiv preprint arXiv:1511.07289.

[133] Choi, S. W., & Lee, I. B. 2004 Nonlinear dynamic process monitoring based on dynamic kernel PCA. Chemical Engineering Science. 59(24), 5897-5908.

[134] Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., & Chen, J. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In Proceedings of the 2018 World Wide Web Conference, 187-196.

[135] Ruby, U., & Yendapalli, V. 2020 Binary cross entropy with deep learning technique for image classification. International Journal of Advanced Trends in Computer Science and Engineering. 9(10).

[136] Belov, D. I., & Armstrong, R. D. 2011 Distributions of the Kullback–Leibler divergence with applications. British Journal of Mathematical and Statistical Psychology. 64(2), 291-309.

[137] Powers, D. M. 2020. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061.

[138] Zhang, H., & Zhu, Q. 2022 Concurrent multilayer fault monitoring with nonlinear latent variable regression. Industrial and Engineering Chemistry Research, 61, 1423-1442.

[139] Izadi, I., Shah, S. L., Shook, D.S., & Chen, T. 2009. An introduction to alarm analysis and design. IFAC Proceedings Volumes, 42, 645-650.

Appendix A
Calculation of Spectral Density

Spectral density $S_{ii}(f)$ in SGC, which is applied for CSGC in Section 2.3.2, can be calculated as follows [98].

Assume the Fourier transformation is applied for SGC, obtaining

$$\begin{bmatrix} A_{ii}(f) & A_{ij}(f) \\ A_{ji}(f) & A_{jj}(f) \end{bmatrix} \begin{bmatrix} \mathbf{x}_i(f) \\ \mathbf{x}_j(f) \end{bmatrix} = \begin{bmatrix} E_i(f) \\ E_j(f) \end{bmatrix} \tag{A. 1}$$

where $[A_{ij}(f)]$ define the coefficient matrix. Define $H_{ij}(f) = [A_{ij}(f)]^{-1}$ as the spectral transfer function matrix, and Eq. (A.1) is rewritten as

$$\begin{bmatrix} \mathbf{x}_i(f) \\ \mathbf{x}_j(f) \end{bmatrix} = \begin{bmatrix} H_{ii}(f) & H_{ij}(f) \\ H_{ji}(f) & H_{jj}(f) \end{bmatrix} \begin{bmatrix} E_i(f) \\ E_j(f) \end{bmatrix}. \tag{A. 2}$$

Thus, the spectral density matrix is obtained as

$$\mathbf{S}(f) = \mathbf{H}(f)\mathbf{\Sigma}\mathbf{H}^*(f) \tag{A. 3}$$

where $\mathbf{\Sigma}$ is the error covariance matrix of a full model which can be obtained by matrix factorization [99]. Besides, the superscript of $*$ means the corresponding adjoint matrix of a known matrix.

Pre-multiply Eq. (A. 3) on both sides with $\begin{bmatrix} 1 & 0 \\ -\Sigma_{ij}/\Sigma_{ii} & 1 \end{bmatrix}$, leading to

$$\begin{bmatrix} \mathbf{x}_i(f) \\ \mathbf{x}_j(f) \end{bmatrix} = \begin{bmatrix} \widetilde{H}_{ii}(f) & \widetilde{H}_{ij}(f) \\ \widetilde{H}_{ji}(f) & \widetilde{H}_{jj}(f) \end{bmatrix} \begin{bmatrix} E_i(f) \\ \tilde{E}_j(f) \end{bmatrix} \tag{A. 4}$$

where

$$\tilde{E}_j(f) = E_j(f) - \frac{\Sigma_{ij}}{\Sigma_{ii}} E_i(f) \tag{A. 5}$$

$$\begin{bmatrix} \widetilde{H}_{ii}(f) & \widetilde{H}_{ij}(f) \\ \widetilde{H}_{ji}(f) & \widetilde{H}_{jj}(f) \end{bmatrix} = \begin{bmatrix} H_{ii}(f) + \frac{\Sigma_{ij}}{\Sigma_{ii}} H_{ij}(f) & H_{ij}(f) \\ H_{ji}(f) + \frac{\Sigma_{ij}}{\Sigma_{ii}} H_{ii}(f) & H_{jj}(f) \end{bmatrix}. \tag{A. 6}$$

Thus, $S_{ii}(f)$ is expressed as

$$S_{ii}(f) = \tilde{H}_{ii}(f)\,\Sigma_{ii}\tilde{H}_{ii}^{*}(f) + \tilde{H}_{ij}(f)\,\widetilde{\Sigma}_{jj}\tilde{H}_{ij}^{*}(f) \tag{A.7}$$

where $\widetilde{\Sigma}_{jj} = \Sigma_{jj} - \Sigma_{ij}^{2}/\Sigma_{ii}$.